DIALOGUE MODELS FOR INQUIRY AND TRANSACTION

Joris Hulstijn

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Hulstijn, Joris

Dialogue Models for Inquiry and Transaction / Joris Hulstijn.
Proefschrift Universiteit Twente
ISBN 9036514401
Keywords: natural language dialogue systems, formal linguistics

Cover: *Esau selling his birthright to Jacob*, Rembrandt van Rijn,
Amsterdams Historisch Museum/Fodor, Amsterdam

Printed by Febodruk, Enschede.

DIALOGUE MODELS FOR INQUIRY AND TRANSACTION

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof.dr. F.A. van Vught,
volgens besluit van het College van Promoties
in het openbaar te verdedigen
op vrijdag 7 april, 2000, te 16.45 uur.

door

J o r i s   H u l s t i j n

geboren op 13 augustus 1970
te Nijmegen

Promotor: prof.dr.ir. A. Nijholt

Contents

Acknowledgements

Writing a thesis seems a lonely affair. This is not true at all. During my stay at Twente, the Parlevink language technology group has evolved into a truly multi-disciplinary research group. I would like to thank everybody in the group who have contributed to this thesis, and to the fact that it was fun to write it.

In particular I would like to thank my promotor, Anton Nijholt, not just for supervising this thesis but especially for stimulating such wildly diverging topics as computational humour, virtual reality and formal semantics and pragmatics. I want to thank Jan Kuper, who supervised the logical parts of the thesis and took me under his wing for the last year. Without Jan I could not have done it. Many thanks for Gerrit van der Hoeven for backing the research from the beginning. Thanks also to Peter Asveld for his meticulous reading of the manuscript and to Franciska de Jong.

Much of the work reported in this thesis is joint work. I would like to thank Gies Bouman and the Philips research laboratories at Aachen for allowing me to reproduce our joint work in section 6.2. Thanks also to the students who did the STEMSTEM project, which is described in section 6.3. Many thanks to Arjan van Hessen and to Mathieu van den Berk, and also to Hendri Hondorp whom I worked with on speech generation and the animated face, section 6.4. Thanks a lot to Boris van Schooten for our discussions on the limits of formal methods, and for his last-minute help.

I would like to thank everybody who contributed to the SCHISMA project at some stage, in particular Jan Schaake, Stan van de Burgt, Hugo ter Doest, Rieks op den Akker, Jos Buijs, René Steetskamp, Danny Lie en Bas Leijdekkers. Special thanks go to Toine Andernach, who taught me a lot about dialogue. I really enjoyed sharing an office with Toine and later with Djoerd Hiemstra. Teaching software engineering turned out to be very rewarding because of colleagues like Klaas van den Berg, Nico van Diepen and Henk Bonnes.

The discourse community in Amsterdam gave me a second academic home. In particular Henk Zeevat, Robert van Rooy, Jelle Gerbrandy and Paul Harrenstein, thanks for being both enthusiastic and challenging. To my film friends in Enschede and elsewhere, Henno Theisens, Victor Breedveld, Linda Vonhoff, Susan Schreefel, Marieke Jochman, Laila Hermsen and Michiel van Leersum: I have very good memories of evenings spent with wine, food, film and discussion.

And Kati, thanks for your love.

Preview

This thesis is called *Dialogue Models for Inquiry and Transaction*. It is concerned with formal models of dialogue, to be applied in the design and testing of dialogue systems. We restrict ourselves to dialogues of a particular type. We have chosen to model dialogues for *inquiry*, cooperative information exchange, and *transaction*, reaching agreement about some future action.

In chapter 1 an iterative and user-centred development method is recommended. An initial design must be repeatedly evaluated and improved upon by means of data from target users. In order to design and test a system, one needs to relate properties of the design to verifiable dialogue properties. We introduce *usability* as the most important quality factor for dialogue systems. Can the controls of the system actually be learnt, understood and operated in an effective, efficient and satisfactory way? Therefore we study dialogue properties that are related to usability. Since we want a formal characterisation of these properties, most of the thesis is concerned with formal models of dialogue.

A dialogue is a coherent exchange of utterances. The coherence of an utterance with respect to the dialogue context depends on its form, on its content, the information conveyed by the utterance, and on its function in relation to the rest of the dialogue. How can we combine the various aspects of utterances in one framework? Chapter 2 describes general models of *coordination*. It gives a theoretical background to the process of communication. The important notions of context and common ground are explained. A dialogue can be seen as a combination of joint actions that are coordinated at various linguistic levels.

In chapter 3 we deal with *inquiry*: cooperative information exchange. We conceive of information as being structured by issues: questions that are currently under discussion. Inquiry dialogues can be modelled as a constant process of raising and resolving issues. The main contribution of this chapter is a formal characterisation of the notion of relevance. Roughly, an utterance is relevant in context, when the information conveyed by it resolves one of the current issues. Other coherence constraints that can be defined are consistency, informativeness and licensing: not being over-informative. The chapter also provides an account of presuppositions.

Dialogue participants engage in dialogue for a reason: they want to achieve some task. The dialogue properties of effectiveness and efficiency have to do with the underlying task. Part of the function of an utterance is to make a contribution towards completing the task. Chapter 4 presents a task model for *transaction* dialogues. A transaction is the result of a negotiation process which proceeds in a number of phases: open, exchange information, exchange proposals, confirm and close. Each of these phases in a negotiation can be explained in terms of a so called negotiation space. We make use of a logic that specifies agents in terms of their beliefs, preferences and commitments. Based on issues we define a notion of awareness.

Chapter 5 deals with the interaction process itself. The crucial notion is that of *coherence*. Some utterances are initiatives; they require an appropriate response. For example, a question must be followed by an answer. Such interaction patterns can be described by the rules of a dialogue game. The task and interaction related functions of an utterance are related. Dialogue games are mutually known recipes for joint action, available to every competent speaker.

We have not fully evaluated our proposals. The style of the thesis is rather argumentative. Nevertheless, some of our ideas have been implemented and tested in small-scale projects. The design, imlementation and evaluation results of three particular dialogue system *applications* are discussed as extended examples in chapter 6.

The thesis ends with *conclusions* in chapter 7. We highlight again two main themes that run through the various chapters. The first concerns the observation that dialogue games may function as recipes for joint action. The second concerns usability modelling.

Originally, there were three related research issues that motivated our inquiry into dialogue models:

1. to formulate principles for the design of usable dialogue systems, and therefore
2. to investigate dialogue properties that influence the usability of dialogue systems, in terms of which such dialogue design principles can be formulated, and therefore
3. to investigate formal models of the semantics and pragmatics of dialogue, in terms of which a formal characterisation of such dialogue properties can be expressed.

This turned out to be too ambitious. Apart from some recommendations in chapter 1, we have formulated no principles for dialogue design. The characterisation of usability related dialogue properties is at best incomplete. Nevertheless, usability modelling remains one of the main motivations behind the dialogue models described in this thesis. It helps to focus and challenge existing theories.

Methodological Note

It is possible to distinguish roughly three scientific approaches in computational linguistics. Each requires a different kind of justification.

First, the *logical approach* is that of the philosopher or mathematician. A theory is judged by its ability to describe and explain the structure of a certain concept or phenomenon. An example of this type of work is the PhD Thesis by Gerbrandy (1999), which advances the mathematical theory of common knowledge. The logician requires proofs to verify a theory.

Second, the *empirical approach* is used both in psychology and corpus linguistics. In experimental psychology, researchers conduct controlled experiments. Based on the interpretation of the data they can derive conclusions. In general one should be careful with generalising specific results. An example of this line of work is the research of Oviatt into the effect of multi-modal interaction (Oviatt and Cohen 1990; Oviatt 1995). In corpus linguistics, a theory is judged on the way it explains and captures the data in a corpus. An example is the PhD Thesis of Rats (1996) that describes utterance patterns in a corpus of information dialogues in terms of a particular theory of topic and comment. Ultimately, empirical research like this is the only way to justify theories of linguistic competence.

Third, the *engineering approach* is widely used in computer science and engineering. The research should lead to a working prototype, a development tool or a design methodology. The justification for choosing a particular framework is whether it works or not, and how much effort it takes to get it to work. If the engineer is satisfied with the design method, it can be recommended to other people. The fact that it works in one situation is reason to suspect that it might work in other similar situations too. But again, it is difficult to generalise results.

Obviously, combinations of the three approaches are possible and useful. Engineers build a system, which is then empirically validated. Or logicians develop a theory which is then used in the formal specification and verification of a practical system. Much research in software engineering and human-computer interaction is hybrid in this way. In the field of dialogue systems, an example of a fruitful combination of different approaches is Jönsson's (1993) PhD work. Based on empirical investigation of simulated human-computer dialogue, Jönsson proposes a simple and clear dialogue model, which is subsequently implemented and verified.

This thesis is also a hybrid one. It is difficult to choose a methodological position, because dialogue system design does involve all of these aspects: design methodology, formal specification and empirical validation. Unfortunately, the evidence collected in this thesis is mostly second-hand: reports from the literature enlivened with qualitative results of projects at our group in Twente. In particular the claims related to usability and relevance do need further formal, empirical or practical justification. But although the thesis may be lacking in this respect, we hope make up for it by its diversity. To our knowledge, this is one of the first works to connect formal theories of dialogue with the software engineering aspects of constructing practical systems.

Having said that, we would like to raise some self criticism of the scientific method which is often implicitly used by researchers in theoretical linguistics, and which is also used in chapter 3. Constructing a theory from a small number of examples, is inappropriate for research into dialogue phenomena. Naturally, an example can serve as counter-evidence; one instance is enough to falsify a certain generalisation. Examples can be used to motivate and explain a certain concept. That is the function of most examples in this thesis. But examples can not replace empirical data or corpus research. In particular, examples like the standard presupposition triggers (see chapter 3) can never fully justify a theory. One could say that the 'theoretical machinery' should not be more heavy than the phenomena it is trying to explain. A theory can of course be justified by other means. By philosophical considerations, mathematical rigour, practical applications or computational efficiency. But the best justification would lie in studying the whole picture, not some isolated examples.

That means that formal semantics, now that it incorporates more and more aspects of pragmatics, should no longer neglect social and cultural aspects of language use. For example, insights from conversation analysis, anthropology and psycholinguistics on the process of turn taking (Sacks et al. 1974), grounding (Clark and Schaefer 1989) and politeness (Brown and Levinson 1987) should be taken into account. There is an obvious link between theories of intonation and psychological theories of attention. Also the influence of the underlying activity or task on the dialogue can no longer be neglected (Ginzburg 1995; Allwood 1995). We hope that this thesis may stimulate further multi-disciplinary research into dialogue.

Chapter 1

Usability

This chapter forms the introduction to the thesis. In the first half of the chapter we give examples of dialogue systems, introduce the dialogue types of inquiry and transaction, and motivate the use of natural language interfaces in comparison with other interface types. Through the example of the SCHISMA system for theatre information and booking we introduce usability as the most basic notion in dialogue system design. This notion is then further developed in the second half of the chapter. In particular, it is argued that the following dialogue properties are related to the usability of dialogue systems: effectiveness, efficiency and a number of properties that are related to coherence. We argue that these can be affected by the functionality of the system and by the transparency of the system prompts.

1.1 Introduction

This thesis is called *Dialogue Models for Inquiry and Transaction*. It is concerned with the design of usable dialogue systems for inquiry and transaction. A dialogue system is a computer system that is able to engage in an interactive dialogue with a human user about a particular topic. Usually it is designed to help the user of the system to perform her[1] task. Unlike most interactive computer systems, dialogue systems use natural language of some sort. Metaphorically, all interaction can be seen as a kind of dialogue. The interface of a computer system can also be said to establish a dialogue with the user. In this thesis however, we restrict ourselves to verbal interaction that resembles human-human interaction. What are valuable design principles for such systems? Is there a theory of dialogue that could substantiate those principles?

A dialogue can be characterised as a coherent exchange of utterances by the dialogue participants. In this thesis we will take the utterance as the basic descriptive unit. An utterance is a combination of phrases, which again consist of words. A number of consecutive utterances of the same speaker is called a turn. A group of utterances that handles the same topic of conversation, is called a dialogue segment. Part of this thesis concerns theories that describe the structure of dialogue. The theory should tell us what combination of words, phrases, utterances and segments constitutes a coherent dialogue. Obviously, such a theory helps in constructing dialogue systems that are usable in practice.

[1]With respect to the gender of pronouns we use the following convention: the speaker or initiator of a dialogue is a 'she'; the hearer or responder is a 'he'. The user initiates the dialogue and will thus be referred to as 'she'. A system or agent is referred to by 'it', except when it has a name.

There are many types of dialogue systems. Language can be used in many environments in many interaction styles. Spoken dialogue systems use speech as their input and output modality. Both the input to the system, and the output from the system are coded in speech. Spoken dialogue systems can therefore be used over the public telephone. Older dialogue systems typically only use the keyboard and a computer terminal, and thus have linear text as their input and output modality. Many current systems on the Internet use mouse and keyboard input in combination with the buttons and menus of a hyper text environment. The output is typically given in the form of tables, text pages or windows style 'dialogue boxes'. Sometimes, such web based services are combined with spoken output, or even combined with separate but related telephone based systems. Thus, different combinations of input and output modalities are possible. Multi-modal systems that combine speech, keyboard input and graphical displays like maps and diagrams have become more and more common. Particular successful examples of multi-modal systems are interactive maps for tourist information. These allow the user to ask information about places on the map, simply by simultaneously pointing and asking. Another example of multi-modal dialogue arises when interactive dialogue systems are embedded as an autonomous agent in a virtual environment. A virtual world presents to the user a simulated three-dimensional space, in which the user can walk around, observe objects and interact with virtual agents. Interactive dialogue agents can be equipped with an animated 'talking head'. For such systems a very natural combination of speech and non-verbal communication becomes possible. Issues like the role of eye contact, facial expressions and gestures in dialogue, raise interesting new research issues. This thesis concentrates on existing spoken and keyboard-based dialogue systems. However, considering a more natural face-to-face setting, provides useful theoretical insights.

Another way to characterise dialogue systems is by the application domain. For instance, automatic tutoring systems may help the user to learn a foreign language. Automatic instruction manuals help the user apply or repair a particular machine. There are dialogue systems for scheduling international meetings, for controlling a robot and for managing e-mail. The dialogue systems in this thesis are mainly concerned with *inquiry* and *transaction*. Dialogue systems for inquiry are basically enhanced database systems[2]. They translate the user's information request into an appropriate query, search the database and communicate the result to the user. However, many applications involve more than the exchange of information. Usually something is done with the information. Ticket reservation, product ordering, financial account managing and collect calls, are examples of the user making a transaction with the system. A concluded transaction forms an agreement about some future course of action. Transaction dialogues therefore contain more complicated types of interaction: negotiation, deliberation, advice giving and establishing agreement.

1.1.1 Example Dialogue Systems

Throughout this thesis a number of concrete examples of dialogue systems are discussed. Here we briefly introduce them.

PADIS is an automatic telephone directory system developed at Philips, Aachen (Kellner et al. 1996; Bouwman 1998; Bouwman and Hulstijn 1998). Based on speech recognition

[2]The term *inquiry* covers both (i) requests for information and (ii) systematic investigations, often of a matter of public interest (Webster Inc. 1983).

of the first name, last name, gender or title of an employee, the system is able to retrieve telephone numbers, room numbers or e-mail addresses from the telephone directory of a middle-sized company. In addition, the system can make a direct connection to an employee. At first sight PADIS looks like a simple inquiry system. The user has a request for information; the system responds with the required information or by a direct connection. However, establishing a direct connection has some effects that go beyond mere information exchange. A telephone call disturbs people and invades their privacy. Because of this, users want to be absolutely sure they get connected to the right person. Therefore, the application already has some characteristics of a transaction. PADIS and the underlying framework for building spoken dialogue systems, HDDL (Aust and Oerder 1995), are used to illustrate typical problems of spoken dialogue system design, and their solutions. Also several other inquiry systems are mentioned. Most of these are used for public transport information since this has been the most active research area.

STEMSTEM (Vote by Voice) is a simple spoken dialogue system, developed by students at the university of Twente (Ebbers et al. 1999). It allows visually impaired users to cast their votes, by using their voice. The application of speech recognition technology has potentially huge benefits for visually impaired or otherwise disabled citizens that can't use current machines. In effect the voting task consists of the selection of a candidate name from a list of candidates, which is structured into separate lists corresponding to political parties. Because voting has legal consequences and because it is such a personal matter, the system is not allowed any room for error. The Vote by Voice system is used to illustrate the issues that have to do with confirmation of transactions. Because of the specific user group, the project is also used as an example of user-centred development.

SCHISMA is a natural language dialogue system for theatre information and booking, developed at the university of Twente (van der Hoeven et al. 1995; Andernach 1996; Hulstijn et al. 1996). Throughout this thesis the prototype of this application is used as an example of a dialogue system for inquiry and transaction. The core inquiry system consists of a database with the performance schedule of the local theatre: what group or artist plays which production at what time? Users can ask questions about performances. The SCHISMA system also makes ticket reservations. From the perspective of the system a reservation transaction requires a unique performance, a number of tickets and information on discount, preferred seats, as well as some identification of the user. The user however expects detailed information on the price and seats before giving his consent. When the system is equipped with preconceptions of the user's desires and goals it will cooperate and help the user to make a decision. In a way a reservation transaction is a promise, or a commitment of the user to pick up the tickets and pay the agreed price. The commitment is made on the assumption that the provided information is correct. Modelling transactions, user goals and user behaviour with respect to alternatives, decisions and commitments plays a large role in this thesis.

The VMC (Virtual Music Centre) is an example of a virtual environment in which dialogue systems can be embedded. Based on the blueprints of the actual music centre in Enschede, a virtual theatre environment was developed that is accessible over the Internet[3] (Nijholt et al. 1999; Nijholt 1999; Nijholt and Hulstijn to appear). The virtual theatre environment is a framework for trying out new ideas about communication and entertainment. Users may walk around in the virtual space, visit the auditory and view the stage. They can observe and manipulate objects in the environment and interact with

[3] `http://parlevink.cs.utwente.nl/Projects/Muziekcentrum/`

+ **natural**: no need to learn controls
+ **concepts**: translate user request to system representation
+ **speech**: accessibility (physically, socially)
+ **context**: disambiguation, correction, conciseness

Figure 1.1: Advantages of natural language interfaces

virtual agents. The SCHISMA system is embedded as an agent called Karin in the VMC; there is also a notice board and a virtual piano that can be played. In the future, virtual performances will be staged. An interesting aspect of the virtual theatre environment is that it attracts leisure users. The virtual theatre invites users to browse through possibilities simply to be entertained. The task of entertainment is not very structured at all. This contrasts with the tasks of most interactive systems, which are highly structured. Normal usability criteria like efficiency loose their meaning in such a leisure setting; other criteria, like entertainment value or *presence*, the subjective feeling of immersion, become more important. In particular, we discuss the development of an animated face that embodies the reservation agent. Other research triggered by the VMC is concerned with a specification formalism that can deal with multi-modal and situated interaction (van Schooten 1999; van Schooten et al. 1999).

1.1.2 Why Natural Language?

Natural language dialogue interfaces have several advantages over other types of interface. For the specific case of natural language interfaces to databases, it is possible to list a number of distinctive differences between natural language interfaces and command-based or graphical interfaces (Androutsopoulos et al. 1995). The advantages are summarised in figure 1.1. Here they are.

Natural language is *natural* to the user; there is no need to learn an austere command language or to master graphical interface controls. Natural language commands are easy to remember, and in principle make it easier to operate a service.

Natural language expressions remain close to the user's *conceptualisation* of the task. Terminology used by experts in the original database may not be the most natural for the user to express queries in. A natural language interface can provide a translation of the user's query into the database's representation language. The database of a building society for example, might contain information on 'immovables' and 'lots'; a non-expert user would probably prefer the words 'houses' and 'pieces of land'.

When using *speech* the interaction becomes even more natural. Everybody knows how to speak. Major computing companies see this *accessibility* as the main advantage of speech technology. In principle speech interfaces would allow illiterates to use computers too. Speech interfaces can be an advantage in countries with languages like Chinese or Thai that have an orthographic representation which is difficult to approximate on a keyboard. Also for visually handicapped people speech interfaces are an obvious advantage; the VotebyVoice application discussed in chapter 6 is an example of this. Speech makes it possible to use the telephone, so accessibility is no longer constrained by the presence of a computer. An advantage of a spoken telephone service over DTMF or touch-tone,

– **ambiguity**: ambiguous at all levels of processing
– **flexibility**: not immediately clear what can and cannot be done
– **costs**: need expensive linguistic resources, skilled developers and motivated subjects for corpus collection and evaluation experiments
– **portability**: need redevelopment for each new application
– **adaptability**: local changes often have a global effect
– **maintainability**: vulnerable to external changes

Figure 1.2: Disadvantages of natural language interfaces

where you have to dial digits, is that a hands-free microphone set-up can be used. This allows car drivers for example to use the service without taking their hands of the wheel. A more obvious advantage of speech over DTMF is its efficiency. We do not have to listen to a list of films and remember the number we have to dial in order to get tickets for that film; we can simple state the title of the film, or the genre.

The use of *interactive dialogue*, finally, makes it possible to correct misunderstandings. The circumstances of the conversation and the previous utterances can play a role in interpretation: the dialogue context. The dialogue context can in principle be used, not only for disambiguation, but also to make information access more easy to use and more effective. Users may continue with queries about the previous topic, using 'telegram style' utterances with pronouns and other context-dependent expressions. If both the system and the user can rely on the context, dialogue interaction may turn out to be very efficient, even for experienced users. In principle, the information that is present in the context need not be repeated.

Given these advantages and given the great effort that has gone into developing natural language interfaces, it is strange that so little of that research has been applied in practice. Apparently, natural language interfaces also have disadvantages (figure 1.2). Natural language expressions are *ambiguous*. A single expression can mean several things. In many cases ambiguity is only apparent. Given the context the intended meaning should be clear. But despite all the research into robust parsing (van Noord 1997) or under-specified meaning representations (Pinkal 1995), dialogue system designers still have difficulty in designing algorithms that deal with natural language in unconstrained contexts. For this reason, most successful dialogue systems to date have a limited application domain.

Natural language interaction is *flexible*. But if anything goes, nothing may work. Freedom to choose or to take the initiative is an advantage only when it is clear what the possible alternatives are. Flexibility turns into a disadvantage, when the behaviour of the system is not transparent, i.e. when it is not clear what the boundaries of the system's capabilities are (Oviatt 1995). And, unlike a graphical user interface with menus that hide or shade an impossible option, a natural language interface has no direct way of indicating what it can and cannot do. What can be done is to guide the user by means of the system prompts. Wording, terminology and intonation may help to suggest alternatives and recommended options. This is one of the major challenges in the design of a usable dialogue system.

According to Androutsopoulos et al. (1995) one of the main reasons that natural language interfaces have never caught on, is the neglect of *software engineering* aspects in the development of natural language interfaces. Very often research prototypes are build by academic institutions, to show that some linguistic theory has been covered. Such systems typically focus on the linguistic aspects and neglect the maintenance, portability, re-usability and reliability of the resources that are developed[4]. But natural language interfaces do in fact need extensive, complex resources such as a knowledge base, a lexicon, a grammar and a dialogue model. Speech recognition, parsing or dialogue management have their own problems and peculiarities, so building these linguistic resources is work for an interdisciplinary team of specialists. This makes the development of a natural language interface for a given application very slow and very expensive, at least compared to the development of a graphical interface. Moreover, linguistic resources are heavily interdependent; adding a simple concept to the knowledge base affects the lexicon, the grammar and the dialogue model as well. Therefore, most of these resources have to be redeveloped for each new application domain: natural language interfaces have low *portability*. But more seriously, every time something changes in the real word, the whole system needs to be adjusted too. This makes dialogue systems vulnerable to external changes: they have low *adaptability*. The reliance on specialist resources makes natural language interfaces difficult to maintain: low *maintainability*.

1.1.3 Usability

Can we validate claims about the advantages and disadvantages of natural language systems? It seems that most advantages have some negative side effects as well. Is there a way to compare the two, for a given application?

To sum up the advantages, one could say that the use of a natural language interface, in comparison with a command-based or graphical user interface, potentially increases the usability of a system. Usability is an abstract concept, that includes both usefulness for some given purpose or task, and ease of use. Several definitions of usability exist in the literature, but most include the following aspects: *effectiveness* and *efficiency* with respect to the task, as well as *learnability*, *memorability* and *operability* of the system's controls, and an overall judgement of *user satisfaction* (ISO 9126 1991). User satisfaction is difficult to influence by a design. It depends on all aspects of a system, including aspects of personal taste such as the preference for a female rather than a male voice. The other usability aspects have to do with the way the functionality can be controlled by the user (Nielsen 1993). For example, if users can express a query in their own words, this will increase learnability and memorability: it is easier to learn how to ask queries, and you have to remember less.

However, the advantages of natural language dialogue only come out, when the disadvantages have been dealt with. Flexibility and freedom in the interface turn into a disadvantage, when the user is not guided. This may lead to insecurity and irritation, as well as misunderstanding. One of the main indicators for low usability is the number and impact of misunderstandings. How effective, efficient, satisfactory or learnable a system may be, when faced with a misunderstanding the usability rate will soon drop. The reason is simple: a misunderstanding prevents a user from accomplishing his or her task, thus damaging effectiveness. To resolve the misunderstanding the dialogue will

[4] In the last five years things have changed in this respect. Witness for example the tools for typed feature structures at DFKI, the LREC conferences on resources and evaluation, and the various special interests groups that stimulate re-usability of resources.

take much longer, damaging efficiency. It is unpleasant to be misunderstood and it is extremely hard to learn and remember how to deal with the types of misunderstanding that occur in human-computer interaction, because the computer makes unexpected mistakes.

A misunderstanding typically arises when the system's behaviour does not correspond to the user's expectations. We assume that the user has a *mental model* of the system's behaviour, capabilities and competence. The mental model is based on conventions, on the current interaction and on previous experience with systems or humans performing a similar task. The mental model is adjusted continuously during interaction. A dialogue system designer can influence the user's mental model in two ways: before the system is used and while the system is used. Before the user starts to interact, we may use marketing techniques and an interface design that suggests a certain interaction metaphor. An example is the form filling metaphor that is used in some multi-modal systems (Oviatt 1995). In this way we can re-apply existing conventions to guide the user. The only other means we have of influencing the user's mental model while using the system, is by a careful design of the system's utterances and responses. Not only the content of the interaction matters, what information is exchanged and how, but also the underlying task and the form of interaction itself. In general, a design should be *transparent*: it should exhibit the same behaviour in similar circumstances and make clear what is going on and why. Providing feedback is the most important means for that.

Designing a good dialogue system is no different from designing a door handle. There are systematic constraints on door handles – imposed by human biology, by the function of the door and by the way doors are conventionally used – that limit the possible shapes and sizes. Yet there is ample room for variation. We may vary the colour and material. Similar considerations apply to dialogue systems. There are systematic constraints on dialogue – imposed by cognitive processing limits, by the underlying tasks and by the way language is conventionally used – that limit the space of possible dialogue designs. Yet again, there is room for choice. Should we take a male or female voice? Do we prefer a 'cheesy' or a terse system? Much of these design issues depend on the application or on the target users, but some are simply a matter of taste.

How can we avoid misunderstanding? What are these systematic constraints on dialogue interaction? How can we model them? How do they shape dialogue design? These and similar questions are addressed in this thesis. But before we discuss these questions, we start with a concrete example of the kinds of design errors that lead to misunderstanding.

1.2 A Motivating Example

In the SCHISMA project we investigate different possible architectures and techniques for building natural language dialogue systems (van der Hoeven et al. 1995; op den Akker et al. 1995; Andernach 1996; Hulstijn et al. 1996). It serves as a motivation and testbed for various approaches and theories. SCHISMA is both the name of the project and the name of our prototype system: a dialogue system for theatre information and booking. The acronym is derived from the Dutch *Sch*ouwburg *I*nformatie *S*ysteem. SCHISMA started as a joint research project of KPN Research and the Parlevink group of the University of Twente. Our local theatre, the *Twentse Schouwburg*, provides the information to fill the database with. We have chosen the theatre domain because of its richness. The kind of background knowledge needed in a system about theatre is more complex and therefore more interesting than knowledge about, say, public transport. For instance the model of genres, like classical music, rock, music theatre, ballet and cabaret is complicated. For practical reasons we assume a fixed hierarchy of genres. Yet each year when a database with a new theatre schedule is adapted, many descriptions used by the local theatre to classify performances do not fit the existing structure of the hierarchy. The diversity of performance types is simply too much. We have chosen the application of ticket reservation because it is a form of transaction. This too makes the interaction more complex. As we said, a transaction involves commitments from both system and user. Roughly, a user is committed to pick up the tickets and pay the price. The system is committed to present all relevant information for the user to make a fair choice.

Here is a sample dialogue taken from evaluation sessions with one of the SCHISMA prototypes (Lie et al. 1997). As you can see, it works, but it is far from perfect. Even if all the right choices are made in the design, much of the performance hinges on details: lexical coverage, the way the prompts are formulated, the way dates are referred to and generally the way information is presented. This example illustrates that how much theory one develops for building dialogue systems, actually building a good one remains a craft[5].

(1) S1: Welkom bij het SCHISMA systeem. Waar kan ik u mee van dienst zijn?
Welcome to the SCHISMA system. How can I help you?

U1: kan ik bij u kaartjes krijgen voor vanavond?
can I get tickets from you for tonight?

S2: Momentje... Ik kijk even voor u in het programmaboekje. Er zijn de volgende voorstellingen op 26 maart 1997:
Moment please... I'll look it up for you in the programme. We have the following performances on March 26, 1997:

VOORSTELLING	ARTIEST	DATUM	GENRE
Mozart	Fred Delfgaauw	26 maart 1997	muziektheater
Romantiek	Victor Piano Quartet	26 maart 1997	kamermuziek

Voor welke voorstelling wilt u reserveren?
For which performance would you like to reserve?

U2: ik wil naar het piano quartet
I'd like to go to the piano quartet

S3: Momentje... Ik kijk even voor u in het programmaboekje. Er zijn geen pianisten voorstellingen van Deep River Quartet.
Moment please... I'll look it up for you in the programme. There are no performances by Deep River Quartet.

U3: maar ik bedoel ook hwt Victor Piano Quartet
but I mean thw Victor Piano Quartet

S4: Momentje... Ik kijk even voor u in het programmaboekje. Victor Piano Quartet speelt alleen de voorstelling "Romantiek"op 26 maart 1997.
Moment please... I'll look it up for you in the programme. Victor Piano Quartet only plays the performance "Romantiek"on March 26 1997.

U4: daar wil ik naar toe
there I'd like to go

[5]The (lack of) punctuation is taken over from the log-files. For instance, questions are not always marked with a question mark.

S5: Ik begrijp niet wat u bedoelt.
I don't understand what you mean.

U5: ik wil reserveren voor die voorstelling
I'd like to make a reservation for that performance

S6: U wilt dus reserveren voor de voorstelling "Romantiek". De voorstelling "Romantiek"speelt op 26 maart 1997. Ik heb de volgende prijsinformatie over de voorstelling "Romantiek":
So you would like to make a reservation for the performance "Romantiek". The performance "Romantiek"is on at March 26, 1997 I have the following prices for the performance "Romantiek":
PRIJS f 27,50 — pas65 f 25,00 — cjp/jtk f 20,00
Hoeveel kaartjes wilt u reserveren?
How many tickets would you like to reserve?

U6: vief
fouve

S7: Ik begrijp niet welke hoeveelheid u bedoelt. Hoeveel kaartjes wilt u reserveren?
I don't understand what quantity you mean. How many tickets would you like to reserve?

U7: vier
four

S8: Welke kortingskaart heeft u?
What type of discount-card do you have?

U8: pas65
pensioner

S9: Op welke naam wilt u reserveren?
On what name would you like to make the reservation?

U9: gerrit
gerrit

S10: U wilt dus 4 kaartje(s) reserveren voor "Romantiek"op 26 maart 1997
op naam van gerrit met tarief: pas65?
So you would like to reserve 4 ticket(s) for "Romantiek"at march 26 1997
on the name of gerrit with discount: pensioner?

U10: ja, hoe duur waren de kaartjes zei u
yes, how expensive did you say the tickets were

S11: U heeft gereserveerd.
Reservation completed.

U11: maar hoe duur zijn nou de kaartjes
but how expensive are the tickets then

S12: Ik heb de volgende prijsinformatie over de voorstelling "Romantiek":
PRIJS f 27,50 — pas65 f 25,00 — cjp/jtk f 20,00
I have the following prices for the performance "Romantiek":
PRICE f 27,50 — pensioner f 25,00 — student f 20,00

U12: dank u
thank you

S13: Graag gedaan.
You're welcome.

The following dialogue phenomena are illustrated by the example.

It is a *mixed-initiative* system. Both system and user may take initiative and ask for information. The system opens in utterance S1 with a non-specific question: "Waar kan ik u mee van dienst zijn?" (How can I help you?). This type of opening is notorious in the dialogue community for causing problems (Fraser and Gilbert 1991). When meeting the system for the first time, users do not know what to expect or how to respond. In ticket reservation it is better to ask directly for the performance that has the user's preference. In this way so-called granny utterances can be avoided. A granny utterance contains many words that fall outside the common vocabulary of the application domain. For instance, "Ja, ik wilde graag mijn oma verrassen met haar verjaardag en ik vroeg me af is er ook ballet?" (Yes, I would like to surprise my granny at her birthday and I was wondering is there any ballet?).

In U1 the user asks for tickets for tonight. The request takes the form of an indirect question: "kan ik" (could I). In this version of SCHISMA we choose to deal with indirect questions by simply throwing away the 'superfluous' elements of the utterance. All utterances are rewritten into a kind of normal form, that takes the shape of an imperative (Lie et al. 1997). The normal form is supposed to trigger the same system response as the original user utterance would have. So a rewriting phase before the actual parse reduces the complexity of expressions that the parser has to deal with. On the other hand, important information may be lost in the rewriting phase.

The response of the system, S2, is simply a table of the performances that were found in the database. So the system does not directly answer the question; a proper answer would have taken the form "Ja dat kan, er zijn de volgende voorstellingen." (Yes, that is possible, we have the following performances). The information from the database is followed by a wh-question from the system. So now the system has taken initiative.

Utterance U4 cannot be rewritten or parsed by the system. Yet, it is the sort of utterance that one would expect in a Dutch reservation dialogue. The use of a corpus of dialogues collected in a so-called Wizard of Oz experiment, to test the coverage of lexicon and grammar, can greatly improve performance on such domain-dependent expressions.

Exchange U2-S4 illustrates the importance of the preprocessing of utterances. The theatre domain of SCHISMA contains many strange strings as titles or names. Names have to be recognised before parsing. However, users often do not use the full name to refer to a title or artist. Therefore our preprocessing module contains a partial-match function. In this case the substring "Quartet" triggers "Deep River Quartet". This is wrong of course, especially since the system has just mentioned "Victor Piano Quartet" itself! It shows that the system does not have a good idea of what has been said.

The same problem is illustrated by the utterances that form S6: the name of the performance is repeated three times. First as feedback to the user on the selected performance, then as feedback related to the date, and finally as feedback related to the price. The system must have an account of what has been said: the dialogue context. Not only to avoid irrelevant repetitions but also to find out what some of the referring expressions actually refer to. So in this case "the piano quartet" refers back to "Victor Piano Quartet". To complicate things, users do not perceive reference resolution as difficult. So even when talking to a computer, replies to prompts like "I did not understand you; could you please rephrase your question?", tend to be shortened and more concise. In other words, when humans want to be clear they leave out words. For natural language understanding such condensed, elliptical utterances are usually more difficult to process than full sentences.

This further illustrates the importance of a good model of the dialogue context. All the information that the user thought could be left out, should in principle be available from the context. The user should be able to control the system by keywords or telegram-style commands. The system must be able to fill in the missing blanks.

Utterance U6 shows a clear typing error. It should have been detected and corrected by a spelling correction program, running as part of the preprocessing. System utterance S8 is not fully cooperative: it asks for a discount card, but the user cannot be expected to know what types of discount are valid. A cooperative system would first mention the alternatives in case of such an alternative question.

Exchange S10–S12 shows an error in the transaction behaviour of the system. The system asks for confirmation. The user agrees in principle, but wants to know, in addition, how expensive the transaction will be. Intuitively, that indicates a conditional agreement: the user agrees provided it is not too expensive. However, after S11 the transaction is completed. Now suppose that the user didn't realize earlier that four times the pensioners price makes Dfl 100,- and now wants to withdraw. This dialogue system does not give any chance for that. The topic is closed. The user would feel 'tricked'.

Example (1) introduced the functionality of a dialogue system and some of its components. It shows that such a first prototype can and must be improved. In general it is impossible to get it right the first time around. There must be room for making mistakes in the development process. In the following section we discuss the various components of a dialogue system. After that we turn to specific development methods.

1.3 Architecture

Dialogue systems can be implemented in many ways. Figure 1.3 depicts the typical components of a spoken dialogue system. In this pipe-line architecture, each component captures a basic processing step: speech processing, preprocessing, parsing, interpretation and dialogue management, response generation and text-to-speech conversion. We discuss the proposed architecture of a spoken version of the SCHISMA system (Hulstijn and van Hessen 1998). With the exception of the first and the last modules, the same architecture is used in the keyboard-based SCHISMA system (Hulstijn et al. 1996). An architecture with similar components appears in almost any dialogue system, e.g. (Veldhuijzen van Zanten 1996; Scha 1983). This is no surprise since the basic interpretation and generation functions are universal. In human language production one distinguishes three levels: deciding what to say (planning), how to say it (generation) and actually saying it (production)(Levelt 1989). If we add the corresponding modules for processing the meaning of what was said (understanding), how it was said (parsing) and what was actually said (recognition), we get the presented architecture, given that both the understanding and planning functions are carried out by the dialogue manager.

Figure 1.3 also represents a particular philosophy to the design of dialogue systems. On the left we find the actual implementations of the modules. The middle depicts the resources that specify the behaviour of the implemented modules. On the right we see the domain and interaction models that guide the design and play a role in verification and validation[6]. Also a corpus can be seen as a model of typical user behaviour.

[6]An arrow depicts 'data flow', a solid line 'specification' and a dashed line indicates a 'captures' relation.

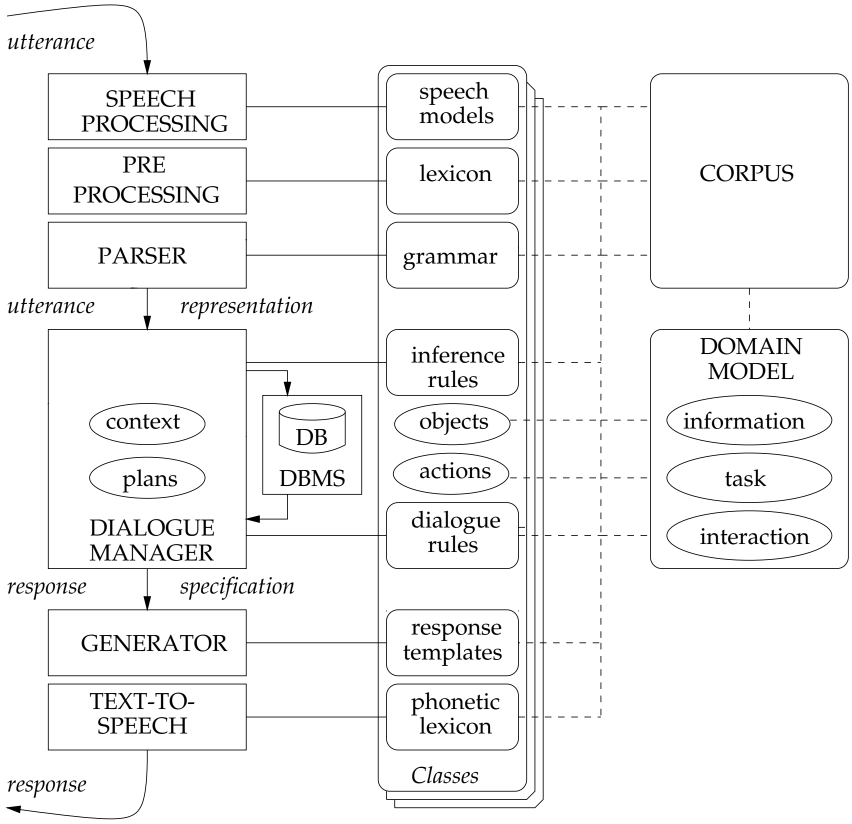


Figure 1.3: Dialogue system components

In this architecture, each module corresponds to a basic processing step. But modules do not have to be scheduled one after the other. In a so called *black-board architecture* more interaction between modules is possible (Engelmore and Morgan 1988). A black-board architecture uses a central data-structure, called the black-board, for each module to take its input from and to write its results on. Such a non-linear configuration is especially useful when the behaviour of modules is mutually dependent, for instance when the speech recognition may benefit from pragmatic expectations of the dialogue manager. For a multi-modal system which combines several input and output channels, a non-linear architecture is compulsory to integrate the different input sources, and synchronise the output. More radical than a black-board architecture is a completely distributed architecture. Each module is pictured as an autonomous entity, which chooses to cooperate with the other modules according to certain processing protocols. In general, the simpler types of architecture use the implicit cooperation protocols of the underlying operating system; more complex distributed architectures use special purpose coordination protocols.

Apart from the decomposition into processing steps, it is usually a good idea to cluster parts of resources that have to do with the same domain object. For instance, a *date* object needs specific lexical entries – "Wednesday", grammar constructions – "a month from now", inference rules – "if unknown, the date is today by default" and utterance templates "Which day would you like?". Ideally, this clustering leads to a library of re-usable classes that specify objects in the domain. Obviously, not all aspects of a system belong to a re-usable class. Good candidates for domain objects that will generate a re-usable class can be found by studying example dialogues for re-occurring elements.

1.3.1 Speech Recognition and Preprocessing

For spoken dialogue systems, each user utterance must first be recognised by the speech recognition module. Input signals in other modalities, like pointing gestures, also have to be processed in this stage. Also keyboard-based systems like SCHISMA need preprocessing. In a process called part-of-speech tagging, numbers and named entities like artist names, titles, and dates are detected. For example, 'Twelfth Night' should be interpreted as a Shakespeare play and not as a date: the twelfth of the current month.

Now, just like the behaviour of a parser is determined by a grammar, the behaviour of the speech recognition modules is determined by two speech models. The first is the *word model*. It provides a mapping of sound patterns onto possible phonemes and therefore onto possible words. A combination of different possible phonemes can be represented as a word-graph: a directed graph in which labelled arcs represent possible phonemes. A path through the graph represents a possible sequence of phonemes. To filter the most probable sequence from these possible sequences, we need language dependent knowledge about what phonemes are usually followed by what other phonemes. This resource is called the *language model*. The language model is typically trained and tested on a domain dependent corpus. All forms of preprocessing are typically dependent on the contents of a domain dependent *lexicon*: a list of words and phrases with their expected meanings.

1.3.2 Parsing

After preprocessing, the utterance is parsed and interpreted. The function of a parser is to analyse a user utterance, segment it into meaningful chunks and produce a meaning representation. The behaviour of a parser is determined by a grammar. A grammar is a specification of linguistic observations about the relation between words, phrase structure and meaning, usually in the form of grammar rules. A parser is a program that uses these linguistic insights to convert actual utterances into meaning representations, which can then be applied by the dialogue manager.

1.3.3 Dialogue Management

Utterance representations are interpreted by the dialogue manager relative to a data-structure that represents the dialogue context. The interpretation process is subject to resources with domain specific inference rules. In a context representation the dialogue manager monitors the current status of the dialogue and the transaction. The plan stack may contain intended future actions, given the current goal of the system. It is the function of the dialogue manager to decide on a response action and to start planning future actions, on the basis of the utterance representation, the context and the current plan. For example, a reservation involves a number of information items that need to be agreed: the performance itself, the number of tickets and the total costs. However, for each of these items, the system depends on the user to cooperate. At the beginning of a reservation the system must have a rough plan of the steps that need to be accomplished, but the details can only be filled in as the dialogue progresses.

A system action is usually nothing but a combination of database manipulation with dialogue acts. A dialogue act is the action of making an utterance with a particular function in the dialogue. Dialogue acts are characterised by their content, the information they are supposed to convey, and a communicative function. Dialogue acts are used both in the interpretation of user utterances and to motivate the generation of system utterances. Take for example the utterance "How many tickets would you like for Twelfth Night?". Its communicative function is to ask for a number and to initiate a particular response. It conveys the information that there is a performance called Twelfth Night and that the asker is interested in the number of tickets the addressee wants.

The decisions of the dialogue manager are prescribed by a set of dialogue rules. Dialogue rules are of the form *if Condition, then Action*. Conditions refer to a certain state of the dialogue, stored in the context representation, but may also refer to semantic, syntactic and phonetic features of the user utterance, or to the current plan of the system. The rules are ordered by specificity. Actions with the most specific conditions are tried first; last is a default that can always be applied.

Each utterance, both of system and user, is characterised by three aspects: *form, content* and *function*. Form encodes phonetic features, words, word order and syntactic features. Content encodes the information conveyed by the utterance. A typical representation structure is a list of *information items*. An information item is an attribute-value pair together with a status variable which indicates whether the value is unknown, known or confirmed. For SCHISMA typical attributes are features of a domain object, such as the performance title, artist or date. Function indicates the pragmatic role the utterance plays with respect to the task and the previous and future interaction steps.

1.3.4 Utterance generation and production

The specifications of the system's response actions are translated into actual utterances by an utterance generation module which uses a long list of utterance templates. For each communicative function a particular template can be selected, that specifies the word order of the system's response. A template contains slots or variables which can be substituted with the linguistic correlates of the information items that need to be conveyed.

Finally, for the spoken versions of the system, response utterances are pronounced by a text-to-speech system. For spoken systems the templates must be prosodically annotated with the right pauses and intonation contour. For example, a yes/no question has a rising intonation.

1.4 Iterative Development

Designing a usable dialogue system is considered an art. Current texts on dialogue system development are based on 'best practice' (Gibbon et al. 1998; Bernsen et al. 1998). There is no established theory of dialogue development that can be applied in the specification and evaluation of dialogue systems. Existing linguistic theories of dialogue are not directly applicable to a 'dialogue engineer'. Theories are difficult to learn, and difficult to apply in practice.

On the one hand, dialogue systems are special. Natural language technology requires special resources, such as a lexicon, a grammar, an interaction model or an extensive domain dependent knowledge base. So general modelling techniques like OMT or UML

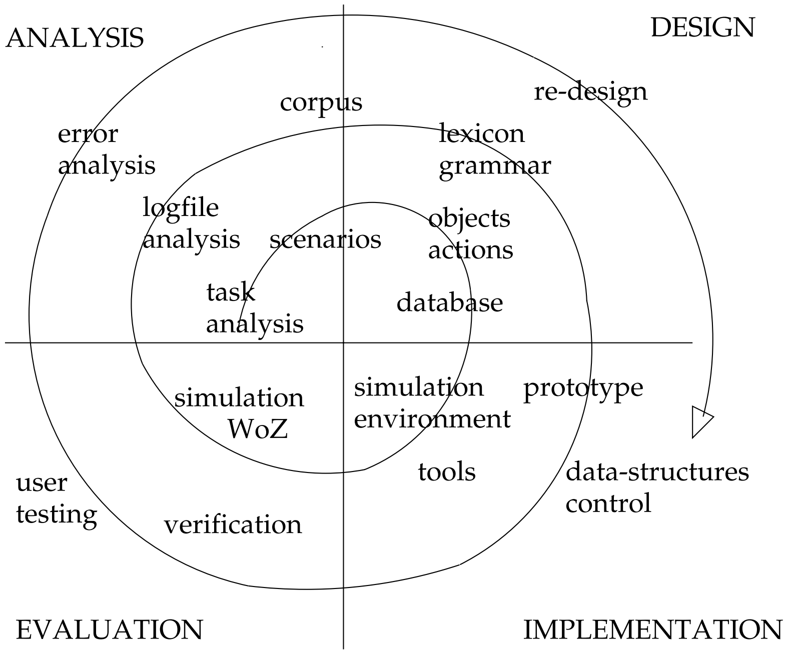


Figure 1.4: Iterative development cycle

(Rumbaugh et al. 1991; Fowler and Scott 1997) are not easily applicable. On the other hand, dialogue systems are just another example of complicated interactive software. Their success depends as much on general software quality factors like *functionality, usability, reliability, performance, maintainability* and *portability*, as on their linguistic coverage. But because of their special linguistic resources natural language interfaces are relatively complex, unstable and difficult to maintain or adapt (Androutsopoulos et al. 1995). The challenge is to formulate a dialogue system development method, that facilitates the design of a usable dialogue system and supports reliable evaluation. Because of the interactive nature and acclaimed advantages of natural language interfaces as being natural and easy-to-use, *usability* is the most important quality factor for a dialogue system. Can the system actually be learnt and used in an effective, efficient and satisfactory way?

Dialogue systems are interactive; their performance crucially depends on user behaviour. User behaviour is generally unpredictable. Because there are many types of users, different and often opposing requirements have to be met. Novice and expert users, business or leisure users and users from different social and cultural backgrounds have potentially diverging expectations of a system. Ideally, a consistent mixture of all expectations should be modelled in the system. Therefore, a development method that is both *iterative* and *user-centred* is recommended. The idea is to involve target users in the whole development process, not just in the testing phase (Nielsen 1993). Iterative development proceeds in cycles. In each cycle, the system is evaluated and improved. Early in development, mistakes are not yet very expensive. Simulations or early prototypes allow developers to experiment with radical new ideas. Feedback from user focus groups, surveys or user experiments, will detect bugs, design mistakes or conceptual errors.

Iterative design proceeds in cycles (figure 1.4). The complexity increases as the development spirals out. There are four types of activities that are carried out for each development cycle: *analysis*, *design*, *implementation* and *evaluation*. These activities are mutually dependent and can take place simultaneously.

1.4.1 Analysis

Development starts with a detailed domain and task analysis. The functionality of a system should reflect the tasks for which it is used. A task is essentially a generalisation of a pattern of actions that is common to a large number of social activities. For each task or activity type, like buying-and-selling or enquiring, there are conventions that regulate the behaviour of participants in a particular role and determine the meaning of words and phrases. The activity type also determines the meaning of artifacts, such as coins or tickets. Dialogue developers should be aware of these social, cultural and linguistic conventions. The function of task and domain analysis is precisely to bring out the conventions and expectations that target users will have.

Analysis may begin with *scenarios*, simple descriptions of the projected behaviour of dialogue participants. Scenarios should be based on observations of real life transactions, because users will inherit intuitions from previous experience. Now often the development of a system does not start from scratch. Usually there is already a system in place, or there is some human service or semi-formalised procedure that performs the same activity. In that case, the experiences of users and designers with the earlier system must be taken into account. Often, a dialogue system is merely an interface added onto an existing database system. In the SCHISMA case, this meant that we simply had to use the theatre data and price conventions of the local theatre.

Scenarios are applied in two ways: they serve as a basis for a more systematic domain and task analysis and they serve as a basis for the collection of a corpus.

Corpus based analysis

A corpus is a way of modelling the interactive behaviour of users on a particular task. If no appropriate corpus exists, a corpus can be collected in a simulation or 'Wizard of Oz' experiment (Fraser and Gilbert 1991). A human experimenter simulates the behaviour of the system, using an environment with semi-automated responses. A group of representative subjects is asked to enact several scenarios. Scenario descriptions should not trigger a particular way of responding. To this end, often diagrams and pictures are used. The interaction is recorded and logged. The recorded data is edited and annotated at different linguistic levels. Speech recognition modules are tested and trained on annotated speech data. Other low-level resources like lexicon and grammar can be automatically generated by machine learning techniques. Also descriptions of re-occurring interaction patterns, such as greeting exchanges, question-answer pairs and confirmation sequences, can be automatically derived from annotated data.

However, there is a risk to corpus based analysis. With each alteration of the system, it is likely that the behaviour of users will change too, which would make the corpus inadequate. Therefore corpus-based development can be rather conservative. Moreover, the set-up of the wizard, the database and the scenarios affect the nature of the corpus and therefore already determine a large part of the system. Evaluation of the system against a corpus thus runs the risk of merely confirming preconceived choices. Therefore corpus-based evaluation must be extended with other methods to get early feedback on projected functionality, like user surveys and questionnaires, or structured discussions with user-focus groups. These methods can also be used when no prototype or simulation environment is available.

Use case analysis

A particular method to manage and document the process of domain and task analysis, applies so called *use cases* (Jacobson et al. 1992). We have good experiences with use cases in the design of simple interactive systems. A use case generalises over a set of related scenarios. A use case is a document that specifies the projected roles of the user and system and their private and public goals, the setting, success and failure conditions, and the form, content and function of possible interaction sequences. In addition, there are specific parts of a use case to specify management information, such as version control, deadlines, and a list of open problems. Figure 1.5 shows an example of a use case for ticket reservation in a theatre application. Figure 1.6 is an example of the additional management information.

Depending on the stage of development use cases can be filled in as the analysis progresses (Cockburn 1997). Initially, they record possible scenarios of user behaviour. Typically, use cases are included in the requirement specification document that serves as a contract for the system developer. Because they can be intuitively understood, use cases facilitate communication with the service provider, with domain experts and with user focus groups. In object-oriented development methods, use cases are recommended to find common data elements that can be modelled as objects in the conceptual model (Jacobson et al. 1992). Test case designs for validation tests can be derived from use cases.

Use cases look much like scripts or plans (Schank and Abelson 1977). The recursive structure and the trigger, pre, success and failure conditions suggest an underlying plan based theory of task analysis. Each use case roughly corresponds to a sub-task or goal. Use cases may be nested and are often sequentially ordered. Use case templates allow variations and extensions to be defined as the development progresses. In this way, motivated design changes can be monitored in the documentation of the development process.

Use cases are not flawless. The modelling techniques for interaction that are supported by use case tools, namely flow graphs and sequence diagrams, are not directly suitable to model mixed-initiative dialogue. Part of the purpose of this thesis is to find a better suited design framework to support information, task and interaction analysis. Other criticism comes from within the software architecture community. Tekinerdogan, (to appear, ch3) criticises the relationship between the domain model and use case design. The required level of detail is difficult to assess, it is difficult to select the relevant use cases for defining architectural abstractions and the variation, extension and clustering constructs are not well defined.

1.4.2 Design and Implementation

During the design phase the domain analysis has to be converted into data structures and control mechanisms. Together these comprise the design of a dialogue system. The design can then be implemented using a programming language, or by combining existing modules. Not all of the domain model has to be implemented in the final version of the system! Domain models can play a role in the background, to motivate choices and to ground specification, evaluation and testing effort. For example, a plan-based task analysis does not need to be implemented using a planning algorithm. Because plans can be pre-compiled into recipes for action, the procedure-call mechanism of an imperative programming language often suffices, with the help of a stack.

Use Case C	Ticket Reservation
Actors	primary: user, system secondary: seat reservation database
Goals	User wants to make a ticket reservation. System wants to sell tickets.
Context	User called the system. User may know of performances.
Scope	Primary task, transaction
Trigger	User indicated the wish to make a ticket reservation.
Precondition	User has preferences and a limited budget. Enough seats are available.
Success	User and system agreed on title, artist, date, time and price of performance. System knows user's name and address and number and type of tickets. User and system agreed on total costs. The seats are marked as reserved for user. User is committed to pick up the tickets.
Failure	No seats marked as reserved for user. User made no commitments.
Description	1.System assists user to select performance within budget (use case B). 2.System informs user of title, artist, date, time and price. 3.System gets user name and address and also number and type (discount, rank) of tickets. 4.System informs user of total costs. 5.System gets explicit confirmation from user. 6.System marks seats as reserved. 7.System informs user of reservation and suggests commitment.
Next	D: Closure, B: Theatre Information or again C: Ticket Reservation.
Extension	xa. User misunderstands. System initiates help (use case E: Help). xb. User ends this reservation attempt (use case D: Closure). 1a. Selection not available. System apologises and offers alternative (1). 1b. User changes mind. System follows (1). 3a. User corrects number or type of tickets or corrects name and address. System follows (3). 5a. User does not confirm. System tries again, else (1). 6a. Desired seats not available. System apologises and offers alternative (3). 7a. User changes mind. System cancels reservation and informs user that no commitments have been made (1).
Sub-variation	2.Information already discussed is not repeated. 3.User satisfied with information only. (Use case D: Closure)

Figure 1.5: Use case for ticket reservation. Template adapted from Cockburn(1997).

Use Case C	Ticket reservation
Version Author Modified Due date	THIS v3.0 Hulstijn 15 06 98 01 09 98
Performance	total time ≤ 5 min, number of vacuous repetitions ≤ 3
Input Output	natural language text, direct manipulation (mouse) natural language text, tables, pictures
Open issues	deixis, multi-modal input
Super Sub	– B: Theatre Information (1)

Figure 1.6: Additional information corresponding to use case C.

In many spoken dialogue systems the interaction is controlled by a a *slot-filling* strategy (Aust and Oerder 1995). It operates on a simplified representation of the dialogue structure. Essentially this is a hierarchical structure of attributes of objects in the domain, combined with the status of their values: unknown, known or confirmed. Such attribute-value pairs are called *information items* in this thesis. In the slot-filling paradigm a system utterance is selected and generated by matching the required attributes against the representation structure, and then selecting a system prompt that asks or confirms the first missing or unconfirmed values. So in such applications, the information model is taken to be primary. The task and dialogue models are fixed on the basis of this model. But in the STEMSTEM voting application, the task and interaction model are primary; the information structure of a candidate list is rather poor. What matters is the confirmation of the vote.

1.4.3 Evaluation

In iterative, user-centred development, testing and evaluation is done in experiments similar to the 'Wizard-of-Oz' experiments used in analysis. Subjects are asked to perform assignments with the system. The assignments can be used to focus testing on precisely defined test objectives. The interaction is recorded, logged and annotated, producing a corpus for further analysis.

To achieve improvement, generally a combination of objective and subjective evaluation should be used. Objective metrics, like dialogue length, are used to filter problematic from unproblematic dialogues. Log-files of the problematic dialogues can then be analysed in detail. Problems are classified according to the apparent source of the problem. Subjective measures are used to set objectives for re-design and to assess the relative importance of different problem types. An example of a such an evaluation and re-design cycle is dealt with in chapter 6, where we discuss usability experiments carried out on the PADIS system (Bouwman and Hulstijn 1998).

Dybkjaer and others have developed a detailed theory of diagnostic evaluation for spoken dialogue systems (Dybkjaer et al. 1998; Bernsen et al. 1998). The theory consists of a list of neo-Gricean *maxims*, that specify principles of cooperative dialogue (Grice 1975). Figure 1.7 shows these principles. Some principles, indicated by a '*', are taken over directly from Grice. Others are adapted specifically for the case of man-machine dialogue. For instance, in man-machine dialogue the partners are not equal. Maxims are defined to deal with this asymmetry. Particular attention has to be paid to background information and misunderstanding and error repairs.

A disadvantage of this type of diagnostic evaluation is that user experiments are needed to collect evaluation data. User experiments are costly: experiments have to be organised, subjects selected, log-files annotated and analysed, and all of these tasks must be done by skilled system developers. It would save time and money if the evaluation effort could be reduced. We know that faults become more expensive when discovered later in development. Mistakes propagate and lead to more mistakes. Moving the detection of design errors forward in the design process is more effective and ultimately saves money. How can this be achieved?

The neo-Gricean maxims can also be used as design guidelines, early in development. When design choices must be motivated by at least one of the maxims, choices are likely to be better. When a design feature proves wrong, it will be easier to retrace and alter

Aspect	Generic or Specific Principle
informative-ness	* Make your contribution as informative as is required (for the current purpose of the exchange) Be fully explicit in communicating to users the commitments they have made. Provide feedback on each piece of information provided by the user. * Do not make your contribution more informative than is required.
truth and evidence	* Do not say what you believe to be false. * Do not say that for which you lack adequate evidence.
relevance	* Be relevant, i.e. be appropriate to the immediate needs at each stage of the transaction.
manner	* Avoid obscurity of expression. * Avoid ambiguity. Provide the same formulation of the same question or statement to users everywhere in the dialogue turns. * Be brief (avoid unnecessary prolixity). * Be orderly.
partner asymmetry	Inform the dialogue partners of important non-normal characteristics which they should take into account in order to behave cooperatively in the dialogue. Provide clear and comprehensible communication of what the system can and cannot do. Provide clear and sufficient instructions to users on how to interact with the system.
background knowledge	Take partners relevant background knowledge into account. Take into account possible (and possibly erroneous) user inferences by analogy from related task domains. Separate whenever possible between the needs of novice and expert users. (user-adaptive dialogue) Take into account legitimate partner expectations as to your background knowledge. Provide sufficient task domain knowledge and inference.
repair and clarification	Initiate repair or clarification meta-communication in case of communication failure. Provide ability to initiate repair if system understanding has failed. Initiate clarification meta-communication in case of inconsistent user-input. Initiate clarification meta-communication in case of ambiguous user-input.

Figure 1.7: Neo-Gricean maxims for man-machine dialogue (Dybkjaer et al. 1998).

it. But we would go even further. Given a sufficiently formal model of the task and the application domain, it should be possible to verify some crucial dialogue properties, already on the basis of the dialogue design. It seems that whatever the application, dialogue properties like consistency, informativeness, and coherence are crucial to usability. If it were possible to guarantee these properties, for instance by embedding verification methods in development tools, the evaluation effort could be reduced. But before we can continue to pursue this possibility, we must first discuss the different aspects of the modelling involved.

1.5 Models

The three major aspects of a domain model are the *information model*, the *task model*, and the *interaction model*. These aspects are closely related, but they are not the same. Each of these structures constrain the dialogue structure in their own way.

1.5.1 Information Model

The *information model* provides an ontology of the basic concepts and relations in the application domain. Each concept is characterised by a number of attributes. For example, the theatre performance concept is characterised by the attributes *title*, *artist*, *date* and *genre*. Concepts with their specific attributes can be implemented as objects using object-oriented class hierarchies, using records and types in a conventional programming language, or using typed feature structures. The concept hierarchy acts as a blue-print for the database design. It indicates which attribute combinations can serve as a key to select a unique instance of an object. Default expectations and functional dependencies between attributes must also be indicated in the information model.

Each concept is a potential dialogue topic. A concept is associated with a 'topic space' of attributes and related concepts that may act as sub-topics, thus forming a hierarchy. The idea is that during the dialogue, some dialogue representation structure keeps track of the active topics of conversation. This is what Grosz and Sidner (1986) call the attentional structure. Such a topic representation is severely constrained by the conceptual hierarchy. Here is an example of a dialogue with a complicated topic structure. It shows that information increase is not always monotonic. Users may jump back and forward between alternatives.

(2) U1: What's on at the 18th of March?
S1: On the 18th you can see 'Deelder Denkt' or 'Indonesian Tales'.
U2: At what time does Deelder start?
S2: The performance by Jules Deelder starts at 8 o'clock.
U3: How much is it?
S3: We give student discount. Do you have a student card?
U4: No
S4: The ticket for 'Deelder Denkt' without discount is f30,-.
U5: Oh, that's expensive. How much is the Indonesians?
S5: One ticket for 'Indonesian Tales' without discount is f30,-.
U6: Give me Deelder then after all.

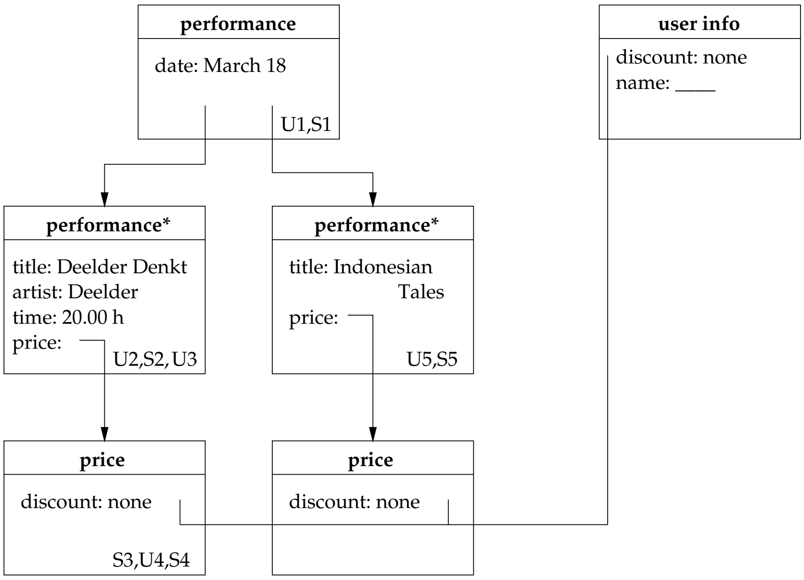


Figure 1.8: Topic Structure of example (2).

After mentioning the date, 18th of March, there are two alternatives. The user can switch between them. In utterance U2, the user asks information about Jules Deelder. This limits the topic to the Deelder performance. Via the calculation of prices, discount becomes a subtopic. However, in utterance U5 the user switches to the other alternative. Backtracking is needed because the phrase 'Indonesians' does not match the discount related topic space, nor the Deelder topic space. It does match the '18th of March', so it produces a new separate topic space. In this new context the discount information seems to be lost. However, discount is not only a subtopic of a performance by the attribute price, but also of the dialogue control topic user. Because the attribute is shared between these concepts, the information is not lost. However, the time information does get lost. This is as it should be because 8 o'clock time is associated with the Deelder performance. In general, a topic space contains attributes that are conceptually related, and that are therefore mutually dependent. There are two distinct topic spaces in this dialogue: one with information related to performances and one with user information.

1.5.2 Task Model

The *task model* relates to the typical actions carried out by a user in a given application type. The task should be matched by the functionality of the system. A task can be expressed by a hierarchy of goals and sub-goals, or else by a plan, a combination of actions, to achieve those goals (Allen et al. 1991; Carberry 1990). This is what Grosz and Sidner (1986) call the intentional structure. For dialogue systems, the basic actions from which complex plans can be constructed are either database operations or dialogue acts like requests or assertions that initiate an exchange of information.

Also the task constrains the dialogue, in two ways. First, the task determines what objects are relevant. The attributes needed by a sub-task become relevant when that sub-task is addressed. So a topic space roughly corresponds to a task, and a sub-topic to a sub-task. Therefore an utterance like "3 for tonight, please" can be interpreted unambiguously as a request for three tickets for tonight's performance given the two sub-tasks of performance selection and specification of the number of tickets, of the underlying ticket reservation task. Answers to questions that were never explicitly asked, are still felicitous because of the underlying task. Second, the task may put constraints on the order of dialogue act exchanges that address sub-tasks. For simple task-oriented dialogue, there are two types of constraints: *precedence*, and *dominance* (Grosz and Sidner 1986). Task *a satisfaction-precedes* task *b* when *a* must be completed before *b* can start. Task *a dominates b*, when completion of *a* is part of completion of *b*. A third type of constraint, that two tasks be carried out *simultaneously*, becomes relevant for multi-modal systems where different output modalities like speech and gesture must be carefully synchronised.

The structure of the different steps in the use case example of figure 1.5, corresponds to the task structure of a ticket reservation task. The general reservation task dominates steps 1-7. The precedence constraints can be partly derived from the preconditions for successful completion of the various sub-tasks. Step 1 is a precondition for steps 2, 4, 5, 6 and 7, and must therefore be completed before them. But step 3 is independent of 1, and can thus occur before it, despite the preferred order of steps initially indicated in the use-case analysis. What's more, steps may be combined, as in "3 for tonight, please".

These task level constraints are for a large part based on functional dependencies in the information model. Selection of a performance depends on the attributes that function as a database key: title, artist, genre and date. That means these four attributes are interdependent. One cannot change one, without having to change the others. The user's name or the requested number of tickets are independent of the selected performance. As we saw in example (2) user related issues constitute a separate topic space. So after retracting the preference of a user for a title, the system should also retract the related dates and artist, but still remember the user's name.

1.5.3 Interaction Model

The *interaction model* relates to the linguistic realisation of a task. It describes the segmentation and annotation of dialogues into exchanges of related dialogue acts. For example, a question is an initiative. It must be followed by some kind of response: an answer, or else by a counter question, which is then followed by an answer, or by a rejection to answer etc. Either way, the hearer is obliged to react in some appropriate way to the question. Similarly, a proposal must be followed by an acceptance, a counterproposal or a rejection. A greeting must be returned. Such typical interaction patterns can be described by dialogue games (Mann 1988), by dialogue grammar rules (Polanyi and Scha 1984) or by coherence relations (Asher and Lascarides 1998a).

How are these aspects of dialogue models related? The answer is: by the dialogue representation structure. At each point in dialogue, the dialogue context is modelled by a hierarchical representation structure, much like a parse-tree. At the nodes of the tree we find a representation of the information carried by specific utterances. On the branches of the tree we find the coherence relations. As the dialogue progresses, a representation of each utterance is attached to the appropriate part of the tree by means of a plausible coherence relation. The coherence relation indicates the function of an utterance with respect to the task, and to the surrounding utterances. Typical coherence relations are *elaboration*, *causation* (because) or *contrast* (but). For dialogue, coherence relations can correspond to initiative-response units like *answering*, *accepting* or *greeting*.

Because dialogue segments consist of utterances that are related, the dialogue representation structure guides resolution of context dependent expressions like ellipsis or anaphora and determines the scope of operators like correction or denial. The objects represented along the 'right frontier' of the tree are accessible for anaphoric reference. These constitute the active topics alluded to above. Naturally, this aspect of the representation is constrained by the conceptual hierarchy of the information model.

In many cases a dominance relation at the task-level corresponds to a coherence relation between segments (Moser and Moore 1996). In the reservation example of figure 1.5, the reservation task dominates the sub-tasks that correspond to steps 1-7. If the system says, "To make a reservation I will need some information about you. What is your name, please?", this indicates that step 3 is a *condition* for the reservation task.

1.5.4 On the Role of Dialogue Models

In figure 1.3 we pictured the domain model for a certain application on the right. The model is not necessarily identical to the resource that gives a specification of the behaviour of a component, let alone to the implementation. Ideally, a resource specifies the desired behaviour of a specific implementation. The resource in turn, should capture the general principles and observations of the domain model.

In this philosophy, the domain model acts as a kind of semantics to the design specification. Given a domain model we can in principle verify whether the resources are internally coherent and whether they are 'correct' and 'complete' with respect to the domain. Correctness would mean for example that each expression that is well-formed according to the lexicon, grammar and dialogue rules, should be interpretable as a concept or action in the domain model. And completeness would mean that each concept or action must have a preferred way of expressing it in system behaviour. Such constraints would ensure among other things that all expressions used by the system are also understood when applied by the user. If such constraints were built into toolkits for dialogue system development, they would help users to learn a transparent mental model of the system's behaviour.

What does it mean to say that the domain model could serve as a kind of semantics? What level of detail is required? Distributed systems for example, are verified relative to an abstraction based on process algebra. A system is judged correct when all projected system states are reachable and there are no deadlocks: no process is waiting for another to terminate. To determine a similar notion of correctness for dialogue system design, we need a modelling abstraction which focuses on the crucial aspects of dialogue systems. We argue that those will be aspects that affect usability. Therefore we try to find verifiable dialogue properties that are related to properties of dialogue system designs which affect usability. In principle a dialogue system design is usable when it has design properties that produce dialogues with usability related dialogue properties.

This is a very idealistic view, especially with regard to the practical dialogue systems of today. Often much simpler design or evaluation characteristics can be used to good effect. On the other hand, a theory about design and evaluation which integrates different aspects of a dialogue system is still lacking. The framework developed here attempts to provide the beginning of such a theory. Moreover, this idealisation might help to challenge, focus and motivate ongoing theoretical research into the semantics and pragmatics of dialogue. Dialogue system design generates interesting puzzles for theorists.

$$\text{usability}\left\{\begin{array}{l}\text{effectiveness}\\ \text{efficiency}\\ \text{understandability, learnability, operability}\\ \text{satisfaction}\end{array}\right.$$

Figure 1.9: Combined usability properties of ISO 9126 and Nielsen (1993).

1.6 Usability Properties

We argued that because of its acclaimed advantages the quality factor that is most important for natural language interaction is *usability*. In general, the assessment of software quality involves three levels of analysis (Pressman 1997). The first level describes *quality factors*. For example, maintainability, how easy it is to keep the system adapted to a changing environment; portability, how easy it is to port the system to a new application; or reliability, how unlikely it is that the system breaks down. These factors describe general aspects of a system. The second level describes *properties* which are still abstract, but can be made concrete for different application types. For example, maintainability and portability are both related to modularisation: the extent to which software is developed in separate modules. For dialogue systems we will argue that the relevant properties are effectiveness, efficiency, coherence and transparency. These properties are concrete enough to guide both the design and the testing efforts. Finally, at the third level we need observable features or *metrics* to collect data about the design and evaluation properties. If features or metrics are measured quantitatively, they can be combined in a weighted sum: a summation of the measurements, multiplied by their respective weights. Weights indicate the relative importance of a feature. For each application type specific metrics and weights will have to be decided on and tuned to fit the test objectives. If the metrics are concerned with qualitative concepts they can often be turned into numbers by indicating a value on a scale of preference. In isolation, it is difficult to interpret the meaning of a metric. Metrics have shown to be especially valuable when historical data exists for comparison. Metrics can indicate a trend, or the relative improvement after a re-design.

Usability is one of the defining characteristics of software quality, along with such factors as functionality, maintainability, reliability, efficiency and portability. Many of these characteristics have been named in different configurations by different authors. Nielsen defines usability by learnability, efficiency, memorability, error recovery and satisfaction (Nielsen 1993). The ISO standard for software quality lists understandability, learnability, and operability as aspects of usability (ISO 9126 1991). We propose to combine these theories as follows (figure 1.9). Effectiveness and efficiency are listed as sub-aspects of usability, although one could argue that they are quality factors in their own right. After all, the software quality factor of functionality which assesses the features and capabilities of a system is obviously related to effectiveness. However, this choice means that the other quality factors all have to do with general system characteristics and management. All user related aspects are now listed under usability.

We now discuss these properties. For each usability property we briefly describe how it can be influenced by design properties, such as transparency, and how it can be assessed by dialogue properties, such as coherence (figure 1.10). Dialogue properties are defined for completed dialogues, but we also refer to the contribution of a single utterance.

usability property	design property	dialogue property	metrics
effectiveness	functionality	task success	success rate
		coherence	see [1]
efficiency	functionality	complexity / effort	number of basic actions, duration, number of utterances
understandability learnability operability	transparency	coherence	[1] number of topic shifts, corrections, superfluous repetitions
satisfaction	several aspects, e.g. appearance	greetings, thanks	surveys, actual usage

Figure 1.10: Usability, design and dialogue properties for task oriented dialogue systems.

1.6.1 Effectiveness

Effectiveness concerns the accuracy and completeness with which users achieve their goals. Obviously, the effectiveness of a design is influenced by the extent to which the system's functionality matches the desired task. Effectiveness can be be assessed by measuring the *task success rate* of a set of dialogues and partly by analysing the relative *coherence* of individual utterances with respect to their context. Part of coherence concerns the contribution of an utterance to the general task. The dialogue property of coherence is dealt with below.

The task success rate of a dialogue system in a particular user experiment is the proportion of dialogues in the corpus in which the user managed to achieve her goal. By what metrics does one measure task success? For the most thorough kind of analysis utterances in a dialogue have to be segmented and annotated with their apparent goals. We can observe if that goal is met by the end of the dialogue, and if not why not. What counts as a success or a failure, should have been specified in the requirement specification of a system. In order to get a fair distribution of the different sub-tasks throughout a corpus, one should design evaluation experiments in such a way that for each sub-task a test case is carried out. A test case here corresponds to an assignment or test scenario for subjects to complete. In particular, tests should be designed to deliberately generate misunderstandings. Note that this type of analysis is very laborious, and depends on the subjective judgement of the individual annotators. In diagnostic evaluation, it suffices to do this kind of analysis only for a number of problem dialogues, which are selected by means of objective evaluation measures like dialogue length or the number of repetitions.

It is difficult but possible to measure task success quantitatively. The PARADISE evaluation framework uses confusion matrices that represent the deviations from some fixed attribute-value matrix that gives the expected values for a task (Walker et al.1997). By calculating agreement between the actual and the ideal values, they derive a general measure of task success. Unfortunately, a task can not always be represented as a fixed set of attributes to be filled in. Why not? First, the number of attributes depends on the contents of the database. For instance, on some days the date is enough to select a performance, but on other days also the artist or title is needed. The scheme could be adjusted to accommodate this flexibility, but this accommodation would require a way to normalise for the current task complexity. Second, we have seen that some attributes are interdependent. Attributes that determine a unique object, like title, artist, genre and date, are clustered into objects. The measure should be normalised for such objects. Third, a dialogue is dynamic. Not only the content of the dialogue, but also the user's goal may change. Recognising the user's goal is a crucial part of a dialogue system's function. In a controlled evaluation experiment, some of these parameters can be fixed temporarily to make quantitative assessment possible.

1.6.2 Efficiency

Efficiency concerns the relative effort in relation to effectiveness. Efficiency can be influenced by carefully adapting the interaction strategy to the task. An example of this will be discussed in chapter 6 (Bouwman and Hulstijn 1998). Efficiency of a design can be computed by dividing the relative success by the effort needed to achieve it. Effort can be assessed by objective dialogue measures, such as the number of turns, number of utterances or dialogue duration. In user experiments indicators of human processing effort can also be measured, such as the number of spoken disfluencies or the number of topic shifts (Oviatt 1995). Obviously, the effort needed to accomplish a task depends on the complexity of the task. Task complexity is a measure that combines the number of levels of embedding in the task, with the number of basic actions needed to complete the task under ideal circumstances. Depending on the likelihood of a misunderstanding, which largely depends on the quality of the speech recognition, we can also estimate average task complexity under different degrees of misunderstanding. Such task analysis is similar to the GOMS method used in human-computer interaction (Card et al. 1983). A GOMS model describes the knowledge, effort and time necessary to complete a task in terms of *goals*, *operators*, *methods* and *selection rules*. Research has shown that the time an average skilled user needs to select a method or apply a single operator is rather constant. Thus, GOMS models can predict the average duration of a task for a particular interface design.

Misunderstanding and error have a great impact both on effectiveness and efficiency. Therefore the error rate and error recovery rate are good indicators of usability. A system response is classified as an error, when it is incoherent with the user's actual utterance, the task, the information in the database or with the dialogue structure. For dialogues that are not goal-directed or that are designed for leisure users, efficiency and effort are less crucial. For leisure users, usability has to do with the relative fun of the interaction, not with achieving the task as quickly as possible. Still, the number of misunderstandings and errors remains a good indicator of usability. Misunderstandings are irritating, take effort to correct and lead to behaviour that is difficult to understand, learn or avoid.

1.6.3 Understandability, Learnability and Operability

Understandability, learnability and operability are aspects of usability (ISO 9126 1991). They indicate how easy a user can understand, learn and handle to control a system. For natural language these aspects should be easier: understanding and learning a natural language system is not a problem, provided the system's resources match the application domain. The occurrence of out-of-domain errors shows that this is not always the case. Operability becomes a problem in case of misunderstandings and when the expectations of users turn out to be wrong. The design property related to understandability, learnability and operability is transparency; the corresponding dialogue property is coherence.

Transparency

For natural language interfaces the limits of the system's capabilities, both with respect to linguistic understanding and functionality, are not immediately visible. The user has to rely on her own mental model of the system and of the task, which is largely based on previous experiences. The capabilities of a system must coincide with the expected mental model, or if that is not possible, the mental model should be influenced, to bring it in line with the system's capabilities. Apart from marketing techniques, there is not much to go on. Users often do not remember the introductory advice about how to use a system. The only other way to influence the user's mental model is by a careful design of system prompts: what to say, when and how. A system is called *transparent* to the extent that it does not allow the user's mental model to deviate from the system's capabilities. A system tends to be transparent when it always indicates what it is doing, and why. Transparency is a kind of consistency in behaviour. Obviously, transparency can not be added to a system's design. It must come from a systematic and principled way of designing system prompts. It is difficult to assess how transparent a system is. Only in case of comparative testing can one come to limited conclusions. For example, the dialogue redesign for PADIS described in chapter 6 probably made the system more transparent (Bouwman and Hulstijn 1998).

Coherence

Coherence concerns the way utterances are combined into a well formed dialogue. An utterance is *coherent* with the context, to the extent that it 'fits' the dialogue representation structure at the relevant linguistic levels. Incoherence signals a lack of understanding and because misunderstandings decrease usability, coherence is a good indicator of usability.

Coherence involves aspects of form, content and function (figure 1.11). *Form* aspects are intonation, lexical choice and syntactic and semantic parallelism. For instance, when the user refers to the date as "Wednesday", it is confusing if the system replies with "On August, 25, ...". In general, adjacent utterances tend to be similar both in structure and wording. Objects that are given in the context, are referred to by a pronoun or even left out. Getting the intonation right is important too. New or contrastive information must be accented; agreed or old information must be deaccented. In particular information to be confirmed by the implicit confirmation strategy must be deaccented. With respect to *content*, information must be consistent and accurate (maxim of quality). Moreover, an utterance should be relevant and should be just informative enough for the current purposes (maxims of relevance and quantity). An utterance is not *licensed* when it is over-informative. These aspects can be given a formal characterisation (Ch 3; Groenendijk 1999; Hulstijn 1997). The *function* of an utterance must also fit the context. This can be assessed relative to the task model and to the interaction patterns. An utterance is incoherent when it cannot be attached to some part of the dialogue representation with a coherence relation (Asher and Lascarides 1998a). How can we measure coherence ? In general, anaphoric references, semantic and syntactic parallelism between adjacent utterances and verb-phrase ellipsis increase the perceived coherence. On the other hand, unnecessary repetitions, interruptions and corrections decrease coherence. Many topic shifts are perceived as incoherent. These aspects must be annotated by hand. Repetitions and corrections can be automatically calculated on the basis of the system's decisions.

coherence	form:	lexical choice, parallelism, intonation
	content:	consistent, informative, relevant, licensed
	function:	task, interaction

Figure 1.11: Coherence aspects

1.6.4 Satisfaction

User satisfaction is the usability factor that is most difficult to influence directly in a design. Yet in a way, it is the most important feature. Ultimately it determines if users are actually going to use the system. Satisfaction combines the user's impression of most aspects of a system. So for example, if the task is important to a user and the system is effective, satisfaction will be expected to be high. In that sense, user satisfaction corresponds to *perceived usability*. It is the subjective and observable correlate of the abstract quality factor usability. Therefore ways of measuring user satisfaction, i.e. by questionnaires, surveys, think-aloud protocols and simply counting the numbers of actual usage of a system can be used to calibrate the objective usability measures and the relative weights. In the PARADISE evaluation framework this is exactly what is done to tune the parameters of the evaluation (Walker et al. 1997).

Ideally, one would have a theory that provides for each type of application a number of suitable metrics that measure the different usability properties. There would be a general formula that tells us how to combine the input of the different metrics. Then the weights in the formula can be tuned, by correlating the calculated usability measure with the outcome of a survey for user satisfaction. What makes user satisfaction so difficult to predict, is that it is affected by all aspects of a system, also by apparently non-functional ones. For example, the satisfaction of many spoken dialogue systems is related to the quality of the synthesised voice, or to whether it is a male or a female voice. What is decisive for users to actually use a system is its added value. How much better is this system compared to an existing system? For example, in the case of PADIS users often preferred the simple paper list for retrieving telephone numbers. Only when users were not at the office, they appreciated the possibility of calling the system.

As we said in the beginning of the chapter, building a usable dialogue system is a bit like designing a door handle. Within the limitations set by human biology, the functionality of a door and the way the handle is to be used, there is still a lot of room for creativity and non-functional differences. Still, those 'style issues' will make a difference for users to decide to use one product rather than another. There is already a trend towards dialogue systems with a personality. More and more interactive systems are developed for leisure users. Developments like affective computing, intelligent computational toys and interactive games make it interesting to see how one should evaluate systems which are by nature less functional.

This concludes our brief overview of the various usability properties and ways they can be influenced and measured.

1.7 Conclusions

This chapter addresses the software engineering aspects of developing a dialogue system. We discussed the advantages and disadvantages of a natural language interface. The main advantage is that natural language is easy to use, and flexible. The advantages only come out when the disadvantages has been dealt with. Natural language interfaces have no natural way to indicate the boundaries of their capabilities. Therefore, the only thing to do is to make sure that the user's mental model corresponds with the capabilities of the system.

Because dialogue systems are examples of complex interactive systems, we argue for an iterative development method, in which a prototype or system is constantly evaluated and improved. Furthermore the development process should be user-centred. It should involve end-users in order to capture pre-existing expectations about the task and the application domain. Use cases are recommended for a task analysis process on the basis of scenarios: descriptions of projected interaction. Use cases facilitate discussion with user focus groups, and test case design.

We advocate the separation of implementation, specification and modelling aspects in the development of a dialogue system for a particular application. Ideally, a domain model could function as a semantics to the design specification, which in turn determines the behaviour of the implementation modules. However, for such a semantics to be effective in specification and verification, it should be based on an abstraction that captures the right kinds of properties: properties that affect the usability of dialogue systems. The properties are effectiveness, efficiency, understandability, learnability and operability, and satisfaction. We indicated that these can be affected by the functionality of the system and by the transparency of the system prompts. They can be assessed by task success rate, a measure of the relative effort such as duration, and by coherence measures such as the number of anaphoric references, the number of superfluous repetitions and the number of topic shifts.

Chapter 2

Coordination

This chapter discusses the notion of coordination in communication. The design of interactive agents that allow multi-modal and situated communication in a virtual environment triggers a view of communication as a combination of joint actions between speaker and listeners. This requires coordination between participants at and within different communication levels. Methods of grounding play an important role. An account of synchronisation and planning constraints on the composition of joint actions is presented.

2.1 Introduction

Consider a virtual environment with agents and objects. Agents can walk around, perceive objects and communicate with other agents. Agents can communicate with each other by means of speech, gestures, direct manipulation and written messages, or by combinations of those. Based on the blueprints of the actual music centre in Enschede, such a virtual environment called VMC (Virtual Music Centre) was constructed (Nijholt et al. 1999; Nijholt 1999; Nijholt and Hulstijn to appear). It is meant as a framework for trying out new ideas about communication and entertainment. The theatre environment can be visited by human users. It houses a number of artificial agents that can perform all kinds of functions related to the theatre. There is a theatre information and reservation agent called Karin who answers enquiries about performances and makes ticket reservations. We are building a navigation agent that helps the user navigate the building. It may give the user a route description or take the user directly to the requested location. There are also some less interactive agents like a notice board, posters and a virtual piano that can be played. Figure 2.1 shows Karin behind the information desk.

When designing interactive agents in a virtual environment a number of different aspects of communication become important, as compared to spoken dialogue systems for example. Aspects like the relative position of speaker and listener, their movements and the objects that are visible to them become crucial to the communication process. So in principle such an environment supports *situated communication*. The virtual space allows agents to make movements or gestures which are perceived by other agents or users. Moreover, the agents can talk with each other or can simulate talk by means of a keyboard interface. Interaction that allows the simultaneous use of multiple interaction modes, like speech and gesture, is called *multi-modal interaction*.

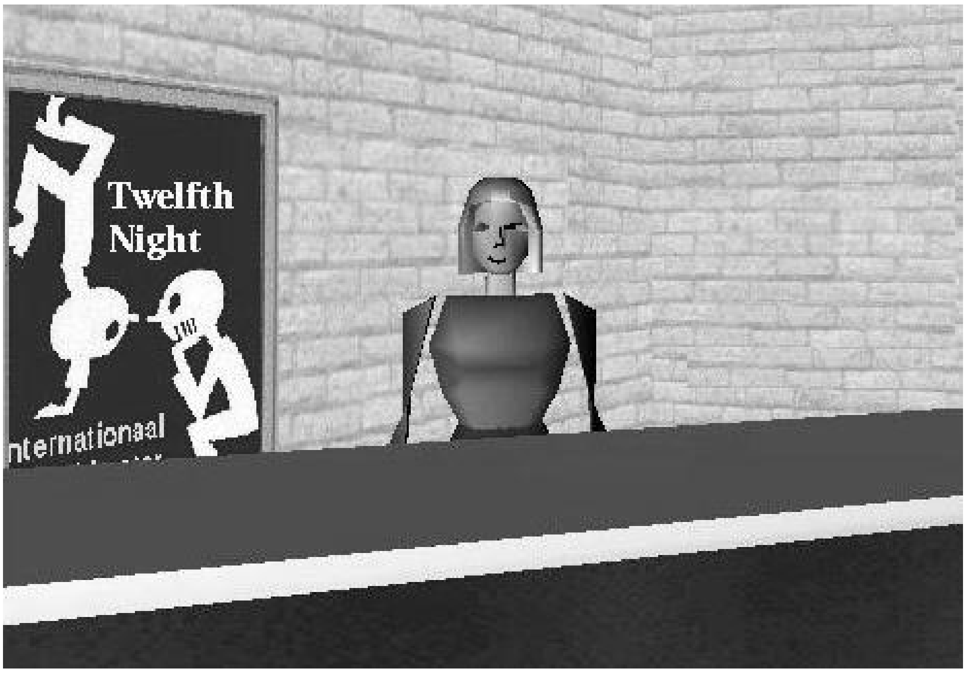


Figure 2.1: Karin in the virtual theatre environment.

In a virtual environment we can try to approximate face-to-face conversation, which is arguably the most basic and direct type of communication. Agents that participate in a conversation must face each other and be within each other's hearing distance. This will identify them as being participants in the conversation. The spatial configuration may change; agents may enter or leave the immediate surroundings of the conversation at all times. When referring to an object, the object must be visible to all participants. Participants must be aware of each other being aware of the object. For example, assume the user is pointing at a poster of "Twelfth Night". In order to understand this gesture as a way to refer to a play, Karin must be aware of the movement being a deliberate pointing gesture, of the fact that there is a poster there and that it represents a particular play. Thus, situated communication requires temporal and spatial coordination between participants.

The setting also helps to determine the social roles played by the participants in the activity that they are engaged in. For each activity there are very basic rules that can be described in terms of a 'language game' or 'script' to guide the interaction. Based on a role, different rules apply. Karin is situated behind an information desk so she is likely to be staff. The user is situated in front the information desk and thus plays the role of a customer. Because the whole environment is a theatre, we can assume that Karin can answer enquiries about performances, make reservations and sell tickets. Similarly we can assume that the user is interested in theatre, in performances or even wants to make a reservation. Such assumptions constrain the way the interaction will develop. Thus, situated communication also involves social coordination between participants.

Multi-modal interaction is interaction that involves a combination of different interaction modes or modalities, such as speech and gesture. The combined interaction requires careful synchronisation between the composite actions. Consider for instance Karin when she is uttering "Welcome!" to a new visitor to the virtual theatre. The welcome can be accompanied by a professional smile. It requires a combination of two interaction modes: speech and facial expression. If the smile comes too late or if she would frown, the welcome is regarded as insincere. So, multi-modal interaction requires careful synchronisation between different communicative processes within one agent.

We take coordination as a fundamental notion in modelling communication processes. Coordination is meant to solve *coordination problems* (Schelling 1960). A coordination problem arises when a group of people share an interest or goal and the success of each person's actions to achieve that goal depends on the actions of the others. To a large extent communication is just that: people solving coordination problems (Lewis 1969). The notion of coordination is therefore related to what Clark and Schaefer (1989) have called *joint actions*. A joint action is an action in which several agents cooperate to achieve a common goal. Typical examples of joint actions are carrying a piano or playing a duet. The success of a joint action crucially depends on the combined success of the actions of the participating agents, called *participatory actions*. It is because of this mutual dependency that coordination is necessary.

Suppose for example, that the user intends to book some tickets. Because Karin controls the reservation database, the success of the user's actions to reach that goal depend on Karin. The user must collaborate with Karin. So the user enters a conversation with Karin, thus establishing a common goal to maintain the conversation itself as well as a common goal to make a ticket reservation. In order to start the conversation, the user must attract Karin's attention by approaching her. On the other hand, Karin must indicate that she notices the user's approach, for example by smiling and greeting. After that particular tickets are requested and booked. So we find joint actions with common goals and coordination at several distinct levels: the task level of the ticket reservation, the level of the content of the conversation and a level of attracting and notifying mutual attention.

That suggests the following hypothesis: communication and in particular conversation or dialogue can be seen as a complex joint action of speaker and listeners, with participatory actions scheduled in parallel at different linguistic levels. At each level we find coordination on some aspect of the communication. Speakers adjust the wording and presentation of their utterances to the listeners. But listeners also adjust; listeners' expectations and interpretations are adapted as the conversation progresses. Speaker and listener attract and keep each other's attention. They use both verbal and non-verbal signs, like intonation, nodding, gazing or posture to coordinate on the timing, form, content and function of each other's contributions. Only when communication fails at some level, we realise how much mutual coordination usually goes unnoticed.

The chapter is structured as follows. In section 2.2 we present a layered account of coordination in communication. Based on this picture, we look for general constraints and structures that make the coordination possible. We introduce the basic coordination devices: convention, explicit agreement and salience (section 2.3). Then we study synchronisation of communicative events. This provides external and observable constraints on the temporal coordination of participatory actions (section 2.4). We look at what makes joint actions joint in section 2.5, where we look at cooperativity, the 'glue' that holds joint actions together and give an account of joint planning and action. The chapter ends with a summary and preview of chapter 4 and 5, in which the theory of action, and the theory of interaction patterns are further formalised.

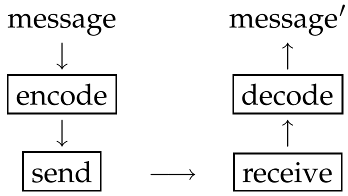


Figure 2.2: Sequential model of communication

2.2 Layered Communication Models

There is a striking resemblance between the idea of communication as coordinated joint action at and between various distinct levels, and the layered communication protocols of distributed and concurrent programming, e.g. Tanenbaum (1995). Such protocols are typically specified in a process algebra such as CSP (Communicating Sequential Processes) (Hoare 1985), CCS (Milner 1980) or the OSI standard specification language LOTOS (Bolognesi and Brinksma 1987). We have chosen CSP for the design of some coordination protocols in the virtual theatre environment (van Schooten et al. 1999). A brief introduction to CSP is given in appendix A to this chapter.

The ontology of a process algebra is fairly simple. There are two types of entities: *processes* and *events*. Processes are continuous, until they stop. Events are instantaneous. Event names are assumed to be unique. So if processes share event names, that means that they must synchronise on those events. Processes can stop correctly, or they can be blocked waiting for another process to synchronise on. This is called a dead-lock situation. Common event names are called *channels*: they define the interface between different processes. Some channels are used to align processes temporally; others are used to send and receive data between processes. From an external point of view a joint action consisting of several participatory actions can be seen as a complex process consisting of several concurrent sub-processes that share synchronisation channels. Because of the shared channels, participatory processes can no longer be described as independent; coordination is at the heart of a concurrent process specification.

By contrast, in the traditional sequential model of communication, that is implicit in much linguistics research, the activities of message production and message understanding are seen as independent. Communication is perceived as in figure 2.2. There is a sender that encodes and sends a message, mirrored by a receiver that in turn receives and decodes the message. Such a message model means that the production and understanding of a message can be studied in isolation. This greatly simplifies the model. Now of course this picture is a caricature. The processes are not completely independent; after all the message must be encoded in a commonly agreed *code*. Scheduling processes in sequence is an effective way of coordinating their delivery. However, the code is usually agreed beforehand, and remains fixed during interaction. Sequencing is a rigid protocol and does not allow for overlapping and simultaneous actions.

We argue that this independence assumption is no longer tenable for real-time, situated, multi-modal communication. To show this we briefly repeat the earlier example of a customer pointing at a poster in a ticket office. The example illustrates that multi-modal interaction typically consists of joint actions; not only joint actions between speaker and listener, but also joint actions between different communication channels of one agent.

2.2.1 Ticket selection example

Suppose I am in a ticket office and say "Two tickets for that one please!" to the lady at the counter, while pointing at a poster displaying a performance of 'Twelfth Night'. My combined speech and gesture taken together with the lady's understanding it, form a complex joint action of performance selection. Only when I speak and point at the poster simultaneously, the deictic expression "that one" will be interpreted as referring to a performance of the play represented by the poster. If there is no salient gesture the utterance will be interpreted as referring to a previously mentioned object. So the meaning of the deictic expressions depends on the temporal coordination of utterance and gesture. Ideally the culmination point of the gesture, the moment at which the index finger is fully extended, should coincide or slightly precede the intonation centre of the phrase. The intonation centre of a phrase is the point at which the pronunciation reaches the highest pitch level; it indicates what is most salient in the utterance. When the gesture culminates too early or too late, the deictic meaning is lost. And also when the lady does not pay attention or when she can not see the poster, the selection would not succeed. Because this act of performance selection in turn is part of a joint action of buying-and-selling tickets, we would soon notice the failure because of conventions in ticket selling and mutual expectations about the way to proceed. If the lady hands me the wrong tickets, I must correct her. And again, there are commonly known conventions for correction that will guide my behaviour. Thus, coordination in this joint selection action assumes a so-called *common ground* (Lewis 1969; Lewis 1979; Stalnaker 1979; Clark and Marshall 1981) among dialogue participants. The common ground is both a prerequisite for and the result of successful communication. It contains information that is commonly known among the members of some well-defined group. In this case shared background knowledge – ticket sales conventions –, shared physical surroundings – the poster – and a shared dialogue context – my utterances – are crucial for accomplishing the joint action. Not only the coordination between agents can be modelled by joint actions, but also the synchronisation of communicative processes within an agent. The culmination of my pointing action – the moment my index finger is fully extended – must be simultaneous with the intonation centre – the moment where the pitch is highest – of the phrase "that performance".

2.2.2 Communication Levels

In human communication, several levels have been identified. According to Goffman (1981) there are three roles in speaking: the principal who decides what to say, the formulator who composes the messages and the vocaliser who turns the message into sound waves or ink patterns. Thus the process of speech production decomposes into three levels: conceptualising, formulating and vocalising. These modules are psychologically real, as can be shown by empirical evidence (Levelt 1989). Language comprehension involves the same three levels: attending to the sound patterns or movements, identifying the message and comprehending the content. According to Clark (1996) each of the pairs vocalising-attending, formulating-identifying and conceptualising-comprehending comprise a joint action at that level. Alternatively, if we take a less procedural view, level three may be referred to as the level of meaning-understanding. It defines the important concepts of *speaker's meaning*[1] and *addressee's understanding* (Grice 1957). Obviously,

[1]Dutch: bedoeling, German: Bedeutung

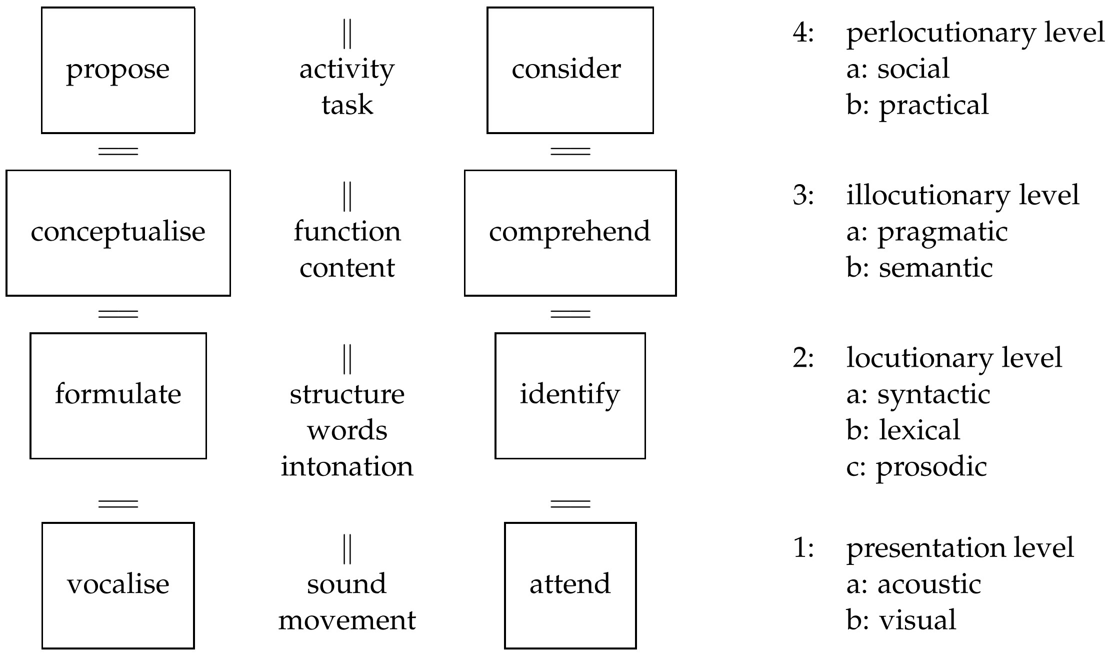


Figure 2.3: Coordination *at*, *between* and *within* different levels of communication

meaning and understanding are the result of the processes of conceptualising and comprehending. In addition to these three classical levels, we add a fourth level: a level of proposing-considering aspects of a social activity. When analysing activities, we can define particular activity types, or tasks, in which certain generalisations and assumptions can be made. The dialogue acts of the third level are motivated by the stage of the task at the level above.

There is a different source of terminology that may be useful in distinguishing communication levels (Austin 1962). For each utterance the *locutionary act* is the act of actually uttering it: formulating and pronouncing the words. For gestures, the deliberate production of movements is also to be found at this level. An *illocutionary act* is to be achieved *in* the utterance. It involves the exchange of information. For instance, the information that the speaker wants the listener to understand that she wants to select Twelfth Night. The term illocutionary act has become more or less synonymous to the term speech act or *dialogue act*. A *perlocutionary act* is to be achieved *by* the utterance. For instance to select a performance as part of a ticket reservation. We will borrow this terminology, extending the notion of 'acts' to include all kinds of communicative joint actions.

There are a number of concurrent processes running at any one time during a conversation. The $\|$ sign of parallel composition between processes in figure 2.3 illustrates how coordination may occur at each of these levels. The tags in the middle indicate the communicative aspect that agents coordinate on at that level. Processes within one agent are coordinated between different levels, indicated by the $==$ sign. Note however, that this picture does not present a final theory. For each application domain levels may be chosen differently or be given different names.

Perlocutionary level

At the *perlocutionary level* the perlocutionary function of utterances is defined. At this level speaker and listener coordinate on the extra-linguistic *task*. A task is a generalisation of an social activity; it defines a particular activity type. For example, the activity of ticket selling is an instance of a transaction task. Activity types are determined by the roles of the participants, their goals, social institutions that may also play a role, the physical and social settings with relevant artifacts and a number of phases that determine the progress of the activity. In ticket sales we have a provider and a client, who want to sell and buy a product or service. The social institution is that of ownership. The typical setting is that of a shop or other public building with a counter. Important artifacts represent money and the service or product to be sold. The phases in a transaction are an inquiry phase, of information exchange about the product, a negotiation phase in which there is bargaining on the price and attributes of the product or service and an actual confirmation phase in which the deal is closed.

Illocutionary level

At the *illocutionary level* both the communicative function and the semantic content of an utterance get established. This involves speaker's meaning and listener's understanding. Because utterances are often ambiguous as to their actual content or communicative function, the listener has some room to *construe* the utterance as having a particular meaning. Second-turn corrections like "No, I didn't mean to correct you, I was just asking!" illustrate a misconstrual of the communicative function (Ardissono et al. 1998). Level three is also the level where the semantic content of utterances is coordinated. Speaker and listener tacitly agree on what material will enter into the common ground. Presupposition accommodation (Heim 1983; Lewis 1979) is an example of coordination at the content level. A presupposition trigger is an expression that conventionally indicates that some object exists or that some fact holds. By using a presupposition trigger a speaker reveals her version of the common ground. Being cooperative, the listener usually adjusts his version of the common ground accordingly, at least provided the presupposed information is compatible. If the material can not be accommodated, a misunderstanding has become apparent and must be repaired.

Locutionary level

At the *locutionary level* the locution of the utterance, the act of saying it, takes place. Now it matters how something is said. Dialogue participants coordinate on the intonation patterns, on the lexical choice and on the syntactic and semantic structure of utterances. In fact, these so-called 'form' aspects have a direct influence on the coordination of the function and content of an utterance. So although it is conceptually clear to distinguish form, content and function of an utterance, in processing these aspects are intertwined. With respect to lexical choice consider the use of jargon or baby-talk. People adjust their vocabulary to other participants and to the activity type of the conversation. In large scale empirical research on every day spoken Swedish, Allwood (1999) found very little overlap between the words used in different activity types. This shows that for a given activity type the vocabulary is limited, which validates the common practice of developing dialogue systems for limited application domains.

There is also coordination of the syntactic and semantic structure of utterances. Usually this goes under the name of *parallelism*. For instance, Levelt and Kelter (1982) found that when answering short questions people usually copy the preposition that was used in the question. A question phrased like "At what time does your shop close?" was most often mirrored by answers like "At 5 o' clock", while questions phrased as "What time does your shop close?" are almost exclusively responded to by expressions like "5 o'clock", without the preposition. So apparently, answers tend to be formulated with a structure that is parallel to that of the question. There are many other examples of such syntactic and semantic parallelism between adjacent utterances. Parallelism is crucial for the resolution of elliptic expressions. An elliptical sentence is a sentence that appears to be 'missing' a constituent, usually the verb. However, dialogue hardly contains any complete or grammatical sentences. Generally utterances are shortened, because much information is already contained in the context. It is the parallel structure of adjacent utterances that helps to connect the ellided or 'missing' material to the context in the right way (Shieber et al. 1996). Corrections are a special case of ellipsis. Research on corrections shows that parallelism is crucial for determining the scope or antecedent of a correction, i.e. which constituent needs to be corrected (van Leusen 1997; Gardent et al. 1996).

Syntactic structure in general helps to project the end of a contribution. Anticipation makes processing easier and is crucial for turn taking and synchronisation of contributions (Sacks et al. 1974), i.e. for deciding who is allowed to take the next turn and when to start speaking. For example, when the speaker begins an utterance with a wh-word the utterance is likely to be a wh-question. There is a general rule that after a question the turn will be directed to whoever is addressed by the question. Whoever is addressed by a question is indicated by other means. One can address someone explicitly, saying "John, ...?", or implicitly by non-verbal communication signals such as nods or eye-gaze. Often the addressee is simply the person who was speaking before the current speaker. This means that the projected next speaker can already start to prepare an answer to the question as soon as it unfolds. This explains why only few gaps are found between turns of different speakers in conversation (Sacks et al. 1974). In conjunction with syntactic structure, also intonation patterns and rhythm help to project the placement of utterances in time.

Word order, pauses and intonation patterns are used for what has been called *information packaging* (Chafe 1976). Along with the delivery of information elements, it is indicated how they relate to the context or common ground. Certain information elements are indicated to be considered *new* in the common ground. This is indicated by a marked word order, a higher pitch accent, a higher volume of speech and prolonged pronunciation and pauses. By contrast, elements that are considered to be *given* in the common ground, are deaccented. Deaccented words that would have otherwise gotten a pitch accent because of the general sentence rhythm, shift their accent to another word. Given objects are generally described in a different way too. Instead of full noun phrases, shortened definite descriptions such as demonstratives or pronouns are used, and often given elements are simply left implicit (Rats 1996).

Presentational level

At the *physical* or *presentation level* the presentation and delivery of utterances is coordinated. Usually the medium is sound, but other media are possible, such as facial expressions, eye-gaze, body posture and hand movement. The use of these presentation media in communication is coordinated. In a noisy disco you tend to adjust the volume of your speech, speak slowly and clearly and use gestures or facial expressions when possible. In a telephone conversation the opening and closing sequences are more elaborate than those in face-to-face conversation. Because of the lack of non-verbal signals, telephone conversation requires more verbal feedback.

At this presentational level one also finds the coordination of mutual awareness and attention. Gestures for instance, are ostentatious movements. It is not just the movement that counts; other participants must be made aware of it. Therefore gestures are often exaggerated to attract attention. Attention in dialogue is managed partly by intonation; accents draw attention to new, contrastive or otherwise important aspects. Eye gaze is also used to attract and to keep attention. Eye gaze concerns what a participant is looking at; whether he is looking at the speaker or looking away makes a difference (Cassell to appear; Cassell and Thórisson 1999). For example, Vertegaal (1998) shows that eye-gaze information is more crucial to the flow of a conversation than the actual image of a participant's face when talking. A fixed stare usually indicates a misunderstanding (Clermont et al. 1998). Gaze and nodding are also used in a deictic way, to indicate objects.

The account of coordination at different levels illustrated by figure 2.3 is really nothing but a hypothesis. That this picture provides a workable theory is yet to be proven. But based on this picture it is easier to look for general constraints and structures that help shape the coordination interface between processes.

2.3 Coordination Devices

Participants coordinate on the timing and delivery of utterances, on the content, what is actually said and intended and on the progress of the conversation, relating the function of utterances to the underlying task. The effect of all this coordination effort is the establishment and maintenance of what has been called the *common ground* (Lewis 1969; Lewis 1979; Stalnaker 1979; Clark and Marshall 1981). The common ground is defined as the information that is commonly known among members of some well-defined group.

As was illustrated by the ticket reservation example, the common ground covers roughly three aspects. First, the common ground covers a shared background of social, cultural and linguistic conventions. Second, there is mutual awareness of the physical and social situation, including the location and orientation of the participants themselves with respect to their surroundings, and their role in the current social interaction. Third, participants have mutual knowledge of the recent interaction history, including public information on contributions and on the current status of the interaction.

What is the relation between common ground and coordination? The answer lies in the way people solve coordination problems. As we said, the common ground is not only produced by coordination, it also facilitates coordination. What means do people have to coordinate on? Roughly, there are three ways to coordinate: by *convention*, by *agreement* or by *salience*.

2.3.1 Convention

A *convention* is a general rule in a community to solve a coordination problem (Lewis 1969). A convention is often arbitrary, but once adopted, it remains fixed. A typical example of a convention is driving on the right-hand side of the road. Most linguistic signs are conventional. Word meaning or gestures get their meanings by convention; the link between the actual sound or movement and its meaning is arbitrary. But once adopted, the sign is recognised by everyone in the community. Because conventions are shared by a cultural or social community, you need to find out if your conversational partners are members of the same group when starting a conversation. To what extend will they share the same cultural and linguistic background? In other words, you need to establish a common ground between you and the other dialogue participants.

2.3.2 Agreement

In the absence of a convention, you can always invent and establish a new solution to a coordination problem. Thus explicit *agreement* is a way to solve coordination problems. Agreement is based on a mutual knowledge of what was said earlier and on a mutual awareness of the implications of what was said. If we agreed to meet at a cafe, that means I am committed to be there, and so are you. We are bound by what have said. Therefore, a record of previous dialogue contributions, and the commitments resulting from those contributions have to be part of the common ground. Agreements are the result of a negotiation process that involves a number of steps: first information exchange about the general conditions of the agreement, then an exchange of proposals and counterproposals each followed by either a rejection and further counterproposals or by an acceptance, and finally an explicit confirmation of the complete deal.

2.3.3 Salience

How do people coordinate in the absence of a convention or explicit agreement? In general, the most obvious solution in a given situation is the best one. Clark (1996) calls this the *principle of joint salience*: people choose a solution to a coordination problem that is most salient with respect to the common ground. When do we call something salient? First, an object or event is salient when it is contextually relevant. Suppose I say "Open the door!" in a room with several doors. Which door do I mean? I probably mean the door closest to me, or the one that is blocking my path, or one that was last mentioned or used. I assume you understand my reference, since we both occupy the same physical space and we are mutually aware of the position and function of the door relative to me and my current activities. Second, an object or event is salient when it attracts attention or when it has recently attracted attention. Red objects are perceptually salient; they attract attention. That is why fire engines are painted red. Loud noise is salient. A person walking into the room is salient. Deliberate movement of the hands of the speaker is salient. When those changes to the physical and linguistic context are publicly accessible, we call them jointly salient. Since it is likely that jointly salient objects or events attract the attention of all participants, they can be used to coordinate on.

One of the reasons of the importance of contextual relevance, salience and attention, is the fact that we humans have limited cognitive processing capabilities. We have a limited working memory and only a limited capacity for managing separate tasks. Those tasks that do receive these limited resources are said to be 'under attention'. Attention needs to be focussed; we would not want to waste processing capability. Thus, salient objects are objects that are expected to be worthy of our attention. Stable situations or continuous processes need little attention. Habitual tasks that we have done so often that we can do them 'automatically' do not need much attention either. But objects or events that are related to the current task, to changes in the environment or to potential danger, do need attention.

In verbal communication, a specific way of attracting attention is prosody. The distribution of pauses and intonation contours over phrases in an utterance conveys the relative salience of information in the utterance. Therefore theories of the pragmatic and semantic impact of prosody, such as the topic-comment and focus-ground distinctions cannot be left out of any serious account of the common ground. Information that is new, contrastive or otherwise contextually relevant, is typically accented. Information that is given, or implicit in the dialogue context is typically deaccented. All utterances have some intonation contour; usually the accent lands towards the end of an utterance. Deaccenting means that the accent is moved from information elements that would normally receive an accent, to other information elements. So if the speaker chooses to deaccent an information, this shows she takes it to be given in the common ground already.

2.3.4 Common Ground

The contents of the common ground have been given many names. For the term *common* we often also find *mutual*, *joint* or *shared* in similar roles, although these terms are used by different authors in different settings and are thus not completely interchangeable. Often common knowledge is related to *public information*; namely, information that is out in the open for everybody to discover, as opposed to *private information* (Ginzburg 1998; Asher and Lascarides 1998a). How do common knowledge, mutual belief or public information relate to individual knowledge, private beliefs and personal information? One of the ways to study this, is to develop a formal semantics of common knowledge by means of a modal epistemic logic (Fagin et al. 1995). In such a framework one can study how the common ground needs to be changed as a result of changes in public information (Gerbrandy 1999).

There are three ways to think of the common ground. The key to these representations is that they involve self-referential links; this is no coincidence, because by definition the common ground relates to mutuality, and mutuality can be established by reciprocal evidence. The first account starts from a situation s that may act as a *shared basis* from which reciprocal evidence can be derived.

(3) Proposition φ is common ground for group A iff there is a *basis* s such that
   a. every $a \in A$ has *direct access* to s,
   b. to every $a \in A$, s *indicates* that every $a \in A$ has direct access to s and
   c. to every $a \in A$, s *indicates* that φ.

By a situation we mean a partial description of some state of affairs. Depending on the force of the inference relation represented by the word 'indicates' and the strength of evidence of the basis, we get different strengths of mutual attitudes, ranging from mutual knowledge, mutual belief, mutual assumption, mutual expectation to mutual awareness. So the type and strength of the information indicated by a shared basis may differ. The salience of elements in the common ground is related to the strength of the basis for them. For example, red objects form a strong basis for mutually knowing them, and so do recent events.

Once the common ground is established, the basis can be dropped. That leads to the following *reflexive* representation.

(4) Proposition φ is common ground for group A iff
 (i) every $a \in A$ has information that φ and that (i).

The third *iterative* account can be derived from (4) given some basic assumptions about reasoning capabilities, in particular if we model the phrase 'a has information that φ' by a modal epistemic operator $K_a\varphi$.

(5) Proposition φ is common ground for group A iff
 (i) every $a \in A$ has information that φ,
 (ii) every $a \in A$ has information that every $b \in A$ has information that φ,
 (iii) every $a \in A$ has information that every $b \in A$ has information that every $c \in A$ has information that φ,
 and so on indefinitely.

Given a number of assumptions about the notion of a situation, about the 'indicates' relation and about the reasoning capabilities of agents, the reflexive and the iterated version can be derived from the version based on a shared basis (Gerbrandy 1999, p 46). Since both the reflexive and the iterated formulations can be derived, the formulation in terms of a shared basis is the preferred one. Moreover, the idea of a basis as a justification for the common ground is conceptually appealing. Clark (1996) calls it the *principle of justification*: in general members of a community take a proposition to be common ground, only when they have a shared basis as justification for taking that proposition as common ground. That suggests that people will always try to find a shared basis and that this will affect their use of language. This is a hypothesis that can be empirically verified. Much research into grounding methods can be seen as a way of verifying this claim.

Despite Lewis' elegant solution some researchers have focussed on the iterative representation of common ground. Endless sequences of "I know that you know that I know that you know..." were taken as a refutation of the hypothesis of a common ground (Sperber and Wilson 1986). One of the criticisms was that characterisation (5) cannot represent peoples' mental states, since that would require a limitless storage capacity. Also it has been argued that it is unrealistic that we go through the process of establishing this information for each grounded proposition or referent. It is rather the other way around one argues: common knowledge should be so obvious it should be trivial and require little processing effort. However, this criticism misses the point of Lewis' solution. A shared basis is in principle directly accessible and can be established without much effort. In practice people will work with the basis itself; only deriving the full common ground when needed. In particular, the idea of a dialogue game board (Lewis 1979) or a commitment slate (Hamblin 1970) is an example of a representation that functions as a basis for the common ground.

The common ground is an abstract notion; in actual conversation participants will have their own beliefs about what the common ground is. Sometimes the individual versions of the common ground will not coincide. This typically happens when a basis was not recognised.

There is a relation between convention, salience and common ground. Imagine the following coordination problem . Suppose we have agreed to meet at some time tomorrow, but we forgot to agree on a place to meet. At first, we may choose any salient solution. For instance, we try the nearest cafe. Once we know more about each other, we will choose accordingly. Suppose that last time we meat at Café Zeezicht, then Zeezicht will be the most salient meeting place given our common ground that includes a record of that meeting. Now, suppose we actually meet. Next time we forget to agree on a meeting place, we will probably choose Zeezicht again. We have established a weak convention. So in a way, a convention can be established by repeatedly solving a coordination problem in the same way. Each time the solution is used, it becomes more salient. Another way to establish conventions is by explicit agreement. For instance, we can agree to meet at Zeezicht always. Usually a combination of expectations and conventions is used. We have a mutual expectation, but we prefer to agree on that expectation explicitly. By the way, salient solutions to coordination problems are easier to remember, and thus easier to keep committed to.

2.3.5 Grounding

The social and linguistic process by which the common ground is established and by which new information becomes part of the common ground is called *grounding* (Clark and Schaefer 1989). For a computational account of grounding see Heeman and Hirst (1995), Traum and Hinkelman (1992), Traum (1994), Traum (1997), Poesio and Traum (1998). Other relevant work on grounding is that by Walker (1996b, 1996c, 1996a) on redundancy and feedback and work by Allwood et al. (1992) on feedback in natural spoken dialogue. The so called *acceptance cycle* of *inform - propose - evaluate - accept/reject* that is often found in negotiation exchanges is discussed in more detail by Di Eugenio et al. (1998). We give a brief overview of the theory of grounding here.

Each dialogue contribution essentially consists of two parts: a *presentation* part and an *acceptance* or *rejection* part. In the presentation part the speaker presents a signal for the listener; in the acceptance part the listener in turn signals that he received and understood the signal, or not. So grounding signals can be either positive ('go on'), or negative ('go back'). Acceptance can be interpreted at different linguistic levels. So an acknowledgement like "uhum" can mean, "yes, I agree", but can also be interpreted in a weaker way: "yes, I see what you mean" or even "yes, I heard". Many acknowledgement signals are actually non-verbal, and include nods, eye-contact and body posture. Acknowledgements do not always fill a separate turn, but when non-verbal feedback is possible, acknowledgements often overlap with the acknowledged material. Repetitions and other types of utterance that are strictly speaking redundant may function as implicit types of positive feedback (Walker 1996b). A sensible continuation of the dialogue counts as positive feedback too. For instance, a way to acknowledge the receipt and understanding of a question is by giving an answer. Because linguistic research has mostly concentrated on records of written and spoken language, the importance of acknowledgements has long been overlooked.

There is an interesting connection between grounding and theories of presupposition accommodation (Heim 1982; Stalnaker 1979). The link is the notion of the common ground, which is central to both. The point of grounding is to establish what has been 'said' in public. This includes the form, content, function and implications of an utterance. There are several techniques for this. One of these can be the use of *presuppositions triggers*. A presupposition trigger is a particular word, expression or construction that conventionally indicates that certain parts of the utterance are already given in the common ground. Thus, by the way the utterance is formulated or pronounced – deaccenting is a presupposition trigger too – the speaker can show what he or she thinks the common ground is. Because the listener assumes cooperativity in the sense of Grice (1975), the listener will usually accept this implicit information. That means that presupposed material will be *accommodated*: added to the common ground, provided it is compatible. A formal account of this type of presupposition accommodation is given as part of chapter 3.

When the presupposed material is incompatible with the listener's version of the common ground, a clash between the speaker's and the listener's version of the common ground becomes apparent. Therefore, an utterance with a failing presupposition is judged infelicitous. A repair is needed to re-establish the common ground before the dialogue can be continued. The listener is therefore expected to indicate a failing presupposition as soon as possible. On the other hand the speaker can assume that the listener has accommodated the presupposition, at least in the absence of any negative feedback.

However, Clark and Schaefer (1989) claim that the absence of negative feedback is not enough to infer acceptance by the listener. We need positive evidence for the successful completion of a communicative act; communication involves joint actions, and all joint actions require positive evidence upon successful completion or failure. That principle produces an empirical hypothesis: we expect to find positive evidence of acceptance. However, much of the positive feedback is communicated non-verbally, or else is implicit in a coherent continuation of the conversation. What counts as coherent depends among other things on the content of the utterance and its intended contribution to the common ground. To assess this one needs to reason about the utterance. So there is no outright contradiction between a Clarkian theory of grounding and a Stalnakerian account of presupposition accommodation. The first stresses the importance of feedback; the latter stresses the importance of reasoning.

The assumption of presupposition accommodation has been applied specifically for confirmation in spoken dialogue systems (Aust and Oerder 1995; Cozannet and Siroux 1994). Therefore, it is interesting to compare the theory of presupposition accommodation with its application in spoken dialogue systems. This is done in empirical research by Krahmer et al. (1999). In chapter 6 we come back to these experiments, and we discuss the advantages and disadvantages of explicit and implicit confirmation in dialogue.

2.4 Synchronisation

A joint action is a combination of participatory actions, unfolded in time. How does this temporal coordination take place? In this section we discuss some external constraints on participatory actions. Clark (1996) distinguishes periodic processes, like dancing or marching from aperiodic processes, like shaking hands, eating dinner or playing football. Conversation is aperiodic. In a periodic process the repeated synchronisation for each

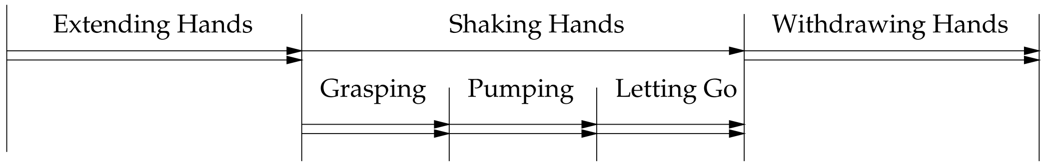


Figure 2.4: Phases of shaking hands (Clark 1996)

period is conventionalised in a rhythm or cadence. Music or beats indicate the rhythm. Pronunciation of words or phrases in an utterance is partly rhythmic. Each utterance must have an intonation centre to help listeners project the end of the utterance. Intonation has many non-rhythmic aspects as well, which have to do with the information packaging referred to above.

There is also a distinction between balanced and unbalanced processes. In a balanced process all participants may initiate actions; in an unbalanced process one participant is leading explicitly. The director leads the orchestra, the man leads the woman in dancing and the speaker leads the conversation. So conversation is unbalanced too. However, these coordination roles are mutually agreed on. The person fulfilling the leading role may change. So although it is the speaker that generally leads the conversation, the roles of speaker and listener alternate. How this happens, is regulated by rules. Rules for initiative management, or rules of turn-taking.

For mixed initiative dialogues we find that usually the participant who is most competent in the particular aspect of the domain that is currently under discussion, has the initiative. For example, in the beginning of a reservation dialogue the user has the initiative; the user asks questions and the system answers them. The user is leading, because in this case the user is most competent in what she wants to find out. Once the desire to make a reservation has been conveyed the system takes over the initiative. The system asks the user for her name, the number of tickets and so on. After all the system is most competent in what it takes to complete a reservation action. This change in initiative also marks a change between the inquiry and the transaction phase of a reservation dialogue. What do we mean by a phase?

In general a joint action progresses in *phases*: periods in which participatory actions are carried out simultaneously. For example, shaking hands can be divided into three phases, as shown in figure 2.4. Since these composite phases are mutually known, participants only have to synchronise on the *entry* and *exit* points of each phase. The entry point of an action, or phase, is the moment at which the action and thus the phase starts. The exit point of a phase is the moment at which the action or phase is completed. In addition we define a *culmination point* that indicates the moment of execution of an action that is most prominent (see section 2.4.3). Participants must be able to project what the entry and exit points will be from what went on before. Therefore we sometimes talk about an entry or exit phase, or opening and closing phase, which is the process leading up to the entry or exit point of the complete process. Synchronisation on entry and exit is easy when there are good reference points: moments in time that are jointly salient. From those the entry and exit points for subsequent phases can be projected.

These assumptions can be formalised to a certain extend in a process algebra like CSP (see appendix A). Consider the process specification (6). Each action α has a public entry and an exit event on which one can coordinate. In between the action is said to be running:

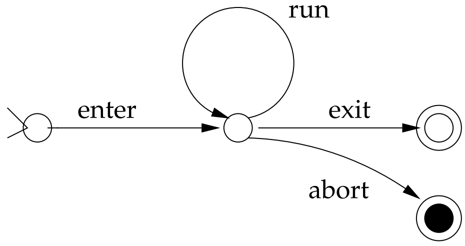


Figure 2.5: Entry and exit points of each action α.

the various sub-actions that compose the body of α are executed. An *exit* leads to a successful completion, an *abort* leads to unsuccessful termination. The corresponding transition system is shown in figure 2.5.

$$
(6)\quad
\begin{array}{lcl}
\alpha & = & enter(\alpha) \longrightarrow Running(\alpha), \\
Running(\alpha) & = & (run(\alpha) \longrightarrow Running(\alpha) \\
 & & \mid abort(\alpha) \longrightarrow Terminated(\alpha) \\
 & & \mid exit(\alpha) \longrightarrow Completed(\alpha)), \\
Terminated(\alpha) & = & \text{STOP}, \\
Completed(\alpha) & = & \text{SKIP}.
\end{array}
$$

2.4.1 Synchronisation Strategies

Clark suggests three common synchronisation strategies (figure 2.6). (1) The *cadence strategy* is limited to periodic activities. Coordination depends on the entry time, duration d of each cycle and for each participant the participatory action to be performed during d. The rhythm determines the duration d of each cycle. Rhythm makes synchronisation easier; think of the activity of digging or cutting down a tree. Rhythm makes sequences easier to remember. Consider the way telephone numbers are structured in triples and pairs of digits. (2) The *entry strategy* can be used for continuous action, consisting of a sequence of phases. This strategy is also called *chaining*. The exit of one phase coincides with the entry to the next. So in this case only the entry time and the participatory action for each participant have to be agreed. (3) In the *boundary strategy* participants coordinate on both the entry and exit points and on the participatory action for each participant. The cadence and entry strategies are special instances of this one. Many unbalanced, aperiodic activities have a jointly salient entry time. Consider the way the director of an orchestra raises his baton to indicate the start of play. Aperiodic phases are often extendible: a phase can be prolonged when necessary, without affecting the other phases. All three synchronisation strategies are characterised by the following *synchronisation principle*: in joint action, participants mainly synchronise on the entry times of phases and participatory actions for each agent.

We can formulate such scheduling constraints by means of CSP. Assume that each joint action α consists of phases α_i coordinated on their entry and exit points, and that each phase again consists of participatory actions a_{ij}. For each series of participatory actions $a_{1j}...a_{nj}$ there is a role x_j, where x_j ranges over agent roles $x_1,..,x_m$. So a_{ij} is the participatory action carried out by x_j in phase i. If we use this terminology we can define for instance the cadence strategy as follows.

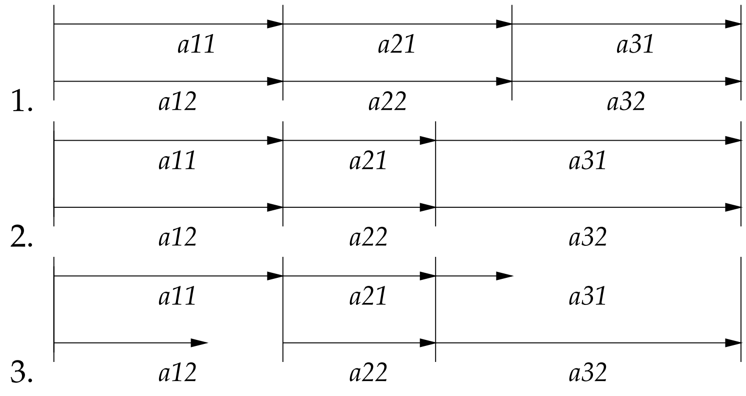


Figure 2.6: Cadence, Entry and Boundary strategies

(7) Cadence strategy

$$
\begin{array}{lll}
\|\alpha & = & \|_{(1\leq i\leq n)}\alpha_i, \\
\|\alpha_i & = & \|_{(1\leq j\leq m)}\, a_{ij}, \\
\text{where} & & enter(\alpha) = enter(\alpha_1), \\
 & & exit(\alpha) = exit(\alpha_n), \\
 & & exit(\alpha_i) = enter(\alpha_{i+1}), \\
 & & t_{enter(\alpha_{i+1})} = t_{enter(\alpha_i)} + d, \quad \text{for some duration } d, \\
 & & enter(a_{ij}) = enter(\alpha_i), \\
 & & exit(a_{ij}) = exit(\alpha_i).
\end{array}
$$

Scheduling participatory actions in phases in general makes the planning activity easier. Planning can be seen as a combination of *task decomposition* and *scheduling*. Decomposition of a joint action produces a combination of sub-actions that are sensible units to be carried out independently. Scheduling means the assignment of an agent, or team of agents to each sub-action as well as a time-slot and possibly other resources. The three phase-boundary strategies mentioned here make scheduling and decomposition a lot more constrained. It could be that the resulting plan is not as efficient as it could have been without these phase-constraints. On the other hand, shared phase boundaries save a lot of coordination effort and therefore reduce the risk of collaboration errors, thus making the result more reliable.

How do these theoretical strategies appear in naturally occurring conversation? Conversation has many distinguished phases with conventionally signalled entry and exit points. Most of these are indicated by prosodic cues, such as intonation, volume and pause. Note that prosody often has a rhythmic quality to it. The rhythm may be supported by beating gestures of the hand. There are distinguished units of conversation at different levels: *discourse* or *dialogue segments*, *turns*, *utterances*, *phrases*, *words*, *syllables* and *phonemes*. The activity of finding the boundaries of these units is called *segmentation*. At most levels it is possible to define a *grammar* or another set of rules or principles that characterises well-formed segments.

The complex process of turn taking, with interruptions and the frequent occurrence of overlaps that go unnoticed, shows that we are experts in synchronising our utterances. There are conventional cues for keeping the turn, like saying ‘uh’. A silence of more than one second already means a surrender of the turn; therefore we try to win time by filling pauses with ‘uh’. Disfluencies like false starts, repetitions and self-corrections are not always a defect. Disfluencies win us time. After all it takes time both to produce and to comprehend an utterance. The relative entry points of an utterance or phrase carry indirect information on participants' mental processing: their current understanding, attention and plans. Clark (1996) relates this to a *principle of processing time*: it is common ground that mental processes take time and that extra processing may delay entry into the next phase. This aspect is related to an aspect of cooperativity that we discuss in section 2.5.1: take your conversational partners into cognitive consideration.

In general, the cognitive processing load of speaking and listening is roughly predictable (Levelt 1989). In language production processing takes longer (a) the rarer the expression , (b) the longer the expression, (c) the more complex the structure of the expression, (d) the more precise the message, or (e) the more uncertain the speaker is about the message. In speech comprehension processing generally takes longer (a) the rarer the expression , (b) the longer the expression, (c) the more complex the structure of the expression, (d) the more concise the expression , (e) the more extensive the implications and (f) the less salient the referents. So processing and content are related: the more complex the content of a message, the longer the processing. All of this nicely correlates with the principle of joint salience. If something is salient, it is generally easy to process, because it does not need any additional memory operations.

Synchrony is needed in conversation because speech is evanescent: it disappears and you can't play it back. So interference must be kept to a minimum. Therefore speech can be seen as a channel that is occupied by one participant only: the speaker. This is the reason that many acknowledgement signs are *collateral*. Nods, gestures or eye-gaze of the listener do not interfere with the speech channel. Such 'go on' signals of positive acceptance are not intended to take over the turn; therefore they can't claim the speech channel. Written language is not evanescent. It does not need to be coordinated on time and there is less room for miscommunication. So in written language the grounding process is less prominent. Yet writing and reading are examples of joint actions too. They are coordinated on aspects like the language, jargon, imagery and purpose of the writing.

2.4.2 Turn Taking

The process by which turns are distributed among participants in time, is called *turn taking*. Sacks et al. (1974) make a number of observations about speaker transitions in naturally occurring conversation. Speakers do take turns, repeatedly. Most of the time, only one participant is talking although overlaps do occur. The majority of transitions are smooth, or have only a slight gap or a slight overlap between turns. Turn size and order are not fixed. The length, the topic and the distribution of turns are usually not specified in advance. Exceptions are institutionalised dialogues, such as court proceedings, meetings or political debates. The number of participants can vary, and may change during the dialogue. New participants must be publicly entered into the dialogue, by glances and greetings. Participants that leave, should publicly announce they are no longer participating, by greetings or ostensibly walking out. In many cases turns are allocated explicitly by the speaker, as when addressing someone with a question, or by looking at the intended respondent. Otherwise, participants may select themselves as the new speaker. Usually the first person to start speaking actually gets the turn. This competition explains the occurrence of overlaps. If no other speaker self-selects, the current speaker may continue. In case several people try to take the turn there is a problem: simultaneous speech is diffi-

1. For any turn, at the first transition relevant place of the first turn constructional unit:
   (a) the speaker may select the next speaker. In this case, the person selected is the only one with the right and obligation to speak,
   (b) else, the next speaker may self-select. The first person to speak acquires the right to a turn,
   (c) else, the current speaker may continue, but need not continue unless someone else self-selects.
2. Rules 1a-c apply recursively for each next transition relevant place, until transition is affected.

Figure 2.7: Turn taking rules (Sacks et al 1974; p 704)

cult to follow. A simple repair mechanism solves this: if you don't manage to get the turn by speaking, because someone else is speaking too, you stop speaking. This explains the brief length of most overlaps. Social relationships play a role in the self-selection process. Dominant persons will usually get the opportunity to take the turn first. If an overlap does occur, dominant persons will usually persist.

There is a very basic set of rules about turn-taking in normal conversation that explains the regularities. These rules are cited in figure 2.7 (Sacks et al. 1974). In rule 1.1 where the speaker selects the next speaker, usually the example of a question is used. We explicitly mention other types of initiatives as well, such as requests, proposals or suggestions: any act that requires an immediate response. Many initiatives are not syntactically marked as a question. For example, a declarative like "I would like to go to Shakespeare" is a very common form of request. The turn taking rules involve two as yet unexplained notions: *turn-constructional unit* and *transition relevant place*.

Turn constructional units are linguistic units from which a turn may be constructed. Almost any type of linguistic unit may serve as a turn constructional unit: words, phrases, utterances or sequences of utterances. Most units are identified by syntactic or prosodic structure. In multi-modal interaction turn constructional units may be accompanied or replaced by non-verbal signs or gestures. Which unit is applicable and which potential boundary is actually used for turn transition depends on the circumstances. That brings us to the second notion. A *transition relevant place* is a unit boundary which is conventionally known to be a point where turns may change. If the turn changes, it changes here. Whether it changes depends on the way the dialogue emerges. The rules are inherently indeterministic.

Consider for example an utterance starting with a wh-word. The wh-word indicates that the utterance is likely to be a wh-question, which indicates the use of a 'current speaker selects next' technique. The next speaker will be the person addressed by the question. Often the addressee is the previous speaker. The end of the question can be projected from the intonation contour. In Dutch, a wh-question can have both a hat-shaped intonation contour, like a declarative, or a rising one, like a yes/no question. In a hat-shaped contour, the pitch stays high and is marked by a sharp drop at the end. As long as the speaker keeps the pitch level high, she indicates that she has not finished speaking.

So far in this thesis, we have been talking of utterances as the basic units. Turns consist of several utterances. What constitutes an utterance? Although there are some features that indicate the boundaries of utterances, such as pauses, intonation contour and syntactic structure, these surface level features are not sufficient to predict utterance boundaries. So instead we use the following semantic stipulation: an utterance is determined by its content and function. The function may be related to the task, or to the surrounding dialogue acts. For example, it may be an acceptance of a previously uttered proposal. So an utterance is a any turn constructional unit that corresponds to a dialogue act. Utterances are the highest level units that can be completed individually. All levels above that involve the grounding effort of other participants for successful completion.

Why do we bother with a theory of turn taking? Hardly any current spoken dialogue systems support *barge-in*, the possibility for the user to interrupt the system, let alone that the system would interrupt the speaker! Still these considerations are important for the design of system prompts. If users project the end of the system's utterance too early, or if a system is to slow in switching from speaking to hearing mode, users may start to speak before the system has opened its microphone. In this way, information gets lost and often a misunderstanding results. Conversely, if the user is surprised by the end of a prompt and starts speaking too late, the system may think the user does not respond at all. Many systems have a delay after which the system will urge the user to respond. Tuning the delay variable and the microphone set-up to a diverse group of users turns out to be difficult, since users may respond in different ways. See the report about the Vote by Voice application in chapter 6.

A very interesting application of turn taking regularities is the automated tutoring system developed by Aist (1998). It is the first spoken dialogue system that can interrupt the user based on the content of what was said. The system impersonates a reading tutor. It asks the pupil to read sentences and complete exercises. The system listens to the user's pronunciation and tries to detect mistakes. Now the amount of time spend before correcting the pupil is of pedagogical importance. A prolonged pause after the pupil's turn may urge the pupil to self-correct. On the other hand, if the pupil makes a mistake and does not realise it, the system should interrupt. In order not to confuse the pupil this must happen at the first turn relevant point after the mistake. This poses an interesting puzzle for theoretical research. For one thing, current speech recognition technology is not accurate enough to judge if the user did make a pronunciation error. Based on reliability measures, estimates of the speech recogniser about its accuracy, a less imposing way of indicating the mistake may be chosen. One way to do this is by cartoon-like face that may frown upon the pupil.

2.4.3 Synchronising Speech and Gestures

So far in this chapter we have talked about coordination between speaker and listeners at different linguistic levels and about coordination between the different levels. However, we have not yet discussed the coordination within a level, between the various interaction channels. Here too, most of the strategies discussed above apply. We now discuss an example of the production of a communicative action consisting of speech and gesture. Take a deictic utterance like "Ik wil die daar!" (*I would like that one*!) , accompanied by a pointing gesture. The timing of the gesture must be aligned with the utterance. We use the following general heuristic.

Like any event, an utterance and a gesture have an *entry* and an *exit* point. Moreover, an utterance can be broken down into the uttering of phrases; each phrase has a so-called

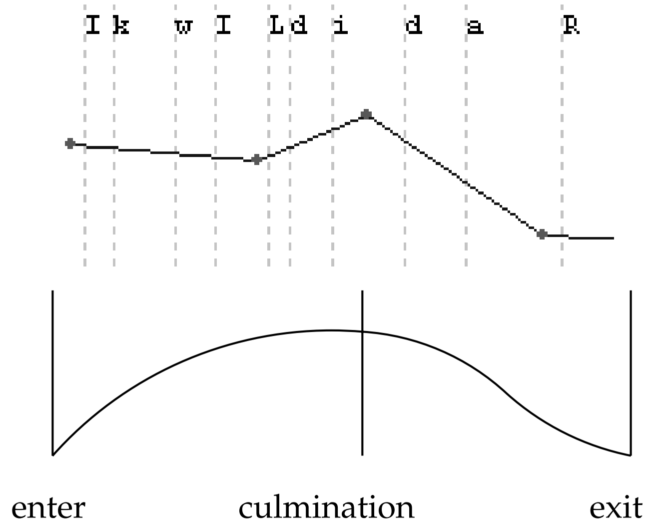


Figure 2.8: Alignment of speech and pointing gesture for the phrase "Ik wil die daar!" (*I would like that one!*).

intonation centre: the moment where the pitch contour is highest. Since pitch accents are related to relative informativeness, we can assume that the accent lands on the most prominent expression. For utterances consisting of several phrases, one phrase has a dominant accent, which is the intonation centre of the complete utterance. The text-to-speech system that we use for pronunciation of of spoken utterances, Fluent Dutch text-to-speech (Dirksen 1997), contains an algorithm for the calculation of intonation centres (Dirksen 1992).

Similarly, each gesture has a *culmination point*. For pointing this is the moment when the index finger is fully extended. In the virtual theatre environment there is a visual animator program that extrapolates a nice curve from the entry point to the culmination and again to the exit point. This trajectory is then played.

Our current working hypothesis is that in general, gestures synchronise with utterances or precede them. So given these programs and the hypothesis, all we have to do is link the gesture's entry and exit points to the entry and exit points of the utterance and make sure that the culmination point of the gesture occurs before or on the intonation centre. An example of this is shown in figure 2.8. The same heuristic can be used to align the raising of eyebrows to words tat are questioned, or that should not be taken literally. For a facial expression like smiling other heuristics are needed. Smiling uses the mouth, and in order to avoid interference with the movement of the lips, smiling only starts after the utterance.

This example is merely used to illustrate that synchronisation issues play a role once you start to combine different output modalities in one interface. Temporal coordination requires very basic principles for alignment. As we have seen, some of these principles can be specified by means of process algebraic methods. However, all of these alignment constraints are external and observable constraints on joint action. They are necessary conditions for successful completion of a joint action, but they are not sufficient. In the following section we look at internal constraints for joint action.

2.5 Joint action

We argue for a view of communication as a joint actions at different levels. A joint action is a complex combination of participatory actions. But what properties make a selection of actions *joint*?

Typical examples of joint actions are playing a duet, or carrying a piano. Carrying a piano assumes careful coordination between the separate actions of the carriers: lifting and lowering must be simultaneous, otherwise someone's back might get hurt. Similarly, voices in a duet need to be coordinated, otherwise the resulting chords are off key. There is an important distinction between a joint action, and a set of actions that merely appear to be coordinated. This is illustrated by the distinction between the (a) and (b) cases of the following examples. Example (8) shows that the distinction is a common one, that is also implicit in normal language use. In general plural sentences in English may have either a collective reading or a distributive reading. The collective reading suggests a joint action. In example (8) background knowledge about the size and use of pianos versus the size and use of guns, indicates which is the right reading. Compare for example, "the artillery men carried a gun" which might be collective.

(8) a. The removal men carried a piano. (collective; joint)
b. The gangsters carried a gun. (distributive; individual)

As a second example, suppose Ann and Ben are students at the music academy who live in the same student house. Ann studies the flute and Ben plays the piano. Both have to audition at the end of the week, on the same piece, a duet. Since they are rehearsing the whole day, it would not be strange to suppose that by accident, at some point they start playing their pieces simultaneously. Both students use a metronome at the same tempo. Now, to a bystander who walks along the opened windows, it sounds like Ann and Ben are playing a duet. Yet, we would not say that a duet – a joint action – is played.

So what is it that turns separate actions into a joint action? The answer lies in the intention behind each of the actions. Actions form a joint action, precisely when they are carried out with the intention of them constituting a joint action. This only works if the participants have common knowledge of a plan that indicates what the participatory actions are, and which agent should do what. In this case, both the plan and the role-assignment are indicated by the musical score. Note that (9) shows the same reciprocal reference as the characterisations of the common ground in section 2.3.4.

(9) a. Ann and Ben are playing a duet.
Ann is playing the flute as part of (9a).
Ben is playing the piano as part of (9a).
b. It appears to a bystander that Ann and Ben are playing a duet.
Ann is playing the flute, adagio
Ben is playing the piano, adagio.

Our running example of a cashier and a customer making a ticket reservation in the theatre is also an example of a joint action. Each participant performs his or her role in the overall activity. Since a ticket reservation can not be made individually – one needs access to both the preferences of the customer and to the reservation database – there is no corresponding non-joint action.

(10) a. Customer and cashier are making a ticket reservation.
Customer is requesting a reservation as part of (10a).
Cashier is booking a reservation as part of (10a).
b. —

Joint action seems paradoxical. People's individual actions are motivated by intentions. You can only intend your own actions; you can certainly not intend other people's actions. So it seems that even though a group can carry out a joint action, it cannot intend it! The paradox is solved because individual actions can be part of a joint action. Such individual actions are special: they are not autonomous but coordinated. Therefore they are called *participatory actions*. Participatory actions are distinguished from individual actions by the intentions behind them. This suggests the first requirement for joint action: each participatory action must be intended to be part of the joint action.

Do we always need to reduce group intentions to individual intentions? Surely, if we allow group action, why not allow group intention as well? Searle (1990) proposes so called *we-intentions*. One of the problems with group intention has to do with the intimate relationship between intention, responsibility and punishment. How should we punish a group that has committed a crime together? People are only accountable for the consequences of actions they commit intentionally. The judiciary system follows the assumption that intentions, and therefore accountability and punishment, are individual. In the Dutch judiciary system individuals can only be punished for their individual actions. A suspect acting as part of a group is therefore only accountable for his share in the group action and for the fact that he did nothing to stop the others. In other words, being an accessory to a crime is a participatory action. This indicates a second requirement for joint action: participants must actively support the other participants in their efforts.

Suppose we would allow group intention as a basic notion. Then we would have to punish the whole group, as a group. And this does happen in some cases, for instance when a whole team gets excluded from a sports competition because one of the athletes used unlawful stimulants. But in the case of a team there is an external authority, the national sports association, that recognises the group as a unity. This suggests that only when there is an institutionalised group structure it makes sense to treat the group as a single agent. Think of a team, a political party, an association, a board of directors or a married couple. However in many cases there is no group structure prior to the joint action. The group is formed in acting jointly. For those groups it does not make sense to ascribe group intention to them. So we agree with Grosz and Kraus (1996) that we-intentions do not make sense as a basic notion and should be reducible to common knowledge and individual intentions, unless the group is an institutionalised unity and can be regarded as an agent in its own right.

So far, we inferred the following principles of joint action. (i) A joint action is characterised by a joint intention, which can be reduced to common knowledge of a plan that defines the participatory actions to be carried out, common knowledge of the assignment of agents to participatory actions, and for each participant the individual intention that their actions are part of the joint action. (ii) Participants are committed to the success of the joint action, and therefore to the success of each participatory action, which means that they must actively support each other. This last aspect is related to the principle of cooperativity.

2.5.1 Cooperativity

If we want to explain communication in terms of joint actions, we must explain why people actually perform their part in a joint communication effort. Why are participants willing to cooperate and coordinate different aspects of their utterances with other participants? For dialogue at least, a common assumption is that participants are cooperative. Grice's cooperative principle reads: "Make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose and direction of the talk exchange in which you are engaged (Grice 1975, p 45)". Focussing on different aspects of the cooperative principle this leads to the well known Gricean maxims of conversation. We have already seen them in chapter 1, were they are used in diagnostic evaluation.

(11)	Quantity	1. Make your contribution as informative as is required.
		2. Do not make your contribution more informative than is required.
	Quality	Make your contribution one that is true:
		1. Do not say what you believe to be false
		2. Do not say for which you lack adequate evidence.
	Relation	Be relevant.
	Manner	Avoid obscurity of expression, avoid ambiguity, be brief and be orderly.

The maxims can be read in two ways. Firstly, the maxims are normative rules the speaker should respect; otherwise she might be misunderstood or become a less reliable conversational partner. But secondly, the maxims are descriptive rules of the expected behaviour of speakers. Therefore, the maxims can be used by the listener as heuristics in understanding. Each of the maxims is a powerful source of expectations in conversation. In this way we can explain conversational implicatures. Conversational implicatures are inferences of the listener about the speaker's beliefs. Or rather, they are inferences about what beliefs the speaker is publicly committed to, based on the fact that she made this utterance and not another. Take the following example of a scalar implicature. Suppose you asked "How did the exam go?" and I replied "It went all-right". From this you can deduce that it didn't go very well. If the exam had gone well, I would have said so. There is an implicit scale, based on the strength of information being conveyed. In this case the scale runs something like very bad; bad; all-right; well; very well. By the maxim of quantity, the speaker is supposed to give an answer that is as strong as she can justify by the maxim of quality. If she doesn't, this indicates that she does not want to commit to a stronger utterance. Another example of an implicature is the case of a speaker blatantly 'flouting' the maxim of quality. If the speaker is obviously not speaking the truth, we can infer that she might be sarcastic or ironic.

Allwood distinguishes four ways to be cooperative in conversation. "Agents are cooperative to the extent that they (i) take each other into cognitive consideration, (ii) have a joint purpose, (iii) take each other into ethical consideration and (iv) trust each other with respect to (i) - (iv)" (Allwood 1995, p18). Now (i) partly corresponds to the Gricean cooperativity mentioned above. It also involves the fact that we reckon with each others' cognitive limitations; hence the importance of salience and relevance. (ii) corresponds to cooperativity at the perlocutionary level. This type of cooperation partially disappears in case of opposing goals. In a political debate by definition participants do not agree. Yet they still have many joint purposes, for instance to respect the chairman and to stage an intellectually challenging coherent argument. The ethical consideration of (iii) comes in because of the repercussions of being impolite, insincere, or unclear. Such defective behaviour is judged wrong; it is more than just inept or inadequate for achieving one's goals. The mutual trust mentioned under (iv) allows listeners to use the maxims as reliable interpretation guidelines; we do not only expect the other participants to adhere to the maxims, we trust them to. When we engage in a conversation, indicated by an exchange of greetings, nods or eye-gazes, we are thereby committed to its successful completion. Walking out of a conversation is a violation of this commitment.

So cooperativity comes at different levels too. One can be cooperative at one level, without being cooperative at a higher level. On the other hand, cooperativity at one level enforces cooperativity at lower levels. The Gricean type of cooperativity which occurs at the illocutionary level and the levels below, is derived. It follows from the common goal at the perlocutionary level and the requirement to coordinate participatory actions. Now it may seem strange that even in adverse situations, when the perlocutionary goals of participants are opposed, participants usually still cooperate at the level of dialogue acts. In a debate, in court or in a cynical student house the Gricean maxims still seem to hold. The reason is that these adverse activity types can still only be carried out by joint actions at the illocutionary, locutionary and presentational levels. Even if you make an insult, you want it to be recognised as an insult. And if you are insulted, it helps to understand the insult so you can counter it. So in adverse activity types you must still coordinate the meaning and presentation of your utterances. By contrast, consider what happens if someone breaks down in crying. In that case he is no longer willing to keep up the argumentation and resorts to other means to have his way: emotional manipulation. Or consider what happens if someone looses control and starts hitting the other. Again, the normal repertoire of interaction appears to be inadequate and the hitter must resort to physical action. Finally, consider a soap-opera couple that is fighting by means of deliberately ignoring each other. Here, non-cooperation at the presentation level is used as a tool for interaction. But even that requires coordination, because the silence must be understood as deliberate.

It seems there is some pattern here: we start from perlocutionary goals and social obligations that follow from the activity type. These lead to joint goals and therefore obligations at the illocutionary level, and again to joint goals and therefore obligations at the locutionary and presentational levels. In each case, cooperativity at a lower level is derived from a common purpose one level higher. On the other hand, the lower level actions instantiate the higher level ones. The locutionary, illocutionary and perlocutionary aspects are all aspects of the same utterance event. We can describe this pattern more systematically.

The levels that we saw depicted in figure 2.3 form a so-called *action ladder*. The following example is from Austin (1962).

(12)		4. John shoots a donkey.	
	evidence ↓	3. John fires a gun.	
		2. John pulls the trigger.	↑ completion
		1. John tenses his index finger.	

Actions in an action ladder occur simultaneously and often at the same location. Action ladders are characterised by a causal link upwards: John is pulling the trigger *in order to* kill the donkey. Causality relations in an action ladder are asymmetric and transitive. We have the following two properties.

(13) *Upward completion*: in a ladder of actions it is only possible to successfully complete an action when all sub-actions in lower levels have been successfully completed.

Downward evidence: in a ladder of actions, evidence that an action at one level is successfully completed, is also evidence that all sub-actions in the levels below it are successfully completed.

Many such action ladders occur in communication too. The causality of communicative intention is a little more complicated. We do not say that we want two tickets *by* uttering "I want two tickets" but *in* uttering that. The uttering itself, the locutionary act and its meaning, its illocutionary force, are two aspects of the same action. Only for obvious perlocutionary acts, like saying that you would like somebody to close the door by means of saying "Oh, it's cold in here" a causality relation holds. Although it is not a causality relation, the properties of upward completion and downward evidence do hold for joint illocutionary acts. Successfully requesting two tickets, is evidence for successfully meaning that two tickets are wanted, which is again evidence for having completed an utterance like "I want two tickets" or "Two, please!". There are more ways of formulating a particular meaning, so many different utterance events may instantiate a particular illocutionary act. The reason that upward completion and downward evidence still hold, is that there exists a *dominance relation*: the higher level action dominates the lower level actions.

Example (14) shows the action ladder involved in the earlier example of the user (u) selecting a performance by pointing at the poster, and the system (s) attending to this communicative action. The example illustrates that the levels of figure 2.3 apply to gestures and deictic reference as well.

(14)
4. (u proposes s to sell tickets for Twelfth Night $\|$ s considers to sell tickets)
3. (u indicates the performance Twelfth Night $\|$ s understands the performance)
2. (u points at poster of Twelfth Night $\|$ s identifies u pointing at poster)
1. (u moves index finger towards x, y, z $\|$ s attends u's finger movement)

So there are systematic relationships between actions at higher and lower hierarchical levels. These hold for the composition of both joint actions and individual actions and also for the composition of joint actions out of individual participatory actions.

2.5.2 Joint Planning and Action

How do scheduling constraints relate to joint actions? We briefly discuss an account of joint planning and action (Grosz and Kraus 1996) . A more elaborate account of planning and action is presented in chapter 4.

There are two ways in which one can be said to 'have a plan' (Pollack 1990). One can have a plan as a recipe for action. This version has been called the 'data-structure' view. A plan is something that can be reasoned about, negotiated, extended and remembered, just like other information or beliefs. We will call this version a *recipe* from now on. Or one can have the plan to act. Here we see an adopted plan as an intention or a commitment to future action. The idea is that adopted plans are carried out, until some success condition or failure conditions is met. The agent's goals are persistent: the agent will continue to try until the action is accomplished or when it becomes infeasible that it will ever be accomplished. Intentions make sure that the agent does not have to reconsider its actions after every step; they produce stability. Both views are needed, as is demonstrated in the BDI logics of agent hood (Georgeff and Lansky 1987; Bratman 1987) that are discussed in chapter 4. The philosophical notions of belief, desire, and intention give a semantics to planning and scheduling algorithms. Goals and plans are the objects of such algorithms. Desires prescribe combinations of states of affairs as they are desired by the agent. Means-end reasoning is applied to the desires, generating a set of possible recipes for action. In the deliberation process, the cost of various actions as well as the likelihood of success are reckoned with. The recipe that is adopted, is the intended plan. So intentions are adopted recipes. The description of the end-state of a recipe is called the goal. So goals are end-states of adopted recipes.

Planning consists of two phases: task decomposition and scheduling. Both are subject to a number of constraints that determine which plans will be well-formed. To assess the suitability of an action, each action must be characterised by a number of conditions. The conditions are defined for each basic action and can be projected for composite actions. We mention *preconditions*, the propositions that must be true at the start of an action in order for it to be successfully completed, and the *postconditions*, the conditions that will be true after a successful completion. So basically, if the preconditions are true, the post-conditions are guaranteed. Often we explicitly define *failure conditions* too. These specify what is the case after an action is aborted. We also specify conditions that determine when an action will be typically attempted: *applicability conditions*. Concrete examples may be found in the use case of chapter 1, and in chapter 4.

In addition to conditions on single actions there are general constraints on the well-formedness of a plan. These correspond to synchronisation and scheduling constraints, as in section 2.4, constraints on the allocation of resources like time, money and locations, and constraints that have to do with the well-formedness of the linguistic realisation of a plan. In general scheduling constraints can be reduced to three basic types: *precedence* and *dominance* (cf. Grosz and Sidner 1986) and *alignment*. Dominance was discussed above in the context of action ladders. It defines the clustering of sub-actions into composite actions. Precedence puts a constraint on the order in which sub-actions can be carried out. The third constraint for *alignment* accounts for coordination between simultaneous processes.

(15) An action α *precedes* action β when α must be completed before β can start.
$t_{exit(\alpha)} < t_{enter(\beta)}$

(16) An action β *dominates* action α, when completion of α is part of completion of β.
$t_{enter(\beta)} \leq t_{enter(\alpha)}$ and $t_{exit(\alpha)} \leq t_{exit(\beta)}$

(17) An action α *aligns* with action β when the entry and exit of α and β are simultaneous, and the culmination point of α precedes the culmination point of β.
$t_{enter(\alpha)} = t_{enter(\beta)}$ and $t_{exit(\alpha)} = t_{exit(\beta)}$ and $t_{culm(\alpha)} \leq t_{culm(\beta)}$

By notation t_{enter} we mean the time point at which the *enter* event occurs. Although we do not explicitly model time, a timed version of CSP can be defined if needed (see appendix A). Notation *pre*(α) and *post*(α) denote the pre- and postconditions of an action α. The precedence constraint is often derived from a logical dependency: *pre*(β) $\rightarrow$ *post*(α), or in other words, the precondition of β is only satisfied when α is completed. The dominance constraint typically derives from a hierarchical decomposition of complex task into easier sub-tasks. Dominance corresponds to the following dependency: $post(\beta) \rightarrow post(\alpha)$. The completion of β involves completion of α. This dependency ensures the properties of downward evidence and upward completion. The alignment constraint is added to account for simultaneous interaction. Here the actions are mutually dependent. The postconditions of both α and β are true only if the preconditions of α and β have been satisfied: $(post(\alpha) \wedge post(\beta)) \rightarrow (pre(\alpha) \wedge pre(\beta))$.

Now we have enough formal machinery to define what a recipe for action is. A recipe is an underspecified specification of the decomposition of a composite action into sub-actions. The recipe can remain underspecified, as long as it conforms to the constraints. So, formally we may equate a recipe with a set of constraints C defined over the set of sub-actions $\beta_1, \ldots, \beta_n$. Each set of constraints C defines a number of possible configurations. Each configuration can be written as a process algebraic expression $\rho(\beta_1, \ldots, \beta_n)$, using the connectives $\longrightarrow$, $\|$, and $|$ (see appendix A). That means that the expression $\rho(\beta_1, \ldots, \beta_n)$ can be used as a notation for a task decomposition: a particular possible configuration that instantiates a recipe for action and that satisfies the constraints C.

In addition, each recipe contains roles. Each role defines the agent that should execute a compatible series of sub-actions. If we use notation $x_1, \ldots, x_m$ for the different roles in a plan, we can write $\rho(\beta_1, \ldots, \beta_n)_{x_1,\ldots,x_m}$ for the recipe with roles attached. Part of the scheduling effort is to assign an agent or team of agents to each role. So collaborating agents must agree on a 'division of labour' $g : \langle x_1, \ldots, x_m \rangle \rightarrow \text{pow}(Agents)$. A similar argument runs for time slots in a schedule and for other crucial resources. For example, if there is only one screw driver, no two agents can be expected to use the screw driver at the same time. Similarly, no two agents can be assigned the speech channel at the same time. Here too roles help to structure the scheduling. The cadence strategy that we described in (7) is a common solution of this general assignment problem.

We can either give a declarative formulation of plans in terms of the end-goals and sub-goals or we can give a procedural representation in terms of actions and sub-actions. An agent can have a plan to achieve a goal φ or a plan to carry out an action α. The first way of specifying plans is useful at a high level of abstraction; the latter is useful at a more implementation oriented level of abstraction. Consider for example the alignment of dancing steps, which would be very difficult in terms of goals. In a formal model, either formulation can be turned into the other. We suppose special actions $bring\_about(\varphi)$ and $achieve(\alpha)$ to translate one representation into the other. We require that the action $bring\_about(achieve(\alpha))$ is identical to the action α and that $achieve(bring\_about(\varphi))$ holds precisely when φ holds.

To summarise, each action is characterised by preconditions, postconditions and applicability conditions. A recipe for a joint action is characterised by a common goal, the *objective* of the recipe. This can be either a proposition φ or an action α. Given an objective φ, there will be a repertoire of applicable recipes: $Recipes(\varphi)$. Within a recipe the possible configurations of sub-actions $\rho(\beta_1, \ldots, \beta_n)$ are determined by a set of constraints C. For joint actions and plans the recipe is structured around roles $x_1, .., x_m$. These roles have to be allocated to agents or teams of agents by an assignment g, and to each role time slots and other resources must be assigned by h. So the following aspects must be agreed for the completion of a joint action with objective φ.

(18) 1. Agree on common goal: φ,
2. agree on possible recipes $Recipes(\varphi)$,
3. agree on optimal $\alpha \in Recipes(\varphi)$ namely
a. agree on decomposition $\alpha = \rho(\beta_1, .., \beta_n)_{\langle x_1,...,x_m \rangle}$, such that $\alpha \models C$,
b. agree on division of labour $g :: \langle x_1, ..., x_m \rangle \rightarrow \text{pow}(Agents)$ and
c. agree on schedule $h :: \langle x_1, ..., x_m \rangle \rightarrow \text{pow}(Resources)$ and
4. commit to $\rho(\beta_1, .., \beta_n)_{\langle x_1,...,x_m \rangle}(g)(h)$.

The question remains how these aspects are coordinated. There are roughly four ways: by negotiation, by some hierarchical command structure, by some protocol that was agreed before or was known by convention, or by salience. The possibilities roughly correspond to the coordination devices convention, agreement and salience discussed in section 2.3.

Negotiation proceeds by what has been called the *acceptance* cycle, which is essentially a way of grounding mutual commitment (Walker 1996b; Di Eugenio et al. 1998). If there is some hierarchical command structure, agents can order other agents to take on a particular task. Another way of *contracting out* certain tasks is by asking other agents; in this case a negotiation will follow to agree on the terms and compensation for the effort. Usually there is already some fixed protocol, established by convention or by earlier agreement, to connect agents to roles. In the case of ticket selling, agents are assigned roles by their physical position relative to the counter. Just like for other aspects of coordination, salience is a last resort. In the absence of any other explicit coordination of roles, participants will usually take the most obvious or salient solution. An example is the assignment of the speaker role in the beginning of a conversation. Social research shows that the person that continues looking at the others for the longest period, will be judged the most dominant; this person will usually also be the one that starts speaking. Note that the assignment of roles does not always succeed. Mistakes can be made here just as well as with other aspects of communication.

An example of a system that makes use of the first three of these methods is the CMUnited Robocup team, developed at Carnegie Mellon (Stone and Veloso to appear). The team has been successful in the Robocup, a virtual football competition. Teams of robots from different universities play soccer matches against each other. There are two types of communication in the Robocup application domain. When the team is 'off-line', full and reliable communication is possible. This is before the game or during half-time. But when the team is playing, communication must happen real-time, at a low bandwidth and over an unreliable channel. That means that negotiation can only be used before the game and during the break. During the game, the team must use a fixed protocol. The solution to the coordination of tasks of the CMUnited team is called a *locker-room agreement*. The team plays in different formations, for instance attacking or defending. A formation is a set of roles, with clearly defined tasks. Roles are grouped into units with a captain. There is a mapping from agents onto the roles. The mapping is flexible; it may change during the game. Changes are established by mutual coordination between those agents that are affected; for instance, a defender covers another defender's zone. Changes may be ordered by the unit captain. In addition, the team formation may change at certain agreed trigger conditions, for instance at set plays (corner, penalty) or when the team is behind and there is little playing time left. Like football, a conversation does not need to be planned all the way down to the level of kicks. A locker room agreement is in effect a convention for scheduling sub-tasks in a joint effort. In the rest of the thesis we hope to find similar conventional recipes for joint interaction.

2.6 Conclusion

In this chapter we explained a number of fundamental notions that are related to the basic concept of coordination. We described natural language dialogue as a combination of joint actions at different linguistic levels, scheduled in parallel. We introduced the notion of common ground and the process of grounding and explained that there are basically three coordination devices: explicit agreement, convention and salience. We described a number of synchronisation strategies based on the entry and exit points to process-phases in a dialogue process. To model synchronisation we suggested process algebra, in particular CSP. The importance of synchronisation is even more pressing for multi-modal interfaces. Design techniques for these kinds of systems are the subject of continuing research at Twente (van Schooten et al.1999). We explained that cooperativity in dialogue is a derived notion; derived from the adoption of a common goal. Finally, we gave a sketchy account of a theory of joint planning and action, and showed the need of a repertoire of commonly known recipes for interaction.

Our talk of joint actions suggests a theory of planning and action. But is this a suitable way to go? It seems not! Communication, conversation and dialogue are essentially *opportunistic*. Dialogue structure emerges from the local decisions of participants to make a particular contribution in reaction to the contributions of others. Moreover, dialogue is dynamic: what is said depends on the context, but also influences the context itself. Although the underlying task may severely constrain the possible course a dialogue might take, dialogue itself is not a planned activity. In a recording, a dialogue appears to have been planned. Every utterance has some function in relation to the general task or goals, at least when looking back and explaining the utterance in the light of the context. But even for highly restrictive tasks the order in which particular sub-tasks are handled is not completely determined beforehand. Misunderstanding may occur, and clarification exchanges may cause a sub-task to be interrupted or suspended until further notice. Participants may pursue several goals simultaneously. Participants postpone making a decision, or display a kind of browsing behaviour touching upon topics without any noticeable plan. Even if they do have a plan or goal, participants can always change their minds, abandon the original goal and start pursuing a new one. That doesn't mean that plans and goals are useless for describing linguistic interaction; it just means that a traditional planning framework, such as STRIPS (Fikes and Nilsson 1971) is not sufficient to capture the diversity of possible interaction patterns.

There are roughly two ways to deal with the opportunistic nature of dialogue. The first is called *partial planning*. It stays within a goal-based planning paradigm. Plans are not required to be complete. A partial plan is a plan 'schema', that is only sufficiently worked out that it makes sense to embark on it. During interaction the plan is further adjusted. Planning involves an underspecified representation of the possible future course of actions. Based on a repertoire of such *recipes for future action*, the expected optimal solution is chosen. At each point many options are kept open. Heuristics decide which option will be chosen for execution or for further adjustment. Several formalisms of partial plans exist. Grosz and Kraus (1996) developed their theory of collaborative planning and action in a framework of partial plans. Although the framework seems sound, a lot of research still needs to be done on the algorithms that decide when a scheme has to be filled out.

A second way to deal with the opportunistic nature of dialogue is the metaphor of a *dialogue game*. Based on Wittgenstein's (1953) notion of a language game, the game metaphor is a powerful one. For dialogue it has been explored by researchers from various backgrounds (Levin and Moore 1978; Mann 1988; Carlson 1983; Kowtko et al. 1992; Carletta 1996; Carletta et al. 1997). Dialogue contributions are like the moves in a game. Lewis (1979) compares the notion of a common ground with a *dialogue game board*, on which a record of the dialogue is kept. Based on the game board, only some moves are allowed and other are prohibited. Based on the game board the player must select a move that maximises the chance of winning, given assumption about what the other players will do. So the dependency on other participants' moves is build right into the formalism.

A dialogue game is a useful structure to describe the rights and obligations, as well as the goals and possible actions of participants at different stages in an activity of a particular type. Like a script or frame (Schank and Abelson 1977), a dialogue game structures human cognitive processing; based on previous moves participants form expectations about new moves. But unlike a script or a plan, the expectations based on a game are essentially normative. You ought to address a question; even when you don't know the answer you are bound to react to it, showing you have understood that a question was asked. Breaking or bending the rules always remains possible, but this will have social effects. When a player breaks the rules, she risks putting herself 'off-side'. Breaking the rules makes one less reliable as a partner. There are certain interaction patterns in communication behaviour which lead to obligations for language users. Dialogue games are build up from such *exchanges*: question-answer, proposal-acceptance, request-compliance, or in general, initiative-response patterns. For this reason dialogue game rules are relatively reliable ways of getting other people to do something. Each successful occurrence of an interaction pattern results in some mutual or joint goal being accomplished.

We believe that the two approaches to the dynamic and opportunistic nature of dialogue, partial plans and dialogue games, are in fact complementary. We need a theory of planning and action to explain how dialogue acts are motivated by goals. On the other hand we need a repertoire of recipes for joint action. It will turn out in chapters 4 and 5 that the interaction patterns which can be described in terms of dialogue games are just that: conventional recipes for joint action. They are in fact the smallest actions that can still be called joint. But before we start to look at the task structure and the interaction, we first turn to the content of utterances in chapter 3.

Appendix A: CSP

CSP is a formalism for reasoning about patterns of communication in distributed systems. Components are described in terms of *processes* which evolve and interact by a series of atomic actions called *events*. Each event means instantaneous communication, and serves as a synchronisation between processes. CSP does not have a direct way of expressing other types of eventuality, like states or culminated processes. These however can be defined. Many extensions of CSP have been defined and used successfully in practice (Hinchey and Jarvis 1995). For instance, there is a timed version of CSP. Here we will do with a rather restricted version.

CSP is a process algebra. Terms are defined using a set of equations. Terms may be rewritten according to algebraic laws. There are several types of semantics to justify the laws. The simplest model is that of a *trace*: a sequence of observable events. Thus, two processes are trace-equivalent, when they are observably equivalent. In many cases it

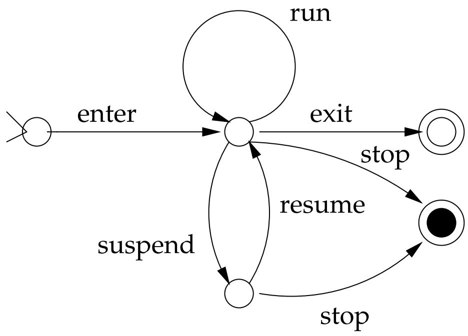


Figure 2.9: Transition diagram of example (19).

helps to think of a CSP expression as a specification of the transition relation of a labelled transition system (LTS). An LTS is a directed graph. Nodes represent possible process states, labels represent possible events. The transition relation is depicted by means of arcs: it defines what states are related by which events in the process.

Here is a simple example of a transition system. The notation will be explained below. Again, it models the fact that every instance of a process consists of phases with explicit *enter* and *exit* points, on which it can be coordinated. In addition, these processes can be suspended and resumed as well. A suspended process does no longer occupy a channel.

(19)
$$\begin{array}{lcl} Process & = & (enter \longrightarrow Running \\ & & \mid stop \longrightarrow Terminated), \\ Running & = & (suspend \longrightarrow Waiting \\ & & \mid exit \longrightarrow Completed \\ & & \mid run(a) \longrightarrow Running), \\ Waiting & = & (resume \longrightarrow Ready \\ & & \mid stop \longrightarrow Terminated), \\ Completed & = & \text{SKIP}, \\ Terminated & = & \text{STOP}. \end{array}$$

The resulting transition system of example (19) is depicted in figure 2.9. As usual, nodes depict process states, possibly labelled with a process label. Directed arcs depict transitions, labelled with their event names. A starting state is indicated by a small $>$ leading into it; successful and unsuccessful end states are indicated by a white and black surrounded circle respectively.

How is the process definition of example (19) built up? Each specification consists of a number of equations of the form $X = P$, where X is a process label and P a process expression. Process labels X act as non-terminals; they are usually written with a capital letter. Process expressions are build inductively from a set of atomic events $\mathcal{A}$, the alphabet. For a given specification P, we can always compute the alphabet of that process $\mathcal{A}(P)$. So for example $\mathcal{A}(Process) = \{enter, stop\}$ and $\mathcal{A}(Waiting) = \{resume, stop\}$. Because $\mathcal{A}(Process) \cap \mathcal{A}(Waiting) \neq \varnothing$ we can see from the alphabets that the two processes share events; they are not independent. In example (19) the end of an entering process is indicated by a SKIP event. In general, there are two basic actions that may end the execution of a (sub)process: STOP and SKIP. They differ in what happens afterwards. After a STOP signal, all activity ceases. A STOP signal explicitly models a dead-lock situation.

After a SKIP signal, other sub-processes may continue. Such an 'alive' process is indicated by a $\surd$ sign. In picturing transition systems we use the convention of painting STOP nodes black, and SKIP nodes white.

Here is the formal syntax definition. The syntax is defined in two stages. First we define process expressions P, and then we define a process description as a set of process equations $X = P$, where X is a process label.

Definition 1 (CSP Syntax)
Given atomic events $\mathcal{A} = \{a, b, \ldots\}$ define a process by
$P ::= \ \text{STOP} \mid \text{SKIP} \mid (a \longrightarrow P) \mid (P|Q) \mid (P\|Q) \quad (P, Q \text{ process }, a \in \mathcal{A})$.
Given process labels $\mathcal{X} = \{X, Y, \ldots\}$, define a CSP specification by
$X_1 = P_1, \ldots, X_n = P_n$. □

The clauses have the following meaning.

$(a \longrightarrow P)$ is a prefixing operation. It is the central language construct in CSP. It models a single event transition: first event a is observed, after that process P continues. Note that a must be an atomic event; this constraint simplifies the processing of expressions.

$(P|Q)$ models non-deterministic choice. It is the environment, i.e. processes external to P and Q that make the decision. In some versions of CSP one also finds an internal version of choice, denoted by $\sqcap$. Roughly $((a \longrightarrow P)|(a \longrightarrow Q)) \equiv (a \longrightarrow (P \sqcap Q))$. For now, this distinction is not relevant. A combination of prefixing with choice, can be called a guarded choice: $((a_1 \longrightarrow P_1)|\ldots|(a_n \longrightarrow P_n))$. Event a_i is called the guard of P_i.

$(P\|Q)$ denotes parallel composition. Processes P and Q run independently, as long as they do not share any atomic events. But when they do have an event in common P and Q must synchronise on that event. The underlying notion of CSP specifications is that of *interleaving*; for modelling the success and failure of parallel processes it does not matter whether processes are really scheduled in parallel. One may run before the other, or they may run interchangeably. The distribution of steps may not even be fair. What matters is that the set of possible observable behaviour contains all interleaved combinations of steps.

Where it does not lead to confusion, parentheses are dropped. Because choice and parallel composition are associative operators, we can freely use expressions like $(P_1\|\ldots\|P_n)$. An expression like $(P_1\|P_2\|\ldots)$ is usually written $\|_{i\in I}P_i$ for some set of indices I. By convention, process labels X that are defined as parallel composition, are prefixed by $\|$ as well.

One of the advantages of CSP is the frequent use of notation conventions. Consider the following input-output convention. Notation $c!x$ is shorthand for a channel c that sends output with the value of the output being x. Likewise, $c?y$ is shorthand for a channel c that receives input, the value of which is y. Note that because the channel c is shared, processes must coordinate $c!x$ and $c?y$, which results in a substitution $y = x$. In the following example we use parameters enclosed in brackets to indicate the various stages of a process, in this case numbers to be added. External functions like '+' can also be applied. The diagram corresponding to this example is shown in figure 2.10.

(20)
$$\begin{aligned}
Add &= A_{\langle\rangle},\\
A() &= in?x \longrightarrow A(x),\\
A(x) &= in?y \longrightarrow A(x,y),\\
A(x,y) &= out!(x+y) \longrightarrow A().
\end{aligned}$$

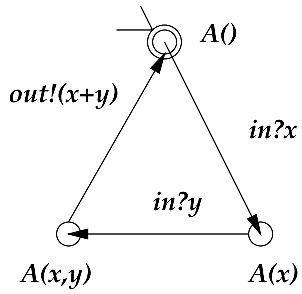


Figure 2.10: Diagram of the Adder example (20)

Often it is convenient to add a notion of time to process algebras like CSP. A common way to do this, is to think of a *Clock* process running in parallel to all other processes. All events a must synchronise on some tick t of the clock process. This t is then defined as the time of a: t_a.

(21) $$\begin{array}{lll} Clock & = & Clock_0, \\ Clock_t & = & tick(t) \longrightarrow Clock_{t+1}. \end{array}$$

The semantics of a CSP specification is defined in terms of labelled transition systems. Each expression P defines a particular labelled transition system M_P. Process names correspond to the names of nodes, events correspond to the labels on the transitions. The process definition itself determines the transition relation. For the details of the semantics, we refer to Hinchey and Jarvis (1995).

This concludes the overview of the CSP specification language.

Chapter 3

Inquiry

This chapter provides a formal model of inquiry: cooperative information exchange. The meaning of an utterance is interpreted as the contribution it can make to an information state. Information states are conceived of as structured by one or more issues: contextually given questions under discussion. An assertive utterance adds factual information; an interrogative adds structure. An utterance can be called relevant when its information content helps to resolve one or more issues. An utterance is licensed when it resolves only current issues. Like consistency and informativeness, which can be defined in terms of logical entailment, relevance and licensing are formal dialogue properties that characterise a coherent dialogue.

3.1 Introduction

A dialogue is an exchange of utterances between dialogue participants. Each utterance depends on the dialogue context for interpretation, but also contributes to the dialogue context. The meaning of an utterance can be seen as the contribution it will make to a dialogue context. This is the basic idea of what has been called dynamic semantics. The meaning of an utterance is its context change potential; a transition relation from context to context. An utterance is characterised by a phonetic and syntactic *form*, a semantic *content* and a communicative *function*. The form of an utterance helps to identify it; it indicates the content and most of its function. The content of an utterance is essentially the information conveyed by it. The communicative function of an utterance indicates the type of effect the utterance will have on the dialogue context. For instance, assertions mean to add information; questions mean to retrieve it. Together, the communicative function and the semantic content comprise a dialogue act. In this chapter we concentrate on the semantic content of an utterance, and how it contributes to the information in a dialogue context.

As we saw in chapter 2 the dialogue context is essentially the common ground: the information that is commonly assumed to be known to all dialogue participants. The common ground is a useful notion, but remains an abstraction. Every participant keeps a personal record of the apparent common ground with evidence from the dialogue as a basis. In making an utterance the speaker conveys part of what she believes. The hearer will subsequently adjust his version of the common ground, to the effect that it is now publicly known that the speaker is committed to the content of her utterance. Depending on the reliability of the speaker and on the relation of the content of the utterance with

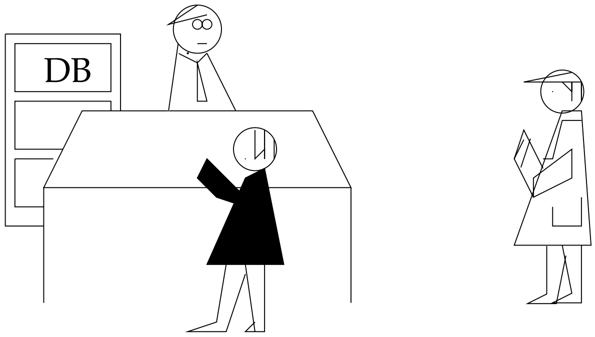


Figure 3.1: Inquiry dialogue situation, with expert, inquirer and observer.

his own beliefs, the hearer will update or revise his private information state accordingly. Because the formal characteristics of this dynamic behaviour of the common ground are complicated, we start by studying a simplified dialogue game, called inquiry (figure 3.1).

The dialogue game of *inquiry* is defined as the exchange of information between two participants: an inquirer and an expert, witnessed by a neutral observer. The inquirer has a certain information need. Her goal in the game is to ask questions about the domain in order to satisfy here information need. The expert has access to a database about the domain. His goal is to answer questions. There are two kinds of utterances: assertions and questions. Both participants are allowed to make one or more utterances in turn, of both types. So the expert may ask questions too. We assume that the expert's information about the domain is correct and that the inquirer is sincere in expressing her information need. Given these assumptions, the information in the dialogue context can be portrayed as the information state of the neutral observer. The game ends when the inquirer has no more questions left to ask; her information need is satisfied.

In this chapter we develop a theory of inquiry which models the content of utterances and the contribution they make to the accumulated information in a dialogue context. The objective is to define a number of logical properties that help to characterise what constitutes a coherent dialogue. Apart from *consistency* and *informativeness* which can be defined in terms of logical entailment, the property of *relevance* is crucial in this respect. The property of *pertinence* expresses that an utterance is not over-informative. These properties are the formal counterparts to certain aspects of the Gricean maxims of *quality*, *quantity* and *relation* (Grice 1975). Formalising the Gricean maxims is one of the motivations of Groenendijk and Stokhof's work on which much of this chapter is based. Groenendijk (1998, 1999) defines a game called *interrogation* which is very similar to inquiry.

The colloquial term 'inquiry' covers all kinds of interactive information exchange, from customer queries in a shopping centre to criminal investigation. The term is also chosen as a tribute to Stalnaker's (1984) work on information change from which the following quote is taken. The quote illustrates the topic of this chapter.

> "To *understand* a proposition – to know the content of a statement or thought – is to have the capacity to divide the relevant alternatives in the right way. To *entertain* a proposition is to focus one's attention on certain possibilities, contrasting them with others. To *distinguish* two propositions is to conceive of a possible situation in which one is true and the other false."(Stalnaker 1984, p4,5)

3.1.1 Relevance

Every day we are bombarded with a multiplicity of information. Books, conversations, e-mails and the Internet might produce an information overload. Yet we are able to structure such large amounts of data. We select the most salient information, information that meets our needs, answers our questions or helps to solve our problems. In short, information that is relevant in the given context. Information is exchanged by means of utterances. This chapter addresses the following question.

> When can an utterance be called relevant with respect to the information in the dialogue context?

The key is to think of information as data that is structured: *information = data + structure*. What kinds of structure do we have available in dialogue? Structure comes in many forms. Obviously it matters how information is 'packaged' (Chafe 1976). Surface aspects like word order and intonation convey how new information relates to information given in the dialogue context. But also the topics addressed in a dialogue and the underlying task have great influence. Can all these aspects of dialogue structure be combined?

Following Ginzburg (1995) we take it that the information in a dialogue context is structured by *questions under discussion*: contextually salient questions, which have not yet been answered at that point. Questions under discussion will be called *issues* here. Different kinds of information structure in dialogue can be expressed in terms of issues.

An issue can be expressed by an explicitly asked question. Once expressed, an issue may be resolved by an answer. Answers convey information and are thus modelled by propositions. Given a question we have strong intuitions about what counts as an answer to the question in general, or not quite the same, what counts as an appropriate response in a certain situation. The idea is that we can base our theory of inquiry on these intuitions. How to interpret short, elliptical or indirect answers? Essentially, each question is associated with a so called *question relation*: an expression that specifies a 'gap' in the information of the asker. The answerer is invited to fill that gap by supplying an answer. For example, the question relation of "Whom does John love?" is given by the following lambda abstract: $\lambda x.person(x) \wedge love(john, x)$. The relation helps to specify possible alternative ways of answering the question. In this case any description of a set of persons loved by John would count, for instance "nobody", "Mary", "Mary and Sue" or "the girls with red hair". The dialogue act of asking a question requires some sort of response. "Well, I don't know" or "Why do you ask?" are not genuine answers, but are possible responses.

According to van Kuppevelt (1995) dialogues are structured by topics. For each dialogue segment the *topic* specifies the object, possibly abstract, that the segment asks or provides information about (Rats 1996). A topic suggests a topic space of related objects and attributes that are available as sub-topics. In this way we get a hierarchical structure. At a global level of dialogue segments we speak of the dialogue topic; at a local level about the topic of an utterance. Within an utterance the term topic is contrasted with *comment*: that part of the utterance that conveys information or asks something about the topic. Now in each dialogue segment the topic can be determined by finding the associated explicit or implicit question that is addressed. The comment typically provides an answer to such a contextually given question. So, topics are closely related to contextually given questions or issues.

For task oriented dialogues it is well known that the dialogue structure depends partly on the task structure (Grosz 1977). The task domain defines which objects are available as potential topics and how they are related. The underlying activity and the goals of the dialogue participants determine what would count as a coherent or cooperative contribution to the dialogue. Participants consider which plans or actions are available to accomplish the task. Alternative actions have different requirements. In order to choose the most appropriate action often more information is needed. Typically, task-related issues will thus be raised as a result of participants deliberating or negotiating a plan to achieve their goals.

Topic and task suggest a global dialogue structure; local features indicate how new or otherwise salient information is to be related to the dialogue context. The *focus* of an utterance contains the most prominent or salient information, or otherwise information that contrasts with expectations. Focus is indicated by syntactic constructs like topicalisation, by other word order differences or by prosodic cues, such as a higher pitch accent. Focus can be contrasted with *ground*: a background of given information against which the focus is to be interpreted (van Deemter 1998; van Deemter 1994; Vallduví 1990; Vallduvi and Engdahl 1996). Taking the semantic representation of an utterance and abstracting over the expression in focus, we derive an expression that indicates the semantic alternatives to the expression in focus (Rooth 1992). Such a lambda abstract looks much like a question relation which expresses the alternative nominal answers. So it appears that the semantics of focus can be modelled by issues too. How the topic-comment and focus-ground distinctions relate to issues will be explained in section 3.8.

It is notoriously hard to derive or predict what the issues are at a given point in dialogue. Moreover, there can be more issues at work at the same time. Only afterwards can we reliably work out the implicit question that actually prompted an utterance. Example (22), adapted from van Kuppevelt (1995), is an example of such an exercise with the implicit questions given between brackets.

(22) Yesterday evening a bomb exploded outside the Houses of Parliament. (*Any victims?*) Nobody was killed. (*Who did it?*) The attack was claimed by separatist rebels of the Liberation Army. (*Any consequences?*) The prime minister condemned the assault, adding that further attacks would endanger the peace process.

In the example background knowledge about news stories and terrorist attacks suggests what the possible issues are. New issues can also be raised by general narrative constraints, for instance about the consequences of an event. But the order in which possible issues are addressed and whether they are important enough to be addressed at all, differs for each particular case. Although it is not possible to predict issues in general, for a fixed domain heuristics can be given to predict the most likely issue. The study of dialogue gives us a slight advantage: issues that would remain implicit in unidirectional discourse are often made explicit by questions in dialogue.

So different types of dialogue structure can be expressed in terms of issues. Issues can be raised explicitly by a participant actually asking a question. But issues can also arise implicitly. Issues can for instance be raised by a topic shift, introducing new potential subtopics for discussion. Or issues can be raised by a change in the underlying task, forcing the agent to consider alternative ways of achieving its goals. Once raised, issues can be resolved by the information conveyed by utterances of dialogue participants. Thus, a dialogue can be pictured as a constant process of raising and resolving issues. There can be more issues at stake at the same time. Ginzburg (1995) suggests a partial order of salience among questions under discussion. Again, this order depends on the general dialogue topic, on the task and on the way the dialogue develops.

Metaphorically, issues open up alternatives: the different answers to a question or the alternatives to an expression in focus are examples of an issue. By adding factual information some alternatives are eliminated. An issue is said to be *resolved* when only one alternative remains. Information resolving an issue is like an assertion answering a question. Now suppose that all contextually given questions that need to be answered, or problems that need to be solved can be represented in terms of issues. Then we might say that information that does not contribute to resolving the issues can not be called relevant. That leads to the following conjecture.

> An utterance can be considered *relevant* with respect to the information in the dialogue context, when its information content (partially) resolves one of the contextual issues.

This characterisation functions as a lower bound. There is nothing to stop the speaker from being over-informative. Groenendijk (1999) defines a constraint that does just that: an utterance is *licensed* when it does not provide more information than is required for current purposes.

> An utterance can be considered *licensed* with respect to the information in the dialogue context, when its information content resolves nothing but contextual issues.

Given this basic idea, a number of notions needs to be specified. How do we model the information content of utterances? How do we model the structure of information? What is an issue? When can we say that information completely or partially resolves an issue? Can we predict what the issues will be from the way the dialogue develops? What is the relation between issues, salience, topics and task?

3.1.2 Update Semantics with Questions

The framework to study inquiry is a combination of update semantics with a semantics of questions and answers. In *update semantics* each formula is interpreted as the change made to the information state of some agent (Veltman 1996). An information state is modelled by a set of possible worlds: the worlds that are compatible with the agent's information. A possible world represents one possible way the world might be. The set of all worlds represents the initial information space of an ignorant agent. When information is added, those worlds that are not compatible with the incoming information are removed from the set. Thus accumulation of information means elimination of possibilities. Similar ideas can be found in Stalnaker (1979), who gives conditions for a successful assertion. If no worlds are eliminated, the information conveyed by the assertion is already *accepted*; the assertion is redundant. If the result is empty, the information conveyed by the assertion is *rejected*; the assertion was internally inconsistent or contradictory with the original information. Given these assumptions, update semantics only accounts for *expansion*, as defined in the framework of belief change (Gärdenfors 1988). *Contraction* or *revision* in the light of a contradiction are briefly discussed at the end of this chapter.

In Groenendijk and Stokhof's (1984, 1996) semantics of questions and answers, the semantic content of a question is modelled by a partition of the information space. The set of possible worlds is divided into disjoint sets of worlds, each corresponding to a complete answer to the question. Such a partition structure will be called an *issue* here. As a matter of fact, the partition structure is called a *question* by Groenendijk and Stokhof. They use the term *interrogatives* for utterances that express a question. Interrogatives are contrasted with *assertives* that express a proposition. We use the word question in a more colloquial sense. Issues are semantic constructs used on a par with propositions.

The mechanism of modelling the increase of information by elimination of possibilities, can also be applied to structures defined over possible worlds. In Veltman's (1996) work, a semantics of a non-monotonic logic is given in terms of an *expectation order*. This is a partial order that ranks worlds according to their 'normality' with respect to the default rules that are in the agent's information state. By adding information about default rules, the structure is further refined. Here, we define updates to refine the issue structure. We model the combined issues in an information state by an equivalence relation. Worlds that are indistinguishable with respect to their answers to all the current contextual questions, are judged equivalent. By grouping worlds that are equivalent we get a partition again.

A combination of update semantics with questions provides a truly dynamic semantics of inquiry. By making assertions participants add factual information. Consequently the set of possible worlds that are still compatible with the agent's information is reduced. By asking questions participants add *structure* to the information. The more issues are raised, the more fine grained the structure of alternative ways of considering the world becomes. Since both data and structure are crucial aspects of information, the model predicts contrary to general opinion, that both assertions and questions can be informative. Assertions are informative when they reduce the set of possible worlds; questions are informative when they further refine or introduce issues.

The development of the material in this chapter is based on work by several people. In presentations Groenendijk suggested the partition semantics of questions to express the alternatives associated with focus, as in the theory of Rooth (1992). The idea was picked up by others, in particular by Jäger (1996) who proposed the combination with update semantics and first used equivalence relations to model the information states themselves. Inspired by Ginzburg (1995) we suggested the terminology of raising and resolving issues (Hulstijn 1997). The interaction of issues with goals was studied in Hulstijn (1998). The theory is especially influenced by later work of Groenendijk (1998, 1999). Modelling interaction as a process of raising and resolving issues suggests a new paradigm for the study of logic and language, based on the Gricean maxims for cooperative information exchange.

The chapter is structured as follows. We start in section 3.2 with the motivation for this particular kind of semantics and a discussion of the different choices. We proceed in section 3.3 with definitions of an update semantics with questions. In section 3.4 we extend the definitions to deal with discourse referents, and discuss the properties of coreference in the logic. Section 3.5 defines notions of resolution and entailment. Section 3.6 discusses the dialogue properties of consistency, relevance and licensing. In section 3.7 we deal with presuppositions. In section 3.8 we discuss how the issue structure is related to topics, salience and task. Section 3.9 contains preliminary conclusions and some topics for further research.

3.2 Questions and Answers

One of the starting points of the theory of inquiry presented here, is that questions actually have a semantics of their own. The meaning of a question can not be reduced to the meaning of another dialogue act. This view needs to be argued for.

Consider the relation between the success conditions and the satisfaction conditions of a dialogue act. A dialogue act is composed of a specification of its communicative function or *illocutionary force* and a specification of its semantic *content* (Searle 1969; Vanderveken 1990). Mental attitudes of agents are composed in a similar way (Searle 1983). The *success conditions* of an act or attitude are determined by the illocutionary force; they specify under what circumstances a speaker has succeeded in performing the corresponding act, or when an agent can be said to have the corresponding attitude. For example, a number of conditions must be fulfilled before a speaker succeeds in making a request like "Two tickets please". The speaker must actually want two tickets; otherwise she would be insincere. There must be a hearer present and tickets must be available. The semantic content determines the *satisfaction conditions*: the conditions under which the act or attitude is satisfied. A request is satisfied when the hearer has brought about the situation described by the content of the act. In our example, when the hearer has issued the tickets. Acts or attitudes are partly characterised by a so called *direction of fit*. Declarative utterances such as assertions or suggestions, and representative attitudes like knowledge or belief make a claim of some sort about the world. Their satisfaction depends on whether that claim is true. Therefore such acts or attitudes have a so called *word-to-world* or *mind-to-world* direction of fit: the 'words' or 'mind' of the agent must fit the world. On the other hand directives such as commands or requests and anticipatory attitudes like intention or desire are focussed on a change in the world. Such acts or attitudes have the *world-to-word* or *world-to-mind* direction of fit. They are satisfied in case the world comes to fit their content.

Now what about questions? Traditionally, questions are lumped with the directives, having a world-to-word direction of fit. Under this view, a question is seen as a request for information. A request succeeds when the hearer has realised the situation specified in the satisfaction conditions; in this case, a situation in which the question is answered. So for questions the satisfaction conditions are essentially *answerhood conditions*: conditions that specify under what circumstances the question counts as answered. The content of a question must specify the desired information. Now of course this can not be a complete specification of the answer; that would imply that the speaker already knew the answer! And the success conditions of a question require among other things that the answer to the question is not yet known. Therefore answerhood conditions specify what would constitute a complete answer to the question, for each possible circumstance. So even if we would want to reduce questions to requests, we would still need something like answerhood conditions.

In fact, there are strong reasons for not treating a regular question as a request. Questions require an independent type of semantics. The main case in point is the treatment of embedded questions, as in "Mary wondered whether John loved Sue". Intuitively, the semantic content of the embedded phrase "whether John loved Sue" is identical to that of the direct question: "Does John love Sue?". If we try to reduce direct questions to performatives, such as: "I hereby ask you whether John loves Sue." or requests like "Bring it about that I come to know whether John loves Sue!", we run into trouble for embedded questions. In that case we would have to embed a request as the complement of 'wonder', which is grammatically impossible. There are many verbs like 'wonder' that can express non-propositional attitudes. Issues are a candidate for modelling the complement of verbs that can have an interrogative complement. The attitudes expressed by such verbs are called *resolutives* by Ginzburg (1996). They do not require their complement to be true or false, but they require it to be *resolved*. For example the verbs 'wonder', 'doubt', 'investigate', 'be aware', 'know', 'discover', 'guess', 'depend on', 'be convinced' and 'matter', can all take an interrogative complement.

3.2.1 Partitions

So questions have an independent type of semantic content. The content of a question is strongly related to its answerhood conditions: what would count as an answer under different circumstances. In each possible world there is a particular answer to the question. Answers convey information, so it is natural to model the semantic content of answers by a proposition, or equivalently by a set of possible worlds. This means that the meaning of a question is modelled by a *propositional concept*: a function from possible worlds into propositions. By grouping worlds according to the proposition they map onto, such a function generates an equivalence relation, or equivalently, a partition of the set of all possible worlds.

There is an important distinction to be drawn between the *sense* or *intension*, and the *denotation* or *extension* of an expression. For example, the denotation of a definite description is an individual, some object from the domain. However, if we abstract over the particular circumstances in which an utterance is uttered and we look at the sense of an expression in general, we get a function from possible words into denotations. The intension of the definite description 'the president' is a so called individual concept: a function from worlds to individuals. At the time of writing the president of the United States is Bill Clinton, but in a counterfactual world in which Clinton had been impeached, the president would be the current vice president, Al Gore. Some verbs, like 'to look for', are intensional constructions in the sense that they typically act on the intension of a description. To say that John is looking for the president, is not the same as saying that John is looking for Bill Clinton. For one thing, John might not know who the president is.

The denotation of an assertive utterance is a truth value; its intension is a function of possible worlds into truth values. Given such a characteristic function, one typically derives a proposition: the set of possible worlds that would make the asserted true, in case it happens to be true, or false in case it happens to be false at the current context of evaluation. This corresponds to the satisfaction conditions of the asserted.

In a similar way, we can draw a distinction between the extension of a question, the particular complete answer in a particular world, and its intension, the function from worlds to complete answers. For those worlds where the question makes sense, the question has a complete answer. For questions the intension corresponds to answerhood conditions. Worlds can be grouped according to their answer. Such alternative answer groups will be typically disjoint. Two different complete answers to a question cannot be true in a single world. Therefore, the resulting structure is a partition: a set of pair-wise disjoint sets which covers the whole space of possibilities for which the question makes sense.

Here are two examples of the construction of a partition. A yes/no question is of the form $?\varphi$, where φ can be any formula. A wh-question is of the form $?x.\varphi$. The notation

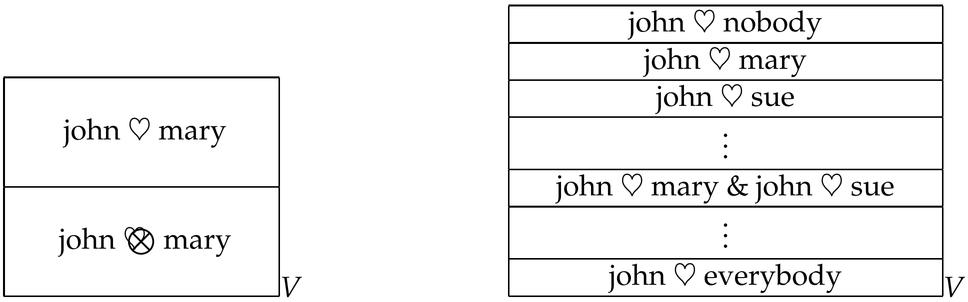


Figure 3.2: The partition of the information space induced by examples (23) and (24).

is further explained in section 3.3. The resulting partitions are pictured in figure 3.2. We assume some information space in which the question makes sense: a set of possible worlds V that is a subset of the set of all possible worlds W. For example, John must be known for these questions to make sense. The partition structure is depicted by a grid.

(23) Does John love Mary? $?love(john, mary)$

(24) Whom does John love? $?x.love(john, x)$

The yes/no question in example (23) partitions the information space into two parts: those worlds that would answer 'yes' and those that would answer 'no'. Assume the current context of evaluation is modelled by a possible world w. Now the 'yes'-worlds are the worlds that agree with w on the truth of the proposition that John loves Mary; the 'no'-worlds disagree.

There are many possible answers to the wh-question whom John loves in (24). He may love no one, he may love Mary or Sue or he may love several people. An answer is a proposition: a description of the set of objects that satisfy the question relation. The question relation is $\lambda x.love(john, x)$. So the extension of a question at some particular possible world w, its complete answer at w, is the set of worlds that agree with w on the set of objects that satisfy the question relation. The intension can be derived by abstracting over w, the current world of evaluation. This produces classes of worlds that agree on the strongest answer they can give to the question. The definitions given in section 3.3.3. Note, that this semantics of questions does not simply produce an *answer set*: a list of the persons John may love. Rather, it produces for every world a proposition that describes the answer set at that world. By grouping worlds according to their answers, we get 'blocks' in the partition that correspond to the members of the power set of the set of persons possibly loved by John: $\{\varnothing, \{mary\}, \{sue\}, ..., \{mary, sue\}, ...\}$.

In the analysis, each of the blocks in the partition corresponds to a *complete answer* to the question. By contrast, a *partial answer* specifies a subregion of the information space that merely reduces the number of blocks in the partition. For example, Mary could teasingly answer that the name of John's loved one does not start with an 's'. In this way, all blocks in the partition with Sue mentioned in it will be eliminated. Depending on the circumstances of the issue being raised, a partial answer may or may not be a sufficient or appropriate answer. In case it was Sue who asked the question out of rivalry with Mary, such a partial answer completely resolves the underlying issue for Sue, whether John loves her or whether he loves Mary.

We believe that similar considerations can be used to account for so called *mention some* readings and for the *relative granularity* of answers. What counts as an appropriate response depends on the apparent goals of the dialogue participants (Ginzburg 1995).

(25) Where can I buy a newspaper?

(26) When is the launching of the Space Shuttle?
  a. – at 19:02:15 GMT (to a space scientist)
  b. – at prime time (to a television producer)

In case you ask "Where can I buy a newspaper?", you do not need an exhaustive list of newspaper stands. You need just one, preferably the closest. However, for a newspaper salesman surveying the competition an exhaustive list would be needed. Such a complete list is provided by the semantics. Based on the apparent goal, to buy a newspaper, this may be relaxed to a particular partial answer. In a similar way the granularity of appropriate answers can be dealt with. In example (26) the scientist expects a precise answer; the television producer contents herself with a much coarser answer, as long as it meets her interests. Our intuition here is that the partition generally induced by a question corresponds to the strongest alternative answers. In practice, it can always be relaxed by the apparent goal of the dialogue participants. How this happens is explained in chapter 4.

3.2.2 Coordination and Entailment

One of the advantages of the partition view of Groenendijk and Stokhof (1984, 1996) that is also advocated here, is that it provides some account of coordination and entailment relations between issues.

Questions or issues can stand in all kinds of logical relationships. Consider the dependency relation in: "Who wins the race depends on who enters it". The same notion of dependence between questions appears as a relevance constraint in dialogue. For example, if you ask me "Who is going to win the race?" I can felicitously respond with a return question "Who is in it?" because knowing an answer to the second question is one of the preconditions for finding out an answer to the first. Precisely these kinds of dependencies help to structure information in a dialogue context. In this example we need background information on horse racing. For dialogue system applications it is important to get the dependencies right. Dependencies can be implemented by conceptual or type hierarchies, as part of the information model that we described in chapter 1. An example of a representation to capture dependencies in an information state is the theory of Veldhuijzen van Zanten (1996, 1998). Dependencies are often related to the task model or to the apparent goal of the dialogue participants.

A relationship between questions or issues that does not depend on the task domain is entailment. The partition view of questions provides a semantics to the entailment relation between questions. A question *entails* another question, $?\varphi \models ?\psi$, when each potential answer to the first question entails an answer to the second. If answers are propositions and set inclusion is used to model entailment of propositions, that means that $?\varphi$ entails $?\psi$ precisely when for each proposition or set of worlds V in φ's partition, there is a U in ψ's partition such that $V \subseteq U$. When issues are expressed as equivalence relations we can use set inclusion on the relations to express the same notion.

Questions can be combined into coordinated questions or decomposed into constituent questions. Consider the following observations. The conjunction of two questions is again a question. On the other hand, a coordinated question like (27a) entails a conjunction of two questions, as in (27b). The conjunction of two issues can be modelled as the intersection of the blocks in the partitions. This property is of central importance to the idea of increasing structure for issues. Issues can be added until the blocks are singleton sets.

Given some background assumptions about traffic lights, a wh-question like (28a) is equivalent to asking for the separate alternatives, as in (28b). First we identify the set of worlds where the question makes sense, i.e. worlds with a traffic light in it. Question (28a) generates a partition of these worlds into three blocks: one for each colour. Each yes/no question generates a partition into two blocks: that colour or another. Partitions can be combined: pairwise intersection of the blocks in a partition produces a partition again. A combination of the yes/no questions in (28b) produces exactly the three-way partition of (28a), assuming that red, yellow and green cover all the possible colours and are mutually exclusive.

(27) a. Where are Mary and John?
b. Where is Mary and where is John?

(28) a. What colour is the traffic light?
b. Is the traffic light red, is it yellow, or is it green?

Notice that the 'or' in (28b) is not interpreted as a we would expect for disjunction, i.e. as a kind of union. A union of the blocks in a partition is not necessarily a partition again. It seems here that 'or' is to interpreted as a *choice* connective at the level of dialogue acts. The speaker asks three alternative questions about the same topic, the traffic light, and leaves the hearer to decide which one to answer. The object talked about remains the same, but different issues are raised about it.

A similar but mirrored effect occurs for continued questions. Consider the following dialogue. The effect of the continuation in U2 is not a restriction of the space of possible performances, as would be expected for a logical 'and', but rather a shift to an alternative part of the space.

(29) U1: What is on on Tuesday?
S1: On Tuesday we have Twelfth Night and Arthur's Eagles.
U2: And on Wednesday?
S2: On Wednesday we have Arthur's Eagles.

The shift can be explained in terms of topics and issues. The question in U1 sets the general dialogue topic to performances, with a specific sub-topic of Tuesday. The corresponding most salient issue is "What is on then?" In utterance U2 we witness a topic shift to Wednesday. The issue remains, but the object it is applied to has shifted. This example suggests a definition of topic as the object or objects that the currently most salient issues centre around. This idea is further discussed in section 3.8.

Give and take the difficulties in mapping naturally occurring dialogue onto the logical language, we can say that the underlying 'logic of questions' can be a rather standard logic of boolean coordination and entailment. Entailment is defined in terms of set inclusion on the blocks in a partition; conjunction is defined as the intersection of blocks in the partition. There are obvious connections between these boolean issue connectives, and the operators of relational algebra applied in the semantics of databases (Ullman 1988). Systematic comparisons between these different semantics of queries is an interesting topic of further research. Coordination and entailment can be applied in the development of dialogue systems. Veldhuijzen van Zanten (1998) proposes an adaptive question selection algorithm. Normally the system will choose a formulation that leaves much of the initiative to the user, such as a wh-question. But in case problems are detected in the dialogue it makes sense to provide more guidance by restricting formulation to a series of closed yes/no questions. That this is possible at all follows from the coordination and entailment mechanisms described here.

3.2.3 Presupposition

We mentioned the notion of presupposition accommodation in chapter 2. By using a presupposition the speaker indicates that she assumes certain information to be in the context. The hearer, being cooperative, will accommodate this information, provided it is compatible (Lewis 1979; Heim 1983). If the hearer's version of the context is not compatible with a presupposition, this indicates a misunderstanding. Utterances with a failing presupposition are judged infelicitous.

There are interesting connections between presuppositions and issues. Obviously, a presupposition can occur inside a question. The resulting partition will then only cover the worlds in which the presupposition is satisfied. In this set of worlds the question makes sense. Consider mentioning 'the traffic light' in example (28), which presupposes that there is a traffic light present.

There is a long standing debate about the *answer presupposition* of a question. Each question, according to this view, presupposes the existence of a positive answer. So in example (28) the question presupposes that the traffic light does have a colour, or in example (24) it is assumed that John does love someone. If we treat this answer-presupposition like any other presupposition, 'nobody' would not be allowed as an answer. It would be classified as a response at a meta-level, that denies or cancels the presupposition. We can also choose to treat 'nobody' as a normal answer. In the first case, we would predict some conversational 'friction' resulting from the apparent misalignment between speaker and hearer. In the second case we would predict no such friction. Which way to go depends on the context and on the precise formulation of the question.

Issues can also be presupposed. Intonation patterns are used to indicate focus. They presuppose a focus structure that can be modelled as an issue. The expression in focus answers that contextual question. Consider the following examples. An utterance like (30a) with a pitch accent on green presupposes a contextually given issue: what is the colour of the door? The speaker indicates that the colour of the door is green and not some alternative colour. An utterance like (30b) on the other hand presupposes the issue which object is green. The speaker indicates that of all objects it is the door that is green.

(30) a. The door is *green*.
b. The *door* is green.

Also without explicit intonation cues issues can be presupposed. This can be explained as follows. To each utterance we can apply the following relevance test: what is it that makes

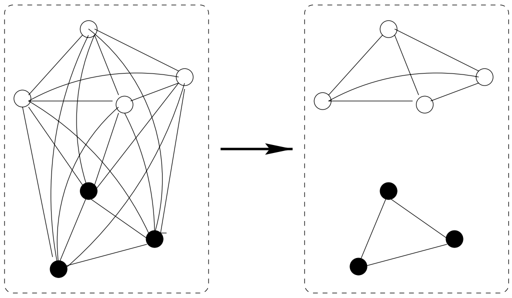


Figure 3.3: u(A) and the partition $A/same\_colour$

the speaker utter this utterance here and now? The result of this test provides the most salient contextual issue. The Gricean maxim of relation also works as an interpretation guide for hearers. Hearers will try to interpret an utterance as being relevant. This is the basic principle of communication according to Sperber and Wilson (1986). It suggests that in the absence of an explicit question, hearers will typically try to adjust their version of the context so that it will contain an issue for the utterance to address. Again, this process of adjustment is called *accommodation*; it occurs only when the accommodated material is coherent with the existing issues in the context. The framework developed here is very suitable to study the interaction of presuppositions with issues. In section 3.7 we define a specific presupposition operator in terms of which both assertive and interrogative presuppositions can be dealt with.

3.2.4 Preliminaries

In order to model the different aspects of information, we need some algebraic notions. An equivalence relation is a binary relation that is reflexive, transitive and symmetric. An equivalence relation R over some set A generates equivalence classes $[a]_R$ for each $a \in A$. The equivalence classes form a partition called the *quotient set* of A over R, written A/R. In general a partition of a set A is a set of non-empty, pair-wise disjoint subsets of A, such that their union produces A again. From the partition we can induce an equivalence relation, by taking all pairs of elements that are in the same 'block' or subset of the partition. An example of a special equivalence relation is the total or universal relation u(A). It combines all elements into a partition of one single block. Another example of a special equivalence relation is the reflexive relation s(A). It produces singleton sets with elements from A as the corresponding equivalence classes. The *domain* of a relation d(R) is the set of objects that occur in the relation. A relation R over A, can be *restricted* to a subset B of A, written $R \restriction B$.

Definition 2 (Algebraic Preliminaries)
Given sets A, B and equivalence relation $R \subseteq A \times A$, define

$[a]_R$	$=$	$\{b \in A \mid \langle a,b\rangle \in R\}$ $(a \in A)$	equivalence class
A/R	$=$	$\{[a]_R \mid a \in A\}$	quotient set
$\mathrm{u}(A)$	$=$	$\{\langle a,b\rangle \mid a,b \in A\}$	universal relation
$\mathrm{s}(A)$	$=$	$\{\langle a,a\rangle \mid a \in A\}$	singletons
$\mathrm{d}(R)$	$=$	$\{a \mid \langle a,b\rangle \in R$ or $\langle b,a\rangle \in R$, for some $b\}$	domain
$R \restriction B$	$=$	$\{\langle a,b\rangle \in R \mid a,b \in B\}$	restriction

□

In a graph, equivalence classes can be depicted by totally connected sub-graphs. Figure 3.3 depicts the partition of a set of totally connected coloured nodes A induced by the relation *same_colour*. All connections between nodes that do not 'agree on colour' have been eliminated. Likewise, in our account possibilities that do not 'agree on the relevant facts' with respect to some issue, are eliminated.

Every function induces an equivalence relation on its domain. Given an arbitrary function f that maps elements of A onto B, we write $\simeq_f$ for the equivalence relation that groups elements of A according to their image under f. This relation is called *equivalence under* f.

Definition 3 (Equivalence under f)
Given a function $f :: A \rightarrow B$, define for any $a, b \in A \quad a \simeq_f b \quad \text{iff} \quad f(a) = f(b)$. □

3.3 Update Semantics with Questions

An update semantics is defined as a tuple $\langle L, \Sigma, [.]\rangle$, where L is a logical language, Σ is a predefined set of information states, and for each $\varphi \in L$, $[\varphi]$ is an update function. Following Veltman (1996) a postfix notation is used: $\sigma' = \sigma[\varphi]$ means that σ' results from updating an information state σ with a formula φ. An information state $\sigma \in \Sigma$ corresponds to the information of a single agent; in case of an inquiry game we model the information state of a neutral observer.

3.3.1 Language

The logical language L is defined in three stages. First, we define the terms of the language T, based on sets of individual variables and individual constants. Second, a predicate logical language L_0 is defined. Third, the language is extended with a question operator '?'. Elements of L_0 and L are called formulas. Formulas with a '?' are called *interrogatives*. We could imagine an assertion operator '.' to express *assertives*. As it stands, all unmarked formulas are taken to be assertive. Formulas of L may be combined by sequential composition ';' to form sequences of assertive and interrogative contributions. The formulas of L model basic dialogue acts for adding and retrieving information. At the same level also *directives*, marked by '!', can be dealt with (chapter 4).

A vector notation is used as a shorthand for a sequence of zero or more different variables or terms. Define $\vec{t} = \langle t_1, \ldots, t_n\rangle$, with $t_i \in T, 0 \leq t_i \leq n$, which is implicitly understood as an empty sequence $\langle\rangle$ in case $n = 0$. A nullary predicate is a proposition letter. For example the proposition that it is raining is expressed as '*rain*'.

Definition 4 (Syntax T, L_0, L)
Given sets of symbols $Pred = \{P, \ldots\}$, $Cons = \{c, \ldots\}$, $Var = \{x, \ldots\}$, define

$$
\begin{array}{lll}
T: & t & ::= x \mid c \\
L_0: & \varphi & ::= P(\vec{t}) \mid (t_1 = t_2) \mid \neg\varphi \mid (\varphi \wedge \psi) \mid (\varphi \vee \psi) \mid (\varphi \rightarrow \psi) \mid \exists x.\varphi \mid \forall x.\varphi \\
L: & \chi & ::= \varphi \mid ?\vec{x}.\varphi \mid (\chi;\zeta) \qquad\qquad (\varphi \in L_0,\ \chi, \zeta \in L)
\end{array}
$$
□

We assume the following standard terminology. A term that contains no variables is a *ground term*. A *ground formula* is a formula that contains only ground terms. A variable that is not bound by a quantifier or by the question operator, is said to occur free in a formula. Such a variable is called a *free variable*. A formula that does not contain any free variables is called a *closed formula*. An atomic formula is a formula that contains no logical connectives or operators. The set of ground atomic formulas is of central importance to the semantics; it is called the *vocabulary* $\mathcal{P}$.

Formulas of the form $?\vec{x}.\varphi$ represent questions. The formula $?x(door(x) \wedge green(x))$ expresses "Which door is green?". Because we restrict ourselves to a first order language here, we can not properly express a plural question like "Which doors are green?". Mapping syntactic forms onto interrogative formulas is a difficult and interesting task. Here are some examples. They are explained below. Obviously, the account would need to be accompanied with a method for translating natural language expressions to the logical language.

(31)	a.	Is it raining?	$?rain$
	b.	Who loves whom?	$?xy(person(x) \wedge person(y) \wedge love(x,y))$
	c.	Which Athenian is wise?	$?x(athenian(x) \wedge wise(x))$
	d.	Is an Athenian wise?	$?\exists x(athenian(x) \wedge wise(x))$ or
			$?\forall x(athenian(x) \rightarrow wise(x))$
	e.	What is John doing?	$?x(activity(x) \wedge do(john, x))$
	f.	What is John's age?	$?x(number(x) \wedge age(john, x))$
	g.	Who is John?	$?x(x = john)$
	h.	Why does John love Mary?	$?x(explain(x, love(john, mary)))$
	i.	How do I get to the station?	$?x(route(x, here, station))$

Example (31a) models a simple yes/no question. Note that time and location and tense and aspect are ignored. (31b) illustrates a multiple wh-question. The type of an answer is indicated by the question, so (31c) shows a simple wh-question that asks for an Athenian. Note the difference between the wh-question in (31c), and the yes/no questions in (31d). The indefinite 'an Athenian' can have both an existential and a universal reading, just like in declarative sentences. For the universal reading think of the biological "A horse has four legs". Higher order properties, as in (31e), are approximated with first order descriptions. So activities are treated just like any other attributes, like *age* in (31f). A who-question like (31g) asks for a person. Note that under this representation, any answer is equivalent to 'John' itself, which would not be an informative response. "That man over there" would be appropriate. Answers like "That man over there" or "the brother of Jack" would be appropriate if it identifies John. So what a wh-question really asks for is a description to identify the answer in the given circumstances. For more on the difficulties of 'knowing who' see Boër and Lycan (1986). Other types of wh-questions like 'where', 'how' or 'why' are even more difficult to capture. 'How' asks for a manner; 'why' asks for a reason. But what counts as a reason? In example (31h), any contextually relevant explanation would count. And what exactly is a 'manner'? When asking directions, like in example (31i), a manner is just a route. For more on how-questions see Asher and Lascarides (1998a).

Although it does not appear in the logical language itself, we make use of n-ary abstracts to represent what we called the question relation. Abstracts function as an auxiliary in the semantics. They are used to define any possible n-ary relation on the basis of predicate logic formulas.

Definition 5 (Abstract)
For each $\varphi \in L_0$ containing free variables $\vec{x}$, $\lambda\vec{x}.\varphi$ is an *n-ary abstract*. □

3.3.2 Framework

The dialogue game of inquiry can be modelled from two perspectives. Apart from the *internal perspective* that is used in update semantics, there is an *external perspective* used in multi-agent systems. From the internal perspective we pick one agent and follow its local information change. The external perspective deals with the global system that may consist of several agents exchanging information. The effect of dialogue acts like an assertion or question on the attitudes of agents is modelled by a multi-modal logic. This is the approach taken in chapter 4. The internal perspective of update semantics can be lifted to the external point of view by relating information states to modal accessibility relations. Suppose for example that the information state of agent b at some dialogue situation s is modelled by σ_b. An agent a uttering a declarative utterance to b with semantic content φ, modelled as a dialogue act assert(a, b, φ), is interpreted as a transition to another dialogue situation s' in which, among other things, the information state of b is updated and now equals $\sigma_b[\varphi]$, provided the utterance is acknowledged in some way.

We assume a model that captures the relations between agents. A model M is a Kripke frame $M = \langle D, A, W, ..., V\rangle$, where D is a set of objects, A is a set of agents, W is a set of possible worlds and the '...' represent a number of accessibility relations over W for each of the agents in A. The definitions are provided in chapter 4. For each $w \in W$, a valuation function V_w maps objects to constants and sets of n-tuples of objects to n-ary predicate symbols.

We make the following provision: the space of possibilities W must cover all possible combinations of facts that are expressible in the language, and each possibility in W must be distinct from the others in some way that can be expressed in the language. This can for instance be achieved by identifying the space of possibilities W with the set of all total valuation functions over ground atomic formulas, and then generating first order models on the basis of these.

A second provision concerns the interpretation of constants and predicate symbols. Because we want to model presuppositions later, we allow expressions to be undefined in certain worlds. For each world w we there is a local domain of discourse $D_w \subseteq D$ that provides the objects that make sense at w. The valuation function is restricted to this domain. Apart from this, constants are interpreted as rigid designators.

Quantification is dealt with by assignments. Assignments carry local information on the values of variables. An assignment g is a partial function from variables to objects. Assignments are partial; in case a variable x has no value, $g(x)$ is undefined. Assignments can be extended with values for new variables. Notation $g[x \mapsto d]$ indicates the assignment that is just like g, except for the variable x, which is assigned the object d. The notation is stretched for sequences: $g[\vec{x} \mapsto \vec{d}] \equiv g[x_1 \mapsto d_1]...[x_n \mapsto d_n]$ for $0 \leq n$, which returns the old g in case $n = 0$. By $g(\vec{x})$ we mean the sequence $\langle g(x_1), ..., g(x_n)\rangle$. In case one of the $g(x_i)$ is undefined, the assignment to the whole sequence becomes undefined.

We start by defining the interpretation of terms relative to a particular world and assignment. The valuation function V is made dependent on assignments too. So the interpretation of a variable depends on the assignment. The interpretation of an individual constant depends on the world of evaluation. The interpretation of a sequence of terms is simply the sequence of interpretations. When one of the terms is undefined, the whole sequence becomes undefined. The interpretation of an empty sequence is the empty sequence.

Definition 6 (Interpretation of Terms)
Given a model M, define for every $t \in T$ its interpretation $V_{w,g}(t)$ at w under g

$$\begin{array}{lcll} V_{w,g}(x) & = & g(x), & x \in Var \\ V_{w,g}(c) & = & V_w(c), & c \in Cons \\ V_{w,g}(\vec{t}) & = & \langle V_{w,g}(t_1), \ldots, V_{w,g}(t_n) \rangle, & \text{if } V_{w,g}(t_i) \text{ defined}, \vec{t} = \langle t_1 \ldots t_n \rangle, (0 \leq i \leq n) \\ & & \text{undefined}, & \text{otherwise} \end{array}$$

□

On the basis of the valuation function we can define when a formula is satisfied at a world, relative to a model and an assignment. An atomic formula is satisfied if the interpretations of the terms are included in the interpretation of the predicate. In case one of the terms is undefined, the formula is simply not satisfied. Two terms are identical, when they denote the same object. Negation and conjunction are defined by meta reasoning, as usual. Existential quantification is defined by extending the current assignment, so that it will contain a value for the variable quantified over. Satisfaction of formulas with implication, disjunction and universal quantification is defined by means of the following standard equivalences: $(\varphi \rightarrow \psi) \equiv \neg(\varphi \wedge \neg\psi)$, $(\varphi \vee \psi) \equiv \neg(\neg\varphi \wedge \neg\psi)$ and $\forall x.\varphi \equiv \neg\exists x.\neg(\varphi)$.

Definition 7 (Satisfaction)
Given a model M define for every $\varphi \in L_0$ its satisfaction at w under g, by

$$\begin{array}{lll} M, w, g \models P(\vec{t}) & \text{iff} & V_{w,g}(\vec{t}) \in V_w(P) \\ M, w, g \models (t_1 = t_2) & \text{iff} & V_{w,g}(t_1) = V_{w,g}(t_2) \\ M, w, g \models \neg\varphi & \text{iff} & M, w, g \not\models \varphi \\ M, w, g \models (\varphi \wedge \psi) & \text{iff} & M, w, g \models \varphi \text{ and } M, w, g \models \psi \\ M, w, g \models \exists x.\varphi & \text{iff} & M, w, g[x \mapsto d] \models \varphi, \quad \text{for some } d \in D \end{array}$$

□

3.3.3 Extension and Intension

Now we turn to the extension and intension of an expression. Recall examples (23) and (24), reproduced here as (33) and (34) and contrasted with an assertion in (32). For readability, reference to assignments is dropped.

First consider the assertion in example (32). The extension of an assertion at a particular possible world is a truth value. The intension is a function that specifies for each world, whether it is true. Based on this characteristic function it is easy to derive what we call the *intension set*: the set of worlds for which the characteristic function is true.

(32) John loves Mary. $love(john, mary)$
extension at w: $V_w(love(john, mary))$
intension set: $\{w \mid V_w(love(john, mary)) = 1\}$

Assume the current context of evaluation is modelled by a possible world w. The extension of the yes/no-question in (33) is the proposition or set of worlds that agree with w on their answer to the question. If John does indeed love Mary in w that would be 'yes'. Otherwise it would be 'no'. The intension of the question can be obtained from the extension by abstraction over the current world of evaluation. So a yes/no question partitions the information space into two parts. Those worlds that agree with w, and those worlds that do not. Therefore a yes/no question produces a bi-partition.

	extension at w	*intension set*	*motivation*
assertion	truth value $V_w(\varphi)$	proposition $\{w \mid V_w(\varphi) = 1\}$	truth conditions
abstract	set of n-tuples $V_w(\lambda\vec{x}.\varphi)$	description $\{w \mid V_w(\lambda\vec{x}.\varphi) = E\}$ $(E \subseteq D^n)$	
question	proposition $\{v \mid V_v(\lambda\vec{x}.\varphi) = V_w(\lambda\vec{x}.\varphi)\}$	partition $\{\{v \mid V_v(\lambda\vec{x}.\varphi) = V_w(\lambda\vec{x}.\varphi)\} \mid w \in W\}$	answerhood conditions

Figure 3.4: Extension and intension of assertions, abstracts and questions

(33) Does John love Mary? $?love(john, mary)$
extension at w: $\{v \mid V_v(love(john, mary)) = V_w(love(john, mary))\}$
intension set: $\{\{v \mid V_v(love(john, mary)) = V_w(love(john, mary))\} \mid w \in W\}$

The question relation of the question in example (34) is $\lambda x.love(john, x)$. The extension at w is the complete answer at w: the set of worlds that agree with w on the objects that satisfy the question relation. Each answer roughly corresponds to a subset E of the set of objects that are initially plausible answer candidates. The intension can again be derived by abstracting over w. Worlds that are indistinguishable with respect to their answers, are grouped together. Therefore a wh-question produces a partition in as many blocks, as there are subsets of the set of plausible answer candidates.

(34) Whom does John love? $?x.love(john.x)$
extension at w: $\{v \mid V_v(\lambda x.love(john, x)) = V_w(\lambda x.love(john, x))\ )$
intension set: $\{\{v \mid V_v(\lambda x.love(john, x)) = V_w(\lambda x.love(john, x))\} \mid w \in W\}$

Now we can motivate definition 8 and 9 of extension and intension for each type of expression in the language. The intuitions are summarised in figure 3.4. These notions can be expressed more elegantly in a two-sorted type-theory (Groenendijk and Stokhof 1989).

For an *assertion*, the extension is a truth value that indicates whether the asserted is satisfied at the world of evaluation. The intension is a characteristic function: a function from worlds to truth values. In general, the intension of an expression is a function that maps worlds onto the extensions. Based on the intension, we can derive what one may call the *intension set*: the set of worlds that share a particular extension value. What value that is, depends on the success conditions of the particular way the expression is used. For an assertion, that value is typically the truth value '1'.

For a *term*, the extension is its interpretation in a world: an object from the domain, as in definition 6. Its intension is a function from worlds to objects: an individual concept. We can again derive an intension set. A minimal requirement is that the term is defined in a particular world. So a natural intension set for a term would be the set of worlds for which the extension is defined. This corresponds to the presupposition of the term.

An n-place *abstract* expresses some (question) relation. Its extension is analogous to the interpretation of a basic predicate symbol. It denotes the set of n-tuples of objects that satisfy the relation at a particular world. Again, we can define an intension set. Any set of applicable tuples $E \subseteq D^n$ may count as the typical extension value of an n-place abstract. The set of worlds with non-empty answer sets corresponds to the conventional presupposition of the question corresponding to the n-place abstract.

Finally, a *question* has as its extension in a world, the complete answer to the question at that world. Answers are propositions, modelled by a set of worlds. In this case, the set of worlds that agree on their extension of the n-place abstract. So the extension of a question equals the intension set of an abstract. The intension of an question is again the functin that maps worlds onto their extensions. The corresponding intension set is therefore a partition of the set of worlds where the question makes sense.

Here is the definition of the extension of an expression relative to a particular context of evaluation: a model, a world and an assignment. Sequences of expressions $\varphi;\psi$ are dealt with when we come to updates, in definition 15.

Definition 8 (Extension)
Given a model M, define for each $\varphi \in L_0$ the extension at w under g

$$\begin{array}{lcl} V_{w,g}(\varphi) & = & \begin{cases} 1, & \text{if } M, w, g \models \varphi \\ 0, & \text{otherwise} \end{cases} \\ V_{w,g}(?\vec{x}.\varphi) & = & \{v \mid V_{v,g}(\lambda\vec{x}.\varphi) = V_{w,g}(\lambda\vec{x}.\varphi)\} \\ & & \text{where } V_{w,g}(\lambda\vec{x}.\varphi) = \{\vec{d} \in D^n \mid M, g[\vec{x} \mapsto \vec{d}], w \models \varphi\} \end{array}$$ □

Note that the definition for assertions is equal to that of abstracts for $\vec{x} = \langle\rangle$ when you take $\{\langle\rangle\}$ as '1' and $\varnothing$ as '0'. Also yes/no questions are an instance of the general case for questions. If the empty sequence $\langle\rangle$ makes the question relation true, this may be classified as a 'yes'. If no sequence makes the question relation true, the answer was 'no'.

Proposition 1 (Yes/No Questions)
For each model M, world w and assignment g

$$V_{w,g}(?\varphi) = \begin{cases} \{v \mid V_{v,g}(\lambda\langle\rangle.\varphi) = V_{w,g}(\lambda\langle\rangle.\varphi) = \{\langle\rangle\}\}, & \text{if } M, g, w \models \varphi \quad (\text{'yes'}) \\ \{v \mid V_{v,g}(\lambda\langle\rangle.\varphi) = V_{w,g}(\lambda\langle\rangle.\varphi) = \varnothing\}, & \text{otherwise} \quad (\text{'no'}) \end{cases}$$

Proof By definition. □

The intension of an expression is the extension abstracted over the context. Given an assignment g, this produces for each φ a representative function $\langle\!\langle\varphi\rangle\!\rangle_g$ from worlds to extensions.

Definition 9 (Intension)
Given model M and assignment g define for each $\varphi \in L$ $\langle\!\langle\varphi\rangle\!\rangle_g = \lambda v.V_{v,g}(\varphi)$. □

Based on the representative function we can derive an equivalence relation. Worlds are grouped according to their extensions. Using the earlier notation for equivalence under a given function we write $\simeq_{\langle\!\langle\varphi\rangle\!\rangle_g}$. The notion of $[\![\varphi]\!]_g$, the set of worlds that satisfies a certain expression φ under g corresponds to what we called the intension set. It defines the *content* of an utterance. For assertives, $[\![\varphi]\!]_g$ is called the *proposition* expressed by φ. For questions we can define a corresponding notion. $[\![?\vec{x}.\varphi]\!]_g$ is the partition generated by $\simeq_{\langle\!\langle ?\vec{x}.\varphi\rangle\!\rangle_g}$. This partition structure is called the *issue* expressed by $?\vec{x}.\varphi$.

Definition 10 (Proposition; Issue)
Given model M and assignment g, define for each $\varphi \in L$

$$\begin{array}{lcl} [\![\varphi]\!]_g & = & \{w \mid V_{w,g}(\varphi) = 1\} \\ [\![?\vec{x}.\varphi]\!]_g & = & \{\{w \in W \mid V_{w,g}(?\vec{x}.\varphi) = V_{v,g}(?\vec{x}.\varphi)\} \mid v \in W\} \end{array}$$ □

Incidently, this notion of content can be seen as a projection of the inverse function of the intension applied to the typical result of an utterance of that type, specified by the success conditions. The typical result of an assertion is the truth value 1; a denial has 0. The typical result of a question is a proposition corresponding to a so called *nominal answer*: a set of tuples of objects that possibly satisfy the question relation. A nominal answer $E \subseteq D^n$ is called *possible* in case there is some world conceivable that has it as a complete answer. Impossible answers are those which are not supported by any world, for instance because they violate some meaning postulate or because some presuppositions in the question are not satisfied.

Definition 11 (Projected Inverse)
For any function $f :: A \to B$ define $f^{-1} :: B \to \text{pow}(A)$, the projected inverse of f, by
$f^{-1}(b) = \{a \in A \mid f(a) = b\}$ $(b \in B)$ □

Proposition 2 (Inverse)
For each $\varphi \in L$, its content is the projection of the inverse intension on a typical result:

$$\text{(i)}\quad [\![\varphi]\!]_g = \langle\![\varphi]\!\rangle_g^{-1}(1)$$
$$\text{(ii)}\quad [\![?\vec{x}.\varphi]\!]_g = \{\, \langle\![?\vec{x}.\varphi]\!\rangle_g^{-1}\{v \mid V_v(\lambda\vec{x}.\varphi) = E\} \mid E \subseteq D^n, E \text{ possible}\,\}$$

Proof

$$\begin{aligned}\text{(i)}\quad \langle\![\varphi]\!\rangle_g^{-1}(1) &= \{w \mid \langle\![\varphi]\!\rangle_g(w) = 1\} && \text{(def)}\\ &= \{w \mid V_{w,g}(\varphi) = 1\} && \text{(appl, def)}\\ &= [\![\varphi]\!]_g\end{aligned}$$

(*) A nominal answer $E \subseteq D^n$ to some question $?\vec{x}.\varphi$ is called *possible*, in case there is some $v \in W$ such that $V_{v,g}(\lambda\vec{x}.\varphi) = E$.

$$\begin{aligned}\text{(ii)}\quad &\{\, \langle\![?\vec{x}.\varphi]\!\rangle_g^{-1}(\{v \mid V_v(\lambda\vec{x}.\varphi) = E\}) \mid E \subseteq D^n, E \text{ possible}\,\}\\ =\ &\{\{w \mid \langle\![?\vec{x}.\varphi]\!\rangle_g(w) = \{v \mid V_v(\lambda\vec{x}.\varphi) = E\} \mid E \subseteq D^n, E \text{ possible}\,\} && \text{(def)}\\ =\ &\{\{w \mid V_{w,g}(?\vec{x}.\varphi) = \{v \mid V_v(\lambda\vec{x}.\varphi) = E\} \mid E \subseteq D^n, E \text{ possible}\,\} && \text{(appl, def)}\\ =\ &\{\{w \mid V_{w,g}(?\vec{x}.\varphi) = \{v \mid V_v(\lambda\vec{x}.\varphi) = E\} = V_{u,g}(?\vec{x}.\varphi)\} \mid u \in W\} && \text{(def, *)}\\ =\ &[\![?\vec{x}.\varphi]\!]_g\end{aligned}$$
□

What to make of this result? For assertives, it highlights the special role played by the truth value '1'. For interrogatives, the function $\langle\![\varphi]\!\rangle_g^{-1}$ applied to propositions that describe nominal answers is the identity function. This means that the two ways of generating a partition are essentially the same: grouping worlds that are indistinguishable with respect to an answer, or generating possible answers on the basis of a distribution of nominal answers (provided they are possible) and then finding the worlds to support them. So for the equivalence relation induced by a question, it does not matter whether you take indistinguishability with respect to the n-place abstract or with respect to the question: $\simeq\langle\![\lambda x.\varphi]\!\rangle_g = \simeq\langle\![?x.\varphi]\!\rangle_g$. The distinction between extensions and intensions of questions is therefore less important for an update semantics with questions, where the equivalence relation is the central notion.

3.3.4 Information States

Now we turn to a definition of the set of information states Σ_W, on the basis of a given information space W. How are the separate *data* and *structure* aspects of an information state to be modelled? The factual information of an agent is modelled by a set of possible worlds: the worlds that are compatible with the information of the agent. This set is called the *data set* here. The semantic content of a question corresponds to a partition. This has been motivated above. We will use an equivalence relation to model the structure of issues. Partitions and equivalence relations are interchangeable.

There is a choice. We can have the equivalence relation range over the complete information space, or over the particular data set of an agent. The choice only makes a difference from the external perspective. The first option makes most sense when issues are public: they are known to each participant on the basis of what was said earlier. Public issues may continue to play a role structuring the information space after they have been resolved for some particular agent. The second option makes most sense when it concerns private issues of an agent. Moreover, questions can contain referential expressions that are only defined in a certain subset of worlds. This context dependency is best modelled by relativising issues to a particular data set. Therefore we take the second option, to limit issues to a particular data set.

The aspects of data and structure can now be combined into one equivalence relation σ over a subset of the set of all possible worlds W. From the equivalence relation, the other aspects can be derived when needed. The data aspect $\mathrm{d}(\sigma)$ is modelled as the domain of the equivalence relation σ: the worlds that occur in the relation. The structural aspect $\mathrm{i}(\sigma)$ is modelled by the partition induced by equivalence relation σ. This *combination trick* is used for instance in definitions by (Jäger 1996; Hulstijn 1997; Hulstijn 1998; Groenendijk 1999). It has advantages and disadvantages. One of the advantages is that it makes some definitions more concise. On the other hand, it makes definitions less transparent; after all, these aspects are in fact different.

Thus, an information state $\sigma \in \Sigma_W$ is identified with an equivalence relation over a subset of the information space. In case the information space is clear from context, reference to W is dropped.

Definition 12 (Information states Σ_W)
Given an information space W, define the set of all information states Σ_W by

$$\Sigma_W \;=\; \{\sigma \subseteq W \times W \mid \sigma \text{ reflexive, transitive, symmetric }\} \qquad \square$$

Definition 13 (Information Aspects)
For each information state $\sigma \in \Sigma$ define

$$\begin{array}{lcll} \mathrm{d}(\sigma) & = & \{w \mid \langle w, v\rangle \in \sigma, \text{ for some } v\} & \text{data set} \\ \mathrm{i}(\sigma) & = & \mathrm{d}(\sigma)/\sigma & \text{issue structure} \end{array} \qquad \square$$

There are two privileged states. The initial *ignorant* state **0** contains all epistemic possibilities and is not yet structured. The inconsistent or *absurd* state **1** contains no epistemic possibilities in its data set. After all, no possibilities are compatible with an inconsistency. The absurd state can trivially be structured in any possible way. Because we require that the equivalence relation is restricted to the data set, the equivalence relation is empty too.

Definition 14 (Initial state 0; absurd state 1)
Given information space W, define

$$\begin{array}{lcl} \mathbf{0} & = & W \times W \\ \mathbf{1} & = & \varnothing \end{array} \qquad \square$$

3.3.5 Updates

Now we have a language L and a set of information states Σ. To get a complete update system $\langle L, \Sigma, [.]\rangle$ we still need an update function $[.]$. In this classic framework, the assignment g is fixed at the top level. So updates are of the form $[\varphi]_g :: \Sigma \rightarrow \Sigma$ for each $\varphi \in L$. In the next section, we will study what happens when assignments are included as part of the information state. There are two kinds of updates: assertives and interrogatives.

The assertives affect the data aspect of information. They eliminate those worlds from the data set that are incompatible with the new information. A world w is incompatible with new information φ if, under the current assignment g, φ is not satisfied at w. The interrogatives affect the structure of the information state. They eliminate all pairs of worlds from the information state that violate the equivalence relation induced by the new issue raised. Two worlds are judged to be equivalent when they *agree* on their answer to the question under the current assignment. The operator ';' is defined by sequential composition.

Definition 15 (Updates)
Given assignment g, define for each $\varphi \in L$, $\sigma \in \Sigma_W$ an update function $[\varphi]_g$ by

$\sigma[\varphi]_g$	$=$	$\{\langle v,w\rangle \in \sigma \mid V_{v,g}(\varphi) = V_{w,g}(\varphi) = 1\}$	assertive
$\sigma[?\vec{x}.\varphi]_g$	$=$	$\{\langle v,w\rangle \in \sigma \mid V_{v,g}(?\vec{x}.\varphi) = V_{w,g}(?\vec{x}.\varphi)\}$	interrogative
$\sigma[\varphi;\psi]_g$	$=$	$(\sigma[\varphi]_g)[\psi]_g$	sequence

□

In all cases, the resulting relation is again an equivalence relation. So updates are well defined. Moreover, the result of an update is either equal to or a subset of the original information state. Since the elimination of possibilities models an increase in information, this shows that updates preserve information increase (section 3.5). In the update setting, a distinction between extension and intension of a question is not a very relevant one. What matters is whether two worlds agree. That means that the interrogative clause could have been defined as $\sigma[?\vec{x}.\varphi]_g = \{\langle v,w\rangle \in \sigma \mid V_{v,g}(\lambda\vec{x}.\varphi) = V_{w,g}(\lambda\vec{x}.\varphi)\}$.

The following equivalence shows that in this basic system, for both assertive and interrogative updates the effect of an update 'out of context', i.e. applied to the ignorant context $\mathbf{0}$, equals the notion of semantic content $[\![\varphi]\!]$.

Proposition 3 (Update Content)
For all $\sigma \in \Sigma_W$, $\varphi \in L$

$$[\![\varphi]\!]_g = \mathrm{d}(\mathbf{0}[\varphi]_g)$$
$$[\![?\vec{x}.\varphi]\!]_g = \mathrm{i}(\mathbf{0}[?\vec{x}.\varphi]_g)$$

Proof By definition 10, 13 and 14. □

Proposition 4 shows that updates are nothing but a 'join' of the information state with the relevant content 'out-of-context'. Together, these two properties show that the system is *additive* in the sense of Veltman (1996). This result can in turn be used to show that the kinds of updates defined here, are expansions in the sense of Gärdenfors (1988). Information can only be added if it is consistent; otherwise the absurd state results.

Proposition 4 (Join)
For all $\sigma \in \Sigma_W$, $\varphi \in L$

$$\sigma[\varphi]_g = \sigma \upharpoonright [\![\varphi]\!]_g$$
$$\sigma[?\vec{x}.\varphi]_g = \sigma \cap \simeq\!\langle\![?\vec{x}.\varphi]\!\rangle_g$$

Proof By definition 9, 10, 15, and the preliminaries. □

3.3.6 Tuple-style framework

There is room for some choice. Suppose that we drop the requirement that the equivalence relation is restricted to the current data set. Then we can compare the data set of one agent with the issues raised by another agent. However, if we separate the data set and the equivalence relation, we can no longer use the combination trick of definitions 12, 14 and 13 above. Information states are therefore redefined to be tuples $\langle S, \mathfrak{I}\rangle$, where S is the data set and $\mathfrak{I}$ an equivalence relation over W. Tuple-style notions are distinguished by a $'$ from their regular counterparts. Most notions are equivalent to the ones defined above, except for the absurd state $\mathbf{1}'$. In the tuple-style framework, there is an indefinite number of absurd information states: all states where the data set is empty.

Definition 16 (Information States (ii))
Given an information space W, define

$$\begin{array}{lcl} \Sigma'_W & = & \{\langle S, \mathfrak{I}\rangle \mid S \subseteq W, \mathfrak{I} \subseteq W \times W \text{ reflexive, transitive, symmetric } \} \\ \mathbf{0}' & = & \langle W, W \times W\rangle \\ \mathbf{1}' & = & \langle \varnothing, R\rangle, \ \text{ for any } R \subseteq W \times W \end{array}$$

□

Definition 17 (Information Aspects (ii))
For each $\sigma' = \langle S, \mathfrak{I}\rangle \in \Sigma'_W$

$$\begin{array}{lcl} \mathrm{d}(\sigma') & = & S \\ \mathrm{i}(\sigma') & = & S/\mathfrak{I} \end{array}$$

□

Definition 18 (Updates (ii))
Given assignment g, define for each $\varphi \in L$, and $\langle S, \mathfrak{I}\rangle \in \Sigma'_W$ an update function $[\varphi]_g$ by

$$\begin{array}{lcll} \langle S, \mathfrak{I}\rangle[\varphi]_g & = & \langle S \cap [\![\varphi]\!]_g, \mathfrak{I}\rangle & \text{assertive} \\ \langle S, \mathfrak{I}\rangle[?\vec{x}.\varphi]_g & = & \langle S, \mathfrak{I} \cap \simeq\langle\!\langle ?\vec{x}.\varphi]\!\rangle_g\rangle & \text{interrogative} \\ \langle S, \mathfrak{I}\rangle[\varphi;\psi]_g & = & (\langle S, \mathfrak{I}\rangle[\varphi]_g)[\psi]_g & \text{sequence} \end{array}$$

□

If we add a condition that $\mathfrak{I} \restriction [\![\varphi]\!]_g$ in the clause for assertives above we get a definition for the tuple-style notation that is equivalent to definition 15. This notation highlights again the classic behaviour of the system. There is no context dependency. In the following section we introduce more context dependency.

3.4 'Dynamic Semantics' and Questions

The meaning of many expressions depends on the context. Consider definite descriptions ('the man who killed the sheriff'), deictic expressions ('this') or pronouns ('he', 'she'). Which object is referred to by such an expression depends on the dialogue context. Because the context changes during the dialogue, what can be referred to may change too. This effect may be called dynamic.

In the previous section we used a single assignment g to link values to variables in predicate logical expressions, and therefore also in interrogative and assertive updates. The assignment remained fixed. In a way, M, w and g represent the context of evaluation from an external perspective. In this section we deal with the context dependency of anaphoric expressions from the internal perspective. Assignments are made part of the information states of agents, and updates account for the changes to assignments.

In discourse representation theory (DRT) (Kamp and Reyle 1993) and file change semantics (Heim 1982) a record is kept of the objects that are explicitly mentioned in a discourse. They are indicated by *discourse referents*. In DRT, factual information is represented as a set of conditions defined over discourse referents. Indefinite descriptions typically introduce a referent; anaphoric expressions pick up previously introduced referents. Dynamic predicate logic (DPL) (Groenendijk and Stokhof 1991) and subsequent developments (Dekker 1993; Groenendijk et al. 1996) take over the distinction between factual information and information about the discourse. Discourse referents are represented by variables. Variables can be assigned values: objects from the domain. Factual information is represented by the set of possible worlds compatible with the information. In this way, a 'dynamic semantics' can also be given in a model theoretic framework.

Because we want to account for context dependent assertions and questions, both aspects must now be included in our definition of information states. We combine worlds and assignments into world-assignment pairs. These may then function as *indices*: epistemic possibilities that make up the information space over which information states are defined. Crucially, this adds another option for increasing information. Information can now be added in three ways: (i) by adding structure, removing possibilities from the equivalence relation, (ii) by adding factual information, removing possibilities from the data set and (iii) by adding discourse information, extending the possibilities in the data set with a new assignment of objects to variables.

Because of this possibility to store information about the values of variables in the information state itself, 'dynamic semantics' in principle allows the scope of existential quantifiers to extend to the right. This mechanism models one aspect of coreference: two expressions, for instance an indefinite and an anaphor, refer to the same object. It does not pretend to model the process of anaphora resolution. Anaphora resolution requires a special algorithm, which takes both syntactic and semantic constraints on coreference into account, as well as information about available referents. See for instance Hobbs (1978) for an early account. Instead, in 'dynamic semantics' it is assumed that resolution has already occurred during the interpretation process.

In the first sentence of example (35) a prince is introduced. And although the sentence is finished, the same prince can be referred to by the pronoun 'he' inthe second sentence. In the logical language, a sequence of utterances is represented by ';'. Note that formally, the x in $brave(x)$ is outside the scope of the quantifier.

(35) Once upon a time there was a prince. He was handsome and brave.
$\exists x(prince(x)); (handsome(x) \wedge brave(x))$

There are also systems that distinguish three levels of analysis: variables, referents and objects (Groenendijk et al. 1996). Variables represent coreference phenomena in the language, at the syntactic level. Referents are motivated as independent objects of discussion or thought. They may or may not turn out to correspond to real objects. A second motivation for separate referents has to do random assignment in DPL (Groenendijk and Stokhof 1991). In DPL the re-use of a variable deletes existing values. This can be avoided by adding the extra level of referents. This objective also can be countered by simply demanding that the utterance interpretation module always produces formulas with a fresh variable for each separate use. Given that so much is already assumed, this seems only a small requirement. For the examples we use, the role of referents can be performed by variables.

3.4.1 Indices

Epistemic possibilities or *indices* i, j are combinations of worlds and assignments. On the basis of indices we can re-use most of the earlier definitions. The information space I is simply the set of all such indices.

We recall the definition of assignments. An assignment is a partial function $g :: X \rightarrow E$, where $X \subseteq Var$ is the *domain* of g, and $E \subseteq D$ is the *range* of g, written dom(g) and range(g) respectively. The set G is the set of all such assignments. In case $x \notin \text{dom}(g)$, $g(x)$ is undefined. Assignments can be extended[1]: $g[x \mapsto d] = g'$, where $g'(x) = d$ and $g'(y) = g(y)$ for $x \neq y$. So $\text{dom}(g[x \mapsto d]) = \text{dom}(g) \cup \{x\}$ and $\text{range}(g[x \mapsto d]) = \text{range}(g) \cup \{d\}$. There is a partial order $\leq$ over assignments that indicates when one assignment *extends* another. Formally, $g \leq g'$ iff $\text{dom}(g) \subseteq \text{dom}(g')$ and $g(x) = g'(x)$ for all $x \in \text{dom}(g)$. Typically $g \leq g[x \mapsto d]$. The *empty* assignment, $\{\}$, assigns no value to any variable. It is the minimal element in the $\leq$-order. Indices too can be extended. If $i = \langle w, g \rangle$, then by $i[x \mapsto d]$ we mean the pair $\langle w, g[x \mapsto d] \rangle$. We say that $j = \langle w_j, g_j \rangle$ extends $i = \langle w_i, g_i \rangle$, written $i \leq j$, when $w_i = w_j$ and $g_i \leq g_j$.

The definitions are adapted accordingly. Given the original model $M = \langle D, A, W, ..., V \rangle$, define an update semantics $\langle L, \Sigma_I, [.] \rangle$ consisting of the language L from definition 4, a set of information states Σ_I and an update function $[\varphi]$ for each $\varphi \in L$. The set of all possible information states ranges over I. The absurd state is again $\mathbf{1} = \varnothing$, but for the initial state it is more complicated. Initially the agent has no discourse information, so in the ignorant state $\mathbf{0}$ assignments are empty.

Definition 19 (Information States (iii))
Given a set of worlds W and a set of assignments G define

$$\begin{array}{lcl} I & = & \{\langle w, g \rangle \mid w \in W, g \in G\} \\ \Sigma_I & = & \{\sigma \subseteq I \times I \mid \sigma \text{ reflexive, transitive, symmetric}\} \\ I_\mathbf{0} & = & \{\langle w, \{\} \rangle \mid w \in W\} \\ \mathbf{0} & = & I_\mathbf{0} \times I_\mathbf{0} \\ \mathbf{1} & = & \varnothing \end{array}$$

□

The valuation function now ranges over indices. For each index $i = \langle w, g \rangle$ define $V_i(\varphi) = V_{w,g}(\varphi)$ for arbitrary expressions φ. Also the notions of proposition and issue can be adapted: $[\![\varphi]\!] = \{i \in I \mid V_i(\varphi) = 1\}$ and $[\![?\vec{x}.\varphi]\!] = \{\{i \mid V_i(?\vec{x}.\varphi) = V_j(?\vec{x}.\varphi)\} \mid j \in I\}$. Indices represent the context of evaluation, which can possibly be abstracted over. So $\langle\!\![\varphi]\!\!\rangle = \lambda i.V_i(\varphi)$ for any expression φ.

3.4.2 Random Assignment

In 'dynamic semantics', the existential quantifier is used to indicate that a new discourse referent is introduced. There are several possible ways of defining such an introduction. The standard solution is called *random assignment*: for each world-assignment pair in the information state, a variable x is assigned an arbitrary object d. A random assignment generates all possible values for x. The rest of the formula, the matrix, is then used to eliminate all incompatible indices. The net effect is therefore the introduction of a referent that meets the constraints in the matrix of the expression.

[1] In some systems the assignments of a value to a variable can also be changed. If that is needed, it can be solved by substituting a fresh variable with the new value.

There is an important choice to be made. We deal with pairs of indices in a equivalence relation. We could (i) assign random objects to the variable in each of them, or (ii) assign the same objects to the variable. This last option is called a *specific assignment* by Groenendijk (1998). In both cases, the resulting information state is again an equivalence relation.

Definition 20 (Random Assignment)
For each $\sigma \in \Sigma$, $x \in Var$

(i) $\sigma[x] = \{\langle i[x \mapsto d], j[x \mapsto e]\rangle \mid \langle i,j\rangle \in \sigma \text{ and } d, e \in D\}$

(ii) $\sigma[x] = \{\langle i[x \mapsto d], j[x \mapsto d]\rangle \mid \langle i,j\rangle \in \sigma \text{ and } d \in D\}$

In case (i), we get a standard random assignment. In case (ii), we keep some more structure. Worlds in the same block will try the same object simultaneously. So the existing partition is further refined with blocks that correspond to the chosen object. So $\sigma[x]$ implicitly generates an issue: $?y.(y = x)$ or "Who is it?". This is in line with our intuitions; if a new topic is introduced, this generates identity questions as to the nature of the topic. Consider the standard beginning to a fairy tale: "Once upon a time, there was a little prince." This makes us curious. Who is this prince? Is he handsome? What kind of adventures will he be in?

In our new definition, all updates are compositionally defined in terms of updates with a smaller complexity. The two-level definition of satisfaction and updates, is dropped. However, there is a slight complication for questions. We would like to define $\sigma[?\vec{x}.\varphi] = \{\langle i,j\rangle \in \sigma \mid V_i(?\vec{x}.\varphi) = V_j(?\vec{x}.\varphi)\}$, but we can not use $V_i(?\vec{x}.\varphi)$ directly, because it depends on the satisfaction definition 7. For one thing, it would mix two styles of definition. Moreover, anaphoric expressions are dependent on the context σ. How do we solve this?

First, we borrow an auxiliary notion from Groenendijk (1998): an index is said to *subsist* in an information state when there is an index based on the same world that extends its assignment. Recall that typically $i \leq i[\vec{x} \mapsto \vec{d}]$.

Definition 21 (Subsistence)
For $i \in I$, $\tau \in \Sigma_I$, i subsists in τ, $i \preccurlyeq \tau$, if there is an $i' \in \mathrm{d}(\tau)$ such that $i \leq i'$.

Second, we define a context dependent extension V_i^σ. For questions it gives the complete answer to the question at an index $i \in \mathrm{d}(\sigma)$, even if that involves temporarily extending the assignment in i. For completeness the corresponding notions for terms and assertives are also redefined. Now, two indices are indistinguishable with respect to their answers, when they have the same extension in context.

Definition 22 (Extension in Context)
Given $\sigma \in \Sigma_I$, define for each $\varphi \in L_0$ the extension $V_i^\sigma(\varphi)$ at some $i \in \mathrm{d}(\sigma)$ by

$$\begin{array}{lcl} V_i^\sigma(t) & = & V_i(t) \\ V_i^\sigma(\varphi) & = & \begin{cases} 1, & \text{if } i \preccurlyeq \sigma[\varphi] \\ 0, & \text{otherwise} \end{cases} \\ V_i^\sigma(?\vec{x}.\varphi) & = & \{j \mid V_j^\sigma(\lambda\vec{x}.\varphi) = V_i^\sigma(\lambda\vec{x}.\varphi)\} \\ & & \text{where } V_i^\sigma(\lambda\vec{x}.\varphi) = \{\vec{d} \mid i[\vec{x} \mapsto \vec{d}] \in \mathrm{d}(\sigma[\vec{x}][\varphi])\} \end{array}$$

□

3.4.3 Updates

Now we come to the definition of a 'dynamic semantics with questions'. The first five clauses concern *assertive updates*: updates that affect the data aspect of an information state, including discourse information. The sixth clause concerns an *interrogative update*: it affects the structure of an information state. The seventh clause for *sequences of updates* defines how updates can be combined by sequential composition.

Definition 23 (Updates)
For each $\sigma \in \Sigma_I$, define for all $\varphi \in L$ a partial update function $[\varphi]$ as follows

1. $\sigma[P(\vec{t})] \quad = \quad \sigma \restriction \{i \mid V_i(\vec{t}) \in V_i(P)\}$
2. $\sigma[t_1 = t_2] \quad = \quad \sigma \restriction \{i \mid V_i(t_1) = V_i(t_2)\}$
3. $\sigma[\neg\varphi] \quad = \quad \sigma \restriction \{i \mid i \not\preceq \sigma[\varphi]\}$
4. $\sigma[\varphi \wedge \psi] \quad = \quad \sigma[\varphi][\psi]$
5. $\sigma[\exists x.\varphi] \quad = \quad \sigma[x][\varphi]$
6. $\sigma[?\vec{x}.\varphi] \quad = \quad \{\langle i, j\rangle \in \sigma \mid V_i^{\sigma}(?\vec{x}.\varphi) = V_j^{\sigma}(?\vec{x}.\varphi)\}$
7. $\sigma[\varphi;\psi] \quad = \quad \sigma[\varphi][\psi]$ □

We discuss the clauses of definition 23 in detail. Disjunction, implication and universal quantification are defined by the usual equivalences.

1. For *atomic formulas* all incompatible possibilities are eliminated from the information state. The $\restriction$ operator makes sure that the resulting information state is an equivalence relation again. Moreover, it makes sure that indices for which clauses with free variables are undefined, are eliminated. After all, if $V_i(\vec{t})$ is undefined, then $V_i(\vec{t}) \notin V_i(P)$.

2. Also the update for *identity* statements is assertive. It makes use of the restriction operator too. Two objects are judged identical if their denotation at each index coincides.

3. Often *negation* is modelled by set complement $\setminus$, but an expression like $\sigma \setminus \sigma[\varphi]$ represents no longer an equivalence relation. Therefore we use the restriction operator $\restriction$. The use of restriction guarantees that the update is assertive: it affects only the data aspect. In case questions could be embedded under negation – which is syntactically excluded – any interrogative structure resulting from the embedding would be lost at the level of the negation. Negation also blocks the extension of assignments. Objects introduced inside the scope of a negation can not be referred to later. And because they are defined in terms of negation and conjunction, disjunction and implication also block anaphoric reference. This behaviour corresponds to observations in the literature of dynamic semantics, e.g. Heim (1982). Witness example (36), (37) and (38). Throughout the thesis, utterances that are judged infelicitous are marked by an '*'.

(36) John does not have a sister. *She is an actress.

(37) John has a sister or a brother. *She is an actress.

(38) If John has a sister, he will not inherit the estate. *She is an actress.

4. *Conjunction* is modelled by function composition on updates. This gives a dynamic notion of conjunction. The word 'dynamic' means non-commutative in this context. So $(\varphi \wedge \psi) \not\leftrightarrow (\psi \wedge \varphi)$. For example $\sigma[\exists x(prince(x)) \wedge brave(x)] \neq \sigma[brave(x) \wedge \exists x(prince(x))]$, because the value for x in $brave(x)$ is still undefined. The order in which clauses are used matters. Defining '$\wedge$' in this way is a bit superfluous when we already have the ';'

operator, although ';' is allowed at the level of L, whereas $\wedge$ only occurs inside formulas of L_0. Alternatively, we could define a commutative conjunction by $\sigma[\varphi \wedge_c \psi] = \sigma[\varphi] \cap \sigma[\psi]$. As it stands, conjunction does not block the projection of new referents. It would not block embedded interrogative structure either, if that were syntactically allowed.

Because *disjunction* and *implication* are defined in terms of negation and conjunction, they are now dynamic too. For disjunction at least this gives some empirical problems. By defining $\vee$ in terms of $\wedge_c$ these problems can be overcome. The following results are similar to definitions of dynamic connectives in Groenendijk et al. (1996) for example.

Proposition 5 (Dynamic $\rightarrow$)
For all $\sigma \in \Sigma$ and $\varphi, \psi \in L$, $\sigma[\varphi \rightarrow \psi] \;=\; \sigma \restriction \{i \mid \text{ if } i \preccurlyeq \sigma[\varphi], \text{ then } i \preccurlyeq \sigma[\varphi][\psi]\}$.
Proof

$$\begin{aligned} \sigma[\varphi \rightarrow \psi] &= \sigma[\neg(\varphi \wedge \neg\psi] \\ &= \sigma \restriction \{i \mid i \not\preccurlyeq \sigma[\varphi \wedge \neg\psi]\} && \text{(def)} \\ &= \sigma \restriction \{i \mid i \not\preccurlyeq (\sigma[\varphi] \restriction \{j \mid j \not\preccurlyeq \sigma[\varphi][\psi]\})\} && \text{(def)} \\ &= \sigma \restriction \{i \mid \text{ if } i \preccurlyeq \sigma[\varphi], \text{ then } i \preccurlyeq \sigma[\varphi][\psi]\} && (\text{meta} \restriction, \preccurlyeq) \end{aligned}$$

□

Proposition 6 (Dynamic $\vee$)
For all $\sigma \in \Sigma$ and $\varphi, \psi \in L$, $\sigma[\varphi \vee \psi] \;=\; \sigma \restriction \{i \mid i \preccurlyeq \sigma[\varphi] \text{ or } i \preccurlyeq \sigma[\neg\varphi][\psi]\}$.
Proof Similar to proposition 5. □

5. *Existential quantification* is used to introduce discourse referents. It is defined by random assignment. Since discourse referents are part of the data aspect of information, a random assignment is an assertive update. But unlike other assertive updates we can not use the restriction operator: extending an assignment is a real extension. Earlier we demanded that the interpretation module produces a fresh variable for each use. Alternatively, we could redefine the clause for existential expressions to include this provision: $\sigma[\exists x.\varphi] = \sigma[y][\varphi\{y/x\}]$for $y \notin \text{dom}(\sigma)$, where $\varphi\{y/x\}$ indicates formula φ with each occurrence of x substituted by y and where $\text{dom}(\sigma) = \bigcap_{i \in \text{d}(\sigma)} \text{dom}(g_i)$, which represents the set of variables that are defined in σ.

Universal quantification is defined in terms of negation and existential quantification. Similar to propositions 5 and 6 we have $\sigma[\forall x.\varphi] = \sigma \restriction \{i \mid \text{ if } i \preccurlyeq \sigma[x], \text{ then } i \preccurlyeq \sigma[x][\varphi]\}$. Universal quantification blocks the introduction of new referents inside its scope. This conforms to some observations from the literature. Witness example (39), but see below for a discussion.

(39) Everybody has a sister. *She is an actress.

6. The clause for *questions* eliminates those pairs of indices that are not indistinguishable with respect to their extension in context, as discussed above. Again, we could have used indistinguishability on the basis of the underlying abstract: $\sigma[?\vec{x}.\varphi] = \{\langle i, j\rangle \in \sigma \mid V_i^\sigma(\lambda\vec{x}.\varphi) = V_j^\sigma(\lambda\vec{x}.\varphi)\}$. The way the definition is set up, questions block the accessibility of referents introduced inside their scope, just like negation and universal quantification. This makes the logic more transparent. Assertives reduce worlds, existentials extend assignments and interrogatives refine structure. And indeed, it seems to conform to linguistic intuition. Witness example (40), but see below for a discussion.

(40) A: Does John have a sister?
B: *She is an actress.

7. Finally, sequences of updates are dealt with by sequential composition.

3.4.4 Discussion

We have a logic that defines three ways of adding information to information states. On the basis of a mixture of empirical observations and technical reasons of symmetry, this logic predicts that in some cases referents are or are not accessible for coreference, and that in some other cases interrogative structure is projected or blocked. The set of predictions that we get now, is rather standard in dynamic semantics and also conforms to other systems of update semantics with questions, in particular Groenendijk (1998). However, these predictions need some discussion.

There are two kinds of remarks. The first concerns *modal subordination* (Roberts 1989). The non-actual contexts which are introduced by modal operators for example, are usually inaccessible for anaphoric reference. The operator 'wants' creates a non-actual context: "John wants to have a sister. *She is an actress". The mechanism of modal subordination can be used to make such contexts accessible for referents after all. The referring expressions are specially marked, usually by a counterfactual (would). The context of evaluation of the second utterance is made subordinate to the first; the hypothetical reasoning continues. Contrast the blocking of examples (38) and (36) with the following examples of modal subordination. In (41) the speaker continues the hypothesis that John has a sister, and speculates about her attitudes. In (42) 'she' could refer to a fantasy sister of John. And even (43) can be felicitous when the topic of conversation is John's acting skills and participants can imagine a sister to John.

(41) If John has a sister, he will not inherit the estate. She would spend the money.

(42) John wants to have a sister. She would be an actress.

(43) John does not have a sister. She would be an actress though.

We do not have a complete theory of modal subordination, but we can predict on the basis of definition 23 and proposition 5 and 6 which contexts are available. When the definition requires temporary calculation of some information state, this becomes available to be continued in modal subordination. So for negation $\sigma[\neg\varphi]$ we expect $\sigma[\varphi]$ to be available. For implication, $\sigma[\varphi \rightarrow \psi]$ we expect $\sigma[\varphi]$ to be available, and later $\sigma[\varphi][\psi]$. For dynamic disjunction, we expect $\sigma[\varphi]$ and $\sigma[\neg\varphi][\psi]$ to be available and for commutative disjunction we expect $\sigma[\varphi]$ and $\sigma[\psi]$. Most importantly, for questions we expect each of the blocks in the partition to provide a context for possible continuation.

Now lets go back to example (39). A variable introduced in the scope of a universal quantifier is blocked, just like in the similar (44a) below. However, if the referring statement is meant in a general or universal sense, as in (44b), the variable can be picked up for reference after all! The variable represents a kind of arbitrary object. The mechanism of continuing with the referent in a 'universal context' resembles modal subordination, but without the special marking. "It would be red" is just as infelicitous. Read (44a) as: "In every case a house has a roof", where phrases like 'in every case' or modal adverbs like 'usually' introduce the non-actual context. Often existentially quantified utterances have an implicit universal reading, as in (44c). Roofs of houses aren't usually red, which explains why a continuation of the universal context is not plausible in (44a).

(44) a. Every house has a roof. *It is red.
b. Every house has a roof. It protects the inhabitants from rain and wind.
c. A house has a roof. It protects the inhabitants from rain and wind.

So it appears that modal subordination, or more generally a continuation of the context in which the referent was introduced, is a possible explanation for anaphoric reference across blocking operators. However, this type of subordinated coreference takes effort: the referring expression must be marked as a counterfactual or must otherwise be recognisable as a continuation of the non-actual context. The mechanism resembles attachment of a discourse segment by means of a coherence relation, see for example (Asher and Lascarides 1998b).

Now, what about questions? Recall example (40) where a referent introduced in a question is blocked. On the other hand it seems we can in fact refer to referents introduced inside a question. Witness examples (45) and (46).

(45) Does John have a sister? Is she an actress?

(46) Who owns a red car? It's headlights are still on.

The position of Groenendijk (1998) is that such cases can be handled by the same mechanisms that handle anaphoric relationships in case of modal subordination. In example (45) the referring phrase is also a question, marking a continuation of the 'question context'. We do not believe that this is a straight case of modal subordination. These question continuations are easier and less marked than the counterfactuals in examples (41), (42) and (43). The question continuation is closer to the universal continuation in (44). In example (46) the question context is not continued. But here the 'explanation for asking', presupposes that there is a specific red car with the lights still on. The referent is introduced outside the question. (46) is equivalent to "There is a red car with the headlights still on. Who owns it?".

A complicating factor is that we are dealing with dialogue here. Witness example (47). If *B* answers 'yes', thereby establishing the existence of a sister for John, then the referent is made accessible. On the other hand, if *B* answers 'no', the non-existence of a sister of John is established and the referent is inaccessible because of the implied negation. So (48) is completely analogous to (36).

(47) A: Does John have a sister? $?.\exists x(sister\_of(john, x));$
B: Yes. She is an actress. $\exists y(sister\_of(john, y))\ ; actress(y)$

(48) A: Does John have a sister? $?.\exists x(sister\_of(john, x));$
B: No. *She is an actress. $\neg\exists y(sister\_of(john, y)); -$

The same mechanism works for wh-questions. Consider (49) and (50), which is one of those examples where the answer presupposition can be cancelled without a problem.

(49) A: Who has a sister? $?x.(\exists y.sister\_of(x, y));$
B: John does. She is an actress. $\exists z.sister\_of(john, z)); actress(z)$

(50) A: Who has a sister? $?x.(\exists y.sister\_of(x, y));$
B: Nobody. *She is an actress. $\neg\exists z.sister\_of(john, z)); -$

First of all, answering 'yes' is not a way to mark non-actual subordinate contexts. In fact, the answer 'yes' makes the answer actual. That suggests that in general inaccessibility of a referent might be caused by *indeterminacy*. In some blocks of the information state there is a referent available, in others not. When the indeterminacy is resolved the referent becomes available. That means that in the appropriate blocks, the referent was already there! Such a view is impossible under definition 22 and 23. The indices are only extended temporarily for calculating a question. Even after answering the question, the extended assignments won’t come back.

But that is only apparent. Consider the way the answers are represented in examples (47) – (50). Incomplete natural language expressions like ‘yes’ or ‘John does’ do not have an incomplete representation in the logic. All answers are propositional and to derive them in the interpretation module we apply the representation of the actual incomplete answer, to the lambda-abstract that underlies the question. In other words, the preceding question is a crucial aspect of the interpretation of an answer. The resulting complete answers carry enough information by themselves to license anaphoric reference later. Technically it works, but intuitively this is a bit unsatisfactory.

The hypothesis of indeterminacy suggests an analogy with the inaccessibility of disjunctive and conditional sentences in (37) and (38), which are also undetermined in this way. The possibility that the referent might not exist given the information conveyed by the utterance, means that the referent is not accessible. But it could become accessible if more information were available.

Taking this analogy the other way, it suggests that disjunctions or implications trigger an issue to be raised. By cooperative principles the speaker must be as informative as possible. So if the speaker chooses a disjunction or a conditional rather than a regular statement, that implicates an issue. In particular, the issue which of the disjuncts, or whether the antecedent of the conditional is true. If this issue is resolved positively, then a referent introduced under the scope of a disjunction or implication can become accessible after all. Otherwise, it is definitely blocked. If we view an existentially quantified expression as a large disjunction of formulas for each of the possible referents, that suggests that an existential expression triggers a wh-question as to the identity of the referent. Witness the relevance and pertinence of B’s remark in (53). And this conforms nicely to the intuition behind our choice for a *specific assignment* in definition 20. It seems we expect a ‘witness’ object for each existential expression.

(51) A: John has a sister or a brother. (Which?)
B: He has a sister. She is an actress.

(52) A: If John has a sister, he will not inherit the estate. (Does he?)
B: He does. She is an actress.

(53) A: Someone has a sister. (Who is it?)
B: John does. She is an actress.

The increase of information proceeds block wise. This is implicated by the licensing condition that will be discussed in section 3.6. What the relevant blocks are, is implicated by the way an utterance is formulated. The fact that B’s answers in the examples above are licensed, indicates that disjunctions, existentials and implications trigger an issue as to the truth of their component formulas.

3.4.5 Cross Speaker Anaphora

There is a more severe complication. In dialogue, referents are introduced by different speakers. That means that anaphoric expressions refer across information states of different speakers. That is why this phenomena is called *cross speaker anaphora* (Dekker 1997). In the introduction of the chapter we assumed ideal circumstances for our inquiry game, and we took the standpoint of a neutral observer, removing the need for explicit grounding measures. But this idealisation is not always warranted. What can we say about cross speaker anaphora using the current machinery?

We stick to the perspective of the neutral observer. However, now the observer is monitoring the different information states that participants are committed to on the basis of their utterances in public (cf. Hamblin 1970). So we assume for each participant a, b a public information state σ_a, σ_b. As we remarked in chapter 2, if the basis for these information states is public and readily accessible, we can assume that information about each participant's information state is common ground. In this case, the basis is a record of what was literally uttered. An important part of the grounding process ensures that the basis is public and accessible.

All utterances that are established as public, are processed as updates. Assertive utterances must be *sincere*. If a utters φ, that means that a is publicly committed to believing φ by the Gricean maxim of quality. So we update a's public information state with φ. If b thinks that a is *reliable* on the matter of φ, b will also update his information state with φ. Now, if a utters a question, or otherwise raises an issue $?\varphi$, again by sincerity a's information state will be updated with the question. And because of *cooperativity* b takes over a's issue. So, for both assertives and interrogatives the public information states of participants align because of the principles of sincerity, reliability and cooperativity. What about the establishment of a referent?

A minimal requirement for the establishment of a referent is some mutual acknowledgement of receipt and understanding of the utterance in which the referent is introduced. In short, the public nature of the utterance in which the referent is introduced must be *grounded*. As we saw in chapter 2 grounding proceeds in two stages: a presentation phase and an acceptance phase. Thus, these question-answer sequences are examples of grounding sequences. We expect a similar pattern for declarative sentences, with an assertion and an acknowledgement. That is why B's utterance in (54) sounds odd in isolation and why (55) is perfectly acceptable.

(54) A: John has a sister.
B: *She is an actress.

(55) A: John has a sister.
B: Uhum. She is an actress.

Now if this is true of declarative sentences, then maybe the data on the basis of which we decided that questions block anaphoric referents is not conclusive after all! Answering 'yes' in (47) is analogous to the acknowledgement in (55) above. If (54) is odd without an acknowledgement, clearly (40) is odd without an answer.

We might say that the grounding process triggers an issue: a *grounding issue*. And indeed Ginzburg, (1996, p 417) suggests that for every assertive utterance with content φ, $?\varphi$ is pushed onto the stack of questions under discussion. The grounding issue is typically resolved when the utterance is acknowledged in some way. This can happen explicitly, by nodding or saying 'uhum', but also implicitly, by responding with an utterance that generally makes sense in the context. For example, the relevance of B's remarks in (51), (52) and (53) indicates that now A's utterance is grounded. If we follow Ginzburg, then the acknowledgement of (55) is a positive answer to the grounding issue. That would make (47) and (55) analogous indeed. Unfortunately, things are not that simple.

As we remarked in chapter 2, grounding may take place at different levels. We distinguish the locutionary level of the utterance event and the illocutionary level with the establishment of the content. A minimal requirement for cross reference between speakers is some indication of receipt and understanding of an utterance event. Speakers do not have to agree on the content to be able to use a referent. We might say that speakers may continue refer to each other's referents 'for the sake of the argument'. The following examples from Dekker (1997) are radical illustrations of this point. Speakers may even correct the conditions that were used for introducing the referent. The corrected material is typically marked by contrastive stress, indicated in italics.

(56) A: Yesterday, Tom went to the Zoo with the youngest son of a colleague of his. The boy enjoyed it very much to spend the day with him.
B: *Sure*, because *he* as a matter of fact was a nineteen year old *girl*, and because they went dancing instead.

(57) R: A man is sleeping on a park bench over there.
N: It is not a *man*, it is a *wo*man, and she is not *a*sleep, she is just *sun* bathing. Besides, it is not a park bench.

The notion of a referent that is established 'for the sake of the argument' is an important one. It may help to explain *cataphora*. In general, a cataphor is an anaphor that refers to an antecedent which is only to be provided later. Because the introduction of a cataphor implies a question as to the identity of the referent, a cataphor is a powerful literary device for capturing the attention of an audience.

(58) He was an inch, perhaps two, under six feet, powerfully built, and he advanced straight at you with a slight stoop of the shoulders, head forward, and a fixed from-under stare which made you think of a charging bull. (Opening sentence of *Lord Jim*, Joseph Conrad, 1900)

Cataphora are also possible in multi-speaker settings. In that case we get so called *cross speaker cataphora*. The following example is from Henk Zeevat.

(59) Het hangt aan de muur en het tikt. ... Ra ra, wat is het?
It's on the wall and it ticks. ... Guess what is it?

Example (59) is the conventional Dutch way of opening a question game. The initiator of the game has a definite object in mind. The inquirer may ask questions about it, which can only be answered with 'yes' or 'no'. If the inquirer manages to guess what the object is, for instance a clock, she wins; otherwise the initiator wins. This game is a special case of the inquiry dialogues introduced in the beginning of this chapter. The sheer possibility of such a question game suggests that there is a mechanism for cross speaker anaphora 'for the sake of the argument'. In other words, an as yet unresolved referent may still be the topic of conversation.

These examples seem problematic. Can we find constraints that define when cross speaker anaphora are allowed? In fact a beginning of a solution was already given above and in chapter 2: levels and layers. First, grounding takes place at different levels. Anaphoric referents can be picked up 'for the sake of the argument', as long as the utterance event that introduced it is established as public. That gives enough foothold for the referents to be anaphorically referred to. The responsibility for the referent remains with the participant that initially introduced it. Such temporary grounding is marked, because the default is that grounding takes place both at the utterance and at the content level. Second, we could say that a stream of information exchange about non-actual topics constitutes a different *layer* of the conversation, in the sense in which Clark (1996; Ch11) uses the word. There is an actual layer and a layer in which participants refer to each other's possibly non-actual objects. Layers are found everywhere where meanings are taken non-literally. A good example is the level of a question game, which is different from the actual conversation level.

What can we conclude from this discussion? Here are the preliminary results of our interpretation of examples (36) to (59).

1. Referents introduced under a negation, implication, disjunction or universal quantifier are available for anaphoric reference only by modal subordination.
2. Referents introduced in a question are available for anaphoric reference, either by modal subordination in another question, or by resolving the issue in the answer. The answer makes a non-actual context, actual.
3. Utterances that are 'pragmatically undetermined' such as disjunctions, implications and existentially quantified expressions, trigger issues to resolve the indeterminacy.
4. Assertives too trigger grounding issues.
5. Constraints for cross speaker anaphora must take the difference between the utterance level and the content level into account.

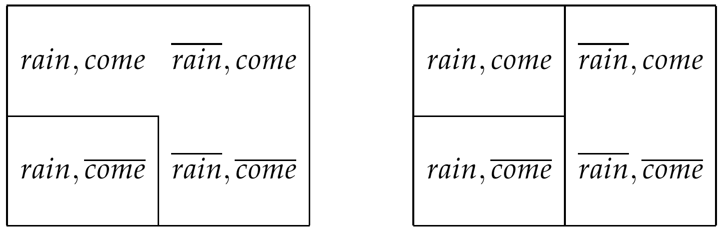
Can we account for this data in one system? The answer is maybe. Here are a number of principles to build an approach on. We demand that conversation proceeds 'block wise'. In the next section, this demand will be called *licensing*. Current issues indicate which blocks to proceed by. Introduction of referents is no problem, as long as it too proceeds block wise. Anaphoric reference is a process which, for each block, tries to find an appropriate referent. The inaccessibility of referents in case of disjunction, implication and questions is explained by the possibility of non-existence for some blocks. Now, if the conversation continues, you have to connect the new information to the old information in a coherent way. In order to refer back, you have to link the anaphoric expression to the particular context or 'topic space' where its referent was introduced, compare e.g. (Asher and Lascarides 1998a).

As a final test of intuitions, consider the following conditional question.

(60) Do you come to the party if it rains? $?(rain \rightarrow come)$

By definition $?(rain \rightarrow come)$ is equivalent to $?(\neg(rain \wedge \neg come)$. Because a negative question is equivalent to a positive one[2], we predict that it is equivalent to $?(rain \wedge \neg come)$.

[2]This is not completely true. A positively phrased yes/no question "Do you want coffee?" indicates some pragmatic preference for the positive answer, whereas a negatively phrased version, "Don't you want coffee?" already anticipates on a negative "No, thanks".



$\sigma[?(rain \rightarrow come)]$ $\sigma[rain \Rightarrow ?come]$

Figure 3.5: Conditional questions in example (60) and the definition of $\rightarrow$ and $\Rightarrow$

But the question could also be paraphrased as: "Suppose it rains. Would you come to the party then?". Clearly an example of modal subordination. The question is concerned with the non-actual situation in which it would rain. Under this view, the question is embedded, $rain \rightarrow ?come$, which is syntactically not allowed. This can be remedied as follows. We define an operator $\Rightarrow$ at the level of updates, to express a *guarded conditional*. First divide the information state in two blocks: one where the antecedent of the conditional is true, and one where it is false. This may be called the 'guard'. Then update the true block with the consequent. The use of $\cup$ in the definition is warranted because the two equivalence classes resulting from the guard are disjoint.

Definition 24 (Guarded Conditional)
For all $\sigma \in \Sigma$ and $\varphi, \psi \in L$ define
$\sigma[\varphi \Rightarrow \psi] \;=\; \sigma[?\varphi][\varphi][\psi] \cup \sigma[?\varphi][\neg\varphi]$ □

The difference shows in figure 3.5. On the left, we see an update of $?(rain \rightarrow come)$ with the traditional definition of implication. On the right an update of $(rain \Rightarrow ?come)$ is depicted, using definition 24. Which notion conforms best to our intuitions? Suppose that you are asked "Will you come to the party if it rains?", by an organiser of a garden party. What does it mean to say 'no'? In our view it means that in case it rains you won't come, and in case it stays dry you haven't committed yourself either way. If you say 'yes' it means, that if it rains you will come. Now, according to the standard definition, at a 'no' we end up in the bottom left corner of figure 3.5, which is correct. After a 'yes' however, we have no definite commitment either way. The 'yes'-block contains both 'come' and 'not come'. For the guarded conditional we get the same result for a 'no'. But for a 'yes', it means a commitment to come in case it rains. So it seems the guarded update resembles our intuitions closer.

Analogous to the guarded conditional, we could define other connectives at the utterance level. These might then satisfy the requirements that we summarised above. We have no definite solution. One of the difficulties is again the interplay between different levels. For grounding effects we have interference with the locutionary level of utterance events. But if we define a guarded conditional, a guarded choice or a guarded assertion which triggers a grounding question, we interfere with the task and the interaction level. Such operators connect utterances at the level of dialogue acts into meaningful sequences. Coherence of dialogue sequences is discussed in chapter 4 and 5. This chapter was meant to focus on properties that define coherence at the content level. For this, we have to go into the resolution of issues.

3.5 Resolution

The approach advocated in this chapter might be described as one of 'raising and resolving issues'. In this section we discuss the notions that have to do with resolving an issue or answering a question. In the introduction we used the metaphor of an issue as a set of alternatives. An issue is resolved, when only one alternative remains. An issue is partially resolved, when one alternative is removed. To make these intuitions formal, we start by defining an information order on information states.

Updates result in an increase of information: factual, discourse or structural information. How can we measure and compare the amount of information after an update? There is an information order $\sqsubseteq$ defined over information states. We say that $\sigma \sqsubseteq \tau$ when τ contains at least as much information as σ. Specific information orders for the data and structural aspects of an information state can also be defined. We say that $\sigma \sqsubseteq_{\text{d}} \tau$ when τ contains at least as much data as σ, and $\sigma \sqsubseteq_{\text{i}} \tau$ when τ is at least as much structured as σ. This is the case, when each block in the partition of σ is either refined, or else eliminated in τ.

Definition 25 (Information order (i))
For each $\sigma, \tau \in \Sigma_W$ define

$\sigma \sqsubseteq \tau$ iff $\tau \subseteq \sigma$,
$\sigma \sqsubseteq_{\text{d}} \tau$ iff $\text{d}(\tau) \subseteq \text{d}(\sigma)$,
$\sigma \sqsubseteq_{\text{i}} \tau$ iff for all $V \in \text{i}(\tau)$, there is a $U \in \text{i}(\sigma)$ such that $V \subseteq U$. □

For the tuple view, we use a similar definition.

Definition 26 (Information order (ii))
For each $\sigma_1 = \langle S_1, \mathfrak{I}_1 \rangle, \sigma_2 = \langle S_2, \mathfrak{I}_2 \rangle \in \Sigma'_W$ $\sigma_1 \sqsubseteq \sigma_2$ iff $S_2 \subseteq S_1$ and $\mathfrak{I}_2 \subseteq \mathfrak{I}_1$. □

For the dynamic version, we need yet another definition. After all, assignments can be extended with new values for variables. Increasing information means either elimination of possibilities or extension of assignments. We make use of the following auxiliary definition. For the classic systems simply read $U \supseteq V$ for sets of worlds, whenever we find $U \leq V$ in the definition.

Definition 27 (Extend)
A set of indices V extends U, $U \leq V$, iff for each $j \in V$ there is an $i \in U$ such that $i \leq j$. □

An information state τ contains at least as much discourse information as σ, $\sigma \sqsubseteq_{\text{disc}} \tau$, if its domain of discourse, the set of variables that are introduced, subsumes that of σ. Recall that $\text{dom}(\sigma) = \bigcup_{i \in \text{d}(\sigma)} \text{dom}(i)$.

Definition 28 (Information order (iii))
For each $\sigma, \tau \in \Sigma_I$ define

$\sigma \sqsubseteq \tau$ iff for all $\langle i, j \rangle \in \sigma$, there are $\langle i', j' \rangle \in \tau$ such that $i \leq i'$ and $j \leq j'$,
$\sigma \sqsubseteq_{\text{d}} \tau$ iff $\text{d}(\sigma) \leq \text{d}(\tau)$,
$\sigma \sqsubseteq_{\text{disc}} \tau$ iff $\text{dom}(\sigma) \subseteq \text{dom}(\tau)$,
$\sigma \sqsubseteq_{\text{i}} \tau$ iff for all $V \in \text{i}(\tau)$, there is a $U \in \text{i}(\sigma)$ such that $U \leq V$. □

From all three definitions it is clear that $\sigma \sqsubseteq \tau$ iff $\sigma \sqsubseteq_{\text{d}} \tau$ and $\sigma \sqsubseteq_{\text{i}} \tau$. Discourse information $\sqsubseteq_{\text{disc}}$ is subsumed under $\sqsubseteq_{\text{d}}$. The information order preserves the amount of data and the amount of structure and conversely, the data and structure orders taken together precisely define the information order. Moreover, the privileged information states **0** and **1** are at the extremes of the ordering. The initial information state is the minimal information state, so $\mathbf{0} \sqsubseteq \sigma$ for all $\sigma \in \Sigma$. The absurd information state is in some sense the maximal information state. Because it is inconsistent, all propositions are derivable from it: $\sigma \sqsubseteq \mathbf{1}$, for all $\sigma \in \Sigma$.

For formulas that do not cause undefinedness, the update function preserves the information order. This holds both for the classic system of definitions 15 and 18 and for the dynamic version of definition 23. Information increases because possibilities, or pairs of possibilities, are eliminated from the information state. For the classic systems, this property is called *strengthening* (Veltman 1996).

Proposition 7 (Information Increase)
For all $\sigma \in \Sigma_W, \Sigma'_W$ and Σ_I and $\varphi \in L$ such that $\sigma[\varphi]$ is defined, it holds that: $\sigma \sqsubseteq \sigma[\varphi]$.
Proof sketch Check the assertives and the interrogatives for (i) the classic system, (ii) the tuple style definition, and (iii) the dynamic version. Sequential composition only combines the results.
(i) The form of definition 15 ensures that the resulting equivalence relation $\sigma[\varphi]$ is a subset or equal to σ.
(ii) The use of $\cap$ in definition 23 ensures that the resulting data set and equivalence relation are subset or equal to the original date set and equivalence relation.
(iii) We have to show that each pair of indistinguishable indices either remains unchanged, is extended or eliminated. The use of $\upharpoonright$ ensures elimination or no change for atomic formulas, negation and identity. For existential quantification the random assignment ensures that all indices are extended. And for questions the definition ensures that indices either remain or are eliminated. □

Once a notion of information growth is defined, it becomes possible to state the following key idea, based on Stalnaker (1979) among others. An information state σ is said to contain φ, accept φ or *support* φ when updating σ with φ will not increase the information in σ. Adding formulas that are already supported would be redundant. Please note that this redundancy constraint heavily depends on the 'standard' rationality assumptions of epistemic logic. See Walker (1996a) for a psychologically more plausible account of the information of agents in dialogue.

Definition 29 (Support)
For all $\sigma \in \Sigma$ and $\varphi \in L$, σ supports φ, $\sigma \Vdash \varphi$, iff $\sigma[\varphi] \sqsubseteq \sigma$. □

By proposition 7 we have that $\sigma \sqsubseteq \sigma[\varphi]$. Because $\sqsubseteq$ is anti-symmetric for the three kinds of systems, definition 29 becomes much stronger: $\sigma \Vdash \varphi$ iff $\sigma[\varphi] = \sigma$. This type of definition of the notion of support is called a fixed point definition.

The notion of support works for assertives and interrogatives alike. On the basis of it, we can define a non-standard notion of entailment. A sequence of utterances conforms to a valid argument, when, after having applied all premisses in the right order to an arbitrary information state, the conclusion is supported. For inquiry the activity is no longer an argument, but consists of cooperative information exchange. Premisses may be utterances from different participants. So for inquiry the conclusion must be accepted after the premisses have been exchanged in the right order. Note that both interrogatives and assertives may act as both premisses and conclusion.

Definition 30 (Entailment)
For all $\varphi_1; \ldots; \varphi_n, \psi \in L$ define $\varphi_1; \ldots; \varphi_n \models \psi$ iff $\sigma[\varphi_1] \ldots [\varphi_n] \Vdash \psi$ for all $\sigma \in \Sigma$. □

What does it mean that issues and assertives may act both as premisses and as conclusions? When does an issue entail another issue? When there is a logical dependency between the two issues. As we said in the introduction, $?\varphi \models ?\psi$ means that every solution to $?\varphi$ will also have solved $?\psi$. When does an issue entail an assertion? In principle never. However, interrogative utterances may add information by means of presupposition accommodation. Recall the mentioning of a traffic light in example (28). We encode presuppositions separately in the formula that represents the utterance content.

When does an assertion entail an issue? Pragmatically, this can occur often as a result of the maxim of quantity. We found that 'underdetermined' expressions like disjunctions 'trigger' an issue as to the truth of its sub-formulas. This effect can not be shown on the basis of the logic. It must be implied by pragmatic principles of quantity; an underdetermined expression indicates that more could have been said. Only for existential expressions the question for a 'witness' can be inferred in the formal system. $\exists x.\varphi \models ?y.(y = x)$ The proof is by definition. This is a result of our choice for specific assignment.

A second notion of entailment, $\models^{\mathbf{0}}$, is restricted to the initial information state **0**. This assumes that all the relevant information in the information state, has been exchanged as part of the inquiry process. In practice that means a kind of 'closed world assumption'. The premisses are all the information. A combination of presuppositions and restricted entailment may result in non-monotonic inference (section 3.7).

Definition 31 (Restricted Entailment)
For all $\varphi_1; \ldots; \varphi_n, \psi \in L$ define $\varphi_1; \ldots; \varphi_n \models^{\mathbf{0}} \psi$ iff $\mathbf{0}[\varphi_1] \ldots [\varphi_n] \Vdash \psi$. □

3.5.1 Complete and partial resolution

In the introduction we said that inquiry can be modelled as a constant process of raising and resolving issues. The raising of issues is modelled by interrogative updates. Resolving issues may happen when assertive updates completely, or partially, reduce the partition induced by the contextual issues. When the data set is included in one of the blocks in the partition, the issue is completely resolved. When the data set excludes some of the blocks in the partition, the issue is partially resolved. This is depicted in figure 3.6. On the left, we see assertive information ψ, completely resolving an issue $?\varphi$. In other words, ψ is a complete answer to $?\varphi$. On the right we see information ψ partially resolving an issue $?\varphi$. In other words: ψ is relevant to $?\varphi$.

So, an information state σ resolves an issue $?\varphi$, written $\sigma \Vvdash ?\varphi$, when its data set is included in one of the blocks of the partition induced by the issue. This definition makes sense both when issues are restricted to the data set, as they are in the main definitions of this chapter, and when issues range over the complete information space. More generally, information ψ is said to resolve $?\varphi$ in context σ when σ updated with ψ resolves $?\varphi$ in the earlier sense, but the old σ doesn't.

Definition 32 (Resolution)
Information state σ resolves $?\varphi$, $\sigma \Vvdash ?\varphi$, when $d(\sigma) \subseteq V$, for some $V \in i(\sigma)$.
Information ψ resolves $?\varphi$ in σ, when $\sigma[\psi]$ resolves $?\varphi$, but σ does not. □

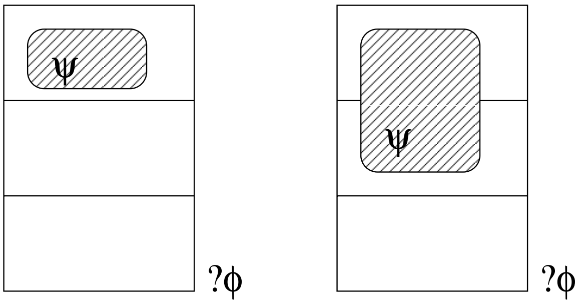


Figure 3.6: Complete and partial resolution of issue $?\varphi$ by information ψ.

Instead of on the partition, the definition can also be based on the equivalence relation: $\sigma \Vvdash ?\varphi$ iff $\mathrm{u}(\mathrm{d}(\sigma)) \Vdash ?\varphi$. The condition $\sigma \Vdash ?\varphi$ by itself is not strong enough for resolution; it simply means that $?\varphi$ is one of the current issues in σ.

Information ψ that always resolves an issue $?\varphi$, irrespective of the information state, constitutes a *direct answer* or *direct solution*. By contrast, an indirect answer depends on the particular information state, for instance because it requires background knowledge

Definition 33 (Direct Answer)
Information ψ is a direct answer to $?\varphi$, when $\sigma[\psi]$ resolves $?\varphi$, for each $\sigma \in \Sigma$. □

An answer does not always have to be complete. In many circumstances a partial answer suffices. And even if a complete answer is required, a partial answer is seen as a good first try. Partially resolving information is always a relevant contribution to the current issues. For yes/no questions, partial and complete resolution coincide. Recall that for the classic systems, $U \leq V$ is read as $U \supseteq V$.

Definition 34 (Partial Resolution)
Information state σ partially resolves $?\varphi$, iff for some $U \in \mathrm{i}(\mathbf{0}[?\varphi])$ there are no $V \in \mathrm{i}(\sigma[?\varphi])$ such that $U \leq V$.
Information ψ partially resolves $?\varphi$ in σ, iff $\sigma[\psi]$ partially resolves $?\varphi$, but σ does not. □

By eliminating at least one alternative block from the partition, partial resolution brings complete resolution closer. A relevant utterance should provide information that matters for the current issues. Partial resolution captures precisely what it means for information to 'matter': it must reduce the partition.

Definition 35 (Relevant Information Increase $\sqsubset$)
For each $\sigma, \tau \in \Sigma$, τ contains more relevant information than σ, $\sigma \sqsubset \tau$, iff for some $U \in \mathrm{i}(\sigma)$ there is no $V \in \mathrm{i}(\tau)$ such that $U \leq V$. □

The reason that this definition makes sense, is that for each block in the partition that is eliminated by partial resolution, some weaker issue that was implicit in the information state is completely resolved. And earlier we stipulated that an utterance is relevant precisely when it answers or resolves at least some issue in the context. We can show that in case of a partial resolution of $?\varphi$ by information χ, there must be some weaker issue $?\psi$, which is completely resolved by information χ. For example, consider a wh-question which leads to a number of alternative blocks in the partition. Answering each of the questions of the form "Is it this block?" or "Is it that block?" brings full resolution closer.

Proposition 8 (Relevance means Resolution)
Whenever $\sigma \sqsubset \tau$, there is an issue $?\psi$ such that $\sigma \Vdash ?\psi$, which is resolved by τ.
Proof Suppose $\sigma \sqsubset \tau$. Then there is a $V \in \text{i}(\sigma)$ such that there is no $U \in \text{i}(\tau)$ with $V \leq U$ (*). This V corresponds to a proposition. Lets call the formula that represents this proposition ψ_V, for instance by taking the disjunction of the conjunction of atomic facts that hold at a world, for each world in V. Now $\sigma[?\psi_V] = \sigma$, because $V \in \text{i}(\sigma)$. And because of (*) issue $?\psi_V$ must be resolved in τ. □

Relevant information resolves at least one issue. We still have to explain how issues are raised. What about the relevance of issues or questions themselves? Obviously that can not be explained in terms of issues.

Asking a question seems relevant in case it addresses the general topic of the conversation. If we suppose that each topic is related to a question in the sense of van Kuppevelt (1995), and there exists a hierarchical dependency relation between questions and subordinate questions, that would suggest that asking those questions on which the general topic related question depends, is relevant. However, there are a problems with this intuitive characterisation in our framework. The only dependency relation that we can consider on the basis of logic alone, is the entailment relation between questions. And as we saw in proposition 8, each of the subordinate questions is automatically supported by an information state once the general question is supported. That means that asking such a subordinate question is strictly speaking not informative! However, other dependency relations might result from a more detailed look at a particular task domain and at the relation between issues, topics and tasks. This will be the topic of section 3.8.

By the nature of the framework we have assumed that agents are perfect reasoners. So the fact that a general question automatically raises the issue of all their subordinate questions, is simply an instance of the familiar problem of deduction (Stalnaker 1984, p25). But as soon as we allow agents to forget things or to be imperfect reasoners, explicitly asking a question makes sense again, even if its relevance could have been inferred all along. Asking a question makes participants aware of the possible answers.

3.6 Properties

This section is based on Groenendijk (1999). We define a number of very general properties for sequences of updates. Since updates model the effect of dialogue acts at the content level, such properties may be part of a more general specification of the coherence of a sequence of speech acts. The properties roughly follow the Gricean maxims of *quality*, *quantity*, and *relation*. The maxim of *manner* must be dealt with at the form level of utterances.

Quality has to do with the reliability of information. From the speaker's point of view, the maxim of quality means saying what you believe to be true, and saying something only if you have adequate evidence for it. From the hearer's point of view, these aspects are strictly speaking inaccessible. Therefore the aspects of *sincerity* and *evidence* have to be dropped as dialogue constraints at the content level. Of course, they do remain valid at a meta level and can be used as principles on the basis of which hearers can infer what the speaker probably believes (Lascarides and Asher 1999). Whether sincerity can be expected depends on the social relationship between speaker and hearer. Moreover, evidence can be asked for in the dialogue.

Violation of these principles can only be brought out by inconsistencies. A reason for not accepting new information is therefore that it is either internally inconsistent, or incompatible with what went on before. Inconsistencies signal that some information is unreliable. Therefore the property that best approximates the Gricean maxim from the hearer's point of view, is consistency. Note that pure interrogatives can not be inconsistent.

Definition 36 (Consistency)
Information ψ is consistent after $\varphi_1; \ldots; \varphi_n$ iff $(\sigma[\varphi_1]\ldots[\varphi_n])[\psi] \neq \mathbf{1}$, for all $\sigma \in \Sigma$. □

The maxim of quantity has to do with the amount of information. Now traditionally, the amount of information in an information state is correlated with the number of things that can be derived from it. According to Stalnaker (1979), assertions must meet two constraints: they must not be inconsistent or incompatible with the existing context, and they must be *informative*, i.e. their content must not already be entailed by the context. So strictly speaking, informativeness is the opposite of entailment. For interrogatives, informativeness is a little more subtle. What does it mean if an interrogative is entailed by some sequence of updates? By the non-classical entailment of definition 30, it means that after the updates, the issue expressed by the interrogative is already supported. Raising it again would be superfluous. Non-superfluous interrogatives are called *inquisitive* by Groenendijk. He reserves the term 'informative' for non-entailment of assertives only. We choose the same definition of informativeness for both assertives and interrogatives.

Definition 37 (Informativeness)
Information ψ is informative after $\varphi_1; \ldots; \varphi_n$ iff $\varphi_1; \ldots; \varphi_n \not\models \psi$. □

Entailment gives us a way of defining the relative strength of information. It defines informativeness from a purely external logical point of view. In dialogue, what matters is how informative some utterance is with respect to the current contextual issues. Such a technical notion of relevance was defined in definition 35.

Definition 38 (Relevance)
Assertive ψ is relevant after $\varphi_1; \ldots; \varphi_n$ iff $\sigma[\varphi_1]\ldots[\varphi_n] \sqsubset (\sigma[\varphi_1]\ldots[\varphi_n])[\psi]$, for any $\sigma \in \Sigma$. □

Now, according to the above notion of relevance, even an inconsistency is relevant: it reduces the partition. There is nothing in the definition to stop the speaker being over-informative. Groenendijk defines a constraint that does just that: licensing. An utterance is licensed when it does not provide more information than is required for current purposes. This corresponds to the second clause of Grice's maxim of quantity: "Do not make your contribution more informative than is required." (Grice 1975, p 45). How much information exactly is required for current purposes, is specified in the contextual issues. The idea is that updates should proceed 'block wise'. If one world is removed from the data set, all the worlds that are indistinguishable from it with respect to the current issues, should also be removed.

Definition 39 (Licensing)
Information ψ is licensed after $\varphi_1; \ldots; \varphi_n$ iff for any $\sigma \in \Sigma$, $i, j \in \mathrm{d}(\sigma)$:
if $\langle i, j\rangle \in \sigma[\varphi_1]\ldots[\varphi_n]$ and $i \not\preccurlyeq \sigma[\varphi_1]\ldots[\varphi_n][\psi]$, then $j \not\preccurlyeq \sigma[\varphi_1]\ldots[\varphi_n][\psi]$. □

Suppose some worlds $U \subseteq V$ remain left in some block V in the partition, after the update. In that case, the assertion did not *only* address the current issues; some other distinction was made between U and $V \setminus U$, which resulted in the worlds of U remaining. Making such additional distinctions is not licensed. If we require that all utterances be licensed, i.e that updates proceed 'block wise', that means that the blocks have taken over the role of epistemic possibilities that can be distinguished by an agent. A clever representation of an information state would only store the relevant facts of a block, i.e. the facts on which all worlds in the block agree, and in which the block differs from at least one other block. Such a representation could take the form of a disjunctive normal form. Because interrogatives do not remove any worlds from the data set, they are trivially licensed.

The notions of relevance and licensing assume that issues have already been raised. Issues can be brought up in various ways: by explicit questions, by intonation patterns or otherwise. But we have said nothing about the relevance of issues themselves. Similar to constraints like relevance and licensing, we could add similar coherence constraints for the content of interrogatives. For example, one could require that issues be related to each other by some dependency relation, or by some shared topic or underlying task. Some proposals along these lines will be given in section 3.8. In particular, relevance or salience of interrogatives can be assessed relative to the underlying goal of an agent (Hulstijn 1998). Roughly, interrogative $?\varphi$ is relevant for an agent a iff a has the goal to achieve ψ and knowing an answer to $?\varphi$ is a precondition for actions that would achieve ψ. This corresponds to the *Q-Elab* coherence relation of Lascarides and Asher (1999). Moreover, in case we assume that agents are not perfect reasoners, or reasoners that only implicitly know how issues are related, asking a question has an important additional function: it makes the hearer aware of some issue, and thereby of the information related to that issue. In chapter 4 we develop an account of awareness based on issues.

The notions of consistency, informativeness, relevance and licensing can be combined into a formalisation of the Gricean maxims. A number of adjustments need to be made. First, one needs to restrict quality to consistency. From the point of view of a credulous hearer, the speaker is assumed to be sincere and reliable, until she contradicts herself. Second, the maxims of relation and quantity are combined in the properties of informativeness, relevance and licensing. The result of this combination of properties specifies when an utterance is *pertinent* after a sequence of previous utterances (Groenendijk 1999). Pertinence defines what it means for a sequence of utterances to be coherent at the content level.

Definition 40 (Pertinent)
Information ψ is pertinent after $\varphi_1; \ldots; \varphi_n$ iff
(i) ψ is consistent after $\varphi_1; \ldots; \varphi_n$, and
(ii) ψ is informative after $\varphi_1; \ldots; \varphi_n$, and
(iii) ψ is relevant after $\varphi_1; \ldots; \varphi_n$, and
(iv) ψ is licensed after $\varphi_1; \ldots; \varphi_n$. □

This technical notion of pertinence concludes the most important part of this chapter on inquiry. We gave a formal account of the dialogue property of relevance. This notion, combined with consistency, informativeness and licensing, can be used in the formalisation of usability. What we haven't done, however, is explain how issues are actually raised, and how thy relate to awareness. This will be dealt with in section 3.8 and chapter 4. Also the notion of presupposition has been left unexplained to a large extend. That is what we will continue with now.

3.7 Presupposition

This section explores the role of presuppositions in an update semantics. We will present several possible strategies for dealing with the interpretation of presuppositions. The work in this section was presented earlier in Hulstijn (1995, 1996). The account is mostly based on Zeevat (1992) and Beaver (1996). They too use an update semantics framework. For a clear exposition of approaches to presupposition in a DRT framework see Krahmer (1995) and Geurts (1994).

3.7.1 Presupposition Triggers

An utterance is not a random collection of words. Utterances are put together following syntactic rules. But not only syntax puts constraints on the use of words. Utterance (61) is wrong or infelicitous because the verb *to eat* expects the subject to be animate.

(61) *The stone ate the cake.

Presuppositions are just such constraints on the use of a certain linguistic construction or choice of words. Constructions carrying such a constraint are called *presupposition triggers*. The presupposed information is part of the meaning of an utterance, just like asserted information. We assume a lexicon that lists both the asserted and presupposed information for all expressions.

There are several types of presupposition triggers. The verb 'to eat' is an example of a lexical presupposition trigger: the presupposition arises from the meaning of the word. Similar to lexical presuppositions are the so called factive verbs, like know and regret that presuppose their complements to be true. On the other side of the scale we have the existential presupposition triggers, such as definite descriptions, proper names, quantifiers and focus constructions like topicalisation and wh-clefts. These referring expressions presuppose the existence of their referents. Another group of triggers is formed by words like 'but' or 'even' that presuppose a certain sentence-structure. For example, 'but' presupposes a contrast. For a good overview of triggers see Gazdar (1979).

(62) John regrets that he killed his wife.
presupposes: John killed his wife.
asserts: John feels remorse.

(63) Most of Jack's children are happy.
presupposes: Jack has children.
asserts: Most of the children of Jack are happy.

(64) Even Fred likes bananas.
presupposes: Fred is the least likely person to like something.
asserts: Fred likes bananas.

The evidence from the semantics of focus that we discussed in the introduction, suggests that also issues can be presupposed. Apart from the general maxim of relevance that urges the hearer to accommodate an issue in case the current utterance does not seem relevant, there are also special triggers. Most importantly, focus expressions and topic shifts. Example (30) is repeated as example (65) and (66) here.

(65) The door is *green*.
presupposes: What is the colour of the door?
asserts: It is green.

(66) The *door* is green.
presupposes: What object is green?
asserts: It is the door.

Such presupposed issues are crucial for dealing with corrections (van Leusen 1997). Witness an example like (67). The intonation, both on the antecedent and on the correction itself, indicate the scope of the correction. If dad speaks with normal intonation each of Johnny's replies can be felicitous. By the choice of his reply, Johnny can manipulate the topic of the conversation. By stressing certain aspects dad has a little more control. If he stresses 'blue' only *a* would be a felicitous response. In other words, when the focus is put on 'blue', this triggers the presupposition that the current issue concerns the colour of the sweater that Johnny wants. Now of course Johnny can override this by explicitly denying that presupposition. But then we would expect some conversational 'friction'.

(67) Dad: Do you want a blue sweater?
Johnny: a. No, a *red* one!
b. No, a blue *cap*!
c. No, I *don't want* anything!

3.7.2 Language

Proper names, definites, factive verbs, some lexical expressions and other quantification expressions like wh-questions are taken to be presupposition triggers. Use of such a trigger indicates that the presupposition that is conventionally associated with the trigger is supposed to be part of the context already. Each assertive or interrogative may contain presupposition triggers. We assume that those triggers are explicitly represented in the object language, indicated by the symbol ∂. For example, a definite like 'the president' would be modelled by the formula $\partial x.president(x)$, where x is a referent. Definite descriptions like 'the president' indicate that a unique individual of that description exists, in the given context. Nevertheless, presupposition of uniqueness is neglected throughout this section. We believe that the accommodation behaviour of a presupposition of the uniqueness of a referent, is tied to the presupposition of the existence of a referent in context.

The utterance "The president is insane" is represented as $\partial x(president(x)) \wedge insane(x)$. Of the presupposed part, the president, it is asserted that he is insane. This does not mean the same thing as $\partial(\exists x.president(x) \wedge insane(x))$, which presupposes the fact that there is an insane president. There is an intimate relation between presupposing and anaphoric binding. This will be explored below.

Incidently, the choice to model presuppositions in the object language by a symbol ∂ is not the only sensible one. One could argue that presuppositions constitute a different kind of update. That would suggest an update semantics with different update functions $[.]_+$ for the normal assertive expansions, $[.]_?$ for interrogatives, $[.]_\partial$ for presupposition updates and possibly even $[.]_*$ for revisions. However, such an approach would run into trouble on the interaction of presuppositions and interrogatives.

We use the following logical language to replace L_0 and L. Note that presuppositions may occur at any level. For instance, inside the scope of a negation, or in the antecedent of an implication. Also questions or issues may be presupposed. Because we do not want questions to be embedded, we need the following cumbersome re-definition.

Definition 41 (Syntax $L_{\partial 0}, L_\partial$)
Given sets of symbols $Pred = \{P, \ldots\}$, $Cons = \{c, \ldots\}$, $Var = \{x, \ldots\}$, define

$$\begin{array}{lll} T: & t & ::= x \mid c \\ L_{\partial 0}: & \varphi & ::= P(\vec{t}) \mid (t_1 = t_2) \mid \neg\varphi \mid (\varphi \wedge \psi) \mid (\varphi \vee \psi) \mid (\varphi \rightarrow \psi) \mid \exists x.\varphi \mid \forall x.\varphi \mid \partial\vec{x}.\varphi \\ L_\partial: & \chi & ::= \varphi \mid ?\vec{x}.\varphi \mid \partial\vec{x}.\chi \mid (\chi;\zeta) \qquad (\varphi \in L_{\partial 0}, \chi, \zeta \in L_\partial) \end{array}$$

□

3.7.3 Accommodation Strategies

The central notion is *presupposition accommodation*: the way in which the hearer's version of the context is changed, when an utterance with a presupposition is added to the context (Lewis 1979; Heim 1983). When seen as a form of inference, presupposition accommodation exhibits strange logical properties. We attempt to characterise these properties in a logical framework, namely update semantics. We use so called accommodation strategies to model several well-known approaches to presupposition from the literature. In this way different proposals can be compared and evaluated in a single framework.

We will study four well-known empirical observations of the behaviour of presupposition to guide the definitions of the accommodation strategies. In this section we will be interested in the propositional part of the language only. Existential presuppositions will have to wait until the section 3.7.6.

Presupposition as Precondition

Presuppositions can be seen as a constraint on their immediate context. A presupposition works very much like a precondition in mathematics or computer science. If the precondition fails, the meaning of the mathematical expression or the result of the program becomes undefined.

(68) a. $\sqrt{a}$ presupposes: $0 \leq a$

(69) a. John's dog is happy. presupposes: John has a dog.
b. John is allergic to dogs. *John's dog is happy.

This notion of 'presupposition as precondition' is very similar to the *satisfaction* strategy: a sentence S presupposes proposition φ in a context σ iff S can only be uttered felicitously provided σ supports φ. Sources of ideas related to what we call the satisfaction strategy are Karttunen (1974), Lewis (1979), Stalnaker (1979), Heim (1983) and Beaver (1996). The presupposition must be true in the context, otherwise the utterance is infelicitous, as in example (69b). In other words, unless the presupposition is satisfied, the conversational effect of the utterance is undefined. Strictly speaking the classic strategy is the strategy of *no* accommodation of the context.

Strategy 1 (Satisfaction)

$$\sigma[\partial\varphi] = \begin{cases} \sigma, & \text{if } \sigma \Vdash \varphi, \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

Since the update function models the conversational effect of an utterance, this means the update function has become a partial function: its value is not defined for all φ. We take it that function composition and the restriction operator of definition 23, and therefore the logical operators $\wedge$, $\neg$ and are *strict* with respect to undefinedness[3]. In other words, a conjunction is undefined if either its first conjunct is undefined, or the second in the context of the first. This behaviour also holds for the other connectives. It is a result of their dynamic nature, recall proposition 5 and 6. Random assignment by itself never leads to undefinedness. Note that under this view the presupposition of a question, the reason it may become undefined, is that there exists some referents to answer it.

(70) For all $\sigma \in \Sigma$ and $\varphi \in L$

$\sigma[\varphi \wedge \psi]$	undefined,	if $\sigma[\varphi]$ undefined or $\sigma[\varphi][\psi]$ undefined,
$\sigma[\neg\varphi]$	undefined,	if $\sigma[\varphi]$ undefined,
$\sigma[\varphi \rightarrow \psi]$	undefined,	if $\sigma[\varphi]$ undefined or $\sigma[\varphi][\psi]$ undefined,
$\sigma[\exists x.\varphi]$	undefined,	if $\sigma[\varphi]$ undefined and
$\sigma[?\vec{x}.\varphi]$	undefined,	if $\sigma[\vec{x}][\varphi]$ undefined.

Given update semantics, the classic projection problem of Karttunen (1973) is therefore partly solved. The undefinedness or the related presupposition inference is usually passed on. Below we will show that in some cases the inference can be blocked in case the presupposition gets bound in the local context.

Accommodation Proper

Preconditions are only part of the story. As we have seen, communication only succeeds on the assumption that all participants are cooperative. Uttering a sentence of which the presupposition is known to be false is uncooperative; therefore the hearer will infer from a presupposition trigger that the speaker takes the presupposition to be true. The hearer will adjust his or her version of the context, by adding the presupposition. This accommodation is only allowed, when the presupposition is compatible with the context. Otherwise the utterance remains infelicitous and its effect undefined.

The following strategy is called the *cautious accommodation* strategy: only add the presupposition when it is compatible.

Strategy 2 (Cautious)

$$\sigma[\partial\varphi] = \begin{cases} \sigma[\varphi], & \text{if } \sigma[\varphi] \neq \mathbf{1}, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Presupposition Test

This adjustment of the context can be seen as a form of inference. Information is inferred from the trigger. It is, however, is a very strange kind of inference. Contrast the presupposition inference in example (71) with normal inference in example (72).

(71)	a.	John's dog is happy.	presupposes: John has a dog.
	b.	It is not the case that John's dog is happy.	presupposes: John has a dog.
	c.	Maybe John's dog is happy.	presupposes: John has a dog.

[3] An operation $\bullet$ on information states σ, τ is *strict* if $(\sigma \bullet \tau)$ is undefined iff σ undefined or τ undefined.

(72) a.	Bella is a cow.	implies: Bella mows.
b.	It is not the case that Bella is a cow.	implies: ...?
c.	Maybe Bella is a cow.	implies: ...? Possibly Bella mows.

In section 3.4 we showed that referents introduced in the scope of a blocking operator like negation, can not be picked up for anaphoric reference later. However, presuppositions can escape from inside such operators. This is one of their defining characteristics. The infelicity prediction or presupposition inference remains available, even if the trigger is embedded under negation or modal operators, as in (71b) and (71c).

This marks a contrast, also with normal entailment. Negating the antecedent of an implication, as in (72b), the conclusion is lost. When the antecedent is weakened by *maybe*, as in (72c), the conclusion is weakened too. None of this happens with presupposition. Therefore this observation can be used as a *presupposition test*. Any inference which survives when its trigger is embedded under negation or a modal operator, may be called a presupposition. Note that the classic strategy already preserves the infelicity prediction under negation.

Cancelling or Anaphoric Binding

As we said, presupposition inference is a peculiar kind of inference. On the one hand, it survives from under a negation; on the other hand, it turns out to be defeasible. The infelicity prediction or presupposition inference will, in some particular embedding contexts, be *cancelled* or *bound*. This may happen for instance in indirect speech, when the antecedent of a conditional or the first of two conjuncts implies the presupposition, in belief contexts, or in some special negated sentences. Traditionally such cancelling contexts are called *plugs*, as opposed to the *holes* that let presuppositions seep through (Karttunen 1973). But unlike Karttunen we are able to deal with the context-dependency of most plugs. As shown by Gazdar (1979) there are no strict plugs. All contexts are holes, which can be turned into plugs if they provide an antecedent.

Consider example (73). The context suggested by John's allergy is incompatible with the presupposition. So as expected, (73a) is infelicitous. But in (73b) it is not. The infelicity prediction and presupposition inference is said to be cancelled or bound. The difference is that in (73b) the antecedent already implies the presupposition. What seems to be the case here, is that interpretation shifts to a temporary context corresponding to the antecedent. Because this context is non-actual, the sentence is marked as a counterfactual by 'had' and 'would'. The temporary context supports the presupposition. There is no conflict with the over-all context. Conjunction works in a similar way, as in example (73c). This effect of binding to the first conjunct or antecedent, is already covered by the classic satisfaction strategy or the cautious strategy, given the update semantics framework. Belief contexts, as in example (73d), also create a temporary context. Since Mary doesn't know of John's allergy, the embedded sentence remains felicitous within her beliefs.

(73) John is allergic to dogs.
a. *If John buys a bone, John's dog is happy.
b. If John had a dog, John's dog would be happy.
c. (But) John has a dog and John's dog is happy.
d. Mary believes that John's dog is happy.

This process of binding or cancelling works for presuppositions and other coreferring and anaphoric expressions alike. In fact, presupposition cancelling and anaphora resolution are the very same process (van der Sandt 1989; van der Sandt 1992; Geurts 1994). In case this process fails, the hearer will try to accommodate a referent. The only condition is that anaphoric reference resulting from a single pronoun, such as 'he' or 'it', is not strong enough to get accommodated. In other words, a pronoun generally does not contain enough semantic content to be able to introduce a new referent. Our earlier remarks on cross speaker cataphora challenge this constraint. In the question game of example (59) or in the opening of a book (58), a temporary referent can be accommodated for the 'sake of the argument' on the basis of a pronoun. However, cataphora only appear in particular genres and contexts. It seems they work only on the assumption that the initiator of the game or the author of the book have a definite individual in mind.

Example (74) is rather tricky. Here the presupposition is explicitly cancelled. That this is possible shows that apparently negation also creates a temporary context for interpretation. The infelicity prediction or presupposition inference at the global context level can therefore be cancelled or bound to the temporary context. Note the influence of the cue phrase 'so'. Without it the example would remain infelicitous.

(74) John does not have a dog. So, it is not the case that John's dog is happy.
presupposes: –

We do realize this example sounds odd. The oddity can be explained as follows. Although the presupposition is cancelled, the utterance does not add any new information. It violates the informativeness constraint. Therefore the utterance does not express a proper assertion and will be judged infelicitous on those grounds (Stalnaker 1979).

Global, Local, Intermediate

We suggested that sometimes interpretation shifts to a temporarily context. This approach was suggested by Heim (1983) and developed by van der Sandt (1989) and Zeevat (1992). How does a temporary context arise? Take another look at the clause for negation in definition 23. First we calculate $\sigma[\varphi]$. Then we subtract all the indices that subsist in $\sigma[\varphi]$ from σ. Suppose φ contains a trigger. Now there are two versions of the context σ that can be adjusted to accommodate the presupposition! The *local context*, inside the scope of the negation, or the *global context*, outside its scope. Here is an analysis of example (74).

(74) Assume $\sigma \Vdash \neg\exists x(dog\_of(john, x))$,

a. $\sigma[\neg(\partial x(dog\_of(john, x)) \wedge happy(x))] =$ (global)
$\sigma[\partial x(dog\_of(john, x))] \upharpoonright \{i \mid i \not\preccurlyeq \sigma[\partial x(dog\_of(john, x))][happy(x)]\} =$
undefined.

b. $\sigma[\neg(\partial x(dog\_of(john, x)) \wedge happy(x))] =$ (local)
$\sigma \upharpoonright \{i \mid i \not\preccurlyeq \sigma[\partial x(dog\_of(john, x)][happy(x)]\} =$
σ.

This explains the possibility of cancelling in case of a negation. In most cases accommodation to the global context is preferred. But when global accommodation is not allowed and there is a local context it is possible to accommodate only locally. In that case the information state remains as it was. When there is no temporary local context to accommodate into, the effect of the utterance becomes undefined.

Presupposition inference, then, is the effect of presupposition accommodation at the original global context level. When assuming that all relevant information with respect to the presupposition is given, (i.e. using $\models^{\mathbf{0}}$) this form of inference turns out to be non-monotonic. Compare examples (71a) and (74) again, but now with a simplified notation: $\exists x(dog\_of(john, x)) = dog$, $\partial x(dog\_of(john, x)) = \partial dog$ and $happy(x) = happy$.

(75) $\neg(\partial dog \wedge happy) \models^{\mathbf{0}} dog$ but $\neg dog \wedge \neg(\partial dog \wedge happy) \not\models^{\mathbf{0}} dog$

A simple consistency check (cautious accommodation) together with the update semantics notion of entailment already produces the non-monotonic behaviour that is characteristic of presupposition. Therefore, we believe, there is no need for a specific calculus of presuppositions based on default logic, as suggested for instance by Mercer (1992).

If we define implication in terms of negation and conjunction by the standard equivalence, this leads to potentially three versions of the context to accommodate into: the *local*, the *intermediate* and the *global* context. Recall proposition 5. Global or intermediate accommodation implies automatic accommodation at the lower levels too. In general, accommodation to the global context is preferred (van der Sandt 1989). If that is impossible, accommodation to the intermediate context is generally preferred to accommodation in the local context only. This principle has been under attack. See the discussion around example (78) below. Therefore, Van der Sandt stresses in later work that these preferences are subject to general constraints on coherence and understandability of the discourse.

Now it may be that one of the temporary contexts already supports the presupposition. In that case, there is no need for further accommodation: the presupposition is said to be *bound*.

(73) a. $bone \rightarrow (\partial dog \wedge happy) \models^{\mathbf{0}} dog$ because for all σ

$$\begin{aligned}
&\sigma[bone \rightarrow (\partial dog \wedge happy)] = \\
&\sigma[\neg(bone \wedge \neg(\partial dog \wedge happy)] = \\
&\underbrace{\sigma}_{\text{global}} \upharpoonright \{i \mid i \not\leq (\underbrace{\sigma[bone]}_{\text{intermediate}} \upharpoonright \{j \mid j \not\leq \underbrace{\sigma[bone][\partial dog][happy]}_{\text{local}}\}\} = \\
&\sigma[dog][bone \rightarrow happy] && \text{global or} \\
&\sigma[(dog \wedge bone) \rightarrow happy] && \text{intermediate or} \\
&\sigma[bone \rightarrow (dog \wedge happy)] && \text{local, and global accommodation is preferred.}
\end{aligned}$$

b. $dog \rightarrow (\partial dog \wedge happy) \not\models^{\mathbf{0}} dog$ because for all σ

$$\begin{aligned}
&\sigma[dog \rightarrow (\partial dog \wedge happy)] = \\
&\sigma \upharpoonright \{i \mid \ \not\leq \sigma[dog] \upharpoonright \{j \mid j \not\leq \sigma[dog][\partial dog][happy]\}\} = \\
&\sigma[dog \rightarrow happy] && \text{bound.}
\end{aligned}$$

We described several approaches to presupposition accommodation in terms of accommodation strategies. Its seems that the cautious strategy already covers most of the data, provided some flexibility in the meta reasoning. We will now discuss various possible extensions.

3.7.4 Stacks and Representations

As a by-product of the definition of implication in terms of negation and conjunction, we get three temporary versions of the context in which the presupposition can be accommodated: locally, in the antecedent and globally. Propositional attitudes and questions also produce subordinate contexts for accommodating into. This coincides with predictions from the DRT theory of presuppositions developed by Van der Sandt (1989, 1992) and Geurts (1994) and observations of Heim (1992) and Zeevat (1992) about the behaviour of presuppositions in the context of propositional attitudes. Moreover, it corresponds with our view on the 'blocking' of anaphoric referents, expressed in section 3.4.

One could imagine that this leads to a *stack* of possible contexts for interpretation (Zeevat 1992). Example (76) would generate a stack of temporary contexts corresponding to John's belief, Mary's utterance, and John's regret.

(76) John believes that Mary said that he regrets being bald.

In fact, discourse or dialogue structure looks more like a *tree*. Contexts of interpretation are organised in some partial order that reflects the development of the conversation. Such trees can be used to indicate the rhetorical, narrative or discourse relations between segments, much like grammatical functions are indicated along the branches of a parse tree. Dialogue or discourse representation trees can be used to indicate narrative structure and temporal relations in text (Asher 1993), but can also be used in dialogue to indicate coherence relations between utterances (Asher and Lascarides 1998a). Similar techniques, are used to account for ellipsis and coordination phenomena (Polanyi 1988; Prüst et al. 1994).

However, looking up from the current context towards the root of the tree, the structure equals a stack of subsuming contexts. Compare the embedding DRSs that are accessible from within a sub-DRS, which also forms a stack structure.

Lets try to investigate this idea a bit further. A stack S is defined as an empty stack (), or as a pair of an information state and a stack (σ, S), which is the result of a *push*. In this pair, σ is the local context. Interpretation always starts at the global level: $(\sigma, ())$. The function *top* returns the top of the stack. The function *pop* is the opposite of push. We sketch the definition of atomic formulas, negation, questions and sequential conjunction. Note that the σ in the result and the σ onto which a copy is pushed, are the very same information state.

Definition 42 (Stacks (sketchy))
For each stack (σ, S) and formula φ define define the update function by:

$$\begin{array}{lcl}
(\sigma, S)[P(t_1, \ldots, t_n)] & = & (\sigma \restriction \{i \mid \langle V_i(t_1), \ldots, V_i(t_n)\rangle \in V_i(P)\}, S), \\
(\sigma, S)[\neg\varphi] & = & (\sigma \restriction \{i \mid i \not\preccurlyeq \sigma'\}, S), \quad \text{where } \sigma' = top((\sigma, (\sigma, S))[\varphi]), \\
(\sigma, S)[\varphi; \psi] & = & ((\sigma, S)[\varphi])[\psi], \\
(\sigma, S)[?\vec{x}.\varphi] & = & (\{\langle i, j\rangle \mid i \preccurlyeq \sigma' \text{ iff } j \preccurlyeq \sigma'\}, S), \text{ where } \sigma' = top((\sigma, (\sigma, S))[\vec{x}][\varphi]).
\end{array}$$

One example strategy that we can explain with the help of stacks, is so obvious that we neglected it in the beginning. It embodies the idea that a presupposition behaves just like asserted information. This makes no sense at the global level, because of the presupposition behaviour under negation, but it does make sense when embedded within the stack.

Strategy 3 (Content)
$\sigma[\partial\varphi] = \sigma[\varphi]$

Employed within stacks, the *content strategy* makes it possible to model Russell's approach to presupposition and negation: a failing presupposition simply leads to falsity. In case of a negation, we have two states in the stack. The presupposition can be projected to the global level, outside the scope of the negation, or not. In case it is projected, we end up with an inconsistent global state, else we end up with a trivial update. The resulting ambiguity of negation is thus modelled by the ambiguous behaviour of projection along the stack.

(77) It is not the case that the king of France is bald.

$$
\begin{array}{lll}
& (\sigma,())[\neg(\partial king \wedge bald)] & \\
& = (\sigma \restriction \{i \mid i \not\preccurlyeq \sigma'\},()), & \text{where } \sigma' = top((\sigma,(\sigma,()))[\partial king][bald]), \\
& = (\sigma \restriction \{i \mid i \not\preccurlyeq \sigma'\},()), & \text{where } \sigma' = top((\sigma,(\sigma[\partial_{content} king],()))[bald]), \\
& = (\mathbf{1} \restriction \{i \mid i \not\preccurlyeq \sigma'\},()), & \text{where } \sigma' = top((\sigma[bald],(\mathbf{1},()))), \\
& = (\mathbf{1} \restriction \{i \mid i \not\preccurlyeq \sigma[bald]\},()) & \\
& = (\mathbf{1},()). & \\
\text{or} & (\sigma \restriction \{i \mid i \not\preccurlyeq \sigma'\},()), & \text{where } \sigma' = top((\sigma[\partial_{content} king],(\sigma,()))[bald]), \\
& = (\sigma \restriction \{i \mid i \not\preccurlyeq \sigma'\},()), & \text{where } \sigma' = top((\mathbf{1}[bald],(\sigma,()))), \\
& = (\sigma \restriction \{i \mid i \not\preccurlyeq \mathbf{1}\},()) & \\
& = (\sigma,()). &
\end{array}
$$

The following strategy comes closest to the ideas of Heim, as reconstructed by (Zeevat 1992; Beaver 1996). There is accommodation proper, but when the global context is incompatible, the resulting update is undefined. So at the global level we use the cautious strategy. Locally, we for example use the content strategy. When an intermediate context already supports the presupposition, it is not projected any further: it is bound.

Strategy 4 (Heimian)

$$
\begin{array}{lll}
(\sigma,())[\partial\varphi] = & (\sigma[\partial_{cautious}\varphi],()) & \\
(\sigma,S)[\partial\varphi] = & (\sigma,S) & \text{if } \sigma \Vdash \varphi \\
& (\sigma[\partial_{content}\varphi], S[\partial\varphi]) & \text{otherwise}
\end{array}
$$

This strategy covers most of the observations. However, it can't deal with the cancelling cases. In particular, example (73d) and (74) would come out undefined, because at the global level the presupposition is incompatible, although there is a local context available at which the presupposition could be accommodated. To be able to deal with the cancelling cases one might propose the so called *full* strategy: presuppositions are accommodated from embedded contexts outwards to the global context, as close to the global context as possible (van der Sandt 1989), except when it is bound. However, in case accommodation turns out not to be possible, one backtracks and is satisfied with accommodation at some embedded context. Accommodation at embedded contexts is usually easier. In (73d) Mary may simply not be aware of the fact that there is no dog of John, so the hearer can easily accommodate one within Mary's beliefs. The clause about binding remains.

Strategy 5 (Full)

$$
\begin{array}{lll}
(\sigma,())[\partial\varphi] = & (\sigma[\partial_{cautious}\varphi],()). & \\
(\sigma,S)[\partial\varphi] = & (\sigma,S), & \text{if } \sigma \Vdash \varphi, \quad \text{else} \\
& (\sigma[\partial_{content}\varphi], S[\partial\varphi]), & \text{if } S[\partial\varphi] \text{ defined,} \\
& (\sigma[\partial_{content}\varphi], S) & \text{otherwise.}
\end{array}
$$

The principle that presuppositions always project outwards as close to the global context as possible, must be modified. Consider the following example, from David Beaver We know that not every German housewife owns a Porsche. So global accommodation is definitely not possible. The full accommodation strategy therefore predicts intermediate accommodation. Intermediate accommodation means in this case, that if a German housewife owns a Porsche, she will wash it on Sundays. However, out of the blue the example sounds odd. For many people the example is odd, because it seems to suggests that most German housewives do own a Porsche. In other words, it suggests that if someone is a German housewife, she will own a Porsche and wash it on Sundays. That reading is the result of local accommodation.

(78) Every German housewife washes her Porshe on Sundays.

$$
\begin{array}{ll}
\multicolumn{2}{l}{\sigma[\forall x(ghw(x) \to \partial y(p\_of(y))) \wedge wpos(x,y))] =} \\
\quad \sigma[\forall x \exists y((ghw(x) \wedge p\_of(y)) \to wpos(x,y))] & \text{intermediate or} \\
\quad \sigma[\forall x(ghw(x) \to \exists y(p\_of(y) \wedge wpos(x,y)))] & \text{local.}
\end{array}
$$

The fact that example (78) sounds odd, is evidence that intermediate accommodation is not always preferred over local accommodation. Therefore, the full strategy needs to be relaxed. The stack merely defines a set of potential accommodation contexts. Other factors, like pragmatic constraints or background knowledge determine the preferred location for accommodation among those. For example, we could differentiate between contexts created by attitudes or questions and by negation. Than we could maintain that example (73d) is acceptable, but example (74) is not. However, the theory does not explain where we might get these pragmatic constraints.

One of the messages of this example of Beaver's is that it is important to remain critical about one's intuitions on isolated examples. Take the infamous "Every farmer beats its donkey". Nowadays, the suggestion that it is plausible that farmers have donkeys is just as odd as the suggestion of (78). The intuitions on which an analysis of the interaction between presupposition, connectives and quantifiers is based, are possibly a by-product of a general process of understanding in context. Being language users we have a strong urge to make sense of an utterance. We try to imagine when somebody would say such a thing. And if this forces us linguists to accommodate donkeys, being cooperative, we do that!

3.7.5 Discussion

The stacks hypothesis holds that for connectives like $\neg$, ? or belief, a copy of the global context is pushed on the stack to calculate temporary results which can then be negated or compared at the global level. In our discussion of the accessibility of anaphora within embedded or subordinated contexts, we used the same notion. Negation, propositional attitudes and questions create contexts that can be accessed by mechanisms similar to modal subordination. Information can be accommodated 'for the sake of the argument' until more information comes in. We evaluate expressions in their local, temporary, non-actual context. At the actual layer, the result of the evaluation is then applied, negated, compared or communicated.

The stacks hypothesis can be turned into formal account (Zeevat 1992; Hulstijn 1995; Hulstijn 1996). However, we no longer uphold the stacks hypothesis. There a are a number of reasons for that. First, the idea of stacks as a by-product of the interpretation process is representational. It finds itself at odds with the model theoretic framework of update semantics. The intuitions underlying stacks of subordinate contexts are much easier presented in DRT or another representational account, such as Asher's SDRT. Second, the account is rather technical. It focusses on theory and not on the function of presupposition accommodation in actual dialogue: to increase coherence. In general it is very difficult to judge the examples on which theories of presuppositions are based. Each example presupposes its own context of interpretation. Although we believe that the general concept of presupposition accommodation can be fruitfully studied by comparing update strategies, the sorts of distinctions made by the stack theory, for example between Heimian or full accommodation, can not be upheld empirically. Third, there is an obvious connection between stacks and modal subordination effects, which would need to be investigated.

Instead, a good theory of presupposition accommodation should be able to make predictions about the usefulness of certain types of presupposition triggers. There are only a few theories that could be of value here, one of them being Asher and Lascarides' (1998b, 1997) account of presupposition inference as a special case of *bridging inference*. Because it is the most important rival theory, we briefly discuss it here.

To Asher and Lascarides, accommodation is but a by-product of interpretation and understanding. New information must be attached to the existing dialogue representation structure, by means of a *coherence relation*. For narrative discourse, typical coherence relations are narration, explanation, and background. Cue words, like 'moreover' or 'because', help to select the appropriate coherence relation. Each coherence relation comes with a number of constraints that must hold for it to be applicable. Asher and Lascarides use a default logic to specify principles to regulate the attachment.

The coherence relation *narration*(α, β) connects the descriptions α and β of two events that happen one after the other. So once we concluded that the narration relation holds, we can infer by default that the events described by α must occur before the events described by β. Such an inference is called a *bridging inference*. The idea is that presupposition accommodation is the result of bridging inferences. An example is "John is late. The car broke down.", where in order to accept the second sentence as an explanation for the first, we must accommodate or infer the 'bridge' that John has a car and usually comes by car. Otherwise the resulting combination is not coherent. We used similar reasoning to explain example (46).

Now, with these principles and default rules in place, Asher and Lascarides take the Van der Sandt theory and turn it upside down. The general constraint that presupposed material accommodates as close to the global context as possible, is now reduced to a last resort, which only applies in the absence of more substantial bridging inference.

Whether a presupposition is accommodated depends on several things. First, it depends on the pragmatic and semantic content of previous dialogue, because this is used to infer which contexts the presupposition can coherently attach to, and which rhetorical relation to use. Second, it depends on the relative plausibility of the various choices of attachment that are coherent. Third, it depends on the relative strength of the rhetorical connections provided by various choices of attachment. Finally, it depends on the default to attach presuppositions close to the global context (Asher and Lascarides 1998b, p 38).

We believe that such a view is essentially correct. However, the notion of information being attached to certain parts of the dialogue representation tree, is impossible to capture in our model theoretic framework. And as we have seen, stacks are a very poor approximation of such representation structures. Ideally, we would need a theory that combines a semantics for the content of utterances, with a representation structure based on coherence, task and interaction patterns.

3.7.6 Presuppositions and Issues

After this excursion to specific strategies for dealing with presuppositions, we return to the main topic: an update semantics with questions. In case we do not make stacks explicit, the cautious strategy already covers most intuitions. So we adapt the cautious strategy for existential presuppositions and other anaphoric expressions. Presuppositions and anaphora are treated alike, except that anaphora do not carry enough additional information to sustain accommodation. However, they do support temporary accommodation of a mere referent, as was suggested by the cross-speaker examples and the cataphora case.

To summarise our account, a presupposition expression or anaphoric expression $\partial\vec{x}.\varphi$ can be treated in one of three ways: (i) the presupposition is already supported by the context. For anaphoric expressions (free variables) that means that they are already bound. In that case nothing further happens. (ii) The presupposition can be added consistently to the context, or a new referent can be consistently created to accommodate the anaphoric expression. In that case, it will indeed be added or accommodated. This requires that suitable assignments are found for the variables $\vec{x}$ occurring in the anaphoric expression. All unsuitable world-assignment pairs are eliminated. This too is a form of accommodation: it increases the discourse information of the information state. Possibly some bridging inferences also need to be made. (iii) The presupposition or anaphoric expression is inconsistent with the context. In that case, the utterance was inappropriate in that context. This is modelled by the resulting update becoming undefined. Presuppositions are the only reason that the update function is a partial function.

Definition 43 (Presupposition Accommodation)
For each $\sigma \in \Sigma_I, \varphi \in L_\partial$ define an update function $[\varphi]$ as in definition 23, where

$$\sigma[\partial\vec{x}\varphi] = \begin{cases} \sigma, & \text{if } \sigma \Vdash \varphi, \\ \sigma[\vec{x}][\varphi], & \text{if } \sigma[\vec{x}][\varphi] \neq \mathbf{1} \text{ and} \\ \text{undefined}, & \text{otherwise.} \end{cases}$$
□

At the illocutionary level where dialogue acts are being exchanged, there is a general policy to avoid undefinedness, or to repair it. One way of avoiding undefinedness, is to restrict to local or intermediate accommodation. Another is to revise one's existing beliefs. Yet another way of avoiding undefinedness is to refuse to consider the utterance at all, and remain in the current information state. If the utterance is integrated with the context without a problem, this should be indicated to the speaker, either explicitly by 'uhuh', or 'yes' or implicitly by continuing the conversation in a coherent way. Only utterances that are acknowledged in this way, can be added to the common ground.

3.7.7 Presuppositions and Questions

Presuppositions and questions interact in interesting ways. Obviously, presuppositions are modelled in questions just as they are in indicative sentences. This is one of the defining characteristics of presuppositions: that they escape from within embedded contexts like questions. In case accommodation is possible, first the presuppositions are accommodated. After that, the partition only covers the worlds in which the presupposition is satisfied. In this way the domain of discourse can be restricted to a subset of the original domain.

Now in general, the set of objects over which a generalised quantifier or other expression ranges is dependent on the context (Westerstrahl 1985). By means of restricting the domain of discourse to contextually relevant objects, many at first unintuitive meanings of quantifiers can be explained. Also it explains the uniqueness implied by definite descriptions. There are many presidents, but given that we are in France, the president is uniquely determined to be Chirac. One of the advantages of combining a dynamic semantics with a semantics of questions and answers, is this process of implicit *domain restriction* (Jäger 1996). Examples (79) and (80) of Jäger illustrate this point.

(79) A: Which Athenians are wise?
B: Only Socrates is wise.

(80) A: Who are wise?
B: Only Socrates is wise.

The operator 'only' is a focus sensitive operator. It contrasts the expression in focus with the available alternatives (Rooth 1992). Earlier we have seen that the set of alternatives that defines the semantics of focus corresponds to an issue, which is either explicitly asked or else presupposed. In example (79) the which-phrase restricts the domain of discourse to Athenians. In example (80) the domain is not particularly restricted. Out of the blue the question ranges over all individuals, including for instance Zeus. So *B*'s answer that only Socrates is wise implies the blasphemous consequence that Zeus is not considered wise.

But what about the presupposition which is traditionally associated with a question? Does a question presuppose that there is a positive answer? Given general conditions on rationality and on the appropriateness of questions, it follows that the asker must consider a positive answer to the question probable. Otherwise it would not make sense to ask it. On the other hand, in many cases a negative answer is possible too.

If we treat the suggestion that there is indeed a wise Athenian in (79) as a presupposition, the answer 'nobody' would be classified as a response at a meta-level that explicitly denies or cancels the presupposition, much like the presupposition of examples (74) and (77) could be cancelled. In that case we predict some 'friction' in the interaction between dialogue participants. Cancelling is really unwanted; it shows that the speaker had wrong expectations. Therefore cancelling requires some repair mechanism, for instance an explanation. If questions do not presuppose a positive answer, the answer 'nobody' is a perfectly acceptable answer. In that case we predict no 'friction' in the interaction, and no repair is needed.

We do not have a definite solution. Whether to treat the answer suggestion as a presupposition depends on each particular case. Some questions clearly carry the presupposition that there is a positive answer, like example (81). Answering 'nobody' then requires at least an explanation. Other examples, like (82) clearly don't. In this case the underlying purpose, to find the culprit or to invite a guest, and the lexical and aspectual properties of the verb phrases 'has stolen' and 'would like' make the difference. Note that the very reason that (82) is regarded as a polite way to invite people to take some cookies, is that it does not impose. It leaves the decision to the guests.

(81) Mummy: Who has stolen the cookies?
Child: Nobody did! It was the dog.

(82) Mummy: Who would like some cookies?
Mummy: ... (puzzled) nobody?

Many questions are ambiguous or underspecified with respect to their typical answer presupposition. Sometimes explicitly denying the presupposition as in (83a) makes sense, so apparently the sentence is otherwise at least ambiguous. However, the possibility of a stronger way of expressing the presupposition, as in (83b), suggests that without it, the presupposition is at least not obvious. Background knowledge again plays a decisive role here. When we replace 'Athenian' with 'Spartan', the non-presupposing reading of the question is more appropriate.

(83) a. Which Athenian, if any, is wise?
b. Which Athenian is the wise one?

So again we find an interesting interaction between presupposition accommodation, coherence principles and background knowledge. Many of these principles are domain dependent. But even in the absence of a complete set of principles, simply being aware of the difference between a presupposition and an assertion can prevent misunderstanding.

Paying attention to the presuppositions of questions is of crucial importance to the usability of dialogue systems. The following example is taken from an early paper on cooperative question answering (Kaplan 1982). It contrasts the answer presupposition with regular presuppositions in a question. If the user's question is straightforwardly translated into a database query, a system response as in (84) is most likely. There are no entries in the database that match the query. Nevertheless, such an answer is misleading, if it results from a failing presupposition. If CS can be recognised as a course an answer as in (85) is more appropriate.

(84) U: Which students got an A in CS in '79?
S: zero

(85) U: Which students got an A in CS in '79?
S: CS was not taught in 1979.

To take one last example, the ambiguity resulting from presupposition failure is one of the disadvantages of the implicit confirmation strategy that is used by some spoken dialogue systems (see chapter 6). The system gives feedback to the user about what it thinks it recognised, by phrasing it as a presupposition which is part of the next system utterance. When the system made a recognition error, the user is often not sure how to respond: to deny the presupposition and correct the system, or to continue and answer the question, hoping that the misunderstanding can be corrected later. Evaluation experiments reveal that many users feel troubled because they do not know how to respond (Weegels 1999). There are two interaction rules that conflict: (i) to respond to a question and (ii) to indicate a misunderstanding as soon as possible after the misunderstanding was discovered.

(86) S: Hoe laat wilt van Amsterdam CS naar Rotterdam reizen?
What time do you want to travel from Amsterdam CS to Rotterdam?
presupposes: you want to travel from Amsterdam CS to Rotterdam at some time.
U: Uh ... om drie uur, maar ik wil vanaf Amsterdam Amstel!
Uh .. at three o'clock, but I want from Amsterdam Amstel!

3.8 Topics, Salience and Task

Our approach is one of raising and resolving issues. We have seen several examples of expressions that trigger issues to be raised. Explicit questions and focus constructions are the most important examples. In the introduction we expressed the hope that the structure of issues may also be used to account for other types of dialogue structure. In particular: topics, salience and task. These aspects of dialogue structure in turn help to determine what the next issues will be are. We give a brief overview.

Dialogue structure enables dialogue participants to process the dialogue efficiently. Humans have limited cognitive abilities, a limited attention span and a limited memory. In linguistics the tricks and mechanisms to manage the limited attention span show up as theories of topic and focus. The main principle is the following: if something is implicit, given or to be expected in the context, you do not have to mention it again. On the other hand, if something is explicit, new or different, you have to mention it with extra effort. This principle leads to a scale of relative importance. However, in linguistics it makes sense to distinguish different scales that are partly orthogonal.

On the one hand we have *focus*, which represents new or contrastive information; focus is generally opposed to *ground*. In examples (65), (66) and (67) we showed how intonation structure and other focus expressions can be used to indicate the focus of an utterance. There has always been a strong connection between focus and questions. So much so that there is a 'question test' (Sgall et al. 1986), to find out the focus of an utterance. In general the item asked for is in focus. This makes sense, because the item asked for is apparently the most relevant or salient at that point. We showed that the semantics of a focus construction is an issue which is presupposed. The presupposition of this issue, what needs to be presupposed in order for the issue to make sense, is the semantics of the ground.

On the other hand we find the notion of *topic* which indicates the object or objects that an utterance or a dialogue fragment is about. Topic is generally opposed to *comment*: the information that is either predicated of, or requested about the topic. Topics too have been associated with questions. We mentioned van Kuppevelt (1995) who more or less equates topics with what we have subsequently called issues. A natural solution would be to define the topic as the object, i.e. not factive information, that is presupposed by all issues. The comment is than either the issue, what is being asked about the topic, or else the information that has recently been provided as an answer to the issue.

A related distinction concerns the relative givenness and newness of discourse referents. This too leads to a scale. Recently mentioned referents are high up the scale; long neglected referents drop off the scale at the far end. This mechanism is related to that of topics. Generally, the topic is given and has recently been mentioned. Related sub-topics are likely to be the next topic of discussion. Such scales are used often in algorithms for automatic anaphora resolution. The best example of such theory is *centering* (Grosz et al. 1995). There is an object, the *backward looking centre*, that constitutes the topic of the dialogue. On the basis of what has been said, and on the basis of the current backward looking centre, predictions can be made about likely new centres. These predictions are ordered on a scale. The most likely next centre is called he *forward looking centre*. After every utterance, the actual centre is determined and a new prediction is made. The scales depend largely on syntactic structure and on the mentioning and re-mentioning of objects.

3.8.1 Topics

How can we characterise the notions of topic and comment in terms of our structure of issues? The best way to introduce topic structure, is by example. Here we present example (79) again, but now continued as a little dialogue.

(87) A1: Which Athenians are wise?
B1: Only Socrates is wise.
A2: And who is brave?
B2: Alcibiades.
A3: What about Spartans?
B3: They're brave but not wise.

After A1 the domain of discourse remains restricted to Athenians. It remains this way, until the topic is changed. In B1 the sub-topic becomes Socrates. Since Socrates is an Athenian, this is compatible. The follow-up question A2 about braveness is therefore implicitly restricted to Athenians as well. A follow-up question about the price of olive oil would not be implicitly restricted to Athenians, because it constitutes a topic shift. In general an utterance leads to a topic shift in case it it is not compatible with the current topic frame, where a topic frame is roughly characterised as a cluster of information that is related to or depends on the current topic. After A1 the topic is Athenians and the general issue is their characteristics. A follow-up question like "What about .." in A3 changes the topic, but leaves the issues. A follow-up question about the price of olive oil would change both the topic and the relevant issues, because issues of wisdom and bravery are not appropriate for olive oil. Wisdom and bravery are applicable to Spartans. Which is why these issues are in fact addressed in B3.

The notion of topic is problematic in the literature. Most accounts share the idea that the topic constitutes what a dialogue segment *is about*. What this aboutness consists in is left largely unspecified. We follow van Kuppevelt (1995) in giving a characterisation in terms of another, less complicated structure: that of questions. There is a direct relation between topic and comment on the one hand, and questions and answers on the other. A topic is contrasted with a comment: the information that is provided about, or requested about the topic.

However, Van Kuppevelt's relation between topics and issues is too tight. The topic is equated with what is being questioned: the possible nominal answers. If we assume an information state σ with a contextually induced issue $?\vec{x}.\varphi$, Van Kuppevelt's topic could be expressed as $\{V_i^\sigma(\lambda\vec{x}.\varphi) \mid i \in \mathrm{d}(\sigma)\}$. So in B2 the topic would be the set of brave Athenians, not just the Athenians as we would have it.

Our notion of topic comes closest to the following: "An entity T is the topic of an utterance U if U is intended to increase the addressee's knowledge about T, request information about T or otherwise get the addressee to act with respect to T (Rats 1996, p37)". However, this definition is too general. What kinds of entities can take the place of T? We suggest to define a topic as a particular object or set of objects, which is represented by a referent or series of referents in the domain of discourse. We allow sets of objects because we envisage plural topics, as 'Athenians' in example (80). So abstract entities are included as well. However, we request that is some discourse referent, possible a plural referent, to refer to it. Which referents become the topic depends on the most recent utterances. For example, the object used to answer an issue may constitute the topic for the next utterance.

Under this account, the comment corresponds to either the current issue, in case an interrogative was the latest utterance, or to the content of an assertion that resolved the latest issue. That shows that the notion of comment is problematic; it is either information that is requested, or information that is provided! Therefore we propose to replace the notion of comment as requested information by the notion of a *topic related issue*, namely the issue that 'centres around' the topic. An issue 'centres around' a topic, when each of the possible answers presupposes the existence of the topic, and asks for some property or characteristic of it. The comment can then be defined as the answer, or solution to that issue.

A precise definition of these intuitions is a subject for further research. The main problem is that different definitions give different predictions, and these have to be tested against empirical data. So although we have a general idea of the notions the kinds of distinctions that can be made by the logic are too fine-grained to be justified on the basis of our small set of examples.

What is the influence of topic structure on the prediction of issues?

Imagine an pre-defined data structure, that specifies, among other things, all potential topics in a particular application domain. This hierarchical structure is often related to the task, because only certain objects are likely to appear in a task. For SCHISMA potential topics are performances, actors, titles, dates, prices etc. The data structure can be represented as a type hierarchy. There are *objects* of certain *types*. Objects have certain *attributes*, based on their type. Attributes may themselves refer to objects again; that would be potential sub-topics. When a topic is introduced, not all of its attributes are known. In the terrorist example in the introduction, the topic 'terrorist attack' raised a number of issues, based on background knowledge about newspaper stories. So in general we expect that a new topic triggers a mass of issues, related to the possible attributes of the topic.

Each dialogue topic carries with it a *topic space* of related objects and attributes. The actual topic space of a given topic, is constrained by the type hierarchy that specifies all potential topics. In general we expect to stay within one topic space. Topic continuations take less effort, cognitively, than topic shifts. When evaluating the success of a dialogue, the number of topic shifts is good indicator of the cognitive processing load. The lower the number, the easier it was to process.

How are topics introduced and referred to in actual dialogue? The following observation are generalised from the empirical work of Rats, (1996, Ch 5). With respect to topic structure user utterances can be divided into four groups depending on how the topic space relates to the previous one. The groups are ordered by the relative complexity of the expression used to refer to the topic.

(88) a. *topic introduction*: a new topic is introduced using long definite descriptions, proper names, (fragments of) titles or complex referring phrases.
b. *topic shift*: a different topic is introduced, when objects or attributes from the utterance do not *match* the topic space associated with the current topic.
c. *topic narrowing*: a subtopic of the previous topic is introduced. Uses the same mechanism as topic shift.
d. *topic continuation*: the topic remains the same. it is referred to with shortened descriptions, demonstrative pronouns, personal pronouns, or left implicit.

Turning these observations around, we obtain powerful heuristics for finding the most likely next topic.

(89) a. When items are described using definite descriptions, proper names or other complex referring phrases, such items are likely to be the new topic.
b. When items are described using shortened descriptions, demonstrative pronouns, personal pronouns or left implicit, the old topic is likely remain.

3.8.2 Salience

By intonation, word order and particular syntactic constructions, speakers indicate the *focus* of an utterance: the constituents that contain either new or else contrasting information. As we have seen, a focus construction is a presupposition trigger that indicates that a certain issue is relevant at that point. For contrastive focus, we can also use issues. Recall example (67). The value in focus must constitute a contrast with some expected value, compare (van Deemter 1994). These mechanisms work well for individual utterances. How do they scale up to a dialogue level?

The material in focus is shown by the speaker to be regarded as the material that is most salient at that point. In chapter 2 we characterised something as salient when it is 'worthy of attention'. However, things can be more or less worthy of attention. The relative importance of issues produces a *salience order*. So far in this chapter we have assumed that all issues are equally important, and that they can be combined into one big contextual issue by means of a join. Although this is technically correct, it is an oversimplification. Our hypothesis is that by defining a partial order over issues, which specifies the relative importance of resolving them at that time, we get a rather general notion of salience.

By ordering issues, we automatically also provide a ordering on possible and actual answers to issues, and therefore on information in general. So apart from the information order $\sqsubseteq$ based on logical entailment, we now propose an extra-logical structure to rank the relative importance of utterance content. Similarly, we can define a salience order over discourse referents that corresponds to the salience order over the issues that the referents are the topic of. Typically, the salience ranking must respect the topic-structure which must again respect the background information on the domain.

The idea starts with the QUD structure of Ginzburg (1995), that contains 'questions under discussion' . This is essentially a stack that keeps track of the most salient issue given the latest move. It helps to determine what to say next. As such it is much more related to an algorithm for a dialogue system, than to a semantic theory. Part of the function of QUD is to keep track of the initiative-response structure, which we handle in chapter 5. Of course the relative importance of issues in dialogue depends on the state of interaction. Responding to a question does not allow a delay, so working out the answer should be most prominent. Algorithms for adding and removing questions from QUD are given in proposals for a semantics of instructional dialogue developed in the Trindi project (Bohlin et al. 1999).

A notion of salience defined over issues, can be generalised to other aspects of the information state. In particular, salience can be expressed by a partial order over discourse referents or objects. Hendriks and Dekker (1996) show that a simple extension of DRT with a partial order over referents allows one to model the distinctions and intuitions behind the given-new and focus-ground partitions described in Vallduví (1990, 1994), without the confusing memory-card metaphors that are applied there. Huls et al. (1995) uses a salience order over objects mentioned in the dialogue for disambiguation for anaphora resolution. In addition, they account for the effect of pointing gestures, by giving objects that have been pointed at a very high salience value. Other ways of influencing the salience order are syntactic structures, or the use of pronouns to re-fresh the salience of an object.

3.8.3 Task

In various places we expressed the conviction that raising issues depends on the task, or on the apparent goals of the participants in dialogue. In chapter 4 we further argue for this view. The approach is based on the observation that goals have a similar effect on information as questions: to add structure. Assume that each goal is associated with a set of possible alternative plans to achieve it. Which plan is best in the given situation, may still be undecided: the agent faces a decision problem. The agent must decide what action to take. Just like a question is modelled by a partition corresponding to the alternative answers, a decision problem is modelled by a partition corresponding to its alternative solutions. Each block corresponds with the preconditions of an action or plan. In other words, blocks contain precisely those possible worlds that do not differ with respect to the preconditions of potentially useful actions.

In addition to the task, there is also the interaction that has to be managed. Some issues are raised on the basis of the interaction process. Consider the question "What did you say?" Another example is the grounding issue that we suggested in the discussion of cross speaker anaphora. One could argue that such issues are not part of the content of utterances and should therefore not be modelled by issues. On the other hand, the interaction itself can become the topic of conversation as well. More on the management of interaction in chapter 5.

How do these aspects interact? As a rule, one might say that the coherence of an utterance that counts as an initiative, depends on the task or goal. This may motivate a new exchange. The coherence of a response depends on its pertinence with respect to the issues raised by the currently active initiative.

3.8.4 Raising Issues

To summarise, by what means can an issue be raised?

(90) In general, an issue can be raised
- by a question.
- by a focus-expression.
- by a topic shift. A topic raises issues about expected attributes and subtopics. In particular, the introduction of a discourse referent raises an issue as to the identity of the referent.
- by underdetermined phrases (disjunction, implication). By the maxim of quantity these trigger issues as to the truth of their component formulas.
- by a dialogue control issue, for example the grounding issue that is temporarily raised for each utterance.
- by an apparent change of goal of one of the participants. Each goal generates a plan with actions the preconditions of which become issues for deliberation.

Can issues also be lowered? Issues can be resolved by the mechanism described in section 3.5. In that way they disappear. However, in some cases issues can also be actively lowered. For goal related issues in particular, when the goal is achieved in some other way or when it has become inachievable, the issues connected to the goal can be lowered. In a similar way dialogue control issues can be removed once the interaction has proceeded in another direction.

3.9 Conclusions and Further Research

This chapter meant to define notions that characterise successful information exchange on the content level. To this end we defined a particular game of information exchange: inquiry. We argued that the relevance of utterances is of key importance. Information is seen as data that is being structured by issues. An issue is the semantic structure that models the content of a question, just like a proposition models the content of an assertion. Information can be modelled in information states, defined as an equivalence relation over a set of possibilities. The content of an utterance is modelled as an update of such an information state. Adding data eliminates possible words from the information state; adding structure eliminates pairs of worlds from the equivalence relation: those pairs that are not indistinguishable with respect to current issues. An issue corresponds to a set of alternatives to select from. Resolution of an issue means that one alternative is selected. Partial resolution means that the number of alternatives is reduced. The notion of relevance can be expressed, although in a rather technical way, in terms of the resolution of issues. Given current issues, an utterance is relevant when it resolves at least one current issue.

The formal properties that correspond to the Gricean maxims of quality, quantity and relation are consistency, informativeness, relevance, and licensing. Given a context consisting of a sequence of utterances, we can say that an utterance is consistent when it's update does not lead to the empty set of worlds. An utterance is informative when its information is not already entailed by the previous utterances in context. This holds for utterances that express issues too. An utterance is relevant, when it at least reduces one issue that has been raised by previous utterances. Finally, an utterance is licensed, when it resolves only issues that have been raised. It should provide exactly the information that is required for current purposes, not more, nor less.

These properties assume that we know what issues have been raised at each point in dialogue. Roughly there are two types of sources for issues to be raised: linguistic cues and task related cues. The linguistic cues have been dealt with at various places in this chapter. Issues are raised by explicit questions, are presupposed by focus constructions or are entailed by other issues that are logically related. Moreover, we showed that the introduction of a discourse referent triggers issues as to the identity of the referent. The other cues are less easy to define. We discussed the relation of issues with topics, salience and task.

3.9.1 Related Research

The most closely related research has been dealt with throughout the chapter: systems of dynamic semantics, update semantics and semantics of questions and answers. But how does our work relate to other views on relevance?

Relevance has been studied from many angles. *Relevance theory* provides an account that relates relevance to cognitive processing load (Sperber and Wilson 1986). When people understand an utterance they try to maximise relevance: they pick that context against which the relevance of the utterance is greatest. Relevance can be quantified, using so-called *extent* conditions: an assumption is relevant in a context to the extent that its effect in this context is large and the effort to process it is small.

As was indicated by De Roeck et al. (1991) relevance theory is not sufficiently worked out to be implemented. However, some of its principles remain valid. They suggest a theory of helpful answers that is loosely based on relevance theory. It is implemented in property theory. The use of property theory makes it possible to link a formal semantics of questions and answers to implemented ontologies. Their approach differs considerably from ours in the details. Its spirit is similar, namely that such notions as relevance should be explained at the content level.

The most widespread approach to relevance in computational linguistics is based on the plans and goals paradigm. Here an utterance is considered to be relevant, when it provides a contribution to a plan that can be used to achieve a current goal. This principle can be used in *plan recognition* (Carberry 1990). The contribution helps to detect what plan,or goal the participant is trying to achieve. A good example of an application of this idea is Ardissono et al. (1998), who analyse misunderstanding in terms of plans and goals. In principle, if we can make a formal link between issues and preconditions of actions to achieve a common goal, our approach can incorporate this one. However, that requires that we go into the details of a theory for planning and action.

There has also been an interest in relevance from logicians. Usually, relevance is mentioned as one of the reasons why material implication is not the best way to model natural reasoning. After all, the conclusion does not have to be related to the premisses. Relevance logic tries to amend this, by using a different way to define conditionals (Anderson and Belnap 1975). It would be interesting to investigate how their purposes could be met by our methods.

The idea to treat information as data that is structured by issues, is very fruitful. Once you can model issues, a lot of other aspects appear to be related. Here we give a selection of ideas for further research. A common denominator of these proposals is the formalisation of non-logical information structure by means of issues, which can then be applied to solve problems that have to do with the fact that agents are not fully rational.

3.9.2 Revision and Contraction

We can prove that the assertive and interrogative updates of section 3.3.5 are in fact a regular *expansion* operator. Information may only be added, not withdrawn. Not having a contraction or revision function is a major drawback of the framework advocated in this chapter. If we reach an inconsistency, all information is lost! How can we deal with revision or contraction of information in the light of a contradiction?

A theory of belief revision describes how to revise an information state when faced with new information, in such a way that the resulting information state is consistent, contains the new information and has lost as little of the original information as possible. Belief revision has been studied in the AGM framework (Alchourrón et al. 1985; Gärdenfors 1988). Postulates have been formulated that specify desired behaviour of the revision operator. Given these postulates, there is a class of operators that are based on *partial meet contraction*. Most of these are based on an *entrenchment order*: the more 'embedded' the information, the less likely it is to be revised. Entrenchment is usually conceived of as an external or at least non-logical relation. That immediately raises the problem how the entrenchment order is motivated.

Our first suggestion for further research is to investigate entrenchment relations derived from discourse or dialogue structure. Suppose that the belief change of the agent takes place as a result of an ongoing dialogue. Then we have additional cues about the relative importance of information, coded by focus-ground, topic-comment distinctions, or by a salience order. Also the dialogue structure may be of influence: what are the coherence relations between the different dialogue segments, corresponding to the information to be updated and revised with? Or, for argumentative dialogue, what is the role of the new information in the argumentation sequence? If a segment is signalled to be a reason or motivation for some other information, this indicates it is to expected by the speaker to be more embedded, or more trustworthy. A first example of the usefulness of dialogue structure for revision, is provided by natural language corrections. The scope of a correction is indicated by intonation and by the parallel syntactic and semantic structure of the correction with the preceding utterance (Gardent et al. 1996; van Leusen 1997). And as we have seen, this structure can be captured by issues too. For such structures, no full revision is needed. Obviously, for more deeply embedded propositions the revision will be more difficult.

A second suggestion for further research is to investigate the relationships between issues and inconsistency tolerant approaches to belief revision (Schöter 1996; Chopra and Parikh 1999). The idea is that a belief set can be divided into relatively independent subsets of beliefs. Inconsistencies are to be kept local to a subdivision, in order not to jeopardise the rest of the beliefs. We suggest that a subdivision based on the structure of the dialogue an agent is engaged in, is a good candidate for providing such a division. Thus the subsets will be related to salience, tasks, topics and issues. Ideally, we might use issues as a semantics for a particular way of subdividing the belief set.

3.9.3 Dynamic Signatures

As the issues change in a dialogue context, in principle also the ontology that underlies the logical language could change. Some distinctions need no longer be made; other distinctions become relevant. Traditionally, a model had a fixed domain and a fixed interpretation function. In dynamic semantics the domain of discourse has become flexible and is made dependent on the dialogue context. In principle the issue structure allows to make the interpretation function flexible too, and make it dependent on the relevant issues in a dialogue context.

We have a logical language that is defined on the basis of symbols *Pred*, *Func* and *Cons*. These sets of symbols together with the set *Type* of basic types of objects in an application domain form a *signature*. The signature plays the role of the *ontology* that is part of the information model that we discussed in chapter 1. For each application domain it describes what the relevant objects, attributes and relations are, at least according to the judgements of the designers who analysed the application domain. In other words, an ontology for a system is made with certain possible issues in mind. That means that in principle, issues can be used to assess the relative strength of different conceptualisations or signatures.

Recall example (26), reproduced here as example (91). In the signature of a television producer we need not distinguish between 19:02 hours and 19:03 hours. Therefore it does not make sense to include all these different time points in the signature. For the scientist however the distinction can be relevant and the time points should be included. The example suggests that the information model and therefore the signature depend on the role of the dialogue participants in an application. Given a role, we can identify a particular task which the participant needs to accomplish in the dialogue. For each task a number of issues are relevant. These will correspond to the preconditions for successful completion of a task. If the modelling was done right, we can assume that task-related issues are captured in the signature; other issues are neglected.

(91) When is the launching of the Space Shuttle?
 a. – at 19:02:15 GMT (space scientist)
 b. – at prime time (television producer)

What happens to the dialogue representation if someone changes role during the dialogue? What happens if a new topic comes up, or a new task is started? For example, the scientist comes home and wants to watch television. The semantics of issues developed in this chapter makes it possible to model the changing distinctions made the scientist.

Earlier we suggested the definition of a vocabulary $\mathcal{P}$: the set of ground atomic formulas. The information space is defined as the set of total valuations on the basis of $\mathcal{P}$, so $W_{\mathcal{P}} = \mathcal{P}^{\{0,1\}}$. Obviously, the vocabulary depends on the signature, and therefore on the issues that the designer of a system had in mind. But during a dialogue the relevant issues may change. Given a way to induce a currently relevant vocabulary $\mathcal{P}(\sigma)$ on the basis of contextual issues, we can work with so called partial representatives of the equivalence classes induced by contextual issues, as if they were total possible worlds. In other words, we can 'zoom in' and 'zoom out' on the structure of possible worlds, depending on what issues are relevant. If we demand that utterances are licensed, no relevant information can be lost by using such partial representatives. This possibility of zooming in and out is reminiscent of Peirce (1867) who distinguishes two ways in which information may be increased. One can increase the set of attributes used to describe known objects, called *comprehension* or *depth*. Or one can increase the set of objects that are described, called *extension* or *breadth*. This may happen as the result of new discourse referents being introduced, or by new issues being raised, whereby different objects may become distinguishable. Consider a topic switch from television programmes to the scientific measurement. This increases the 'depth' of the concept of time and subsequently adds a whole load of new time objects to the relevant domain of discourse. Traditionally it was believed that the product of depth and breadth, called *area*, was to remain constant. Peirce convincingly shows that each dimension can be increased independently of the other. Please note the relevance of this view for theories that try to quantify the amount of information. These notions of depth and breadth can be characterised in our model.

A *situation* is a partial valuation function. The set of atomic formulas for which a situation s is defined is called the *vocabulary* $\mathcal{P}(s)$. We say that situation s subsumes situation t, written $s \leq t$, when t preserves all the truth values assigned by s, and possibly assigns some more. Possible worlds w are total with respect to the original vocabulary $\mathcal{P}$, but a situation s can be made total by restricting it to its vocabulary $\mathcal{P}(s)$. Situation t subsumes s, $s \leq t$, iff $t(p) = s(p)$, for all $p \in \mathcal{P}$ with $s(p)$ defined.

Each partial valuation s *represents* a set of possible worlds: the set of worlds that are subsumed by it. And for each set V of possible worlds, it is possible to find the least underspecified partial valuation, that represents it. This valuation is called the *representative* $\mathrm{r}(V)$ of the set. For each $w \in \mathrm{d}(\sigma)$ there is an s such that $s \leq v$ for all v such that $\langle w, v\rangle \in \sigma$. By definition $s = \mathrm{r}(V)$ for $V \in \mathrm{i}(\sigma)$. If we replace all the blocks in the partition generated by contextual issues with their representatives, we get a kind of minimal representation of the information in an information state. From this minimal representation the Peircian notions of breadth and depth can be defined.

By taking the union of the vocabularies of the representatives we can define the *current vocabulary*. For each $\sigma \in \Sigma$ define $\mathcal{P}(\sigma) = \bigcup_{V \in \mathrm{i}(\sigma)} \mathcal{P}(\mathrm{r}(V))$. Which atomic facts end up in the current vocabulary? For each block: all the facts on which the block agrees. Since blocks are pair-wise disjoint, all the atomic propositions that matter for the current issues, namely those facts that discriminate between blocks are in the current vocabulary. But a lot of other, possibly irrelevant facts too. In particular, all the facts that have been relevant earlier to resolve another issue. Therefore we restrict the vocabulary to those atoms that discriminate between blocks. That gives the *relevant vocabulary* $\mathcal{P}^*(\sigma)$. For each $\sigma \in \Sigma$ define $\mathcal{P}^*(\sigma) = \bigcup_{V \in \mathrm{i}(\sigma)} \mathcal{P}(\mathrm{r}(V)) \setminus \bigcap_{V \in \mathrm{i}(\sigma)} \mathcal{P}(\mathrm{r}(V))$. This represents precisely the ground atomic formulas that are dependent on the current issues. Now the notion of depth seems to correspond to the set of predicates that occur in $\mathcal{P}^*(\sigma)$. And breadth would then correspond to the set of all terms that occur in $\mathcal{P}^*(\sigma)$.

Definition 44 (Depth; Breadth)
For each $\sigma \in \Sigma$ define

$$\begin{array}{lcl} \mathrm{depth}(\sigma) & = & \{P^n \mid P(t_1 \ldots t_n) \in \mathcal{P}^*(\sigma)\} \\ \mathrm{breadth}(\sigma) & = & \{t_k \mid P(t_1 \ldots t_n) \in \mathcal{P}^*(\sigma), 1 \leq k \leq n\} \end{array} \quad \square$$

This is only one possible way to understand these notions. There could be a relation between the objects that can be named by terms, the breadth, and the objects that are mentioned in the dialogue: the discourse referents $\mathrm{dom}(\sigma)$. Also, we could demand a more semantic characterisation, not by terms and predicate symbols, but by objects and sets of objects that represent predicates.

3.9.4 Awareness

The intuition behind the equivalence relation induced by a question is that worlds are indistinguishable with respect to their answer to the question. Interestingly, such an interpretation comes close to the motivation for accessibility relations in modal epistemic logic (Fagin et al. 1995). There, two worlds are related when they are indistinguishable with respect to the information of the agent. A combination of the epistemic accessibility relation with the equivalence relation induced by a question or issue can be used to express indistinguishability with respect to information that addresses the question.

A theory of knowledge along these lines, may somewhat soften the consequences of the counterintuitive *problem of deduction* that arises in a possible worlds analysis. In a possible worlds framework an agent believes all the consequences of its information. The problem then is to explain how the conclusion of a deductive argument can be informative (Stalnaker 1984). And as we have seen, this problem haunts assertive information just as much as interrogative information. Why ask a question, when it is already entailed by a super-ordinate question?

If we drop the assumption that agents are perfect reasoners, we can solve this dilemma. Now there are many ways to relax the standard rationality assumptions. For example, you could put agents under a time limit. Reasoning steps take effort and time, and after a while reasoning simply stops. This approach is taken in recent theories of belief revision for resource bounded agents (Wasserman 1999; Chopra and Parikh 1999).

A more general solution has to do with the notion of *awareness*. An agent is aware of a belief, when it's truth is actively considered. Here is an example to illustrate the difference. You know that Clinton is taller than me. After all, you believe that I am of average height and from television footage you know that Clinton is tall. So in none of the possible worlds that correspond to your information state, I am taller than Clinton. But it is only because the issue was brought up just now, that you realised this! You never actively considered the distinction. Nevertheless, you might say that you knew it all along.

The example suggests an important link between issues and awareness: a reason for actively raising an issue, for example by asking a question, is to make other participants aware of the alternative options. Awareness formally separates explicit from implicit belief (Fagin and Halpern 1988). We believe that such an awareness notion can be derived from the issue structure proposed in this chapter. We will say that an agent is *aware* of a proposition when it contrasts the proposition with alternatives. In other words, the proposition corresponds to one of the alternative solutions to an issue the agent is considering. Note that this definition requires that the partition covers the whole logical space, not just the epistemicly accessible worlds of an agent. Please note that awareness is strongly related to salience. Being aware or not, or being an issue or not are the bipolar extremes of a scale of salience.

In chapter 4 we discuss a logic of explicit and implicit belief based on Fagin and Halpern's proposal, combined with issues to account for awareness. Awareness is a very diverse notion. Issues are not going to solve all the aspects of the deduction problem. In particular, the aspects related to informativeness of issues themselves remains a problem. For that we still need to resort to other non-logical notions of dependency and usefulness for a task. But at least one of the reasons that participants want to raise an issue, is to make the others aware of it.

Chapter 4

Transaction

This chapter discusses a task model of dialogue systems for inquiry and transaction. Extending the notion of inquiry developed in the previous chapter, a formal account of agents in an inquiry and transaction dialogue can be given. Agents are motivated by desires and preferences to act. New information on preferences causes agents to update their preference order. Based on preferences and on the available plan recipes for action, agents deliberate and select a feasible goal. Agreements and commitments can be modelled as the adoption of joint plans for future action.

4.1 Introduction

Dialogue participants hope to achieve something by engaging in dialogue. For many types of dialogue, participants have a particular goal or want to achieving a particular task. Goals guide and structure the progress of a dialogue. Relative to a goal we can assess the success of a participant engaging in dialogue. In chapter 1 we suggested that the usability of a dialogue system is directly correlated with its effectiveness and efficiency in helping the user to achieve that goal. In this chapter we characterise under what circumstances a goal has been achieved. This allows us to formulate measures of effectiveness, and of efficiency: was it indeed achieved and in how many steps was it achieved?

As always in this thesis, we concentrate on dialogues for inquiry and transaction. The underlying activity or task is that of a negotiation. Negotiations share certain characteristics that can be fruitfully studied in general. However, for task models to be successful, they have to be fleshed out for each specific application. In the course of developing dialogue systems one thing we learned is that details do matter. Here is an extended example to illustrate this point.

Compare the SCHISMA system for theatre information and booking (van der Hoeven et al. 1995) with a spoken dialogue system for ticket reservation for movies (van den Berk 1997). The tasks seem to be extremely similar at first glance. Both concern ticket reservation and the exchange of information about performances. Both the databases are centred around the concept of a performance, which can be selected on the basis of title, date, genre and the director and actors involved. However, differences between the activity of going to the movies and going to the theatre and differences in the context of use and the role of the system, result in a different dialogue structure. A movie usually plays a number of weeks consecutively. Therefore people go to the movies when it suits them.

Most people who call the service already know the film they want to reserve for. They typically inquire about the time the film starts, about the length of the film, or about the price. Otherwise, they call to find out if something 'is on tonight' and only make a reservation when that is needed, usually in the weekend. This behaviour can be influenced by the price of the performance: on Monday and Tuesday people get a discount. By contrast, a theatre, dance or music performance, especially in a provincial town like Enschede, plays only one or two nights in a row. Theatre and concert schedules are known months in advance so decisions to go are made relatively early. The SCHISMA service is mainly used to get information about the schedule and about the content of performances. This 'browsing behaviour' is typical. Discounts try to attract specific groups in the audience: students, elderly or people on social benefit. Other differences result from the fact that SCHISMA is a keyboard-based system accessible over the Internet, whereas the movie theatre is a spoken dialogue system to be used over the telephone. Both systems have not been systematically tested with real users, but only with 'in-house' volunteers. In a real evaluation, we would predict that the distribution of dialogue act types and the initiative handling will differ between the two systems. In particular, SCHISMA dialogues will show more browsing behaviour, and will be longer. Movie theatre dialogues will probably be more system directed and more standardised. In this chapter and the next, we study techniques to model these kinds of differences.

Lets start with an overview of the terminology. We often use the terms activity, task, goal, plan and action interchangeably, because they concern similar aspects of an interaction. But they are not identical. By activity type, we mean the structural conventions of a certain social activity like buying and selling. A task describes what participants in different roles do in a given activity type, for example they negotiate or enquire. A goal is an intended future state of an agent, or of several agents in the case of a joint goal. An intention just is a goal. An action is an intentional event carried out by some agent to try and reach a goal. A dialogue act is a special action: to make an utterance with a certain content and function. A dialogue act has a task-related function, the communicative intention or dialogue-act-related goal, and an interaction related function, to respond to, contribute to or initiate a dialogue exchange. By the phrase current goal or current task we usually refer to the perlocutionary goal or task that is to be achieved by an exchange of several utterances. A plan is a combination of several future actions to reach a goal. Actions and plans can be joint too. A reason for using these terms interchangeably is that tasks can be described in terms of goals, or else, in terms of actions or plans to achieve those goals. The notion of goal is viewed from the perspective of the agent; the notion of task is viewed from the outside. Tasks, goals and actions can be combined by the following types of relation: dominance, guarded choice and sequential and parallel composition. This compositional structure is what we call task structure.

4.1.1 Goals and Semantics

At several places we have stressed the importance of task structure on dialogue. To be able to capture the expectations and conventional behaviour of users in dialogue, one needs a model of the task. We argue that the sort of task modelling and activity analysis that is needed in practical dialogue systems, is needed in natural language semantics in general. What we say depends on what we are doing. We adhere to an *activity based* approach to natural language semantics and pragmatics (Allwood 1995). This means that the stage of the task and the public goals of dialogue participants form an important parameter to the natural language interpretation and generation process. The task or goal parameter is an aspect of the dialogue context along with the dialogue history, the roles and background information of the dialogue participants and the physical setting.

That goals are needed as a parameter to semantics has been argued extensively by Ginzburg (1995, 1996), Carlson (1983) and Boër and Lycan (1986) among others. It is best illustrated by examples from the semantics of questions and answers, although it certainly applies to other types of utterance as well. The meaning of a question is often associated with its answerhood conditions: the conditions under which the question counts as being answered. In each of the following examples there is a difference between an appropriate answer in the *a* case and in the *b* case. The goal parameter is supposed to model just that difference.

The following example is similar to the example in chapter 3 that contrasted the concept of time of a television producer with that of a scientist. The question in (92) is appropriately answered by answer *a*, when the asker is a traveller with the apparent goal of catching the train to London. To such a traveller an answer like *b* would be inappropriate. However, answers with the precision of the *b* answer are usually appropriate in scientific contexts. The underlying goal of the scientist requires such precision.

(92) A: When is the train for London leaving?
a. B: in 2 minutes, platform 4.
b. B: at 14 hours, 02 minutes, 35 seconds and 11 hundreds of a second.

Answers *a* and *b* differ in their precision. But answer *a* also differs from *b* in another respect. Answer *a* volunteers some additional information, "platform 4", that is strictly speaking not asked for. Moreover, the *a* answer is phrased relative to the current time: "in 2 minutes". Apparently this will make it easier to deduce the consequences of the answer with respect to the askers goal of catching the train. When respondents recognise the askers apparent goal, they will often adjust the presentation of their answers and add other helpful information; information that was not even asked for.

Questions and answers are related to propositional attitudes in a systematic way, by 'disquotation under know' (Ginzburg 1995). The idea is that an appropriate answer to a question can be translated into a statement about knowledge. So attitude reports like (94) are just as dependent on the goal parameter as the question-answer pair in (93). The *a*-versions would be appropriate at a Dutch school exam, but inappropriate at a fund-raising party. There we would expect something like the *b*-versions.

(93) A: Who is Bill Clinton?
a. B: He is the president of the United States.
b. B: He is that man over there (pointing).

(94) I know who Bill Clinton is.
a. He is the president of the United States.
b. He is that man over there (pointing).

Actually examples (93) and (94) are used by Gerbrandy (1997) to support a similar point. He argues that the context dependency of question-answer pairs and 'knowing-who' can be modelled by parameterising the identity relation of the semantics. In epistemic modal logic, he argues, one could have different types of trans-world identity relations, to indicate when two objects are considered identical under different circumstances. For instance identity based on historical importance at school and physical identification features at a party. Technically this may be correct. However, what still has to be specified is how the identity relation can be found for a particular example; at what level of the framework it should be implemented and how it changes in the dialogue. Therefore we believe that this type of context dependency can be more directly modelled with goals. Given the apparent goal of the dialogue participants, the appropriate identity relation, or rather the appropriate *identification scheme*, should usually follow. Our hypothesis is that the issue structure of chapter 3 can be used to generate an indiscernibility relation between objects. Only objects that differ with respect to the current issue are distinguished. That means, that if we manage to translate the distinctions needed by an underlying goal into an issue structure, in principle we get the identification relation for free.

The following example (95) demonstrates a similar type of context dependency, but this time without explicit questions or attitudes. It contrasts different conversational implicatures, resulting from different underlying athletic goals. By the Gricean maxim of quantity, to make one's contribution as strong as possible, and by the underlying scale of athletic achievements we can infer from *a* that Lewis did not jump further, and from *b* that Lewis did not run in less time. Athletes in general try to jump as far as possible but try to run in as little time as possible. So apparent goals help determine the scale that underlies a scalar implicature.

(95) a. Lewis jumps 7.02 meters (not more).
b. Lewis runs the 100 meters in 11.20 seconds (not less).

Questions like (96) below, that usually expect a *mention-some* reading can be seen as questions that prefer a partial answer. The 'pragmatically best' answer mentions the nearest coffee machine, as in *a*, not a full list, as in *b*. Just like precision in example (92), partiality can be explained in terms of the apparent underlying goal of the asker: to get coffee, and to get it fast. Goals often suggest such optimality: the quickest way, the shortest route, the best beer. Still, for a coffee machine engineer that has come to do the monthly check-up, a list like *b* is more appropriate.

(96) A: Where do I find a coffee machine in this building?
a. B: At the end of the corridor.
b. B: At the end of the corridor, on the 3rd and 4th floor and opposite the library.

To remind ourselves that dialogue often involves social interaction where cooperativity means more than merely fulfilling the Gricean maxims, here is an example from Ginzburg based on a scene in Jim Jarmusch' film *Night on Earth*. The question "Do you know where you are?" in example (97) is properly answered by "Yes, I am in Helsinki" for the businessman in *a*. A compassionate taxi driver in *b* would not be satisfied by such an answer from a drunk in a suburb, given the apparent goal to get home safely and the fact that it is freezing cold. Apparently, the drunk does not know enough of where he is, to be able to get home.

(97) a. B is a businessman who has just arrived at Helsinki Airport.
b. B is a drunk who has just collapsed in a Helsinki suburb.
A: Do you know where you are?
B: Yes, I am in Helsinki.

These examples exhibit a specific type of context dependency, a dependency on the apparent underlying goals of the dialogue participants, in particular the speaker. We believe that the partition framework of chapter 3 can deal with examples of context-dependent precision, identification and partiality. To determine the pragmatic conditions that modify the partition we will need a model of the apparent goal and a model of the available actions to achieve it. The goal directly affects the currently relevant issues. For instance, goals that suggest optimality criteria narrow down the full range of possible answers to the 'pragmatically best' ones. This explains the mention-some readings and other cases where partial answers are sufficient. In the scales example (95), something else is the case. Here it is the preference order that one could assume among possible athletic achievements – answers to the question what was achieved – that can be turned upside down by the underlying goal. The preference triggered by "runs" is that a minimal number is optimal; the verb "jumps" suggests that maximal numbers are preferred. In this example the recognition of underlying goals and the corresponding scales is strongly related to the verb. But in other examples, like (92) or (97), goal recognition is less conventional. We need a general framework for modelling cooperative agents. Obviously, it would be impossible to formulate goals with sufficient detail to account for all of the examples presented above. What can be done and what is done frequently in computer science is to limit oneself to a single task domain and develop an account of the expected goals of dialogue participants within the domain. Developing the goal or task model is part of the art of building applications, just like the acquisition of a conceptual or information model.

4.1.2 Plans and Goals

Plans and goals have been used extensively in natural language processing. The link between goals or intentions, actions and language is naturally made through the theory of speech acts (Austin 1962; Searle 1969). A speech act is the deliberate act of making an utterance with a form that conveys a certain intention: to transmit a certain content and function. Like all acts, speech acts can be characterised by the intended effects, and by conditions that specify when an action is applicable, or when an action can be successfully carried out. Therefore speech acts form a natural starting point for computational models of dialogue processing (e.g. Cohen and Perrault 1979; Allen and Perrault 1980). When the task-related function of a speech act, to exchange information, is extended with the effect on the dialogue itself, the theory of speech acts can be developed into a theory of *dialogue acts*. Dialogue acts explicitly include the so-called *dialogue control acts*, such as greetings and acknowledgements (Allwood et al. 1992; Bunt 1989). Plan based approaches to natural language understanding argue that the key to understanding utterances, is to recognise the underlying intention of the speaker. This can be done by reasoning backwards from the observed behaviour to the plan that must have given rise to that behaviour (Litman and Allen 1987; Carberry 1990).

In natural language generation plans are used to determine the structure and order of utterances in a text. For instance, Moore and Paris (1993) prefer the use of goals and plans over mere schemata for instructional dialogue and context-sensitive help systems. Schemata and templates are considered less flexible and adaptive to different circumstances (Reiter and Dale 1997). Goals in principle make it possible to provide relevant additional information or to leave out redundant information, compare example (92).

Plans and goals are especially useful to repair misunderstanding, it is argued. Having a goal to motivate the utterance makes it easier to back-track and try a different utterance (Ardissono et al. 1998). Similar motivation is reported by Bretier and Sadek (1997). On the other hand, a systematically designed set of schemata can be seen as a library of 'compiled-out' plans for future action. van Deemter et al. (1999) argue that practical systems that make use of templates and schemata can be just as flexible and theoretically sound as plan-based systems.

Most plan based approaches share an ontology inspired by the STRIPS framework (Fikes and Nilsson 1971): there is a set of states with an initial state and one or more goal states and there are operators (acts or actions) that transform one state into another. States are represented by sets of facts, sentences in some logical language. Actions are characterised by a *precondition* that must be true for the action to be successful and a *postcondition* consisting of an *add-list* of facts that will become true after execution and a *delete-list* of facts that will no longer be true after execution. It is assumed that states and facts that are not mentioned remain unchanged. A plan is a well-formed sequence of actions: the preconditions of an action match the postconditions of previous actions. Think of a plan as a path through the state-space.

In this framework, the goal of a particular plan or action equals the description of its intended effect. In order to separate intended effects from actual effects, more fine-grained distinctions can be made. In the model proposed by Carberry (1990) among others, an action is characterised by a number of characteristics. *Applicability conditions* indicate when action is applicable at all. *Preconditions* specify what needs to be the case for successful completion. The *effect* indicates when the action is successfully completed. The *failure conditions* indicate what must remain true after a non-successful execution. Together the intended effect and the failure condition form the postconditions of an action. The *body* of a plan describes a combination of sub-actions to be carried out. Only when the precondition is true or can be made true by sub-actions, the plan will be successful: the intended effect becomes actual. The body of an act may contain complex sub-actions. This leads to a recursive structure. We assume a set of basic actions from which complex actions or plans can be built up. For applications in information-centred task domains, like automated travel inquiry systems or automated booking systems, dialogue acts like *inform*, *request* and *confirm* would be the typical basic actions.

In figure 4.1 a composite action is shown from the SCHISMA system. For each of the preconditions, a sub-action can be activated to make that precondition true. This version is similar to the use case analysis of chapter 1, with one major exception: the preconditions listed here were treated as sub-actions right away in the use case. In the use case we analysed the complete activity of making a reservation. We focussed on the steps needed to complete this complex action. Here we analysed only the intended effects of the final transaction stage: under what conditions would the mutual goal of user and system to make a reservation be achieved?

This example shows, that when the goal of the user is indeed to reserve tickets, and the system cooperatively adopts this goal too, a number of questions naturally arise: which performance, how many tickets, how many tickets with discount, are enough seats available, what is the overall price and is that agreed by the user? Such questions provide a background against which highly context-dependent expressions can be resolved. For instance, an utterance like (98) would be meaningless without such a background.

(98) 3 for tonight, please!

Reservation	
applicability conditions	wish to make a reservation indicated by user
preconditions	unique performance known by system number of tickets known by system number of tickets with discount known by system name and address of user known by system right number of seats available overall costs known by user transaction confirmed by user
intended effect	user and system agreed on performance, tickets and costs seats marked as reserved for user user committed to collect tickets and pay
failure conditions	no seats marked as reserved; no commitments made by user
body	utter verbal realization of questions raised by preconditions adjust reservation database

Figure 4.1: Reservation in SCHISMA system

Note that the earlier phenomena of precision, partiality and identification reappear in this limited task domain. Mentioning "tonight" might be enough information to identify a unique performance. Yet more precise information may be needed. This depends on the particular schedule of the theatre for that week. Since users are not expected to know the theatre-schedule, the system should ask for additional identification features when needed. This example also illustrates that the notions of goals and questions are very much related. Questions are asked in order to achieve goals or sub-goals. But some of the effects of goals or sub-goals can in turn be modelled at the information level. Like questions, goals structure information: they generate issues for consideration.

To get such an account to work one needs a planning module with algorithms to select appropriate plans to achieve a given goal. For dialogue systems, the dialogue manager component can be seen as such a planning module: it decides what to do next. In general, a dialogue system can be seen as an *autonomous agent*: it receives input from the world in the form of user utterances and it acts upon the world by uttering responses or manipulating a database. An agent is called autonomous when its behaviour is not completely controlled or determined by the input it receives. Its decisions to take action are partly based on its own goals.

An agent does not have to recalculate what to do in every possible circumstance. Experienced agents have libraries of so-called *plan recipes*. These are schemata that specify what steps need to be taken, or what intermediate goals need to be achieved in order to reach some final goal. The SCHISMA reservation example in figure 4.1 is a description of a recipe. Recipes can be underspecified with respect to the values of parameters of actions and with respect to the precise order and combination of sub-actions. And this is as it should be. Dialogue is opportunistic. You can't plan everything ahead. The details have to be filled in as you go along. This is the principle of *partial planning* (Grosz and Kraus 1996).

An account of one autonomous agent is therefore not enough. In chapter 2 we argued for an analysis of dialogue as a series of coordinated actions, at various linguistic levels. In principle, for every joint action the participatory actions and time slots need to be assigned to agents. Sometimes negotiation about the distribution of sub-tasks is needed. Our model is rich enough to account for such negotiation behaviour. But usually joint actions are highly conventionalised and the distribution of sub-actions is based on roles. Recipes are 'compiled-out' plans for joint action, that have proven their worth. Every competent speaker has a repertoire of interaction recipes. For natural language interaction, it will turn out that the smallest recipes for action that can still be called joint, are in fact *dialogue games* in the sense of (Mann 1988). This insight resolves the dispute whether to use a planning framework or a dialogue grammar-based framework to describe dialogue structure, e.g. (Jönsson 1993, Ch 4.). Both the intentional aspect of the plans and goals paradigm, and the conventional aspect of game rules are needed.

4.1.3 Overview

This chapter is structured as follows.

In the first section we sketch a model of a particular type of activity that underlies many dialogues: negotiation. As we stressed in chapter 2, agents often need to cooperate in order to achieve something. A common form of cooperation is based on the outcome of negotiations between agents. We study a simple kind of negotiation, namely negotiation about buying and selling a certain product. A negotiation dialogue consists of a number of phases: opening, information exchange, proposing, confirmation and closing. The purpose of this chapter is to define a semantics for the basic dialogue acts that occur in these phases. In particular we study greeting, question and answer, inform and acknowledge, propose and accept or reject, confirm and bye. Our semantics is based on the metaphor of a *negotiation space*: the set of objects under consideration. The space is structured by an issue structure, as developed in chapter 3. Making a proposal restricts the negotiation space. Making a counter proposal overrules earlier proposals, or brings up new issues for negotiation.

In section 4.3 we explain the basic ontology for individual agents. Agents have information about the current state of the world, about recipes for action and about the likely effects of those actions. Agents are motivated to act by desires and preferences. Based on their preferences and their abilities to act, they determine a plan to reach a particular goal. The goal influences and drives the agents behaviour. New information or unexpected events in the outside world may cause the agent to reconsider its goal. How can we model such agents? We follow the influential BDI-paradigm in which agents are modelled by three basic notions: belief, desire and intention.

Beliefs are modelled by the information states of the previous chapter, along with the issues in it. We define a notion of awareness based on issues. The definitions are given in section 4.4. Desires and preferences and the way they influence planning and goals, are studied in section 4.5. For each agent we postulate a preference order over possible worlds. Preference orders respect the so-called conjunction expansion principle (von Wright 1963): an agent prefers φ to ψ , when it prefers $\varphi \wedge \neg\psi$-worlds to $\neg\varphi \wedge \psi$-worlds. We suggest how an agent may update its preference order as new information about preferences becomes available. Intentions form the last important aspect of the BDI paradigm. In section 4.6 we explain that intentions are in fact goals the agent has committed itself to. In section 4.6.2 we give an analysis of commitment based on Wooldridge and Jennings (1999). This section remains rather sketchy. Section 4.7 concludes with a discussion of the contributions made by this chapter and lists topics for further research.

4.2 Negotiation

If we want to follow the adage that 'meaning is use', we have to show how our analysis of utterances in a dialogue depends on the particular type of *activity* in which they are used (Allwood 1995). Here and in chapter 5, we study a particularly important activity type: *negotiation*. Formal models of negotiation can be applied in a description of human dialogue (Walker 1996b; Di Eugenio et al. 1998), in the design and verification of automatic dialogue systems (Alexandersson 1996) and in a conceptualisation of the social level of agent-based systems (Rosenschein and Zlotkin 1994; Wooldridge and Jennings 1999).

Our model of the basic negotiation actions is based on the account of *inquiry* developed in chapter 3. Like inquiry, negotiation can be modelled at a semantic level as a constant process of raising and resolving issues, but this time the issues have to do with the attitudes of the agents towards the transaction that is being negotiated. A *transaction* typically involves an agreement between parties about some future action, and the commitment of each party to perform their part. In the case of distant sales for example, the supplier must deliver the product; the customer should collect the product and pay the price. Therefore the actual closure of a transaction involves a sequence of confirmation steps. For many transactions, the confirmation is represented by some act or token that conventionally signals the importance of the occasion. On a cattle market, the sale is ratified by slapping hands; a property deal ends with a signature on a contract. This fits with a view of agreement as a grounding activity in which the common ground is derived from a publicly accessible *basis* (Lewis 1969). Having an external representation such as a signature makes it more difficult to misapprehend the agreement.

The term negotiation suggests some conflict of interests between participants,the needs to be reconciled. So full cooperativity cannot be assumed: an agent does not necessarily adopt another agent's goal. As argued in chapter 2, we can continue to assume cooperativity at the interaction level. Even agents that serve opposing interests at the task level, can still be expected to be sincere and cooperative at the interaction level. If the supplier gets caught in a lie, he loses credibility as a supplier. For a customer it is not beneficial to lie about what she wants. So it is counterproductive to lie.

Such generalisations can be made in a systematic way (Rosenschein and Zlotkin 1994). They distinguish three types of negotiation domains. (i) In *task-oriented* domains agents have to coordinate a division of labour to minimise the costs. Tasks are independent; actions have no side-effects. (ii) In *state-based* domains agents collaborate some intended state of affairs: a goal. Here negotiation involves possible side-effects on actions of other agents. (iii) For *worth-based* domains the relative desirability of a goal is no longer a binary value. Some goals are more preferred than others. Both the details of the collaboration and the expected worth are a topic of deliberation and negotiation.

For each type of domain, a convention indicates what is considered a good deal, and when it is reached. Usually, a deal is optimal when it can not be made any better for any of the agents, without making it worse for some others. Interesting properties of negotiation protocols in specific domains are the *stability*, whether agents have any reason to deviate, *simplicity*, how difficult it is to conceptualise and program it, *effectiveness*, does it converge on a deal, and *efficiency*, how many steps does it take on average to converge? Exploring the properties of protocols in domains with different characteristics is part of research into multi-agent systems. Such properties play a similar role in system development, as the usability related properties discussed in chapter 1.

What kind of domain is exemplified by ticket reservations? In most reservation scenarios the relation between price and features of a ticket is non-negotiable. In the SCHISMA case, for each performance features like date, artist and genre are fixed, and so are the conventions that determine a price, like discounts and ranks of seats. You can't bargain at your local theatre. Instead, ticket reservation dialogues can be seen as a joint search towards the most optimal transaction given the client's preferences and the supplier's abilities. Thus ticket reservation concerns a worth-oriented domain. It concerns the preferences and possibilities of the participants. But there are a number of simplifying assumptions. Agents are cooperative at the level of information exchange: they can be assumed to be sincere and are generally reliable about their area of expertise: the client about her wishes, the ticket seller about tickets. From a linguistic point of view, the way the joint search is carried out shares many characteristics with true negotiation dialogues, in particular where it concerns the grounding of proposals and suggestions. For this reason we start from accounts developed for true negotiation, such as the appointment scheduling dialogues of the VERBMOBIL project (Alexandersson 1996).

For our purposes, a negotiation can be characterised as an exchange of information about the participants' attitudes towards the performance of some future joint action, which is supposed to lead to an agreement. Obviously, the final agreement depends on the choices made in the negotiation process. To model the choices we use the metaphor of a *negotiation space*. It contains objects that can be subject of a transaction. Objects in the negotiation space can be pictured as vectors in a multi-dimensional space: each attribute constitutes a dimension. Information about the future transaction thus corresponds to a region of the negotiation space. Adding information, for instance by making a proposal, means a reduction of the region. Raising an issue, for instance by asking a question or making a counter proposal increases the dimensionality of the region. In this way new distinctions between objects become relevant.

The negotiation space does not only contain information about available transaction objects, but also about the attitudes of participants towards them. Many attributes of objects carry an intrinsic value, which suggests a preference order or scale. The most obvious example is price: most customers prefer a lower price. Many preferences are not determined beforehand, or differ between participants. In the process of negotiation, participants communicate their preferences. To model attitudes of agents we define a multi-modal logic. In particular, the framework accounts for the *beliefs*, *preferences*, and *goals* of agents, as well as *commitments*. Each of these attitudes can be modelled by modal operators with corresponding accessibility relations which partly correspond to the information states of chapter 3. So technically, think of the negotiation space as a structured set of possible worlds.

4.2.1 Dialogue Acts and Exchanges

A dialogue act is characterised by a semantic content and a communicative function. For the dialogue acts considered for negotiation, the content is either a proposition, an issue, a description of an action or empty, depending on whether the act is an interrogative, an assertive, a directive or a dialogue control act. The communicative function combines two main aspects: task related aspects and interaction related aspects. Because certain social activities involve a combination of task related information exchange with a characteristic interaction pattern, task structure and interaction patterns may be entangled.

Task related aspects have to do with the exchange of information about the attitudes of the dialogue participants towards some future action. Interaction related aspects have to do with the role of the dialogue act in the current dialogue game. A dialogue act can be either an initiative, a response or be neutral. An initiative requires a particular response from the other dialogue participants; a response, such as an answer or an acceptance, expects that a particular initiative was raised. Otherwise, the dialogue is not coherent. A single initiative-response unit or *exchange pattern* constitutes the smallest kind of dialogue game.

For example: a reservation involves a confirmation of the reservation details, in order to make the user committed to pick up the tickets and pay the price. To achieve this the system can initiate a particular dialogue game: a confirmation exchange. Like most exchanges it consists of an initiative move, followed by a response move. In this case the initiative consists of a request for confirmation; the response consists of a positive or negative reply, or else of another move which implies a negative reply. In this example, the task related function of the confirmation exchange is to establish agreement and make the participants committed to the outcome. In case of success, the next sub-task will be to terminate the interaction, which can be achieved by means of a closing exchange (bye – bye). In case of a failure, the next sub-task will be to renegotiate the current proposal. The interaction related function has to do with the initiative-response structure. It specifies that the person initiating the exchange keeps the initiative, and that after successful completion the topic of confirmation is closed.

In chapter 2 we suggested a layered account of communicative actions in dialogue. At the top we find the task level. The task can be modelled by a complex joint action. The steps that need to be taken to achieve the task can be specified in terms of their effects. Intended effects are simply goals, which are usually joint goals: to establish contact, for example, or to reach agreement. So plans to achieve the task will be joint plans. The typical components of these joint plans are the smallest units of interaction which can still be called joint: initiative-response units or exchange patterns. And the intended effect of an exchange pattern corresponds to the joint goals at one level higher. Since exchange patterns are recipes for joint actions, they consist of participatory actions: actions that are ultimately to be carried out by a single agent. These basic participatory actions are dialogue acts.

4.2.2 Negotiation Phases

We distinguish five negotiation phases, shown in figure 4.2. The transitions, depicted by arrows, are the result of joint actions. So usually each transition in picture 4.2 corresponds to an exchange of several dialogue acts. The nodes correspond to states of affairs that successfully end a phase. After the opening stage we reach a state in which there is mutual contact. After the information stage we reach a state in which there is agreement on the parameters of the negotiation space. We now give an informal description of the negotiation phases in terms of the negotiation space. Further analysis of dialogue acts and exchange patterns follows in chapter 5. Please note that the acts that are mentioned in one phase, can very well occur in another phase as well. So information exchange acts can be part of the proposing stage as well.

Except for the opening and closing stages, any result of a stage can be undone (back arrows). Because of corrections or clarifications no progress can be made at all (reflexive

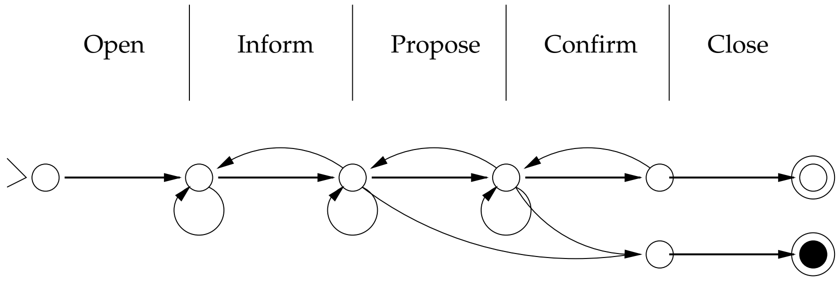


Figure 4.2: Phases in negotiation dialogue

arrows). A stage may not be skipped. If it appears that a phase is skipped – there are no utterances to exemplify it – we assume that the intended effect is conventionally determined. By this 'conventional skipping' we can explain simple interactions as an instance of the more general model. Take for example the STEMSTEM (VotebyVoice) application that is discussed in chapter 6. The activity of casting one's vote can be described as an opening, followed by a proposing stage to vote for a candidate of a particular party, followed by a combined confirmation and closure. The information phase is skipped by convention. Or to take another familiar example, a public transport information dialogue can be analysed as an opening, an information exchange about a train journey, and a closure. The corresponding proposal and confirmation stages, to request and buy a ticket for that journey, cannot be conducted by telephone in the current service.

Opening

During the *opening* phase participants exchange greetings to acknowledge each other as negotiation parties; they establish contact and find a common ground of social and linguistic conventions for interaction. In this opening or entry phase (see chapter 2) participants identify the activity type, the roles of the participants and subsequently their tasks. The general topic of conversation and the corresponding basic distinctions among objects in the domain is established too. In this way a negotiation space is set up. In formal agent communication languages, this is typically done by reference to some standardised knowledge representation language (e.g. LOOM) and by reference to a common ontology (e.g. WordNet). So on the basis of the opening sequence agents can decide whether they will be able to understand each other at all, or not. In our model of the negotiation space, we could compare this step to the establishment of a particular vocabulary of predicates, constants and variables, which constitutes a signature. Since different signatures highlight different distinctions, they generate different negotiation spaces.

Information

In the *information* phase the parameters of the negotiation space are set. Participants agree on the relevant attributes and preference orders, often implicitly. For instance, in the appointment scheduling dialogues of the VERBMOBIL project (Kay et al. 1994), the negotiation space takes the shape of a calendar. Usually, you meet people on weekdays. Weekends are assumed not to be an option. Or when buying an ice cream, you first enquire about the price, before deciding on the actual flavours. In this way the non-negotiable boundaries and dimensions of a negotiation space are established. Setting the parameters is largely dealt with at the beginning of a dialogue, but may be re-opened at a later stage. Consider for example an appointment scheduling dialogue with the following reply to a proposal to have a meeting on Friday: "Friday, that is difficult. What about the weekend?". Another possibility is that one of the participants starts making proposals, and only later realises that some of the parameters are not yet agreed on. This happens often in reservation dialogues, where a number of tickets is requested before the customer knows what is on offer. So the information phase is only logically prior to the proposing phase, not always temporally. The borderline between inquiring and proposing is not easy to draw. The best way to make the decision is by judging the capabilities of agents to change the circumstances. In a ticket reservation, the cashier is not in a position to change the theatre schedule. She can merely inform her customers about what is on offer; that sets the parameters. However, she is capable of selling tickets, so that is part of the proposing stage. Typical dialogue acts that occur in the information phase are questions and answers, assertions and acceptances, or informs and acknowledgements.

Proposing

A *proposal* is a way of reducing the negotiation space. A proposal is either *accepted* or *rejected*. A failure to accept is seen as a rejection. A rejection may lead to a counter proposal, which is simply a proposal that overrules the earlier proposal, or to a redefinition of the negotiation space. We distinguish complete proposals, which uniquely define a transaction object, from partial proposals, which leave some negotiation space for discussion.

A proposal is more than information. It generates a commitment if it gets accepted. Proposals are ordered on a scale. It is implicitly understood that each proposal you make is the strongest one possible. If you make a proposal, you implicitly make all weaker proposals too. The relative strength of a proposal is based on various underlying scales, such as the quality and price of the transaction object that is discussed. In case the interests of the participants are opposed, the scale of strength of proposals equals the reversed preference order. A less preferred deal makes a stronger proposal. Participants do not know of each other exactly how their preferences lie. For this reason the most basic negotiation strategy starts with participants naming their most preferred option. After that they gradually increase the strength of their proposals, reducing the negotiation space until they meet in the middle. Thus, an active proposal of agent *a* is characterised as the least preferred option for *a* that is still acceptable. Active proposals function like issues: they further structure the negotiation.

Requests and suggestions are exactly like a proposal in that they try to get somebody to do something: they are *directive* dialogue acts. In a proposal the speaker is the projected actor; in a request it is the other participant. You request something for yourself. Suggestions sometimes leave the decision who will carry out the action to be decided later, but usually mean to get the other participant to act. Suggestions and requests differ in how much they impose on the other agent. Suggestions leave the choice to the other agent. All three types of directives require some form of evaluation: an acceptance or a rejection. In human dialogue rejections are often phrased indirectly for reasons of politeness. An example of an indirect formulation of a rejection, is: "Friday, that is difficult." There is an account of politeness on the basis of the notion of *face* (Brown and Levinson 1987). Positive face has to do with the desire to be approved of. Negative face has to do with restrictions or inhibitions you put onto other people's liberty to act. The idea is that one should try to maintain face; increase positive face and reduce negative face, in this case by avoiding a downright 'no'. By saying that Friday is difficult, the speaker shifts the decision to the hearer, who is now at liberty to decide to respect the speaker's difficulty, or else to insist. The idea of negative face as a restriction of the liberty to act, can be naturally expressed using issues. In this case, blocks in the issue structure correspond to the possible alternative courses of action that the agent can take. The amount of negative face of an utterance is proportional to the number of alternative blocks that are eliminated by the utterance. So we find another important scale underlying the dialogue: a scale of positive and negative face, where negative face is roughly proportional to the number of alternative courses of action left for the other participant. Politeness conventions vary enormously between countries and communities. Designers of dialogue systems must be aware of such cultural differences. For example, an American 'cheesy' attitude did not work at all in France, where people expect professional distance (Lynn Chase, p.c.).

Confirmation

After acceptance, the consequences of a transaction are stated once more for *confirmation*. In this stage parties have a last chance to correct misunderstandings. Confirmations are initiated by questions or suggestions, and expect either a positive response (confirmation) or a negative response (disconfirm). A disconfirm results in an unsuccessful end of the negotiation, or else in a re-negotiation of active proposals, or a re-setting of parameters. Requests for confirmations differ from requests for normal acknowledgements, in that they need to be responded to explicitly. The difference between a confirmation and an acceptance lies in its scope. An acceptance is always related to an active proposal, usually the latest. A confirmation concerns the total transaction, including possible 'hidden' consequences. The difference is illustrated by the use of a 'shopping basket' metaphor in current applications for home shopping over the Internet. Users browse the catalogue and add the items that they like to their 'shopping basket': a list of items to be purchased. Adding an item to the basket can be compared to a proposal. Acceptance of a proposal by the service provider is indicated by visual feedback: the item is displayed to be in the basket. At the end of a session the user 'proceeds to the check out' and confirms which items of the list are to be purchased, confirms the total price, agrees on the delivery details and method of payment and finally confirms the whole deal. Until this point, all actions are reversible. After this point, the user is bound.

Closure

After the confirmation or disconfirmation of a transaction, the interaction needs to be brought to an end. By exchanging greetings and thanks in the *closure* phase of an interaction participants re-establish themselves as reliable interaction partners for a later occasion. Closing must take place no matter if the current activity succeeded or failed. Even in non-cooperative activity types like arguments, walking out of a discussion is seen as extremely rude. A good example is the closing system prompt of the movie reservation system developed by (van den Berk 1997): "Veel plezier met ⟨*title*⟩!" (Enjoy ⟨*title*⟩!). The user gets some extra feedback. Hopefully the user gets the impression that the system 'cares' about the reservation. This may increase positive face.

4.2.3 Ice cream example

We now discuss a particular example of a negotiation dialogue. The situation is as follows. We are at Antonio's ice cream stand. Agent A (Antonio) is the seller and agent B (Betty) is the buyer. Betty has a particular taste: she likes fruit flavours such as strawberry combined with vanilla, or she likes chocolate with flavours like mocha or hazelnuts. She does not like to mix the two flavour groups. Antonio does not know this. There is no strawberry anymore, but this is not indicated. The sign for strawberry is still visible. Given the situation there are the following choices: a buyer buys one or more ice creams, consisting of two or three flavours, with or without whipped cream. There are five flavours: strawberry, vanilla, chocolate, mocha and hazelnut. This combination of choices generates a negotiation space.

(99)	B1:	(enters) Hi.	greet	open
	A1:	Hello!	greet	
	B2:	(views flavours) Mmm, looks nice.	assert	inform
	A2:	What would you like?	question	
	B3:	How much for three flavours?	question	
	A3:	2.50 for a 3 flavour-cone	answer ‖ suggest	
	B4:	All right, 3 then ... Mmmm	accept	
	B5:	I like vanilla and strawberry.	request	proposing
	A4:	Sorry,	apologise	
		we don't have strawberry.	inform ‖ reject	
	B6:	Oh.	acknowledge;	
		Vanilla and ...	request	
	A5:	(gets cone) Chocolate?	suggest	
	B7:	No, uh	reject	
	B8:	Yes, chocolate and mocha, and ...	accept; request	
	A6:	(moves towards chocolate)	accept	
	B9:	.. hazelnut.	request	
	A7:	Chocolate, mocha, hazelnut then?	suggest	confirm
	B10:	Ok.	accept	
	A8:	(fills cone) Cream?	propose	proposing
	B11:	No thanks.	reject	
	A9:	Anything else?	propose	
	B12:	No.	reject	
	A10:	(put cone in holder) That will be 2.50 then.	assert ‖ request	confirm
	B13:	(hands a 5-er)	accept	
	A11:	(hands back change) Here you are.	acknowledge	
	B14:	(picks up cone) All right, thanks.	thank	close
	A12:	Bye.	bye	
	B15:	(leaves) Bye.	bye	

The utterances of example (99) are annotated with the most prominent dialogue act type, as well as the phase, indicated on the right. If one utterance serves two functions at the same time, this is indicated by '‖'. If one utterance contains two consecutive dialogue acts, this is indicated by ';'. Obviously, this is one of many ways of annotating this particular example. Differences in the way conventions and implicit meanings are dealt with, may well arise. For example, we choose to annotate the physical actions of handing and receiving the money as part of the confirmation phase, but it is equally possible to regard the whole dialogue as a joint plan negotiation, and to mark steps A10 – B14 as an execution of the plan. Or to take another example, A10 is an assertion. Strictly speaking it doesn't add any new information; the price was already agreed in B3 – B4. By making the statement, the seller reminds the buyer of the price, and given buying-and-selling conventions, effectively requests the buyer to pay. A10 could have been phrased as a question with the same function: "2.50 then please?". Handing the money is an acceptance of the request.

These early annotations are not detailed enough to specify the full communicative function of an utterance. This will be provided by the theory in this chapter and the next. For example, in B6, the buyer keeps the vanilla part of the earlier request. To express this, the content of the request needs to be represented. Or in B7 and B8, the buyer first rejects the chocolate suggestion of A5, then hesitates, self-corrects and finally accepts it after all, further adding mocha to the request. To express this, the dialogue control functions of a hesitation, to keep the turn, and of a self-correction have to be specified. In A7 – B10 the early confirmation exchange obviously consists of two moves, an initiative by A, and a response by B. Only together they constitute a confirmation of that part of the transaction. Actually, one possible reason that the request in A10 is phrased as an assertion, listing the outcome of a calculation, is that this may be perceived as less direct. An indirect request decreases negative face.

In this chapter we will propose a logic for expressing the content of proposals, requests and suggestions, as well as the earlier repertoire of questions and assertions. The main motivation for such a formalisation is that the content of utterances and the update effects to the negotiation space are much easier to assess than the actual dialogue act type. The hypothesis is that dialogue act types can be reduced to updates of different sorts.

4.3 Agents

In the coming sections we present a account of agent systems in the tradition of (Bratman 1987; Grosz and Sidner 1990; Cohen and Levesque 1990; Singh 1994; Grosz and Kraus 1996) and others. For an overview of the literature in this field see Müller (1997). To model the attitudes of agents, a modal logic is used with modal operators for *belief*, *desire* and *intention*. Because these notions are taken to be the central ones, this paradigm is called the *BDI-account* of agents. It is inspired by the theory of rational action developed by Bratman (1987), which in turn relies on ideas from Davidson and on accounts of decision theory.

Davidson (1980) argues against a causal theory of action. He accepts that teleological explanations of actions in terms of reasons are just as valuable, although different, as explanations in terms of causal regularities and laws. Actions done by an agent can be distinguished from events that merely happen to an agent. An action is an event which can be described, under some perspective, as being *intentional*. So spilling the coffee is an action, because it can be described under some perspective as an attempt to drink coffee without spilling. Misreading a sign is an action; although the misreading was not intentional, it can be described as a reading event, and the reading was intentional. So Davidson makes it possible to explain rational action in terms of its underlying intentions: objectives, reasons and goals.

Decision theory and game theory[1] provide a quantitative theory for the optimal selection of actions under different circumstances. The changing circumstances are modelled as a game, in which players are allowed to make moves. Based on the state of the game, some moves are allowed and others are not. Of the moves that are allowed, some moves take more effort or costs to complete. Some moves are more beneficial to the agent than others. Now, an agent is called rational in its decisions when it selects those future actions that will produce the highest estimated benefits, against the least estimated efforts and costs. In other words, the agent should select the action with the highest *utility*. The utility of an action is a weighed sum of the expected benefits, minus the expected effort and costs. In decision theory researchers are interested in optimal decision strategies. A strategy is a set of decision rules for a particular game. An agent that possesses an algorithm for correctly estimating the utility of an action in all possible circumstances, potentially has a strategy of winning its game. Often no such algorithm exists, because there is not enough information on which to base the estimation. In these circumstances a set of heuristics may help. Heuristics are rules of thumb that give a crude approximation of a utility estimate. A strategy is optimal when it has heuristics that maximise the chance of winning, against minimal efforts and costs.

In the BDI-theory this model is made relative to the attitudes of an agent at a certain moment. Beliefs capture the agent's information about the current state of affairs. Desires indicate what counts as a success: what is beneficial to the agent. The crucial aspect of a BDI theory is that long-term intentions may suppress short-term incentives to act. The notion of an intention can be motivated by stability in the decision making process. The BDI-approach was implemented and used by Georgeff and Lansky (1987) in the control system of a lunar robot. A lunar robot finds itself in a fluctuating environment. Through sensors new information is constantly pouring in. A general strategy will be difficult to find in such cases. If we assume a simple decision theoretic approach, at each stage the most optimal course of action must therefore be decided. If the agent has no long-term goal, the optimal course of action will be selected on the basis of current information only. However small changes in information from the sensors, may lead to big changes in the decision making. Think of shadows that suggest a rock from a certain point of view. If the robot would go left around a rock, it might go right at the next moment, resulting in an unstable and inefficient behaviour pattern. Intentions stabilise the behaviour of an agent. Once an agent has decided on a particular course of action, this decision itself becomes input to the decision making process. In other words: the intention of completing the plan influences the behaviour of an agent at lower decision making levels. Only when dramatic changes occur the plan may be re-evaluated and a different course of action may be chosen.

We agree with Hobbs (1990) and Bratman (1987) that intentions can be modelled by goals. Intentions play a role in a philosophical theory of rational action; goals play the corresponding role in a programming theory of planning and action. To model intention-in-action (Searle 1983) the technical machinery that is available for plans and goals is of the right type. In particular, the problematic causal role of Searle's intention-in-action on the actual actions being carried out is not a problem for plans and goals. Part of the confusion comes from a failure to see the conceptual difference between desires, preferences and other motivational attitudes on the one hand, and intentions, commitments and goals

[1] Decision theory and game theory was pioneered by von Neumann and Morgenstern (1944). See Russell and Norvig (1995) for its application in computer science and artificial intelligence.

on the other. Desires do not have to be rational (Bell and Huang 1997). Desires may be impossible even or mutually conflicting. The consequences of desires may not be desired at all.

In the process of *deliberation* an agent determines the alternative possible actions or plans, weighs the alternatives according to their respective preference ranking and feasibility and selects the optimal plan as its goal. So the deliberation process is focussed on deciding which desires to pursue and which to ignore. Desires are potential goals; pursued desires have become goals. Once a rational agent has made a decision to pursue a certain desire only good reasons will make it reconsider. In transaction dialogues it is the task of the system to approximate the user's desires within the bounds of the possible. Users can have conflicting desires because they don't know the possibilities and consequences. So it is part of the task of the system to educate users about the possibilities and about the consequences of the actions they might select. In other words, deliberation is not necessarily a private process; it can be one of the topics of conversation. The TRAINS corpus contains dialogues that illustrate a more complex deliberation process between agents to achieve a common goal (Traum 1997; Allen et al. 1995).

Deliberation involves the assessment of the expected effort of possible actions relative to some adopted goal. This process depends on information and therefore possibly on the way the dialogue progresses. To model the joint deliberation process we assume an order among desired worlds. This will be a partial order. Some states are more desired than others; some are simply incomparable. At any stage there will be sets of mutually compatible most-preferred worlds: desires. These function as potential goals. We assume an estimation of the feasibility of a potential goal, based on the feasibility of the actions that make up a plan to reach it. Further we assume a planning mechanism that selects a feasible plan that maximises the balance between desirability and expected effort. When the planning mechanism does not have enough information to make a decision, for example when two conflicting desires are equally feasible, the planning mechanism generates an action to ask the user to resolve the conflict. By adopting a plan its intended effect becomes a goal: "Intention is choice with commitment" (Cohen and Levesque 1990). For single agents the commitment means that the agent will persist in trying to carry out the plan until it is completed, or until the plan has become infeasible; in that case the agent will try another plan. If the reason for adopting the goal is no longer true, for example because has been reached by some other means, the goal is withdrawn. For joint commitments between several agents, a commitment also involves a social dimension to keep the agreement.

4.3.1 Local and Global States

In chapter 3 we used an idealised setting for information exchange. We modelled a dialogue between an inquirer and an expert from the perspective of an observer. Each agent was assumed to be sincere in uttering, and credulous in adopting the content of utterances to their information states. In this chapter we have switched to a negotiation setting of buying-and-selling. Much of the idealisation however is still warranted. For example, it is in the interest of the seller to take over the buyer's preferences and act according to them, whenever that is possible. When it is not possible, he should say so. For buying-and-selling this cooperative principle may be called the 'your wish is my command'-

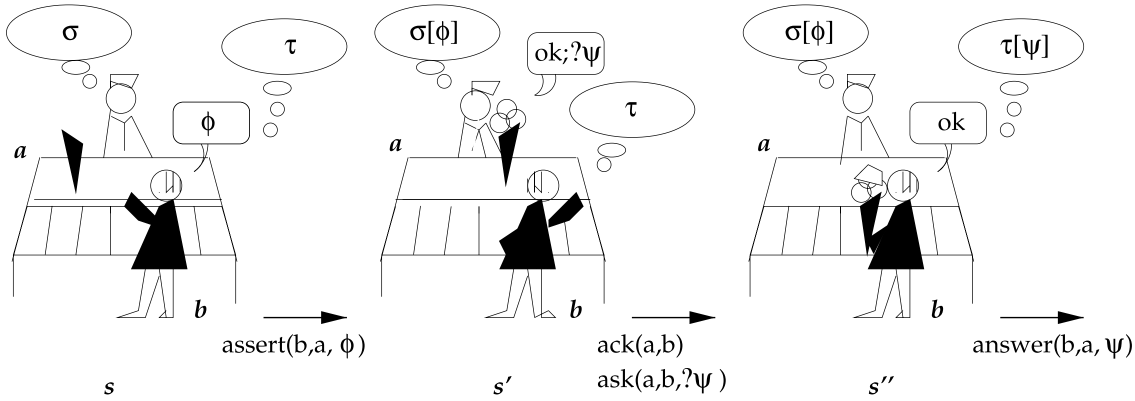


Figure 4.3: Global states with information state updates

principle[2]. The seller determines what is on offer. Because it is usually not in the seller's interest to give no for an answer, the buyer has reason to take the seller's word for it. We continue to model the dialogue from the point of view of an interested but objective observer. That means that information states represent the apparent beliefs and knowledge of agents, as can be assumed on the basis of the dialogue situation, including a record of what was said. Information states can thus be compared with the commitment slates of Hamblin (1970). If necessary, the common ground can be computed on the basis of information states. Typically, this is needed for commitments and agreements.

Information states are described as a structured set of possibilities, which can be updated with assertive and interrogative utterances. In principle, preferences and goals can also be dealt in the update framework as additional parts of information states. This requires among other things that agents have specific knowledge about the preconditions and effects of their actions. At each *global state of affairs* we model the configuration of the objects, agents and their *local information states*. One way to think of global states is as a large tuple $s = \langle w, \sigma_a, \sigma_b, ...\rangle$ with a possible world w to model the current environment and information states σ_a and σ_b for each agent a and b. We will not use tuples like this. Instead we apply possible worlds to index the information states. So $\sigma_{a,w}$ designates the information state of agent a at w.

Figure 4.3 shows three successive global states during a dialogue in an ice cream shop. Like any action, a dialogue act changes the global state. The effect of a dialogue act is reflected in changes of the information states of the individual agents. Let φ be a statement of absolute preference for some ice cream flavour, for instance $\varphi \equiv \text{good}_a(\textit{strawberry})$: "I would like strawberry, rather than other flavours". By asserting φ, agent b invites agent a to update its information state σ. In state s' agent a has in fact updated its information state with the preference statement φ, and acknowledges this to agent b. Furthermore, agent a asks b the yes/no question $?\psi$. Let for instance $?\psi \equiv ?\text{good}_a(\textit{cream})$ express the question "Do you want cream, or not?". In state s'' agent b answers positively to this question, so agent a can update its information state with ψ.

We now define the basic formal framework for the rest of the chapter. A multi agent system consists of a logical language L_M and a semantics in terms of a Kripke model M. We intend this description to be used in a characterisation of ideas on interaction. Therefore definitions may appear somewhat sketchy at places. We trust the reader can

[2] Dutch: "De klant is koning."

substitute his or her favourite theory of action. In particular, we ignore aspects of time. To express time explicitly we can postulate that states of affairs are ordered by a linear, or by a branching time order, as is done by Wooldridge and Jennings (1999) among others.

We concentrate first on the actions and attitudes of individual agents. Joint actions and attitudes like commitment are not yet dealt with in this framework. They can be defined on the basis of the actions and attitudes of individual agents. See section 4.6.

4.3.2 Logical Language

We need a logical language that can capture some aspects of actions in terms of their applicability conditions, preconditions and intended effects. If we want to model reasoning about plans and goals, we need expressions to describe them. That means that we need both procedural formulas to describe actions and plans α as well as proposition-type formulas φ to represent the content of dialogue acts and the pre- and postconditions of actions. A classic version of the combination of action and proposition-type formulas is propositional dynamic logic (Harel 1984). Dynamic logics can be seen as multi-modal logics defined over Kripke-frames (de Rijke 1993): multiple modal operators are modelled in the same framework[3]. Essentially, each atomic action *act* defines a transition relation between states that gives rise to a different modality $<act>$.

Since we focus on dialogue acts, most atomic actions will be like greet(a, b), ack(a, b), ask$(a, b, ?\varphi)$, inform(a, b, φ) or request$(a, b, !\varphi)$. Each action requires some agent a to be its actor. All interactive actions require an actor a and an addressee b. Based on the action type, actions have different types of semantic content. A greeting has no content apart from its being a greeting. An acknowledgement has no content either; it refers by default to the latest active contribution. A question has an interrogative content $?\varphi$ and an inform act has an assertive content $.\varphi$, where the '.' is usually dropped. Proposals, suggestions and requests have a *directive content* $!\varphi$, which initiates an act in order to achieve some future state of affairs.

The idea is that directive updates correspond to goals, just like assertives correspond to the attitudes of belief and 'knowledge that' and like interrogatives correspond to attitudes like wonder and 'knowledge who, what, why, how, where, when and whether'. Updates are the result of the corresponding dialogue acts, and are intended to affect the corresponding attitudes. These relations are summarised in figure 4.4. The effect of actions on agents' attitudes is what we have called the task-related part of the communicative function of a dialogue act. The interaction related functions of the dialogue acts are further discussed in chapter 5 along with the required response acts. For dialogue control acts similar relationships can be drawn. A greeting affects the attitude of contact. An exclamation like 'Look!' affects attention. But such aspects do not have a formal semantics.

There is a close relation between directives and statements of preference indicated above as the as the 'your wish is my command'-principle. So dialogue acts with a preferential complement often function as suggestions, proposals or requests. For each application domain different types of dialogue acts and complements are highlighted. For example, we neglected the aspects of authority, commands and permission. These would be crucial for some types of negotiation, but are not crucial for ticket reservation.

Procedural formulas are combined by '$\|$' for parallel composition, '$|$' for choice, ';' for se-

[3]This sense of multi-modality has nothing to do with the combined interaction modes of chapter 2.

Type	Acts	Content	Attitude
assertives	inform, assert, deny, answer, ...	$.\varphi$	belief, knowledge that
directives	suggest, request propose, ...	$!\varphi$	goal, commitment
interrogatives	ask, check, ...	$?\vec{x}.\varphi$	wonder, knowledge-wh

Figure 4.4: Dialogue act types, updates and corresponding attitudes

quential composition and '*' for repetition. The sequential composition of assertive and interrogative updates ';' is now redundant. We may redefine it in terms of a sequential composition. For any dialogue act type *dialogue_act*, agents a, b and contents φ, ψ, by definition $dialogue\_act(a, b, (\varphi; \psi)) \equiv dialogue\_act(a, b, \varphi); dialogue\_act(a, b, \psi)$. Note that this corresponds to our definition of utterance in chapter 2. An utterance is a linguistic unit that can be interpreted as a single dialogue act.

The content of assertives are simple propositions. They are combined by the usual connectives $\neg$, $\wedge$ and $\exists$. Moreover, we define a number of modal operators to account for the attitudes of agents. Statements about attitudes are just like other statements. For each agent a we have B_a and L_a for explicit and implicit belief respectively, K_a for 'knowledge that', A_a for awareness, W_a for wonder, $k^?_a$ for 'knowledge wh', pref_a for relative preference, good_a and bad_a for absolute preference and dispreference and G_a for goals. Interrogatives are formed from propositions just like in chapter 3. Again we use the notation $?\vec{x}.\varphi$. Directives are formed from propositions by the operator '!'.

In addition to assertives, interrogatives, directives and procedural formulas, there are special operators called *modes* and *projections*, that turn assertives, interrogatives and directives into basic acts, and actions into assertives.

A typical mode is the *test* action. A test is conventionally written as φ?, but will be written as test(φ) here to avoid confusion with interrogatives. It is an action that tests the current state of affairs for the truth of φ. We leave it merely for technical reasons. More practical modes are the dialogue acts of the form $dialogue\_act(a, b, \varphi)$ that are described above. A test can be seen as a yes/no-question directed at the environment, combined with an immediate answer in the form of a truth value. Tests are sometimes used to model observations. Note that observations are coloured by the current issue under investigation too.

A typical projection is an expression like $<\alpha>\varphi$ which indicates that it is possible to execute α, and that afterwards φ is the case. Its dual is the necessity operator $[\alpha]\varphi \equiv \neg<\alpha>\neg\varphi$, that indicates that φ holds after all successful executions of α. From these, we can derive more practical projections like pre(α) that indicates the weakest precondition for an action α to guarantee its postcondition post(α). In particular, $\text{pre}(\alpha) \equiv <\alpha>\top$ and $[\alpha]\text{post}(\alpha) \equiv \top$.

The signature of the logical language consists of a number of sets of symbols. We have predicate symbols *Pred*, individual constants *Cons*, variables *Var* and agent names *Agt*. In addition we have modal operators $Att = \{B, L, A, K, W, K^?, \text{pref}, \text{good}, \text{bad}, G\}$ to form attitude expressions. With the dialogue act types $Act = \{$ inform, assert, deny , answer, ask, check, suggest, request, propose, ack, accept, reject$\}$ we can form atomic actions.

Assertives, interrogatives and directives only occur as the content of an appropriate dialogue act or attitude.

To summarise, the language L_M is a multi-sorted language of dynamic logic, defined by mutual recursion on the basis of terms T, a language of actions $L_{\mathcal{A}}$, a language of propositions $L_{\mathcal{P}}$ and languages $L_.$, $L_?$ and $L_!$ to represent the content of assertive, interrogative and directive updates respectively. As in chapter 3, we usually drop the '.' sign and treat $L_.$ and $L_{\mathcal{P}}$ as identical.

Definition 45 (Syntax L_M)
Given $Pred = \{P, \ldots\}$, $Cons = \{c, a, b, \ldots\}$, $Var = \{x, \ldots\}$, $Agt \subseteq Cons$, define $L_M = L_{\mathcal{A}} \cup L_{\mathcal{P}}$

$$
\begin{array}{lll}
T: & t ::= & x \mid c, \\
L_{\mathcal{A}}: & \alpha ::= & \mathrm{inform}(t_1,t_2,\varphi) \mid \mathrm{assert}(t_1,t_2,\varphi) \mid \mathrm{deny}(t_1,t_2,\varphi) \mid \mathrm{answer}(t_1,t_2,\varphi) \mid \\
& & \mathrm{suggest}(t_1,t_2,\chi) \mid \mathrm{request}(t_1,t_2,\chi) \mid \mathrm{propose}(t_1,t_2,\chi) \mid \\
& & \mathrm{ask}(t_1,t_2,\zeta) \mid \mathrm{check}(t_1,t_2,\zeta) \mid \mathrm{ack}(t_1,t_2) \mid \mathrm{accept}(t_1,t_2) \mid \mathrm{reject}(t_1,t_2) \mid \\
& & \mathrm{test}(\varphi) \mid (\alpha \| \beta) \mid (\alpha;\beta) \mid (\alpha|\beta) \mid \alpha^*, \\
L_{\mathcal{P}}: & \varphi ::= & P(\vec{t}) \mid \neg\varphi \mid (\varphi \wedge \psi) \mid \exists x.\varphi \mid \;\mid <\alpha>\varphi \mid B_t\varphi \mid L_t\varphi \mid A_t\varphi \mid K_t\varphi \mid W_t\zeta \mid K^?_t\zeta \mid \\
& & \mathrm{pref}_t(\varphi,\psi) \mid \mathrm{good}_t\varphi \mid \mathrm{bad}_t\varphi \mid G_t\varphi, \\
L_!: & \chi ::= & !\varphi, \\
L_?: & \zeta ::= & ?\vec{x}.\varphi, \\
& & \text{where } t,t_1,t_2 \in T,\ \alpha,\beta \in L_{\mathcal{A}},\ \varphi,\psi \in L_{\mathcal{P}},\ \chi \in L_!,\ \zeta \in L_?.
\end{array}
$$

□

A number of other connectives can be defined by the standard equivalences: $(\varphi \to \psi) \equiv \neg(\varphi \wedge \neg\psi)$, $(\varphi \vee \psi) \equiv \neg(\neg\varphi \wedge \neg\psi)$, $\forall x.\varphi \equiv \neg\exists x.\neg\varphi$ and $[\alpha]\varphi \equiv \neg<\alpha>\neg\varphi$. Above we already defined $\mathrm{pre}(\alpha) \equiv <\alpha>\top$ and stipulated $[\alpha]\mathrm{post}(\alpha) \equiv \top$, where $\top \equiv (\neg p \wedge p)$ for some ground atomic $p \in \mathcal{P}$, and $\bot \equiv \neg\top$.

In some cases it is easier to restrict to a propositional language. In that case we use the same definitions as above, but restrict it to formulas composed of ground atomic formulas from the vocabulary $\mathcal{P} = \{P(c_1, \ldots, c_n) \mid P \in Pred, c_i \in Cons\ (0 \le i \le n)\}$. Also we need a definition of the alphabet $\mathcal{A}$ on the basis of which action formulas are defined. Let $\mathcal{A} = \{\mathrm{inform}(a,b,\varphi), \mathrm{assert}(a,b,\varphi), \mathrm{deny}(a,b,\varphi), \mathrm{answer}(a,b,\varphi), \mathrm{suggest}(a,b,\chi), \mathrm{request}(a,b,\chi), \mathrm{propose}(a,b,\chi), \mathrm{ask}(a,b,\zeta), \mathrm{check}(a,b,\zeta), \mathrm{ack}(a,b), \mathrm{accept}(a,b), \mathrm{reject}(a,b) \mid a,b \in Agt, \varphi \in L_{\mathcal{P}}, \chi \in L_!, \zeta \in L_?\}$. This list can be extended as the need arises. For a full specification of dialogue acts for negotiation, see chapter 5.

4.3.3 Framework

The Kripke model is of the form $M = \langle W, D, A, R, \sigma, F, V\rangle$. W is a set of possible worlds that includes global system states. Worlds are assumed to have a 'time stamp' which indicates whether they are supposed to be in the past, present or future. D is a set of objects that make sense in the application domain. It includes a set of agents $A \subseteq D$. R assigns to each agent a an appropriate accessibility relation $R_a(Att)$ to account for attitude Att. For example, $R_a(K)$ is an equivalence relation over W. The definitions of the various other accessibility relations are given in the subsequent sections. σ assigns to each agent a in world w the public information state $\sigma_{a,w}$. F defines the transitions of the Kripke frame based on basic actions. For each $act \in \mathcal{A}$ we have that $\langle w, v\rangle \in F(act)$ precisely when executing act in w results in v. Finally, V is a valuation function that for each $w \in W$ assigns objects to constants, sets of tuples of objects to predicate symbols and agents to agent names. Agents and agent names are both written $a, b, ..$ and we usually assign $V_w(a) = a$.

Like in chapter 3, the central definition is again that of satisfaction. Proposition-type formulas are evaluated relative to a world and an assignment; action type formulas are evaluated relative to an assignment and a pair of worlds that indicates a transition. Here too we sometimes need an assignment, because it may be needed to account for the variables that occur in the scope of an attitude.

Definition 46 (Satisfaction)
Given a model M and assignment g, define for each $\varphi, \alpha \in L_M$ the satisfaction relation by

$M, w, g \models P(t_1, ..., t_n)$	iff	$\langle V_{w,g}(t_1), ..., V_{w,g}(t_1)\rangle \in V_w(P)$
$M, w, g \models \neg\varphi$	iff	$M, w, g \not\models \varphi$
$M, w, g \models (\varphi \wedge \psi)$	iff	$M, w, g \models \varphi$ and $M, w, g \models \psi$
$M, w, g \models \exists x.\varphi$	iff	$M, w, g[x \mapsto d] \models \varphi \quad (d \in D)$
$M, w, g \models <\alpha>\varphi$	iff	there is a v such that $M, \langle w, v\rangle, g \models \alpha$ and $M, v, g \models \varphi$,
$M, w, g \models Attribute$	iff	see subsequent sections,
$M, \langle w, v\rangle, g \models act$	iff	$\langle w, v\rangle \in F(act) \quad (act \in \mathcal{A})$,
$M, \langle w, v\rangle, g \models (\alpha;\beta)$	iff	there is an u s.t. $M, \langle w, u\rangle, g \models \alpha$ and $M, \langle u, v\rangle, g \models \beta$,
$M, \langle w, v\rangle, g \models (\alpha\vert\beta)$	iff	$M, \langle w, v\rangle, g \models \alpha$ or $M, \langle w, v\rangle, g \models \beta$,
$M, \langle w, v\rangle, g \models \alpha^*$	iff	there are $u_0, ..., u_n$, $w = u_0, v = u_n$ s.t. $M, \langle u_i, u_{i+1}\rangle, g \models \alpha$,
$M, \langle w, v\rangle, g \models \text{test}(\varphi)$	iff	$w = v$ such that $M, w, g \models \varphi$. □

Frames are assumed to satisfy the following *success constraint*: two worlds or states w and v are related by an atomic action *act* when *act* is possible and successful, i.e. its preconditions hold at the input state w and its postcondition holds at the end state v.

Definition 47 (Success Constraint)
Given a model M and assignment g, define F such that for each $w, v \in W, act \in \mathcal{A}$
$\langle w, v\rangle \in F(act)$ iff $M, w, g \models \text{pre}(act)$ and $M, v, g \models \text{post}(act)$. □

If actions are combined into plans for composite actions, there are rules about the projection of the preconditions and postconditions of the component actions onto larger units. For example pre($\alpha;\beta$) equals pre(α) combined with those preconditions of β that do not follow from the postconditions of α. Similarly, post($\alpha;\beta$) are the postconditions of β combined with those postconditions of α that have not been undone by β. For $\|$ both sets of preconditions must be true simultaneously. So $\text{pre}(\alpha\|\beta) \leftrightarrow \text{pre}(\alpha) \wedge \text{pre}(\beta)$, and also $\text{post}(\alpha\|\beta) \leftrightarrow \text{post}(\alpha) \wedge \text{post}(\beta)$. The connective $|$ for choice brings some indeterminism. Either of the two actions can occur. In fact, that means that at least one of the sets of preconditions must be true, and also that at least one of the sets of postconditions becomes true. So $\text{pre}(\alpha|\beta) \leftrightarrow \text{pre}(\alpha) \vee \text{pre}(\beta)$, and also $\text{post}(\alpha|\beta) \leftrightarrow \text{post}(\alpha) \vee \text{post}(\beta)$. Finally the clause for repetition $*$ works as a repeated sequence. So we can safely assume that for each $\alpha \in L_{\mathcal{A}}$ we can calculate the pre(α) and post(α).

Proposition 9 (Success)
For each action $\alpha \quad M, \langle w, v\rangle \models \alpha$ iff $M, w, g \models \text{pre}(\alpha)$ and $M, v, g \models \text{post}(\alpha)$.
Proof sketch Would have to be done on the basis of distribution axioms of modality $<\alpha>\varphi$, and the projection of pre(.) and post(.) above. □

This gives a very idealised view on actions. If we want to use these notions to model reasoning about planning and action, we need to model the *intended effect*: the goal of an agent in executing the action. But the postcondition of an action, what is true after execution, may not be equal to its intended effects. There can be unintended side-effects to an action. Moreover, agents do not know which preconditions guarantee the intended effect of an action. And even if they do know what would guarantee it, they often do not have access to that information. Consider the precondition of another agent being cooperative. In our view, an agent makes the decision to carry out an action or plan on the basis of *applicability conditions*. These must be true for the action to be applicable at all. They do not guarantee success. If an action fails something changes in the world. Time is lost for example, or patience. Some commitment may have been made during the failed action and this must be undone. Therefore in practical systems often *failure conditions* are specified. These are goals that must be met anyway, in case the intended effect is no longer reachable. Often these will bring back the situation to the state before the action started. For an example of applicability conditions, preconditions, intended effects and failure conditions, see figure 4.1 in the beginning of this chapter.

For practical applications and especially for complex actions and plans, we would have the following additional frame constraint. If an agent decides to carry out an action α in w, then at least the applicability conditions of α must be true. Otherwise nothing happens. Now if they are true, then if the preconditions are also true, the intended effect is guaranteed, possibly with some side effects too. If the preconditions are false, at least the failure conditions apply. So once the applicability conditions are true, either the intended effect or the failure conditions are guaranteed. This disjunction forms the postconditions. So $\text{post}(\alpha) \rightarrow (\text{eff}(\alpha) \vee \text{fail}(\alpha))$. Not the other way around, because of possible side-effects.

Definition 48 (Applicability Constraint)
For a model M and assignment g, define F such that for each $w, v \in W, \alpha \in L_{\mathfrak{A}}$

$$M, \langle w, v\rangle \models \alpha \quad \text{iff} \quad \begin{array}{ll} \text{(i)} & M, w, g \models \text{appl}(\alpha) \text{ and } M, v, g \models \text{post}(\alpha) \quad \text{and} \\ \text{(ii)} & \text{if } M, w, g \models \text{pre}(\alpha), \text{ then } M, v, g \models \text{eff}(\alpha), \\ & \text{else } M, v, g \models \text{fail}(\alpha). \end{array}$$ □

There is a complication here. We have an agent with information about the world. Now in general there are two kinds of updates of information. In the first kind of update the world remains the same, but the agent learns more about it. This is the kind described in update semantics. In computer science the word 'update' is usually meant in the second sense, where the world is changed. As a result, the information of the agent needs to be brought up-to-date too. Now the difficulty is to assess the scope of a change. Which aspects are persistent, and which are affected by a change? In artificial intelligence the group of difficulties in representing the effects of actions is called the *frame problem*. Originally, so called frame axioms were defined for each exception. The above requirements are an attempt to constrain the frame problem. But in general system designers continue meet the *qualification problem*: how do you assess the circumstances under which an action is guaranteed to work? How many exceptions to a rule do you have to take into account? Designers are also plagued by the *ramification problem*: how do you assess the implicit consequences of an action? How detailed must the model be? Consequences that usually are not worth modelling, may turn out to be relevant after all. A word consists of characters. Usually, characters are not important for the meaning. But if I misunderstood the word 'intention', somebody could say "No, I meant intension-with-an-s!".

Even with the success and applicability constraints, agents do not have access to all the relevant conditions. There is possible interference with actions of other agents. In short:

reasoning about planning and action involves reasoning about the future and the future is unknown. The only way to properly solve this, is by default rules of inference. In the current framework, the most obvious way to do that would be to incorporate Veltman's (1996) work on defaults. So where we consider the information of an agent about the future, expressed as a set of worlds considered possible by the agent, it should thought of as restricted to the set of *normal worlds* of Veltman. Normal worlds are maximal in the expectation order: they fulfil as much defaults as possible.

That concludes the discussion of the framework. We continue by discussing the various attitudes of agents in dialogue, and how they are changed.

4.4 Belief

In this section, we look at the 'B' element in the BDI paradigm. To this end we transfer the results of chapter 3 to the multi-agent perspective. Most importantly, we highlight the distinction between *implicit* and *explicit* belief, and define a notion of *awareness* that is based on issues.

4.4.1 Awareness

At the end of chapter 3 we expressed the hope that by means of the issue structure we could solve part of the problem of deduction. If we follow the standard definition of belief or knowledge in a possible worlds framework, a rational agent is supposed to believe all the consequences of its beliefs. In other words, the set of beliefs of an agent is closed under deduction. This becomes a problem when you start to use such models to account for non-rational agent behaviour. For example, if you want to model mathematical reasoning of humans, you must explain why the solution to a mathematic equation is not immediately believed, once the axioms and premises are believed. For one thing, it takes time and effort to discover a solution. In a dialogue model the same problem arises, because we can't explain why an agent would explicitly raise an issue, if that issue is already deducible from the general topic of the dialogue. And similarly, we cannot yet explain why an agent realises that Clinton is taller when the issue is brought up.

One way to treat the problem of deduction is to make agents *resource bounded*. This very influential idea originates with Herb Simon, see for example Simon (1955), and has been applied in various areas of artificial intelligence. Recent examples of logical theories of resource bounded agents are Singh (1994) and Huang (1994). The main idea is that agents only have limited processing powers. Therefore, the set of beliefs can no longer be assumed to be closed under deduction. The set of beliefs that can be deduced depends on the available beliefs and on the available resources, like time or memory space. The more resources are allowed, the larger the deducible set. And for infinite resources the deducible set is again closed under deduction. A good example of work in belief revision that proceeds along these lines is Wasserman and Hansson (1999).

A more general solution to the problem of deduction starts from the concept of *awareness*. According to this idea an agent may believe some information, but as long as it is not actively considering the information, the belief is only implicit. On the other hand, a belief is explicit when it is believed and the agent is actually *aware* of it. That means that implicit belief can be modelled by the standard KD45 modal logic of belief. Explicit belief is then by definition equal to implicit belief plus awareness (Fagin and Halpern 1988). That puts the burden onto the as yet unexplained notion of awareness.

Here we define a logic for both explicit and implicit belief based on a general notion of awareness, suggested by (Fagin and Halpern 1988). After that, we show what happens if we use a particular notion of awareness, namely, awareness based on the current issues in an information state.

The logical language is the language as defined in definition 45 above. We focus on the attitudes $B_a\varphi$ for explicit belief, $L_a\varphi$ for implicit belief and $A_a\varphi$ for awareness, where $\varphi \in L_{\mathcal{P}}$. We discuss definitions of $K_a\varphi$ for knowledge-that, $W_a?\varphi$ for 'wonder' and $K_a^??\varphi$ for knowledge-wh as well.

For the semantics take a Kripke model $M = \langle W, D, A, R, F, \sigma, V\rangle$ as above, where for each $a \in A$ and $w \in W$ we have structures $R_a(L)$ and $\mathfrak{A}_{a,w}$ to model implicit belief and awareness respectively, and where the information state of an agent a at w is given by $\sigma_{a,w}$. We make the following stipulations.

First, $\sigma_{a,w} = \langle S_{a,w}, \mathfrak{I}_{a,w}\rangle$. Information states are defined in the tuple-style, with $S_{a,w}$ the data set containing accessible worlds, and $\mathfrak{I}_{a,w}$ an equivalence relation over W. That means that issues that are resolved for one agent, may still continue to structure the information in the dialogue. Without this requirement the characterisation of awareness in terms of issues would not make sense.

Second, $R_a(L)$ is a transitive, serial and Euclidean relation over W, which produces exactly the properties of KD45 modal logic of belief. Apart from the distribution axiom those are $L_a\varphi \rightarrow L_aL_a\varphi$ (positive introspection), $\neg L_a\varphi \rightarrow L_a\neg L_a\varphi$ (negative introspection) and $\neg L_a\bot$ (consistency). In case we require that relation $R_a(L)$ is reflexive too, we get the so called *truth axiom*: $L_a\varphi \rightarrow \varphi$. This models the fact that knowledge is *factive*. In that case the accessibility relation becomes an equivalence relation and the logic is $S5$. This is in fact the semantics for knowledge-that. So $R_a(K)$ is an equivalence relation. There is an obvious relation between the data set of an information state and the accessibility relation for belief. For each $a \in A, w, v \in W$ we require that $\langle w, v\rangle \in R_a(L)$ iff $v \in S_{a,w}$.

Third, $\mathfrak{A}_{a,w}$ contains the set of formulas from $L_{\mathcal{P}}$ that the agent is actively aware of (Fagin and Halpern 1988). Please note that this is a set of formulas, not a model-theoretic notion. By choosing different awareness functions we can account for different aspects of the problem of deduction. For example, we could make $\mathfrak{A}^m$ the set of formulas that are derivable in a number of steps that falls within a given time or memory bound m. Another proposal is based on Levesque's (1984) idea to use situations, partial valuations, instead of total possible worlds to account for awareness. For each situation s we can define the local vocabulary $\mathcal{P}(s) = \{p \in \mathcal{P} \mid V_s(p) = 1 \text{ or } V_s(p) = 0\}$, the set of ground atomic facts for which the truth value is defined. These are the facts the agent is aware of in s. Now, an agent is aware of a complex formula φ, when it is aware of all the atomic facts that occur in φ. So $\mathfrak{A}_{a,s} = \{\varphi \mid \text{atoms}(\varphi) \subseteq \mathcal{P}(s)\}$. In case we wanted to follow this lead, note that sets of possible worlds U always correspond to a situation s_U, where $s_U(p) = 1$ if $w(p) = 1$ for all $w \in U$ and $s_U(p) = 0$ if $w(p) = 0$ for all $w \in U$, and undefined otherwise. In particular, blocks in the partition induced by the current issues are good candidate situations. They typically agree on facts relevant to the current issues. In that case awareness becomes close to the account of Peircian depth that we sketched at the end of chapter 3.

However, we opt for a more direct relation between awareness and issues in information states. For each $a \in A, w \in W$ we require that $\mathfrak{A}_{a,w} = \{\varphi \mid \sigma_{a,w} \Vdash ?\varphi\}$. So an agent is aware of a proposition, when it is actively considering whether that proposition is an answer to a current issue. Each wh-question implies a number of yes/no questions as to the the truth of the different answers to it. For example, $?x.\varphi \models ?\varphi\{t/x\}$ for contextually available terms t. In general issue $?\varphi$ entails $?\psi$, $?\varphi \models ?\psi$ iff $\psi \models\equiv ?\varphi$, so whenever an assertive ψ is a possible direct answer to $?\varphi$. Therefore this definition takes care of all issues, not only of binary ones[4]. Now we can define a logic of belief based on Fagin and Halpern (1988).

Definition 49 (Belief Constraints)
For each $a \in A, w, v \in W$

$$R_a(L) = \{\langle w, v\rangle \mid v \in S_{a,w}, w \in W\}$$
$$\mathfrak{A}_{a,w} = \{\varphi \mid \sigma_{a,w} \Vdash ?\varphi\}$$
□

Definition 50 (Semantics)
For each $\varphi \in L_{\mathcal{P}}$ extend definition 46 as follows

$M, w, g \models L_a\varphi$ iff $M, v, g \models \varphi$ for all v such that $\langle w, v\rangle \in R_a(L)$
$M, w, g \models A_a\varphi$ iff $\varphi \in \mathfrak{A}_{a,w}$
$M, v, g \models B_a\varphi$ iff $M, v, g \models L_a\varphi$ and $M, v, g \models A_a\varphi$ □

Given this particular definition, we get the following result. It relates the epistemic attitudes belief and awareness to the meta-logical notion of support for information states.

Proposition 10 (Issues and Awareness)
For each $\varphi \in L_{\mathcal{P}}$

$M, w, g \models L_a\varphi$ iff $\sigma_{a,w} \Vdash \varphi$
$M, w, g \models A_a\varphi$ iff $\sigma_{a,w} \Vdash ?\varphi$
$M, w, g \models B_a\varphi$ iff $\sigma_{a,w} \Vdash ?\varphi$ and $\sigma_{a,w} \Vdash \varphi$

Proof By definition. □

Similar definitions can be made for attitudes with interrogative complements. As an example we define the attitude W_a for wonder. An agent is said to wonder about?φ when it is currently entertaining issue $?\varphi$. For example, Sherlock is wondering who did it, when he considers possible answers to that question. That means there is a relation between awareness and wondering: $M, w, g \models \mathrm{W}_a?\varphi$ iff $M, w, g \models \mathfrak{A}_a\psi$ and $\psi \models\equiv ?\varphi$, so for all ψ that are direct solutions to $?\varphi$. The factive counterpart of belief is knowledge. Is there also such a counterpart to wondering? The attitude of 'knowing wh' is called a *resolutive* by Ginzburg (1991). Its interrogative complement needs to be resolved. So we say that Sherlock knows who did it, when he knows the answer to the question. It is not enough that Sherlock knows an answer; he must know the correct answer. The current world of evaluation must make the answer true. We get the following relationship between knowledge-wh and knowledge-that: $M, w, g \models K^?_a?\varphi$ iff $M, w, g \models K_a\psi$ and $\psi \models\equiv ?\varphi$.

Definition 51 (Wonder; Knowledge-wh)
For each interrogative $?\varphi \in L_?$

$M, w, g \models W_a?\varphi$ iff $\sigma_{a,w} \Vdash ?\varphi$
$M, w, g \models K^?_a?\varphi$ iff $\sigma_{a,w} \Vvdash ?\varphi$ and $w \in \mathrm{d}(\sigma_{a,w})$ □

[4]In chapter 3 we defined $\sigma \Vvdash ?\varphi$ iff $\mathrm{u}(\mathrm{d}(\sigma)) \Vdash ?\varphi$, and $\psi_1; \ldots; \psi_n \models\equiv ?\varphi$ iff $\sigma[\psi_1]\ldots[\psi_n] \Vvdash ?\varphi$ for all $\sigma \in \Sigma$.

4.4.2 Discussion

This is a nice proposal. Does it work? Are issues the right kind of structure to model awareness? Is it awareness that makes the difference between explicit and implicit belief? We do not have conclusive answers. It is difficult to develop intuitions about awareness, without some non-logical observations. What we can do is study the properties of awareness by playing around with the awareness set $\mathfrak{A}_{a,w}$. We try some closure conditions discussed by Fagin and Halpern and compare them with properties of issues.

(1) Awareness is closed under negation, so $\varphi \in \mathfrak{A}_{a,w}$ iff $\neg\varphi \in \mathfrak{A}_{a,w}$. This conforms to the issue based notion, because for yes/no questions $?\varphi \models ?\neg\varphi$ and vice versa.

(2) Awareness could be closed under subformulas: if $\varphi \in \mathfrak{A}_{a,w}$ and ψ is a subformula of φ then $\psi \in \mathfrak{A}_{a,w}$. The assumption makes sense for an algorithm that needs to compute the truth of all subformulas in order to compute the truth of the whole. On the other hand, to compute the truth of $\varphi \vee \neg\varphi$ you do not have to compute the subformulas of φ. This parallels the (inconclusive) reasoning in chapter 3, where we suggested some pragmatic reasons for disjunctions and implications to trigger issues as to the truth of their disjuncts and antecedents respectively. Recall "John has a brother or a sister" which suggests the issue "Which?". However, because we are dealing with model theoretic notions, the subformulas may not always be accessible. Consider "John has a sibling", which does not trigger any issues. So in general this subformula restriction does not hold, which conforms to the existing behaviour of issues.

(3) We could restrict $\mathfrak{A}_{a,w}$ to all formulas that only contain atomic propositions from some specific vocabulary $\mathcal{P}(w) \subseteq \mathcal{P}$. Under this assumption we would get an awareness notion that is very similar to the suggestion by Levesque (1984) discussed above. In this way the differences induced by different application domains can be captured. Recall the example about television time versus scientists' time.

(4) The awareness set could be closed under some conceptual dependency relation, derived from the task domain or from the topic structure. The same restriction was discussed for issues.

(5) We could restrict the awareness set to those formulas that are recently mentioned in the dialogue history. This makes awareness dependent on memory. Or else we could restrict it to those formulas that are most important or salient in the context. We could restrict $\mathfrak{A}_{a,w}$ to formulas mentioned by one particular participant b, to model that a is only focussed on b. Finally, we could restrict awareness to formulas that are related to a particular object: the topic. These four mechanisms are similar (though not identical) to the raising, renewing and withdrawing of issues that we sketched can be modelled by an issue structure ordered by salience. Especially when we take the analogy between the issue structure and a datastructure like Ginzburg's (1995) QUD into consideration, such constraints can help to maintain this datastructure.

Is there is a fundamental relationship between issues and awareness? We have a double connection: (i) raising an issue makes agents aware of facts, namely those facts that are related to the issue by some dependency relation. By means of the general mechanism of raising and resolving issues, we have a way of filling out the awareness functions of Fagin and Halpern. (ii) The notion of awareness and the distinction between explicit and implicit belief can be used to explain when asking a question is relevant. Without the distinction all issues related to a general topic are automatically raised.

With respect to (i) it seems we have succeeded. Because of the characteristics of issues we get a number of properties for $\mathfrak{A}$ that are mentioned by Fagin and Halpern as desirable properties for at least some notions of awareness.

With respect to (ii), consider the following 'Socratic Principle' in which the teacher lets the pupil discover what he knew already by means of questions. Whenever $L_a\varphi$ but not yet $B_a\varphi$ (i.e. $\neg A_a\varphi$) asking $?\varphi$ can be appropriate in case there is some general topic which induces issue $?\psi$ and $?\psi$ depends on $?\varphi$. This mechanism is found very often in the dialogues studied by Walker (1996b). Because people are not rational agents, they need to be reminded of certain facts, even if they did know them. That means that in principle this notion of awareness removes some of the counterintuitive effects from the standard rationality postulates in epistemic logic. In general however we can not use the issue structure itself to explain when explicitly raising an issue is appropriate, given that the issue was already deducible on the basis of the general topic. If the set of issues is closed under 'conceptual dependency' as we suggested at various places, all possibly appropriate issues are automatically raised. The obvious solution is to require that an agent must be aware of the dependency constraints as well. In that case, explicitly raising an issue makes sense, whenever you are not sure the hearer is aware of the dependency between the issue and the general topic.

4.4.3 Belief Updates

In update semantics we modelled the increase of information by eliminating (pairs of) possible worlds from the information state. Can we describe the effect of dialogue acts in terms of updates? Consider the exchange in example (100). A says something while B is listening, B and signals that he received and understood the message. So information state σ_b is updated with the information conveyed by A's utterance. For this we use the regular assertive updates of chapter 3. We are modelling the apparent information of an agent based on what was said. Since we assume that A is sincere, we also update A's information state.

(100) A: John buys flowers for Mary. inform$(a, b, buy_flowers(j, m))$;
B: (nods) ack(b, a)

(101) $M, \langle w, v\rangle g \models \text{inform}(a, b, \mathit{buy\_flowers}(j, m)); \text{ack}(b, a)$,
so $\sigma_{b,v} = \sigma_{b,w}[\mathit{buy\_flowers}(j, m)]$ and $\sigma_{a,v} = \sigma_{a,w}[\mathit{buy\_flowers}(j, m)]$.

In the above example, both information states are updated in world v resulting from the joint inform-acknowledge act. We could have dealt with them in more detail, expressing the update of σ_a already in an intermediate world u, just after the inform. In that world we find also an update with the 'grounding issue' discussed at the end of chapter 3. So $\sigma_{b,u} = \sigma_{b,w}[?\mathit{buy\_flowers}(j, m)]$. In other words, B considers whether the content of A's utterance is compatible with what he knows, privately. It is in fact compatible, which is indicated by the acknowledgement. The result of the combined inform-grounding issue-acknowledgement sequence is equal to the one above.

What are the appropriateness conditions associated with an inform act? The applicability conditions are (1) that there is contact between speaker and hearer, and (2) no other initiatives are active, or else that this inform act provides information that is relevant to active initiatives. The intended effect of an inform act is to make its content publicly available and thereby to get the hearer to believe the content of the inform act. An inform act requires a (weak) acknowledgement: the hearer must indicate that he receives and understands the message in good order. The preconditions of success of an inform act are that (1) speaker is sincere, (2) hearer trusts speaker to be competent, and (3) hearer has no information to the contrary. Failure conditions are that (1) contact is maintained, (2) a motivation is given for the failure that explains what is wrong, i.e. which precondition is violated. For example, "You must be joking", "I don't believe you" or "I don't think so."

What about updates with information about other agents' information? Consider the belief report in (102). Note that we use implicit belief here, because that best conforms to belief reports: "Judging from the way he acts, John believes that Mary loves him".

(102) A: John believes that Mary loves him. inform$(a, b, L_j(love(m, j))$;
B: (nods) ack(b, a)

(103) $M, \langle w, v\rangle g \models \text{inform}(a, b, L_j(love(m, j)));\text{ack}(b, a)$,
so $\sigma_{b,v} = \sigma_{b,w}[L_j(love(m, j))]$ and $\sigma_{a,v} = \sigma_{a,w}[L_j(love(m, j))]$.

Suppose (102) is said by some trustworthy A to agent B. B acknowledges so we update the information state for B. In each possibility where John exists, there is a pointer to the information state of John. Now there are two options (Zeevat 1997): the first option is to test for each of the projected information states of John, whether the alleged belief is supported. All the worlds where it does not are eliminated in the regular update fashion. The second option is similar in spirit but a little more complicated. It is attributed to Stalnaker. It suggests to take the union[5] of the possible worlds in the information states according to each of the possibilities. This construction represents the information state of John according to B's global information state. This construction is then updated with the fact that Mary loves John. Now B must eliminate those possibilities that indicate that John's information state does not contain at least as much information as the updated global construction.

(104) (i) $\sigma[L_j\varphi] \;=\; \sigma\restriction\{w \in \text{d}(\sigma) \mid \sigma_{j,w} \Vdash \varphi\}$
(ii) $\sigma[L_j\varphi] \;=\; \sigma\restriction\{w \in \text{d}(\sigma) \mid \bigsqcup_{v\in \text{d}(\sigma)} \sigma_{j,v}[\varphi] \sqsubseteq \sigma_{j,w}\}$

The first of the two options is the simplest. The second option is equivalent to the first, for distributive updates. An update $[\varphi]$ is called distributive, when its behaviour depends only on the individual possible worlds. For a distributive update $[\varphi]$ $(\bigsqcup_{v\in \text{d}(\sigma)} \sigma_{a,v})[\varphi] = \bigsqcup_{v\in \text{d}(\sigma)}(\sigma_{a,v}[\varphi])$. Updates with presupposition, $[\partial\varphi]$, which check if the resulting state would be inconsistent, are typically not distributive. Decisions about the pertinence of answers are not distributive either; they proceed 'block wise'. For these, the update styles do make a difference. We realise there is a potential problem, but it would take too far to explore it here. For further discussion see Zeevat (1997) and Gerbrandy (1999) on distributed knowledge. We simply assume that option (i) is selected.

The approach generalises to reports about interrogative attitudes such as wonder.

[5]We can't really take the union of an equivalence relation. For the tuple-style define $\langle S_1, \mathfrak{I}_1\rangle \sqcup \langle S_2, \mathfrak{I}_2\rangle = \langle S_1 \cup S_2, \text{rts}(\mathfrak{I}_1 \cup \mathfrak{I}_2)\rangle$, where rts$(R)$ returns the reflexive, transitive and symmetric closure of a relation R. Now if we also $\langle S_1, \mathfrak{I}_1\rangle \sqcap \langle S_2, \mathfrak{I}_2\rangle = \langle S_1 \cap S_2, I_1 \cap \mathfrak{I}_2\rangle$, then $\sqcup$ and $\sqcap$ are the meet and join with respect to $\sqsubseteq$.

(105) A: John is wondering whether Mary loves him. $\text{inform}(a, b, W_j(?love(m, j)))$
B: (nod) $\text{ack}(b, a)$

(106) $M, \langle w, v\rangle g \models \text{inform}(a, b, W_j(?love(m, j))); \text{ack}(b, a)$,
so $\sigma_{b,v} = \sigma_{b,w}[W_j(?love(m, j))]$ and $\sigma_{a,v} = \sigma_{a,w}[W_j(?love(m, j))]$.

(107) $\sigma[W_j?\varphi] = \sigma \restriction \{w \in \text{d}(\sigma) \mid \sigma_{j,w} \Vdash ?\varphi\}$

Obviously, one would need a systematic analysis of these kinds of attitudes to see what extra constraints mark the difference between for example wonder, ask yourself, know, find out or discover. There are also relationships between interrogative attitude reports and the raising of issues. Because of the use of restriction $\restriction$ in the analysis, attitude reports now block the projection of embedded interrogative structure to the overall context. But this seems wrong. Example (105) is not just a report about John; it makes other participants wonder whether Mary loves John, too. Indeed, doubts or questions are often formulated in this indirect way for reasons of politeness or out of consideration. Consider a conversation between friends.

(108) John: I wonder whether she loves me.
Bill: Yeah, I don't know.

Bill's cooperative attitude makes him take over John's issue too, otherwise the reply would not be pertinent. We do not know whether this is the result of the cooperative setting, or whether it is a general property of the attitude verb 'to wonder', comparable to the factive nature of 'to know that' or 'to regret'. Compare the example of a sign saying: "We regret to inform you that dogs are not allowed", where the factive verb regret is used to indirectly phrase a polite prohibition. This analogy suggests that by presupposing issues we can deal with this projection of embedded interrogatives. For more on the semantics of interrogative attitude verbs see Ginzburg (1991).

4.5 Preference

In this section we look at the 'D' element of the BDI paradigm. How can we model desires? In the introduction we argued that desires should not be confused with goals. Desires may function as potential goals; but unlike goals they need not be feasible or consistent. We present a model that describes the desires of agents in terms of a preference order over (sets of) possible worlds. The preference order can be updated in the course of a dialogue. Again, adding information about preferences means refining the order.

4.5.1 Preference Notions

Consider the following three groups of related notions. The first group involves normative notions like obligation, permission and prohibition (must). The second group contains value concepts such as 'right and wrong' and ethical considerations (should). The third group involves desire, choice and intention and a concept of rational action (would). Von Wright's (1963) purpose is to try and study a central abstraction that underlies these fundamental notions: *preference*.

In general, Von Wright argues, preference concepts form a *scale*: a range of values between the two extremes of 'good' and 'bad'. Relative to such a scale the concepts 'better' and 'worse' are each others converse: to say that x is better than y, means to say that y is worse than x and vice versa. The two are definable in terms of each other because they involve a single concept: relative betterness. The terms 'good' and 'bad' are defined as the extremes of the scale: to say that some action is good, means to say that not doing it would be worse; to say that some action is bad, means that it would be better if it were not done. Preference is context dependent. Each statement of preference is essentially a relative statement: x is preferred over y. A preference is always somebody's preference, which is relative to certain circumstances at a certain moment in time. Moreover, preferences may change over time.

Preference is related to the notion of choice. A scale of preference typically ranges over alternative choices for future action. Even a preference for a certain state of affairs involves a hypothetical comparison. Often such comparisons are of the form: "If I were the queen, I would ...". That means we can use agents' choices to test the account, as for instance in the ice cream example. There is a distinction between extrinsic and intrinsic preference notions. When we prefer something because the doctor tells us it is better for us, there is an external reason for the preference. Such preferences are called *extrinsic*. However, if we prefer something simply because we like it better, than that is an example of *intrinsic* preference. It is redundant to a say that you prefer something because you like it better: the betterness constitutes the preference.

Preference orders underly the three groups of normative, ethical and motivational notions. There is also a non-preferential or binary boundary to what actions can be chosen. An agent is guided in its actions by what it desires to do, but is restrained by what it is capable of, by what is permitted and by what it feels it ought to do. Under certain idealisations the notions collapse. For example, an obedient agent only desires what is allowed. In a highly regulated environment, finding out what is allowed is quite difficult. So in legal expert systems the normative dimension is dominant. Similarly, a practical agent will not attempt the impossible. In an adverse environment doing what you are capable of is already difficult enough. So in robotics, the practical dimension is the most restrictive. In general that kind of preference relation takes precedence, that is most restrictive in a certain application.

Think of a space of possible choices for action, resulting indifferent states of affairs. The space is structured by different dimensions (figure 4.5). Along each dimension there is a binary distinction between 'possible' and 'impossible' or between 'good' and 'bad', as well as a scale or order among the remaining possibilities. For the normative, ethical and motivational axes this is a preference order. For other notions different orders apply. For example, for the practical axis that represents the abilities of an agent, there is a level of *skill* that ranks actions. Or along the epistemic axis that represents information, i.e., belief and issues of an agent, there is a scale of *salience* that indicates the relative importance. An important axis is the teleological axis, where we placed intentions and plans. Once an agent has decided to pursue a plan, which is of course checked for practicality, physical possibility, permission etc, the intention to pursue it is in itself a motivation to continue to pursue it, unless one of the preconditions becomes invalid. For example, when the action is no longer permitted. In this picture the notion of intrinsic preference is only on one of many notions that affect choice. However, it is the only positive motivation for action; the other notions are negative in the sense that they restrain possibilities.

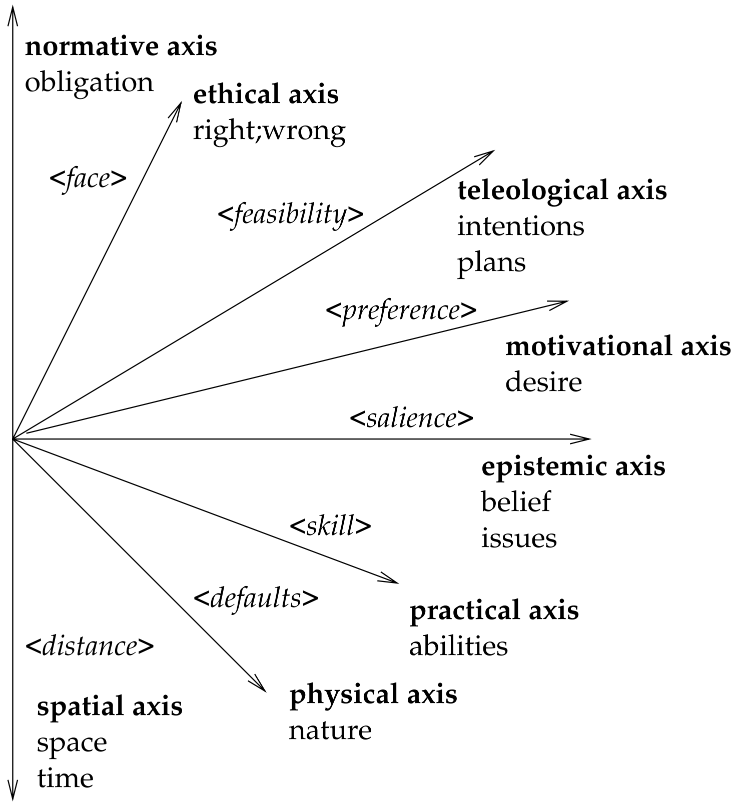


Figure 4.5: Dimensions of choice in rational action

Earlier we characterised a transaction as the establishment of a mutual commitment to future action. Where should we place commitment in this figure? Cohen and Levesque (1990) characterised intention as 'choice with commitment'. An intention is a choice to pursue a certain course of action, with an additional feature: commitment. It means that the agent must persist until either the goal is reached or until it becomes infeasible or unnecessary to reach it. So commitments are placed on the teleological axis of intentions and plans. But commitment is more than persistence. A promise generates a commitment, as does a task that is assigned to you, or a task that you proposed to do. A mutual commitment involves an obligation, a social force, to keep your part. In some environments you could be punished for not keeping a commitment, e.g., for not delivering the goods or for failing to make a payment. That puts commitments on the normative axis too. Even in the absence of an authority that might punish you, there is still the feeling that it is wrong not to keep a promise. Basic decency, sincerity or complacency are building blocks of cooperation. There is a link here with the notion of *face* (Brown and Levinson 1987) as it is used to explain politeness. This involves two aspects. Participants are generally agreeable and they are considerate of other participants (Allwood et al. 1992; Jokinen 1996; Bunt 1996). The tendency to be pleasant or kind can be explained as an attempt to maintain positive face. Being considerate, a tendency not to impose, can be explained as an attempt to reduce negative face. Why do agents keep face? From a social point of view it is beneficial to be cooperative. If you don't, you might be disqualified as a conversational or social partner for the next time around. But this tendency does not explain why an individual agent would be nice an considerate. Ethical considerations might be the drive that keep the agent complying. So commitments have an ethical component to.

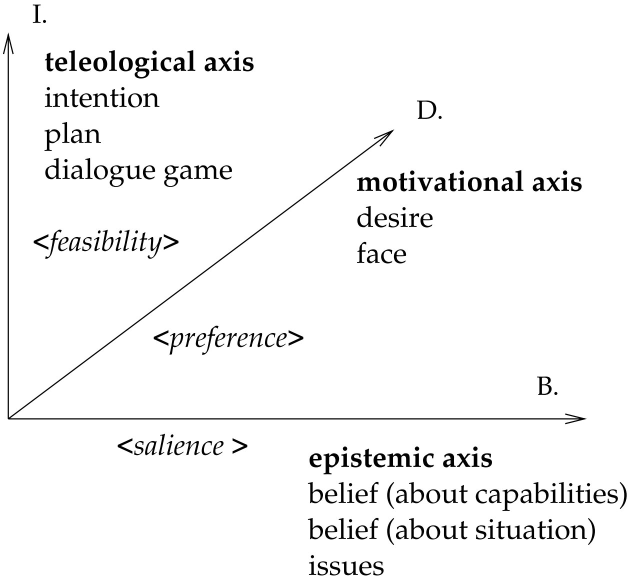


Figure 4.6: Limited dimensions of choice for dialogue agents

Having said that, this picture is too complicated if we want to study inquiry and transaction dialogues. For different application domains different dimensions take precedence. In dialogue the physical abilities of an agent are not as important as they would be in lunar exploration for example. The capabilities of an agent depend on the particular dialogue game. For example, I am not capable of marrying two people, but the official at the town hall is capable, provided she speaks the right words. Given dialogue principles that are related to the various aspects of cooperativity, we can reduce the picture to a standard BDI-picture of agenthood (figure 4.6). Ethical considerations and repercussions of breaking an obligation are made part of the preference order. An agent simple prefers not to violate an obligation. An agent prefers to maintain positive face and minimise negative face. That means that preference is now no longer only intrinsic, but also partly extrinsic. Permissions and prohibitions, are just like the capabilities of the agent 'hardwired' into what the agent believes it can do.

Studying normative, ethical and motivational notions in terms of preference orders is a form of what has been called *qualitative decision theory*. Obviously, the scales along the axes can be expressed by numerical values too. In economic theory they are often expressed by *probability* values (degrees of belief, feasibility) or by *utility* values (preference). However, in decision theory some properties of preference, such as asymmetry and transitivity, are simply taken for granted. These properties must be motivated. Moreover, numerical values reduce a preference order, which is a special kind of partial order, to a linear order. For multi-agent systems the way a preference changes during interaction is especially interesting. Updates of preference orders may therefore be a good source of inspiration for quantitative research. Thus, qualitative decision theory may provide motivations for principles which can then be applied in numerical theories for rational action.

Other accounts of preference orders that change during interaction, exist too. The most notable one is Van der Torre and Tan's (1997, 1999) account of obligation in update semantics. Bell and Huang (1999) re-apply much of their theory developed for desires and goals to an account of obligation, permission and prohibition. For an overview of deontic logic in computer science see Meyer and Wieringa (1993). Another interesting parallel is found in work by Lewis (1979), who suggests to model permission and prohibition just like beliefs: as a set of possible worlds. i.e. worlds that are possible with respect to the permissions and prohibitions granted by some agent. We expect that the permitted sets of Lewis would correspond to the maximal elements in a betterness relation modelled as a preference order.

4.5.2 Preference Constraints

The account of preference that we present has been worked out in detail by Huang and others (Huang 1994; Huang et al. 1996; Bell and Huang 1997; Bell and Huang 1999) and is based on an insight of Von Wright (1963), called the *conjunction expansion principle*. We say that an agent prefers φ to ψ , when the agent prefers $\varphi \wedge \neg\psi$-worlds to $\neg\varphi \wedge \psi$-worlds. The general idea is to add a binary modal operator, $\text{pref}_a(\varphi, \psi)$, to the logical language to represent that φ-worlds are preferred over ψ-worlds by an agent a. This is interpreted on a Kripke frame, where a partial preference order over possible worlds takes the place of an accessibility relation. Apart from the conjunction expansion principle, the preference order has to respect a number of other constraints. Such constraints correspond to axioms in the preference logic.

Before discussing intuitions about constraints, we first need a definition of what a preference order is. A preference order is a relation over states of affairs. There are two possibilities: to let the preference order range over individual possible worlds, or to let it range over sets of possible worlds, which correspond to propositions.

The first option best fits the framework of update semantics. We define a *weak preference order* as a particular type of partial order over the set of possible worlds W. The intuition is that $w \preceq_a v$ iff agent a considers world v to be *at least as much preferred* as world w. Initially, the agent does not have any particular preferences: all the worlds are equally preferred. So initially $\preceq_a = W \times W$. When we update the preference order with preference statements like $\text{pref}_a(\varphi, \psi)$, all the pairs of worlds that are not compatible with the preference statement are eliminated, and the relation is closed under the other constraints. The second option corresponds more directly to the intended semantics of the $\text{pref}_a(\varphi, \psi)$ operator. We define a *comparison relation* as a particular type of binary relation over the set of sets of worlds pow(W). Intuitively, $X \lhd_a Y$ means that agent *a prefers* each world in Y to each world in X. The intuitions behind the two options can be combined. We start with a partial order $\preceq_a$. In case both $w \preceq_a v$ and $v \preceq_a w$ we say that v and w are equally preferred, written $w \simeq_a v$. Equally preferred worlds naturally form equivalence classes. We can show that the weak preference order over worlds generates a comparison relation between these sets of equally preferred worlds (proposition 12).

Von Wright (1963) and Huang (1994) define a number of constraints that correspond to axioms in the preference logic. We first discuss these constraints here, before continuing to formalise them. Preference statements are *irreflexive*[6]: $\vdash \neg\text{pref}_a(\varphi, \varphi)$. Intuitively, you don't prefer apples over apples. Note that the weak preference relation of the first option is in fact reflexive. So reflexivity marks a difference between comparison and weak preference. Another property of comparisons is *asymmetry*[7]: if I prefer apples to candy, than necessarily I do not prefer candy to apples, all else being equal. So we have

[6] A relation R is reflexive when $\langle x, x\rangle \in R$ for all x and irreflexive when $\langle x, x\rangle \notin R$.

[7] A relation R is asymmetric when for all x, y, if $\langle x, y\rangle \in R$ then $\langle y, x\rangle \notin R$.

$\vdash \text{pref}_a(\varphi, \psi) \rightarrow \neg\text{pref}_a(\psi, \varphi)$. Note again that asymmetry conflicts with the reflexivity of a weak preference order. The constraint of *transitivity*[8] applies to comparison relations and weak preference orders alike: if I prefer chocolate to apples, and apples to candy, then necessarily I prefer chocolate to candy, at least when all else remains equal. So we have $\vdash \text{pref}_a(\varphi, \psi) \wedge \text{pref}_a(\psi, \chi) \rightarrow \text{pref}_a(\varphi, \chi)$. Below we deal with a possible counterexample to transitivity.

The most important constraint for both comparison relations and weak preference orders is the *conjunction expansion principle*. The intuition is that preference is a relative notion. What does it mean to say that you prefer apples to candy? Consider a situation in which you would have both apples and candy. Does this make the preference true? You don't really know, because you don't have to choose. Now suppose that you must drop one item, which one would you chose to drop? Yes, the candy. And similarly, suppose that you have neither apples nor candy. Does this situation conforms to the preference stated above? Well, you don't know, really. Having candy or having apples are hypothetical in that situation. But if you had to choose, you would choose the apples. The principle reads: $\vdash \text{pref}_a(\varphi, \psi) \leftrightarrow \text{pref}_a((\varphi \wedge \neg\psi), (\neg\varphi \wedge \psi))$. To say that you prefer apples to candy, really means that you prefer situations in which you have apples and no candy, over situations in which you have no apples and candy. The statement does not say anything about situations in which you have both, or neither.

Straightforwardly applying the conjunction expansion principle in a modal logic, leads to some standard paradoxes (Meyer and Wieringa 1993). If φ is preferred to ψ then also $\varphi \vee \chi$ is preferred to ψ (weakening) and also, if φ is preferred to ψ, then φ is preferred to $\psi \wedge \zeta$ (strengthening). Translate φ as apples, ψ as candy, χ as diesel-oil and ζ as a million dollars, and it is clear that we have a paradox.

The intuition behind a solution is that apart from the preferences, worlds or situations should otherwise differ as little as possible. Such *minimal change* requirements are known from the theory of counterfactuals (Lewis 1973). A counterfactual is a sentence of the form "If Clinton had been impeached, Al Gore would be president". Counterfactuals are indicated by a subjunctive mood, in English expressed by 'had been' and 'would'. The antecedent of the counterfactual is obviously not true. Nevertheless the speaker invites us to consider what would have happened. In order to do this we need to make a mental comparison between the current world and a world that differs from the current world, only in the fact that Clinton has been impeached. Now of course, a lot of things follow from that. Worlds are closed under deduction. So in a world where Clinton would have been impeached, Hillary would no longer be the president's wife. However, other aspects that have nothing to do with the impeachment, would have to remain constant in the comparison. So we make a comparison between the current world w and the set of worlds that are 'close enough' to w, but where Clinton is impeached. This set is expressed as $cw(w, [\![impeached(clinton)]\!])$. In that set, we check if Al Gore is president. Note that this is exactly the kind of mental comparisons that we mentioned earlier to highlight the relation between preferences and choice.

We use a *selection function* that indicates a minimal change between worlds. We define a function $cw :: W \times \text{pow}(W) \rightarrow \text{pow}(W)$. Intuitively, $cw(w, [\![\varphi]\!]_{M,g})$ returns the set of worlds 'close enough' to w that satisfy φ, where 'close enough' needs to be defined in some way dependent on the framework. For example, we could count the number of atomic facts on which worlds differ and minimise that number. Obviously, the closest

[8]A relation R is transitive when for all x, y, z, if $\langle x, y\rangle \in R$ and $\langle y, z\rangle \in R$, then $\langle x, z\rangle \in R$.

worlds function needs to conform to a number of constraints. To explain them would take us too far into the logic of conditions. Here we just follow the choice made by Huang (1994) to take Lewis' system called VC. The system allows one to define a conditional '$\leadsto$' as follows: $M, w, g \models \varphi \leadsto \psi$ iff $[\![\psi]\!]_{M,g} \subseteq cw(w, [\![\varphi]\!]_{M,g})$. In this system, the cw function conforms to the following constraints.

Definition 52 (Closest World Constrains)
For a model M, assignment g and for each $\varphi \in L_{\mathcal{P}}$

1. $cw(w, [\![\varphi]\!]_{M,g}) \subseteq [\![\varphi]\!]_{M,g}$.
2. If $w \in [\![\varphi]\!]_{M,g}$, then $cw(w, [\![\varphi]\!]_{M,g}) = \{w\}$.
3. If $cw(w, [\![\varphi]\!]_{M,g}) = \varnothing$, then $cw(w, [\![\psi]\!]_{M,g}) \cap [\![\varphi]\!]_{M,g} = \varnothing$.
4. If $cw(w, [\![\varphi]\!]_{M,g}) \subseteq [\![\psi]\!]_{M,g}$ and $cw(w, [\![\psi]\!]_{M,g}) \subseteq [\![\varphi]\!]_{M,g}$, then $cw(w, [\![\varphi]\!]_{M,g}) = cw(w, [\![\psi]\!]_{M,g})$.
5. If $cw(w, [\![\varphi]\!]_{M,g}) \cap [\![\psi]\!]_{M,g} \neq \varnothing$, then $cw(w, [\![\varphi \wedge \psi]\!]_{M,g}) \subseteq cw(w, [\![\varphi]\!]_{M,g})$. □

Other choices to deal with the hypothetical comparisons are certainly possible. An obvious possibility is Veltman's (1996) logic of non-monotonic reasoning. Here expectation patterns rank worlds according to a set of default rules that specify what is 'normally' the case. A set of worlds can be called 'closest' by reference to that expectation pattern. A second possibility is to investigate the relationship between issues and 'close enough'. In the Clinton example, the issue is who is president. Issues related to that, like who is the president's wife, do not add any extra structure. These aspects are allowed to change in a mental comparison. Issues that are non-related go across blocks. So the *cw* function should select worlds that are similar, except for those aspects in which one block differs from the next. Worlds are 'closest' when they differ only in aspects that are relevant to the current issues.

Now consider a potential counterexample to transitivity:

> "Suppose a person is offered, 'on a tray', a choice between an orange and a banana. He chooses the orange. He is then offered a choice between a banana and an apple. He chooses the banana. Finally he has to face a choice between an orange and an apple. Is he then 'by the laws of logic' bound to choose the orange? Does he contradict himself, in thought or action, if he chooses the apple?" (von Wright 1963, p22).

Notice that the choices are made in different consecutive situations. As we remarked earlier preferences are context dependent. How do we account for context dependency without begging the question? First, the person may come up with some additional reasons for choosing the apple. For example, she already had an orange and she prefers as much different fruits as possible. In that case the choice we witnessed was not motivated by an intrinsic preference; there were several preference scales at work. Second, the person may actually change her mind. The preference is intrinsic but changes. Preferences depend on the situation, including for instance the food the person has already eaten. A change in preference does not violate the transitivity axiom when the change in circumstances is larger than that judged to be 'close enough' by the selection function. Our judgement is that if the first fruit was eaten, the situations are not 'close enough'. Then the example is not a counter example. If the fruit is not eaten, the example concerns a mere choice game. Then the situations are 'close enough' and the person would be irrational in choosing the apple.

4.5.3 A Preference Logic

The logical language is defined by definition 45, with expressions $\text{pref}_a(\varphi, \psi)$, $\text{good}_a(\varphi)$ and $\text{bad}_a(\varphi)$ for each agent name $a \in Agt$ and $\varphi, \psi \in L_{\mathcal{P}}$. Intuitively, $\text{pref}_a(\varphi, \psi)$ means that agent a prefers φ-worlds to ψ-worlds. It expresses relative preference. The operator good_a expresses absolute preference. $\text{good}_a(\varphi)$ means that agent a prefers φ-worlds to non-φ-worlds, so by definition $\text{good}_a(\varphi) \equiv \text{pref}_a(\varphi, \neg\varphi)$. The operator bad_a expresses the opposite: $\text{bad}_a(\varphi) \equiv \text{pref}_a(\neg\varphi, \varphi)$.

The semantics is defined with respect to a model M as defined in section 4.3, where $M = \langle D, W, A, \lhd, cw, \sigma, F, V\rangle$, where cw is a function that assigns to each $w \in W$ the set of worlds $cw(w, V)$ that is closest to a given set of worlds V. For each $a \in A$ and $v \in W$ there is a binary *comparison relation* $\lhd_{a,v}$ over sets of worlds. Intuitively, $X \lhd_{a,v} Y$ means that in world v agent a prefers each world in Y to each world in X. The world v acts as an index; it represents the global system state.

There are a number of constraints for $\lhd$. The most important constraint is the conjunction expansion principle. In addition, we need to make sure that in making comparisons, we compare only on the basis of the preferred facts. That means that we compare situations that are otherwise as close as possible to the current world of evaluation. The satisfaction definition is therefore: $M, w, g \models \text{pref}_a(\varphi, \psi)$ iff $cw(w, [\![\neg\varphi \wedge \psi]\!]_M) \lhd_{a,w} cw(w, [\![\varphi \wedge \neg\psi]\!]_M)$. These two constraints, the *conjunction expansion principle* and the *closeness constraint*, are embodied in the following abbreviation[9].

Definition 53 (Abbreviation)
Given a model M define for each $X, Y \subseteq W$, agent a and $w \in W$:
$X \lessdot_{a,w} Y$ iff $cw(w, X \cap \overline{Y}) \lhd_{a,w} cw(w, Y \cap \overline{X})$. □

By means of this abbreviation the other constraints, *irreflexivity*, *transitivity*, *asymmetry* and *left-* and *right disjunctiveness*, can be specified in a simplified way.

Definition 54 (Constraints)
For each $X, Y \subseteq W$, $a \in A$, $w \in W$, the following constraints hold:

(irr) $X \not\lessdot_{a,w} X$,
(asymm) If $X \lessdot_{a,w} Y$, then $Y \not\lessdot_{a,w} X$,
(trans) If $X \lessdot_{a,w} Y$ and $Y \lessdot_{a,w} Z$, then $X \lessdot_{a,w} Z$,
(disl) If $X \lessdot_{a,w} Z$ and $Y \lessdot_{a,w} Z$, then $X \cup Y \lessdot_{a,w} Z$,
(disr) If $X \lessdot_{a,w} Y$ and $X \lessdot_{a,w} Z$, then $X \lessdot_{a,w} Y \cup Z$. □

The plausibility of the disjunctive principles (disl) and (disr) follows from the definition of a comparison relation, in conjunction with some meta reasoning on sets. Recall that $X \lessdot Y$ iff each world in Y is preferred to each world in X. Now assume that $X \lessdot Z$ and also $Y \lessdot Z$. That means that each world in Z is preferred to each world in X but also to each world in Y, so it must be the case that each world in Z is preferred to each world in $X \cup Y$, so $(X \cup Y) \lessdot Z$. That motivates (disl). Similar reasoning applies to (disr). Apart from such a technical motivation, the principles can also be defended on intuitive grounds. Take left disjunction. Suppose you prefer apples to candy, and you also prefer chocolate to candy. That means that you prefer to be in situations where you have apples or chocolate or

[9] By notation $\overline{X}$ we mean the set complement relative to W, so $\overline{X} = W \setminus X$. For an agent a, it makes sense to define $\overline{X} = \text{d}(\sigma_a) \setminus X$.

both and no candy, to situations where you have candy, but no apples and no chocolate. Please note that left and right disjunction are limited forms of weakening, or closure under consequence. This is allowed as long as it respects existing preferences. In general weakening of the preferred proposition is not allowed, to ban the paradoxes.

Now we define the satisfaction conditions for the preference operator. The definition closely follows the conjunction expansion principle, as captured in the abbreviation. The satisfaction conditions for the other connectives are taken from definition 46.

Definition 55 (Semantics)
For each $\varphi, \psi \in L_{\mathcal{P}}$ extend definition 46 as follows:
$M, w, g \models \text{pref}_a(\varphi, \psi)$ iff $[\![\psi]\!]_{M,g} \lessdot_{a,w} [\![\varphi]\!]_{M,g}$ □

Given the constraints of irreflexivity, asymmetry, transitivity, left- and right disjunction on the preference orders, it follows that we have the corresponding axioms for a logic of preference, in addition to the axioms of propositional logic. The axiom CEP represents the conjunction expansion principle. This replaces the distribution axiom of a normal modal operator.

Definition 56 (Axioms)
For each $\varphi, \psi \in L_{\mathcal{P}}$

IRR	$\vdash \neg\text{pref}_a(\varphi, \varphi)$,
ASYMM	$\vdash \text{pref}_a(\varphi, \psi) \rightarrow \neg\text{pref}_a(\psi, \varphi)$,
TRANS	$\vdash \text{pref}_a(\varphi, \psi) \wedge \text{pref}_a(\psi, \chi) \rightarrow \text{pref}_a(\varphi, \chi)$,
DISL	$\vdash \text{pref}_a(\varphi, \psi) \wedge \text{pref}_a(\chi, \psi) \rightarrow \text{pref}_a((\varphi \vee \chi), \psi)$,
DISR	$\vdash \text{pref}_a(\varphi, \psi) \wedge \text{pref}_a(\varphi, \chi) \rightarrow \text{pref}_a(\varphi, (\psi \vee \chi))$,
CEP	$\vdash \text{pref}_a(\varphi, \psi) \leftrightarrow \text{pref}_a((\varphi \wedge \neg\psi), (\neg\varphi \wedge \psi))$. □

Various versions of this logic has been shown to be sound and complete (Huang 1994; Huang and Masuch 1995; Bell and Huang 1997). We will not repeat the proofs here.

Because of the selection function a number of properties do *not* hold. This may come as a surprise when you think of preference as a modal operator. The distribution axiom and consistency axiom of a modal operator enforce deductive closure. For a binary modal operator, as preference, the intuitions are already less obvious. Nevertheless, it makes sense when you distinguish desires or preferences from intentions and goals. Preferences need not be feasible or closed under inference. On the other hand, the non-standard behaviour complicates the model.

(109) These principles do *not* hold:

necessitation	If $\models \varphi$, then $\text{pref}_a(\varphi, \psi)$
	If $\models \varphi$, then $\text{pref}_a(\psi, \varphi)$,
deductive closure	If $\models (\varphi \rightarrow \psi)$, then $\models \text{pref}_a(\varphi, \chi) \rightarrow \text{pref}_a(\psi, \chi)$,
	If $\models (\varphi \rightarrow \psi)$, then $\models \text{pref}_a(\chi, \varphi) \rightarrow \text{pref}_a(\chi, \psi)$.

The fact that necessitation does not hold, shows that you need not like nor dislike tautologies. Because the principle of closure under logical consequences does not always hold, the paradoxes of deontic logic are solved (Huang and Masuch 1995). It shows that preference is an example of a hyper-intensional context. In general, an operator creates a hyper-intensional context, when a proposition in its scope cannot be substituted by a logically derived proposition, without a change in truth value. For one thing, the agent may not be aware of the inference rules, or it may disapprove of the conclusion and therefore deny it. For example, you may like candy better than apples. It is well known that eating candy produces dental caries. But you don't prefer caries over apples. The comparison concerns only some specific, in this case non-caries related, aspects of candy. In fact, according to the disjunctive principles above, the preference concerns precisely those aspects of the candy that are already preferred, such as its sweetness. It depends on the selection function. Under some *cw* it is impossible to compare apple and candy situations without taking the caries into account as well. The two cannot be logically separated. So preferences do not have to be rational. Actions and plans on the other hand, and therefore eating habits, do have to be rational. It would be irrational for an agent to have a diet of candy and coke and later complain about the consequences.

Based on the relative preference operator pref_a we can define states that are absolutely preferred, or 'good' and states that are absolutely dispreferred, or 'bad' (von Wright 1963, p 34). $\text{good}_a\varphi \equiv \text{pref}_a(\varphi, \neg\varphi)$ and $\text{bad}_a\varphi \equiv \text{pref}_a(\neg\varphi, \varphi)$ Now this is interesting, because in dialogue people give either positive or negative feedback to a proposal or suggestion. Only rarely do they make the underlying preference explicit. So although the underlying concept is a comparative scale, we only get to know about the ends of the scale. From statements about good or bad we can reconstruct the underlying preferences.

The following properties show that these operators behave as expected. They really are at the ends of the scale.

Proposition 11 (Good and Bad)
For each model M the following properties hold:
(i) $\models (\text{pref}_a(\varphi, \psi) \wedge \text{good}_a\psi) \rightarrow \text{good}_a\varphi$
(ii) $\models (\text{pref}_a(\varphi, \psi) \wedge \text{bad}_a\varphi) \rightarrow \text{bad}_a\psi$
(iii) $\models \text{good}_a\varphi \leftrightarrow \text{bad}_a\neg\varphi$
(iv) $\models \text{good}_a\varphi \rightarrow \neg\text{bad}_a\varphi$
(v) $\models \text{bad}_a\varphi \leftrightarrow \neg\text{bad}_a\neg\varphi$

Proof sketch Proof of (i) is by definition and application of TRANS, CEP, TRANS and the definition. Proof of (ii) is similar. The others follow directly from the definition. □

As a matter of fact, Huang and Masuch (1995) use preference orders to define a deontic logic on top of it, with operations *oblige*, *forbid* and *permit*. Something is forbidden when it is judged bad by some super-agent. The super-agent personifies the rules or legislation. We will not follow this approach. For us preferences motivate goals and rational action.

4.5.4 An Update Semantics for Questions and Preference

We saw that we can model a preference operator by a comparison relation over sets of possible worlds. In this section we will incorporate such a preferences into an update semantics. There is a weak preference order for each information state that can be updated as more information about the agent's preferences comes in.

We define an update semantics for questions and preference. As any update system, it is a tuple $\langle L'', \Sigma'', [.]\rangle$, where L'' is the logical language, Σ'' a set of information states and $[.]$ an update function. A double $''$ is indicates update notions adapted for preference.

The logical language is defined by $L = L_{\mathcal{P}} \cup L_?$. A special kind of assertives are preference statements $\text{pref}_a(\varphi, \psi)$, $\text{good}_a(\varphi)$ and $\text{bad}_a(\varphi)$ for each agent name $a \in Agt$ and $\varphi, \psi \in L_{\mathcal{P}}$. In the internal perspective of an update system, the reference to an agent a is dropped.

The general framework is again a model $M = \langle D, W, A, \sigma, F, V\rangle$, as defined in section 4.3, but now for each $a \in A, w \in W$ we have $\sigma_{a,w} \in \Sigma''$. We require that each $\sigma_{a,w} = \langle S, \mathfrak{I}, \preceq\rangle$. So the definitions are based on the tuple-style framework, extended with a weak preference order $\preceq \subseteq W \times W$. The relation $\preceq$ is not limited to S. We indicated earlier that preferred states may be known to be untrue.

Definition 57 (Information States (iv))
Given an information space W, define

$$\begin{aligned}
\Sigma'' &= \{\langle S, \mathfrak{I}, \preceq\rangle \mid S \subseteq W;\ \mathfrak{I} \subseteq W \times W \text{ refl, trans, sym};\ \preceq \subseteq W \times W \text{ refl, trans}\},\\
\mathbf{0}'' &= \langle W, W \times W, W \times W\rangle,\\
\mathbf{1}'' &= \langle \varnothing, R, R'\rangle, \text{ for any } R, R' \subseteq W \times W.
\end{aligned}$$

□

The relation $w \preceq v$ means that world v is at least as much preferred as w, and possibly more. Initially, there is no ordering structure on worlds: every world is as much preferred as another one. The order is updated by 'weeding out' those pairs of worlds that are no longer compatible with the stated preferences, and by keeping the resulting relation closed under the preference constraints. Sets of equally preferred worlds form equivalence classes in the order. As a matter of fact these correspond to blocks in the partition resulting from the question "What do you prefer as much?". The comparison relation was defined over precisely these sets. Moreover, if those sets are closed under deduction, and if there is some plan that makes it feasible to achieve them, they may function as potential goals.

The following line of reasoning justifies this equivalence. We start with a partial order[10] $\preceq$. From this a strict order $\prec$ can be derived: $w \prec v$ iff $w \preceq v$ but $v \not\preceq w$. In case both $w \preceq v$ and $v \preceq w$ we say that v and w are equally preferred, written $w \simeq v$. Equally preferred worlds form equivalence classes: $W/_{\simeq}$. These equivalence classes can be compared: $[w]_{\simeq} \lhd [v]_{\simeq}$ iff $w \prec v$. Therefore $W/_{\simeq}$ is ordered by $\lhd$. This is exactly the comparison relation that we described in the previous section. And from a comparison relation $\lhd$ we can define a weak preference order $\preceq$. Suppose $X \lhd Y$, now for all $w, v \in X \cup Y$ we have $w \preceq v$ iff $w, v \in X$ or $w, v \in Y$ or $w \in X, v \in Y$.

Proposition 12 (Comparison and Weak preference)
For each $X, Y \subseteq W$ and $w, v \in X \cup Y$
$X \lhd Y$ iff $w \preceq v$ and either $w, v \in X$ or $w, v \in Y$ or $w \in X, v \in Y$.
Proof Follows the above discussion. □

That settles the equivalence, at least for reflexivity and transitiveness. Left- and right disjunction follow from the definition, by general set theoretic principles and additional constraints on the *cw* function. The other properties, conjunction expansion and the closeness constraint, are dealt with below.

In order to update the recorded preferences of an agent in dialogue, we need the concept of a *refinement*. We say that partial order $\preceq'$ is a refinement of $\preceq$ iff $\preceq' \subseteq \preceq$, and if the resulting order is again a partial order. This is forced by an rta(.) function that returns the reflexive, antisymmetric, transitive closure of a relation.

[10] A partial order is reflexive, transitive and antisymmetric. A relation R is antisymmetric when it holds that if $\langle x, y\rangle \in R$ and $\langle y, x\rangle \in R$, then $x = y$.

Definition 58 (Refinement)
For $\preceq$ a weak preference order and $X, Y \subseteq W$ define

$$\begin{aligned} \preceq\circ(X,Y) \;=\; & \text{rta}(\{\langle w,v\rangle \in \preceq \;\mid\; \text{if } w,v \in X \cup Y \\ & \text{then either } w,v \in X \text{ or } w,v \in Y \text{ or } w \in X, v \in Y\}). \end{aligned}$$ □

Now we define an update function [.] for our update semantics with questions and preferences. The definition is analogous to that of the tuple definition in chapter 3, with a clause added to deal with preference statements. Note that this clause only applies to preference statements of the agent it concerns. The definition follows the conjunction expansion principle. Formulas φ and ψ are swapped because preference statements prefer the left-hand formula but refinements prefer the right-hand set. With respect to the closeness constraint, we can no longer use a closest world function *cw* based on a single world of evaluation w. It is the agent that is doing the mental comparison, not the observer, and as far as the agent is concerned it could be in any of the worlds in $\mathrm{d}(\sigma)$. See below for a discussion.

Definition 59 (Updates (iv))
Define for each $\varphi \in L''$ and $\langle S, \mathfrak{I}, \preceq\rangle \in \Sigma''$ an update function $[\varphi]_g$ under g by

$\langle S,\mathfrak{I},\preceq\rangle[\varphi]_g$	$=$	$\langle S \cap [\![\varphi]\!]_g, \mathfrak{I}\rangle$	assertive
$\langle S,\mathfrak{I},\preceq\rangle[?\vec{x}.\varphi]_g$	$=$	$\langle S, \mathfrak{I} \cap \simeq\langle\![?\vec{x}.\varphi]\!\rangle_g\rangle$	interrogative
$\langle S,\mathfrak{I},\preceq\rangle[\text{pref}(\varphi,\psi)]_g$	$=$	$\langle S,\mathfrak{I},\preceq\circ(cw(w, [\![\psi \wedge \neg\varphi]\!]_g), cw(w, [\![\varphi \wedge \neg\psi]\!]_g))\rangle$ for all $w \in S$	preference
$\langle S,\mathfrak{I},\preceq\rangle[\varphi;\psi]_g$	$=$	$(\langle S,\mathfrak{I},\preceq\rangle[\varphi]_g)[\psi]_g$	sequence □

We define an information order $\sqsubseteq$ again. Based on that order we get a support function, $\sigma \Vdash \varphi$ iff $\sigma[\varphi] \sqsubseteq \sigma$, and a notion of entailment: $\varphi_1;\ldots;\varphi_n \models \psi$ iff $\sigma[\varphi_1]\ldots[\varphi_n] \Vdash \psi$ for all $\sigma \in \Sigma''$.

Definition 60 (Information order)
For each $\sigma_1 = \langle S_1, \mathfrak{I}_1, \preceq_1\rangle, \sigma_2 = \langle S_2, \mathfrak{I}_2, \preceq_2\rangle \in \Sigma''$
$\sigma_1 \sqsubseteq \sigma_2$ iff $S_2 \subseteq S_1$ and $\mathfrak{I}_2 \subseteq \mathfrak{I}_1$ and $\preceq_1 \subseteq \preceq_2$. □

In proposition 12 we showed how a weak preference order $\preceq$ gives rise to equivalence classes with a corresponding comparison relation $\lhd$. What about the constraints on a comparison relation? Check that as long as $\preceq$ respects the conjunction expansion principle, the same will be true for the corresponding comparison relation $\lhd$. In the initial state conjunction expansion holds trivially, because then there are no equivalence classes to be compared. Because of definition 59 updates of the $\preceq$ order also respect conjunction expansion. The properties of left-and right disjunction make no sense for individual worlds. They follow from general set theoretic principles.

What about the closeness constraint? Earlier we defined a function *cw* on the basis of the world of evaluation w, but now we can not do this anymore, since the agent does not know in what world it is. This brings out a distinction of Von Wright between *actual preferences*, where the comparison is based on the current world of evaluation; *ceteris paribus* preferences, where the comparison is based on any world as long as it is the same in both cases; *conditional preferences* where the compared sets of worlds are restricted to some condition χ; and finally *absolute preferences* (Huang 1994, p103). The current definition uses ceteris paribus preference, limited to the data set.

(110) $M, w, g \models \text{pref}_a(\varphi, \psi)$ iff
(i) $cw(w, [\![\neg\varphi \wedge \psi]\!]_g) \lhd_{a,w} cw(w, [\![\varphi \wedge \neg\psi]\!]_g)$ actual,
(ii) $cw(v, [\![\neg\varphi \wedge \psi]\!]_g) \lhd_{a,w} cw(v, [\![\varphi \wedge \neg\psi]\!]_g)$, for all $v \in W$ ceteris paribus,
(iii) $cw(w, [\![\chi \wedge \neg\varphi \wedge \psi]\!]_g) \lhd_{a,w} cw(w, [\![\chi \wedge \varphi \wedge \neg\psi]\!]_g)$ conditional,
(vi) $[\![\neg\varphi \wedge \psi]\!]_g \lhd_{a,w} [\![\varphi \wedge \neg\psi]\!]_g$ absolute.

It depends on the application domain what kind of preference is most suitable. Because the agent does not know what world it is in, we can not use actual preferences. As we said, the mental comparison is not made by the observer, but by the agent itself. For some legal or moral applications, preferences can be thought of as absolute. Under all conceivable circumstances it is wrong to kill somebody. Most other comparisons allow mitigating circumstances. Think again of the paradoxes, where a million dollars or diesel oil can be added. So we rule out absolute preferences too. Think of a travel agency where a couple is trying to decide to go to Greece or to the Bahamas. This is an example of a ceteris paribus choice. The couple may vary any assumption as long as they do that for both options. Otherwise it is not a fair comparison. So they compare the distance to the beach, the price and the photos in the brochure. In the ice cream example, Betty prefers strawberry to chocolate in combination with fruit flavours and vanilla. But in combination with mocha, she prefers chocolate. This could be captured either by the ceteris paribus or by the conditional definition. However, conditions can be captured by extending the formulas in the object language, so the ceteris paribus option is more general. That explains our choice in definition 59.

We have the following equivalence, which is analogous to proposition 10 for belief.

Proposition 13 (Preference Statements and Updates)
For each $\varphi, \psi \in L_{\mathcal{P}}$
$M, w, g \models \text{pref}_a(\varphi, \psi)$ iff $\sigma_{w,a} \Vdash \text{pref}_a(\varphi, \psi)$
Proof First note the correspondence between definition 58 and proposition 12:
$(*)$ If $\preceq \circ(X, Y) = \preceq'$ then $X \lhd' Y$ for the corresponding comparison relation.
Suppose $\sigma_{w,a} \Vdash \text{pref}_a(\varphi, \psi)$, so by definition $\sigma_{w,a}[\text{pref}_a(\varphi, \psi)] \sqsubseteq \sigma_{w,a}$, so we have $\preceq_{w,a}\circ(cw(w, [\![\psi \wedge \neg\varphi]\!]_g), cw(w, [\![\varphi \wedge \neg\psi]\!]_g)) \supseteq \preceq_{w,a}$, and by the definition of $\circ$, the antisymmetry of $\sqsubseteq$ and observation $(*)$ we have that $cw(w, [\![\psi \wedge \neg\varphi]\!]_g) \lhd_{w,a} cw(w, [\![\varphi \wedge \neg\psi]\!]_g)$, for the corresponding $\lhd_{w,a}$ and therefore $M, w, g \models \text{pref}_a(\varphi, \psi)$. □

We can make a further restriction. There is a general constraint on preference statements: the two things compared must be two of a kind. So it is in fact allowed to 'compare apples and oranges' as they say; it is not allowed to compare apples with a holiday on the Bahamas, or apples with oranges and diesel oil. Such conditions on the kinds of objects to choose from correspond to the answer presupposition of the issue that is typically at stake in a choice situation. Something like: "Which holiday do you prefer: Greece or the Bahamas?" or "What fruits would you like?". This is exactly the same context dependency as we saw in chapter 3 for the "Only Socrates is wise" examples. The domain of discourse is restricted to fruits or holidays. The minimal change requirement should respect the facts on which blocks in the partition 'agree'. And these facts are stored in the data set. That means that we can replace $cw(w, [\![\varphi]\!]_g)$ in definition 59 with $cw(\sigma, \varphi)$ which is defined as $cw(\sigma, \varphi) = \text{d}(\sigma[\varphi]) = S \cap [\![\varphi]\!]_g$. This suggestion gets rid of the mysterious closest world functions and deal with the paradoxes in a more familiar way: by context dependency.

Definition 61 (Updates (iv))
For each $\sigma \in \Sigma''$ and $\varphi, \psi \in L_{\mathcal{P}}$ redefine the update function [.] of definition 59 as follows

$$\langle S, \mathfrak{I}, \preceq\rangle[\text{pref}(\varphi, \psi)]_g = \langle S, \mathfrak{I}, \preceq \circ (S \cap [\![\psi \wedge \neg\varphi]\!]_g), S \cap [\![\varphi \wedge \neg\psi]\!]_g)\rangle. \qquad \Box$$

However, in this case the data set which models the context must also contain information about future or hypothetical situations. Unwanted exceptions like diesel oil must be ruled out. And this is exactly the kind of problem that occurs in non-monotonic logics for reasoning about planning and action. Therefore we now believe that a treatment of the mental comparisons in preference statements requires a proper treatment of the counterfactuals and conditionals involved. In this framework, the most obvious solution would use Veltman's (1996) expectation patterns. These are partial orders that represent the priority among possible conflicting default rules. We could require that all non-actual possible worlds considered by the agent, be ordered by these expectation patterns. So unless it is part of the comparison to explicitly deny normal circumstances, comparison should be between sets of worlds that are as normal as possible, considering the information of the agent about the current state of affairs and about default rules.

4.5.5 Preference Updates

There is a relation between updates and utterances of preference statements. Recall example (100). We now deal with a similar example for preference statements taken from the ice cream example. The example is rather tentative. We need to make a lot of assumptions to get it to work; it is only meant to stimulate discussion. We do not claim that this is a final solution. The hypothesis is that absolute preference expressions like "I want ..", "I would like" or "No, I don't like ..." can be interpreted by expressions of the form good and bad. Given the buying-and-selling setting with the 'your will is my command' principle, the buyer's preference statements are adopted by the seller, and subsequently interpreted as a request.

Note that utterances A1 – B4 have set the stage. Betty agreed to a three flavoured cone. Given the setting, Antonio is committed to let her have that cone. The issue is now which combination of three flavours she wants to order. We can summarise the outcome of this information phase by assuming something like $?xyz.\mathit{three\_cone}(x, y, z) \equiv \exists u(\mathit{cone}(u) \wedge \text{commit}(a, b, \mathit{have}(b, u))); ?xyz(\mathit{flavour}(x) \wedge \mathit{flavour}(y) \wedge \mathit{flavour}(z) \wedge \mathit{with}(x, u) \wedge \mathit{with}(u, y) \wedge \mathit{with}(u, z) \wedge \text{good}_b(\mathit{taste}(b, x)) \wedge \text{good}_b(\mathit{taste}(b, y)) \wedge \text{good}_b(\mathit{taste}(b, z)) \wedge \text{good}_b(\mathit{combine}(x, y, z)))$. The semantics of the commitment operator is briefly explained in section 4.6.2.

From the acceptance of this information a number of domain-dependent inference rules follow which guide the selection of flavours. For example, that if Betty says she likes a flavour, it is interpreted in this context to mean that she likes to have it on the ice cream cone. The issue what three flavours Betty prefers over other combinations of flavours continues to structure the dialogue. For any flavour f we use the following shorthand: $\text{good}_b(f)$ means that f is a solution to this issue: f is one of the flavours on the cone, in combination with flavours already mentioned. That solves the literal ambiguity of $B5$ "I like vanilla and I also like strawberry" or "I like vanilla and strawberry together". So whenever in these circumstances $\text{good}_b(f) \wedge \text{good}_b(g)$ then also $\text{good}_b(\mathit{combine}(f, g))$. Such modelling decisions also depend on the type of preference relation. For absolute preferences, we must make the provisions about combinations of flavours in the object language; for conditional and ceteris paribus choices, we can rely on the framework and use $\wedge$ for combinations.

(111) B5: I like vanilla and strawberry.
$\text{inform}(b, a, \text{good}_b(\mathit{vanilla}) \wedge \text{good}_b(\mathit{strawb}))$
$\rightsquigarrow \text{request}(b, a, !\exists z.\mathit{three\_cone}(\mathit{vanilla}, \mathit{strawb}, z))$
A4: Sorry, we don't have strawberry.
$\text{apologise}(a, b); \text{inform}(a, b, \neg \mathit{available}(\mathit{strawb}))$
$\rightsquigarrow \text{reject}(a, b)$

(112) $M, \langle w, u \rangle g \models \text{inform}(b, a, \text{good}_b(\mathit{vanilla}) \wedge \text{good}_b(\mathit{strawb}))$ and
$M, \langle u, v \rangle g \models \text{apologise}(a, b); \text{inform}(a, b, \neg \mathit{available}(\mathit{strawb}))$
$\sigma_{b,u} = \sigma_{b,w}[\text{good}_b(\mathit{vanilla}) \wedge \text{good}_b(\mathit{strawb})] = \sigma_{b,w}[\text{good}_b(\mathit{vanilla} \wedge \mathit{strawb})]$
$= \langle S_{b,w}, \mathfrak{I}_{b,w}, \preceq_{b,w} \circ (S_{b,w} \cap [\![\mathit{vanilla} \wedge \mathit{strawb}]\!]_g, S_{b,w} \cap [\![\neg(\mathit{vanilla} \wedge \mathit{strawb})]\!]_g) \rangle.$
$\sigma_{a,u} = \sigma_{a,w}[\text{good}_b(\mathit{vanilla}) \wedge \text{good}_b(\mathit{strawb})] = \sigma_{a,w}[\text{good}_b(\mathit{vanilla} \wedge \mathit{strawb})]$
$= \langle S_{a,w} \cap [\![\text{good}_b(\mathit{strawb} \wedge \mathit{strawb})]\!]_g, \mathfrak{I}_{a,w}, \preceq_{a,w} \rangle$
$\sigma_{b,v} = \sigma_{b,u}[\neg \mathit{available}(\mathit{strawb})] = \langle S_{b,u} \cap [\![\neg \mathit{available}(\mathit{strawb})]\!]_g, \mathfrak{I}_{b,u}, \preceq_{b,u} \rangle$
$\sigma_{a,v} = \sigma_{a,u}[\neg \mathit{available}(\mathit{strawb})] = \langle S_{a,u} \cap [\![\neg \mathit{available}(\mathit{strawb})]\!]_g, \mathfrak{I}_{a,u}, \preceq_{a,u} \rangle$

For the first update there is a difference between the speaker *b*, for whom the update results in a refinement of her apparent preference order, and hearer *a* for whom it means an elimination of all the worlds in which *b* does not like strawberry. This is analogous to the way we dealt with belief updates in the previous section. However, it is strange that in one case we calculate the update, whereas in the other case we assume the result is given and eliminate all the worlds that do not comply. This points to a general deficiency in our combination of updates and modal logic. We need principles that tell us whether an agent has information about its own preferences. If we assume positive and negative introspection with respect to preferences, then each time $\preceq$ is refined S must be adapted too. Such principles of interaction between beliefs, issues and preference statements are a topic for further research.

(113) $\models \text{pref}_a(\varphi, \psi) \rightarrow L_a \text{pref}_a(\varphi, \psi)$ positive introspection
$\models \neg \text{pref}_a(\varphi, \psi) \rightarrow L_a \neg \text{pref}_a(\varphi, \psi)$ negative introspection

What happens if we combine preferences and interrogatives? Here is an example of a yes/no question. Again we assume shorthands. In particular $\mathit{cream} \equiv \exists u(\mathit{cone}(u) \wedge \text{commit}(a, b, \mathit{have}(b, u)) \wedge \mathit{with}(u, \mathit{whipped\_cream}))$.

(114) A8: Cream? $\text{ask}(b, a, ?\text{good}_b(\mathit{cream})) \rightsquigarrow \text{propose}(b, a, !\mathit{cream}))$
B11: No thanks. $\text{answer}(b, a, \neg \text{good}_b(\mathit{cream})) \rightsquigarrow \text{reject}(b, a)$

(115) $M, \langle w, u \rangle g \models \text{ask}(b, a, ?\text{good}_b(\mathit{cream})$ and
$M, \langle u, v \rangle g \models \text{answer}(a, b, \neg \text{good}_b(\mathit{cream}))$
$\sigma_{a,u} = \langle S_{a,w}, \mathfrak{I}_{a,w} \cap \simeq \langle\![?\text{good}_b(\mathit{cream})]\!\rangle_g, \preceq_{a,w} \rangle$
$\sigma_{b,u} = \langle S_{b,w}, \mathfrak{I}_{b,w} \cap \simeq \langle\![?\text{good}_b(\mathit{cream})]\!\rangle_g, \preceq_{b,w} \rangle$
$\sigma_{b,v} = \langle S_{b,u}, \mathfrak{I}_{b,u}, \preceq_{b,u} \circ (S_{b,u} \cap [\![\mathit{cream}]\!]_g, S_{b,u} \cap [\![\neg \mathit{cream}]\!]_g \rangle$
$\sigma_{a,v} = \langle S_{a,u} \cap [\![\text{bad}_b(\mathit{cream})]\!]_g, \mathfrak{I}_{a,u}, \preceq_{a,u} \rangle$

With these examples we end the discussion on preference relations. As is clear from the sections above, a lot of research into the combination of preference relations and update semantics, and preference relations and issues still needs to be done.

4.6 Intention

In the previous sections we stressed the different nature of desires and preferences on the one hand, and goals on the other. For goals we do expect a certain rationality. Goals are the intended effects of the actions an agent is currently engaged in. Goals are motivated by desires: the maximally preferred sets on the preference order. So preferred sets of worlds function as potential goals. In this section we briefly discuss what additional assumptions are needed to characterise goals.

Assume a logical language along the lines of definition 45. We focus on expressions $G_a\varphi$ for all $a \in Agt$ and $\varphi \in L_P$. The semantics is based on the general Kripke model $M = \langle W, D, A, R, \sigma, F, V\rangle$, where $R_a(G)$ represents a serial[11] accessibility relation for each $a \in A$, which is meant to model goals.

Definition 62 (Semantics)
For each $\varphi \in L_{\mathcal{P}}$ extend definition 46 as follows
$M, w, g \models G_a\varphi$ iff $M, v, g \models \varphi$ for all v such that $\langle w, v\rangle \in R_a(G)$. □

A serial relation produces a *KD* modal logic, which is the basic modal logic, extended with the principle of distribution. That means that we have the following axioms. Distribution means closure under consequence. Consistency and distribution are enforced by seriality; there must be some world that supports the goal.

Definition 63 (Goal Axioms)
For all $a \in Agt$, $\varphi, \psi \in L_{\mathcal{P}}$

(i)	$\models G_a\varphi \wedge G_a(\varphi \rightarrow \psi) \rightarrow G_a\psi$	(distribution),
(ii)	$\models G_a\varphi \rightarrow \neg G_a\neg\varphi$	(consistency),
(iii)	If $\models \varphi$, then $\models G_a\varphi$	(necessitation).

□

Property (ii) is also true of absolute preferences; recall proposition 11. By further restraining preferences, we can turn them into potential goals. The main advantage is that we get a preference order among potential goals. After all, deliberation about different possible courses of action is not trivial. Preferences for one goal over the other may change in the course of dialogue.

Here is an example from the ice cream setting. Suppose Betty only has 3,20 in her purse. Initially, she would like a three flavoured ice cream with whipped cream on top. However, in the beginning of the dialogue she finds out that a three flavour cone costs 2,50. She knows that cream would cost her another 0,75. So she can't have both. Now she must decide between a three flavoured cone, or a two flavoured cone with cream. She decides for the first option.

Earlier we stressed the differences between desires and goals. With respect to goals an agent must be rational. When is the decision of an agent called rational?

First, an agent's goals must be *realistic*. In other words, goals must be considered possible by the agent. The accessibility relation for goals is included in that for implicit beliefs, $R_a(G) \subseteq R_a(L)$, and they interact: if $\langle w, u\rangle \in R_a(G)$ and $\langle w, v\rangle \in R_a(L)$, then $\langle v, u\rangle \in R_a(G)$, for each $a \in a$. In other words, possible goal states are included in the data set. We use implicit beliefs here because agents may not realise the possibilities of the future.

[11] A relation R is serial when for all x, there is a y such that $\langle x, y\rangle \in R$.

Definition 64 (Belief and Goals)
For all $a \in Agt$, $\varphi, \psi \in L_{\mathcal{P}}$
(i) $\models G_a\varphi \rightarrow L_a\varphi$ (realism),
(ii) $\models G_a\varphi \rightarrow \neg L_a\neg\varphi$ (consistency),
(iii) $\models G_a\varphi \rightarrow L_aG_a\varphi$ (introspection). □

Second, the beliefs of an agent about what is possible or feasible depend on its knowledge of the available actions, both for itself and for other agents. In general we demand that a future state of affairs is considered possible,ifthere is an action or plan to reach it. So for all $w, v \in W$, if $M, \langle w, v\rangle \models \alpha$ for some $\alpha \in L_{\mathcal{A}}$, then $\langle w, v\rangle \in R_a(L)$ too, for all $a \in Agt$. Because goal states are included in the belief states, this constraint also enforces that potential goal states are realisable by some action. This constraint is known as *strong realism*. If an agent has a feasible goal, then it believes it to be feasible. This constraint makes sure that the practical axis in figure 4.5, which represents the agents abilities, is properly projected onto the epistemic axis.

Definition 65 (Belief, Goals)
For all $a \in Agt$, $\varphi \in L_{\mathcal{P}}$
$\models G_a\varphi \rightarrow L_a{<}\alpha{>}\varphi$, for some $\alpha \in L_{\mathcal{A}}$ (strong realism) □

Third, we consider the *desired* worlds. By analogy with Veltman's normal worlds, these are are maximal in the preference order. They correspond to statements of the form good(φ). So w is desired when for all v $w \preceq v$ only if also $v \preceq w$. A preference order is called *coherent*, when there are indeed such desired worlds: the desires are mutually consistent. Coherence is not a constraint for mere preferences; it is a constraint for potential goals, as we have seen above. As a central constraint, we expect the agent to act in an *optimal way*. It should be rational in the sense that it maximises its preferences within the bounds of the possible. Potential goals are therefore defined as those realisable states that are maximally preferred. So w is *optimal*, when it is realisable and there is no other realisable world that is strictly more preferred.

Now it could be that there are more sets of optimal worlds. After all, the preference order is a partial order. Whenever there are several maximally preferred reachable sets of states, which are mutually incompatible, the agent faces a *decision problem*. We will require that decision problems induce an issue for the agent: which of the optimal potential goals to pursue. In order to resolve this issue the agent may ask a question, make observations or otherwise try to reduce the number of alternatives. However, the agent must make a choice. Indecision is irrational.

These constraints on the model could correspond to the following principles. This is only a first suggestion. If φ is a goal and there is a goal ψ preferred over φ, then ψ must be unrealistic. And further, if there are two desires φ and ψ, which are both possible but incompatible, the agent must wonder which one to adopt as a goal.

(116) $\models G_a\varphi \wedge \text{pref}(\psi, \varphi) \rightarrow L_a\neg{<}\alpha{>}\psi$, for any $\alpha \in L_{\mathcal{A}}$ (optimality)
$\models (\text{good}_a\varphi \wedge \text{good}_a\psi \wedge L_a(\varphi \leftrightarrow \neg\psi) \rightarrow W_a(?G_a\varphi; ?G_a\psi)$ (decision).

Throughout this chapter we have treated intentions and goals as synonymous. This is not entirely correct. Intentions are philosophical notions which can be modelled by the intended effects, goals, of the plans that agents have adopted to achieve them. Intentions are the links between goals and the plans to reach them. The fact that a plan is adopted to reach a goal, creates an intention. So we could add an operator 'intend' to express that an agent has adopted a particular plan: intend(a, α, φ) means that agent a has adopted plan α with intended effect φ. It can be defined as follows: intend$(a, \alpha, \varphi) \rightarrow G_a\varphi \wedge L_a[\alpha]\varphi$.

Now we come to an interesting constraint on goals and intentions: *persistence*. Earlier we argued that intentions stabilise the behaviour of an agent in fluctuating environments. Only when there are good reasons to re-evaluate a plan, the plan may be changed. In particular, a plan is changed when the motivation for adopting it is no longer valid, when its goal is achieved or when the plan has become impossible to execute. And because we analysed goals as the intended effects of a plan, the same holds for goals: only if there are good reasons to re-evaluate a goal, the goal may be changed. In particular, a goal is dropped when it is satisfied or when it becomes clear that it can never be satisfied.

4.6.1 Directive Updates

We defined are three types of dialogue acts, corresponding to three types of update: $.\varphi$ for assertives, including preference statements, $?\varphi$ for interrogatives and $!\varphi$ for directives. Each results in a particular kind of attitude. Directive updates correspond to dialogue acts that try to get somebody to do something. In particular: *propose*, *suggest* and *request*.

What is the type of content of a directive dialogue act? The correspondence in figure 4.4 suggests that they contain goals. Goals must be feasible, realisable by some plan and ordered on a preference scale. In other words, we need a structure that corresponds to the accessibility relation $R(G)$, just like the data set corresponded to $R(L)$ and the preference order corresponded to the comparison relation $\lhd$. Instead one could argue that the content of a directive is formed by action descriptions or plans, i.e. by the means to achieve those goals. The verbs that express directive dialogue acts can have both states and actions as their complement. As a matter of fact, they often have a complement that describes an object which is the intended result of an action. Compare for example "John requests a reservation for tonight" (object), "John requests that the clerk makes a reservation for tonight" (action) and "John requests to have a reservation for tonight" (state). Before we get into lexical semantics, just remember from chapter 2 that we can always turn an action α into a goal by some achieve(α) operator and we can turn a goal φ into an action by the bring_about(φ) operator. As a matter of fact, a good definition of achieve is the following: achieve$(\alpha) \equiv <\alpha>\top \wedge [\alpha]\top$. In a similar way, a good definition of bring about is the following: bring_about$(\varphi) \equiv <\varepsilon>\varphi \wedge [\varepsilon]\varphi$, for any arbitrary act $\varepsilon \in L_{\mathfrak{A}}$.

Consider a data-structure for scheduling actions: a plan-stack or *agenda*. To accept a proposal or request means to adopt a goal. By the link of goals and actions through intentions, that means that an appropriate action must be put on the agenda. In some cases however, what is put on the agenda is a delayed action. A reservation gives me the right to obtain tickets later. There are conventions about such delays. In our local theatre for example, the right to obtain tickets persists until half an hour before the show. What is a delayed action other than a commitment? A correspondence result similar to proposition 13 and 10 would show that the commitments an agent makes in dialogue, expressed by directives $!\varphi$, must be translated into appropriate planning and scheduling incentives. On the other hand, information states concern public information. Many of the planning and scheduling deliberations are typically private. If you make a commitment, I usually don't care how you manage, as long as you deliver. What matters is the commitment itself. This expresses again a difference between an external and an internal point of view.

4.6.2 Commitment

In the introduction to this chapter, we stipulated that an agreement reached in a transaction dialogue can be seen as a joint commitment to future action. In this section we give a brief account of commitment to fit this description. In chapter 2 we argued against 'we-intentions'; the kind of joint intention and commitment that characterise joint action, such as playing a duet, is to be defined in terms of individual intentions and common ground. Individual intentions were clarified above. That leaves only common ground to be defined. Despite the characterisation of the common ground from chapter 2 based on a situational *basis* (Lewis 1969), we use an iterative definition here. This is only for technical reasons.

The logical language is defined along the lines of definition 45, with the following additions, for each group of agents N: MB_N for mutual belief, JG_N for joint goal and commit_N for commitment.

Definition 66 (Mutual Syntax)
For every $N \subseteq Agt$ and $\varphi, \psi, \chi \in L_{\mathcal{P}}$ and ρ a set of rules, extend definition 45 with
$\varphi ::= MB_N\varphi \mid JG_N\varphi \mid \text{commit}(N, \varphi, \psi, \chi, \rho)$ □

The definition makes use of auxiliary notions $EB_N\varphi$ and $EG_N\varphi$. to mean that every agent in N believes φ or has the goal φ.

Definition 67 (Mutual Belief; Joint Goals)
For each $N \subseteq Agt$ and $\varphi \in L_{\mathcal{P}}$ define by recursion

$$\begin{array}{lllllll} EB^0_N\varphi & = & \varphi & EG^0_N\varphi & = & \varphi & \\ EB^{k+1}_N\varphi & = & B_a EB^k_N\varphi, & EG^{k+1}_N\varphi & = & G_a EG^k_N\varphi, & \text{for all } a \in N \\ MB_N\varphi & = & EB^k_N\varphi, & MG_N\varphi & = & EG^k_N\varphi, & \text{for all } k \in \{0, 1, \ldots\} \end{array}$$ □

The mutual belief and joint goals definitions have similar properties to single agent beliefs and goals. So MB has a kind of $KD45$ logic, with distribution, consistency, positive and negative introspection and necessitation. MG is again KD kind of logic with distribution, consistency and interaction axioms with MB. The factive counterpart of belief, knowledge, can also be given a mutual counterpart: common knowledge. This is a version of an $S5$ logic. See Wooldridge and Jennings (1999) and Fagin et al. (1995) for proofs.

With those definitions, we can characterise commitments. Earlier, we discussed the social and ethical 'glue' that keeps commitments together. The obligation to abide and the feeling that it is wrong not to keep committed. The result of this, is that commitments *persist*. This persistence element holds both for single-agent commitments and for joint commitments. In fact, it was taken to be the defining element of single agent intentions by Cohen and Levesque (1990).

We now sketch the account of commitments given in Wooldridge and Jennings (1999). Each joint commitment of a group of agents N is characterised as a complex notion $\text{commit}(N, \varphi, \psi, \chi, \rho)$, that involves an immediate *goal* φ, a long-term *motivation* ψ, *preconditions* χ and set of *conventions* ρ to define the interaction protocol. Commitments concern goals that are meant to be kept, but only relative to a motivation, and as long as it is practical. If we assume that commitments can be nested, just like goals and subgoals, the commitment structure takes the role of an agenda, as suggested above. In case the motivation is achieved, the commitment for the immediate goal may be dropped. And similarly, if another plan becomes available which is better for achieving the over all goal, again the commitment to the immediate goal may be dropped. Commitments may also be dropped when its motivation becomes undesirable, because of a chance in preferences. That means that both the immediate goal, related to a plan, and the underlying motivation must be part of the notion of commitment.

Interestingly, Wooldridge and Jennings (1999) explicitly make use of *conventions* to indicate the general circumstances under which a commitment is to hold, and when it is to be abandoned. This parameter can be filled in differently for different applications. For example, it may come to contain the sanctions upon not keeping a commitment. A convention is a list of n rules of the form: if condition, then action. Rules are made specific for different roles $x_1, .., x_n$in the ineraction. So formally, $\rho = \{\mu_k \longrightarrow \gamma_k \mid k \in \{1, ..., n\}\}$. The idea is that each time a trigger condition μ_k is true, the agent that performs role x_k will adopt the corresponding goal γ_k.

Here is a rough approximation of Wooldridge and Jennings account. Because we have no explicit temporal operators we use ε as a stand-in for any arbitrary action. Thus $<\varepsilon>\varphi$ expresses that φ is feasible and $[\varepsilon]\varphi$ expresses that φ is inevitable. The operator $(\varphi \text{ U } \psi)$ indicates that φ will be the case *until* φ becomes true[12].

Definition 68 (Commitment (sketchy))
For ρ a set of rules $\{\mu_k \longrightarrow \gamma_k \mid k \in \{1, ..., n\}\}$, define

$$\begin{array}{lll} \text{commit}(N, \varphi, \psi, \chi, \rho) & = & \chi \wedge ((p \wedge q) \text{ U } r), \quad \text{where for all } a \in N \\ p = G_a\varphi, & & \text{(motivation)} \\ q = \bigwedge k(B_a\mu_k) \rightarrow (G_a\gamma_k \text{ U } r), & & \text{(maintain convention)} \\ r = \bigvee_k \gamma_k & & \text{(termination).} \end{array}$$

□

By means of example Wooldridge and Jennings recreate a Cohen and Levesque-type persistence as the conventional conditions. Agents must believe the immediate goal φ is not yet achieved, and that as soon as they find out that it is achieved, or that has become impossible or that it is no longer motivated by ψ, they must tell the other agents.

(117) A joint commitment commit$(N, \varphi, \psi, \chi, \rho)$ persists when

$$\begin{array}{lll} \chi & = & \neg L_a\varphi \wedge L_a<\varepsilon>\varphi, \\ \rho & = & \{B_a\varphi \longrightarrow MB_N\varphi, \\ & & B_a[\varepsilon]\neg\varphi \longrightarrow MB_N[\varepsilon]\neg\varphi, \\ & & B_a\neg\psi \longrightarrow MB_N\neg\psi\}. \end{array}$$

These rules correspond to cooperativity principles: if you believe the joint action is completed, tell everybody; if you believe it has become impossible, tell everybody and if you believe it is no longer needed, tell everybody too. The fact that these conventions turn up in the middle of a characterisation of joint commitments, is further evidence that the kinds of conventional interaction patterns that are modelled by dialogue games are nothing but recipes for joint action. The rules specify coherence constraints at the level of beliefs and intentions. Other constraints apply at other levels. Obviously, a more precise comparison between interaction patterns and the conventions used by Wooldridge and Jennings, or indeed between interaction patterns and the recipes of Grosz and Kraus (1996), need further research.

[12]Define $(\varphi \text{ U } \psi) \equiv \text{O}((\text{test}(\neg\psi); \text{test}(\varphi))^*; \text{test}(\psi))$, where $\text{O}(\alpha) \equiv \langle\alpha\rangle\top \wedge [\alpha]\top$.

4.7 Summary and Further research

In this chapter the task structure of dialogues was explored. We argued that an account of the apparent goals of dialogue participants is crucial in understanding the structure of dialogue. We gave an account of the activity type of negotiation and presented a formal model of the attitudes of agents in dialogue, based on the standard BDI-paradigm, but influenced by update semantics, semantics of questions and answers and by preference orders.

We showed how tasks can be modelled by complex actions. Both complex and atomic actions are characterised by applicability conditions, preconditions for success, success conditions that define the intended effect and failure conditions that indicate what must remain true even if the action fails. Moreover, we argued that many of the aspects of language understanding and generation that have been relegated to the 'background', are in fact dependent on the underlying goal of the communication. Goal supply additional structure, such as the scales that are used to rank the relative strength of an answer. In particular an activity based goal analysis helps to explain the granularity of appropriate answers. It explains why in some cases partial answers are sufficient. In general when a question is asked to resolve some issue that is related to the precondition of a underlying goal, the answer to that question is appropriate in case it enables the asker to pursue its goal further: the precondition is satisfied. That explains why a partial answer is enough for many 'mention some' readings of wh-questions. The goal requires only a single answer. It also explains additional helpful comments, that were strictly speaking not asked for, as when the train attendant adds the platform number.

As an example of a particular activity type, we studied the five phases of a negotiation: open, inform, propose, confirm and close. Each of these phases can be characterised by the change it brings to a negotiation space. Opening establishes contact, and sets up an initial space. After that information is exchanged about the non-negotiable properties of the space. The negotiation proper takes place in the proposing phase, where requests, suggestions proposals and counter proposals are exchanged. Now crucially, proposals are ranked on a preference order. The strongest active proposal that remains, is finally confirmed in the conformation phase. The closure phase ends the contact and maintains social obligations, even if the activity itself was unsuccessful. These phases were illustrated by an example dialogue in an ice cream parlour.

The chapter continued with a formal account of the attitudes of agents in dialogue. We developed a modal logic for belief, preferences and goals alongside an update semantics to account for the changes in the apparent attitudes of agents. The effect of a dialogue act is an update of the information states of agents. Belief can be accounted for directly by means of information states. As a nice side-effect attitudes with interrogative complements, such as wonder, can also be directly accounted for. Moreover, we defined a notion of awareness that marks the difference between explicit and implicit belief. An awareness function as suggested by Fagin and Halpern (1988) is provided by issues. An agent is aware of a proposition, whenever that proposition is a possible direct answer to one of the issues the agent is entertaining at that point. In other words, explicitly raising an issue has as additional effect that agents become aware of the various alternatives. In this way we could partly solve the open problem of chapter 3: when is raising an issue appropriate or relevant? An issue is relevant when it is related to the current general issue, topic or task, and when the participants are not yet or no longer aware of this dependency.

With respect to preferences, we relied on the insights of Von Wright (1963). Analysing the desires and preferences of agents, as well as the priorities among obligations or ethical considerations is an example of qualitative decision theory. Out of the many dimensions of choice for an agent only some remain because of the idealised circumstances of dialogue. In particular, various aspects of cooperativity make it possible to reduce normative and ethical aspects, as well as aspects of ability, to the BDI-dimensions.

To model preference, we used both an irreflexive and asymmetric comparison relation among sets of worlds and a reflexive weak preference order among individual worlds. One can be turned into the other. Weak preference orders are made part of information states of agents to account for the changes in the apparent preferences of an agent. Initially all worlds are equally preferred. An update with a preference statement of the agent itself results in a refinement of the order: all pairs of worlds that are incompatible are eliminated. We use the conjunction expansion principle of Von Wright that analyses a relative preference of φ over ψ as a preference of φ and non-ψ worlds over non-φ and ψ worlds. In addition we require that the order remains reflexive and transitive. Relative preference often involves a mental comparison between non-actual situations. Therefore, we use selection functions to enforce a minimal change requirement, to avoid paradoxes of deontic logic.

Preferences motivate goals. Unlike preferences, goals must be rational. They are consistent and closed under consequence. They must be realistic too: goals are included among the beliefs of an agent. Moreover, goals must be realisable: the agent must believe that there are actions to reach the goal. Finally, they must be optimal: among the realisable worlds no other worlds are more preferred. Since the preference order is partial, there may be more sets of mutually consistent and maximally preferred goal states. In that case the agent is faced with a decision problem, which corresponds to an issue which potential goal to pursue. The agent must choose eventually; indecision is irrational because then neither of the potential goals is reached. We predict that in such cases the agent should ask a question for more information that may help to resolve the dilemma.

One of the aspects that distinguishes a goal or intention from a mere desire is persistence; the agent will continue to try to achieve it, until it is achieved or until it has become clear that it is not achievable. This aspect may be called a commitment (Cohen and Levesque 1990). More often than not, a commitment is made to other people. Such joint commitments are the result of a transaction dialogue, or are implicitly agreed by conventional means. The 'glue' that makes participants keep their commitments to other people is a social mechanism. Keeping commitments is a matter of obligation (Traum and Hinkelman 1992), but also common decency or consideration play a role (Allwood et al. 1992).

4.7.1 Further Research

Some of the material presented in this chapter remains rather sketchy. Partly it borrows techniques from modal logic; partly it borrows techniques of update semantics. The logical properties of the combination of assertive, interrogative and directive types of update with the modal attitudes, have not been explored well enough. For example, the logics for each of the attitudes have not been shown to be sound and complete. More importantly, we have not fully investigated the interaction of the various attitudes. We need principles such as the positive or negative introspection principles between belief and goals. Are agents aware of what they are wondering? Are preferences believed?

As we mentioned in various places, the feasibility requirement for goals, and also the mental comparisons involved in preference judgements are speculative. One makes a prediction about a non-actual world based on expectations and default rules of inference. Therefore a proper account of both practical reasoning and preferences should incorporate some non-monotonic logic. And as we claimed earlier, the most obvious candidate in the current framework is Veltman's (1996) work on expectation patterns.

Interestingly, for dialogue many of the default rules of inference will be concerned with the expected behaviour of other dialogue participants when a certain dialogue game is initiated. Because a particular exchange pattern conventionally requires a particular continuation, for example a greeting requires a greeting in return, participants can presume that they will continue that way. Such dialogue games are the topic of the following chapter. To accurately model reasoning with such conventional recipes for joint action we need default rules. Interestingly, this is exactly the approach taken by Asher and Lascarides (1998a). They define defeasible rules of inference based on coherence relations. Other evidence along the same lines is found in Walker (1996b). Walker needs default rules of inference to account for non-explicit acceptances and rejections. Often, implicit acceptance or rejection can be inferred as a form of scalar implicature. More research is needed to link the dynamically changing preference orders of this chapter with the scales of strength that help to identify acceptance and rejection.

The ice cream example showed how difficult it is to accurately annotate an example in full semantic detail. In particular, the redundancy between the various kinds of dialogue acts and attitudes, and the assertive, interrogative and directive updates suggests a theory that might explain how a dialogue act could be reduced, in the end, to a mere label for a certain combination of task-related function, interaction related function and semantic content. Every researcher has a favourite list of dialogue acts for each application. We believe that an analysis of the primitive function and content types can help to motivate the choice of dialogue acts for a given application. Each act can be assembled from primitive components, as it were.

In chapter 2 we stressed the importance of partial planning techniques to explain joint actions, such as the actions involved in communication. We hypothesised that dialogue games, which model conventional interaction patterns, function as recipes for joint action. In this chapter we dealt with a lot of the machinery to get this hypothesis to work. Nevertheless, we largely remained within a single agent view point of what is essentially a multi-agent dialogue situation. The joint aspects clearly need to be worked out in more detail. The mechanism by which agents negotiate the division of labour and scheduling of different sub-tasks in various application domains, still needs to be explored (Grosz and Kraus 1996). That involves for each of the agents deliberations of the kind that we have modelled in this chapter: balancing personal preferences and abilities against the commitments made to other agents.

Finally, in chapter 1 we set out to find theoretical properties that contribute to usability. Given the formal system developed in this chapter, do we get the right formal properties to characterise usability with? In chapter 1 we hoped that we could characterise coherence as the relative measure of 'fit'. Does an utterance fit the context with respect to the form, content, task and interaction? Fitness with respect to the task was the topic of this chapter. Roughly, we can say that an utterance fits with respect to the task, when it contributes to resolve the various issues that are raised by a particular task or sub-task in that phase of the dialogue.

Chapter 5

Coherence

This chapter explains when a sequence of utterances can be called a *coherent* dialogue. An utterance is coherent with respect to a dialogue context, when it can be attached to the dialogue context by means of a coherence relation. For dialogue, particular coherence relations correspond to exchange patterns, that indicate well-formed sequences of initiatives and responses. Exchange patterns can be described by the rules of a dialogue game. We argue that dialogue games are conventional recipes for joint action. The moves in a game correspond to dialogue acts. For each of the phases in a negotiation dialogue, the typical dialogue acts are given a characterisation.

5.1 Introduction

A dialogue is defined as a coherent sequence of utterances by the dialogue participants. When can a dialogue be called coherent? In chapter 2 we characterised a dialogue as a combination of joint actions scheduled at different linguistic levels. This requires careful coordination. In this chapter we are going to look in detail at the exchange patterns needed to accomplish this coordination process. The result of the coordination process is a coherent dialogue. Coherent dialogues exhibit a number of properties. For example, the syntactic and semantic structure of adjacent utterances, is often parallel. The choice of words and intonation patterns are adjusted to the function of an utterance in context. In language generation, these properties can be formulated as coherence constraints. In a way, coherence constraints are but a by-product of the need to be understood with as little effort as possible.

In this chapter we focus on the *form* of utterances, including the prosodic characteristics, and on the *structure* of the dialogue context. Coherence involves topicality and salience and it involves the earlier aspects of content and task too. In chapter 3, a dialogue was studied as a constant process of raising and resolving issues. This provides a rather technical characterisation of the notions of consistency, relevance, informativeness and licensing and thus of the pertinence of an utterance given the current issues. A characterisation of the relevance of a particular issue at a certain point in dialogue could not be given, at least not by logical and semantic means alone. For this we need extra-logical aspects, in particular the task structure and dependency relations which are related to aspects of salience and topicality. In chapter 4 we described dialogue acts as contributions towards some underlying task or goal. However, this still leaves the interaction between utterances and dialogue structure unaccounted for. That is the topic of this chapter.

When we look in detail at the communicative functions of utterances in dialogue, it is obvious that some utterances are not (only) used for information exchange to achieve some goal. They serve to control the interaction process. For this reason, some theorists have extended the notion of a speech act to become a dialogue act, covering also the so called dialogue control acts (Bunt 1989; Allwood et al. 1992). Greetings and thanks, for instance, serve to establish a reliable interaction channel and to keep social relations between participants in order. Acknowledgements indicate that some information was indeed communicated. So it seems that dialogue control acts are actions that are part of a plan to manage the dialogue. However, we do not need to analyse each interaction in terms of the underlying intentions and motivations alone. In a corpus, greetings and thanks simply mark the beginning and end of a dialogue, acknowledgements just follow assertions or suggestions and answers usually follow questions. Such stereotypical sequences of dialogue acts are called *exchange patterns*. It is certainly possible to postulate specific dialogue control related intentions and plans and to analyse exchange patterns in terms of these plans. However, it seems that exchange patterns are 'compiled-out' plan recipes, that are no longer consciously deliberated by the agent, but conventionally triggered by the circumstances. At the surface level such exchange patterns can be fruitfully studied and applied. The only exception may be misunderstandings. Because all utterances are motivated by some goal, a plan-based account has in principle a method for backtracking and finding a repair mechanism (Ardissono et al. 1998). So although exchange patterns are best studied and applied at the surface level, providing them with a plan-based justification, may help in the case of misunderstanding.

The chapter is structured as follows. In the following section we give a brief overview of approaches that can be used account for exchange patterns. In section 5.3 we examine dialogue game parameters based on work by Mann (1988). In section 5.4 we discuss the activity type of negotiation. The phases of chapter 4 are discussed again, with for each phase a characterisation of the dialogue acts that are most typical. In section 5.6 we examine implicit acceptance and rejection. The conclusion adds some preliminary ideas on the notion of coherence, as can be applied in chapter 1.

5.2 Approaches to Dialogue Structure

In this section we discuss different approaches to dialogue structure and exchanges: dialogue games, discourse grammars and Asher's coherence relations. We argue that they are similar and that their insights can in principle be combined and incorporated in a dialogue game framework in a wide sense.

5.2.1 Dialogue Games

A useful metaphor for studying exchange patterns is that of a *dialogue game*, in which each participant plays a role and expects the others to play their respective roles. Each participant can only make the moves that are allowed for its role at that point in the game. The state of the game, the dialogue game board if you like, determines which moves are allowed. But each move in turn determines the state of the game. The dialogue game board thus functions as a semantics to moves in a game. Moreover, there is the aspect of winning the game. From all the moves that are allowed, some are more useful than others with respect to the objective of the game. A move is useful when it is allowed and when it contributes most to the underlying task. So the dialogue game metaphor involves all aspects of coherence. The form indicates how to recognise moves and which moves are possible at all. The content determines how the move relates to the dialogue game board. The function of the move in relation to the game has both an interaction related aspect, is the move allowed at this point, and a task related aspect, how does the move contribute to winning the game? Therefore we argue that a representation based on dialogue games in a wider sense, is suitable to incorporate all these levels.

We start our overview with Wittgenstein's (1953) notion of language games. A Wittgensteinian language game involves all aspects of language use, in contrast to a view of language in which words are merely names or descriptions of objects used to talk about them. "Whereas in fact we do the most various things with our sentences. Think of exclamations alone, with their completely different functions. Water! Away! Ow! Help! Fine! No! Are you inclined still to call these words 'names of objects'?"(Wittgenstein 1953, §27). Many expressions get their meaning from the function they perform in the language game. This principle can be paraphrased in the slogan: *meaning is use*. Crucially, the same words or utterances have different meanings in different activities. Compare a court room with a kitchen table conversation. The term 'language game' is meant to illustrate that the speaking of language is part of an activity or form of life. Earlier we referred to such an approach as an activity based approach (Allwood 1995).

Different aspects of the game metaphor have been highlighted in different areas of research. Starting with von Neumann and Morgenstern (1944) there has been extensive mathematical and economic research into game theory. Here one is interested in the rules of a game, which moves are allowed, and in strategies for selecting moves that contribute to winning the game. There is also an elaborate logical theory of games, in which a winning strategy is compared to a valid argument structure. Games can thus be used as a semantics to logical expressions. Similar to semantic tableaux, a game representation is constructed. Inspired by such logical theories, Carlson (1983) uses games for a semantics and pragmatics of utterances in dialogue. Lewis (1979) develops the notion of a *dialogue game board* as a metaphor to understand the development of a basis for the common ground. In that respect Lewis' ideas are similar to Hamblin's (1970) *commitment slates*: representations of the commitments each participant has publicly made in the dialogue. Moves can be described in terms of the changes to a dialogue game board.

Dissatisfied with the focus of speech act theory on single sentences, researchers in the seventies proposed dialogue games as a way to combine different speech acts (Levin and Moore 1978; Mann 1988)[1]. This use of the idea of a dialogue game stresses the importance of patterns of interaction which can be identified at the surface level. Such patterns indicate which moves are allowed. However, initiating a game and making a move do have a distinct motivation behind it in this theory, formulated in terms of goals. Naturally, the theory lacks a formal account of plans and goals, let alone joint plans and goals. But when we examine the *illocutionary point* of a dialogue game, the motivation for the initiator of a game to start it, we find that it is usually a joint goal. That further motivates our hypothesis that these kinds of dialogue games are precisely the sorts of recipes for joint communicative action that are needed if we want to apply theories of joint planning and action (Grosz and Kraus 1996) to communication. We come back to this particular version of the dialogue game metaphor in section 5.3.

[1]The article by Mann (1988) was first presented in the seventies, but only published in 1988.

A more recent wave of researchers use the metaphor of a dialogue game for empirical research on exchange patterns in large corpora of spoken dialogue (Houghton and Isard 1987; Kowtko et al. 1992; Carletta et al. 1997). We mention the MAPTASK corpus, collected at the HCRC in Edinburgh, the TRAINS corpus, collected at SRI Cambridge and corpus research for the spoken translation system VERBMOBIL. The set-up of the MAPTASK is as follows (Anderson et al. 1991). There are two subjects separated by a screen. Both have maps, which differ slightly. One participant, the leader, has a route indicated on the map. Her task is to communicate the route with the other participant, the follower, so that he can follow the route and draw it on the map. Because maps differ, misunderstandings frequently occur. Acknowledgements and grounding behaviour are very important in the MAPTASK. Aspects that have to do with coreference resolution in dialogue, the way objects are described and the different function of utterances have been described. The TRAINS set-up involves two participants who have access to a railroad yard with toy trains (Allen et al. 1995). Several types of cargo are placed at different locations. The task is to deliberate and negotiate a joint plan for moving and combining trains, in order to get certain cargo from one location to another. Because the task is complicated and requires extensive communication, again acknowledgements are important. Note that for the TRAINS task, plans and goals are part of the topic of the dialogue, as well as a potential means to describe the dialogue. The set-up of VERBMOBIL involves two users, neither of whom are native speakers of English (Kay et al. 1994; Alexandersson 1996; Alexandersson et al. 1995). They want to schedule an appointment. Therefore they negotiate a date and time, using English initially. The system follows the negotiation and keeps track of the proposals. At any time, a user may request a spoken translation of an utterance in her native tongue, into English or into the language of the other participant. Keeping track of the proposals and the reasons for acceptance and rejection are the major challenges of this application.

One of the research goals behind these corpora is to find a consistent annotation scheme. Utterances in the corpus are labelled or annotated with their communicative functions, at several levels: what *move* an utterance represents within what *exchange*, described as a conversational game, and what *activity* the exchange is again part of. A second research goal is to find a segmentation scheme that allows one to cluster utterances into separate dialogue segments with a single topic. Typically a segment corresponds to an exchange, but often the boundaries are hard to draw. One of the applications of this segmentation and annotation effort, apart from increased insight, is that machine learning techniques can be used to make generalisations and predications. On the basis of the sequence of move types one can predict the most likely next move type (Alexandersson 1996; Andernach 1996). Such predictions may be generalised into a set of empirically validated conversational game rules. It is clear that such expectations and generalisations can help to improve the response selection of a dialogue manager in a dialogue system application.

However, first the corpus must be annotated by human experts. A full speech act theory is inadequate for this purpose. If a theory needs to be suitable for annotation and segmentation, there must be clear surface level elements on the basis of which classification decisions can be made. In this empirical line of research, people have come up with a number of hierarchically ordered exchange patterns that are called *conversational games*. Games are sequences of moves. Moves can be initiatives or responses. Carletta (1996) defines a decision tree for annotators, based on surface and content elements. Carletta et al. (1997) have studied the reliability of such annotation schemes. They propose the *kappa metric* to calculate inter-annotator agreement. The kappa coefficient κ measures pair-wise agreement among a set of annotators making judgements, correcting for chance agreement: $\kappa = (P(A) - P(E))/(1 - P(E))$, where $P(A)$ is the proportion of agreements and $P(E)$ is the proportion one would expect agreement by chance. For moves in the MAPTASK corpus, inter-annotator agreement was about 85 % for expert coders and about 70 % for novice coders, depending on the different categories. There are a number of reasons for these figures. First, there is genuine ambiguity. Second, some of the categories do not differ in the kind of response that they predict; humans know how to respond but not how to categorise. For example checks and yes/no questions were frequently confused. Third, the ambiguity in the function of an utterance has a purpose. It is up to the hearer to take the utterance in a certain way. Which way it is taken, is revealed by feedback. Still one might ask: if humans already find it difficult to decide on the communicative function of an utterance, how could a machine?

One way to increase the suitability of an annotation scheme for automatic detection is to use surface level features only: lexical entries, word order and cue words. There are machine learning techniques for clustering utterances on the basis of surface features, such that the resulting classes, when applied, give an optimal segmentation of the dialogue into separate chunks. The resulting classes, described by a combination of surface properties., can then play the role of a move or dialogue act. Andernach (1996) applied this approach to the SCHISMA corpus. Initial results on the clustering were promising; indeed surface level information generated sensible distinctions. The experiments were not continued. One of the difficulties concerned the amount of application dependent information that needed to be annotated.

A similar concept to the empirical conversational games is used for specification of the interaction model in the influential SUNDIAL project (Bilange 1991) and subsequent implementations of dialogue systems (Jönsson 1993; Jönsson 1997). Here an exchange is called an *IR-unit*. It consists of an *initiative* by one of the speakers and an expected *response* to that initiative. There is also a separate *evaluation* step in which the recognised material is confirmed. Despite the focus on surface level aspects, this model still reserves a large role for a separate task model. The dialogue control structure described by IR-units does not provide a specification of the content of the dialogue acts. The decision which information is missing and needs to be asked for, must be made on the basis of the task. Bilange proposes a model that interprets utterances in context and attempts to resolve coreference links. In case of a reference failure, the module may initiate a repair.

5.2.2 Discourse Grammar

Another way of studying discourse structure, is that of a *discourse grammar* (Polanyi and Scha 1984; Polanyi 1988; Prüst et al. 1994). The hypothesis is that such techniques also work for dialogue. Just like a sentence, a discourse can be segmented into constituents. For a text these would be paragraphs and sentences. For a dialogue the constituents are dialogue segments, utterances and possibly phrases. Wherever a phrase like an NP or VP is needed as a separate level constituent in the discourse grammar, it could be argued that the phrase constitutes an utterance in its own right. Discourse grammar rules can describe and prescribe what sequences of constituents are well-formed. By parsing the discourse we get a parse-tree, which forms a discourse representation structure. Ideally, we would have incremental parsing: new constituents can be continuously attached to the discourse representation structure.

What does this representation structure look like? It is a tree-like structure. At the leaves of the tree we find the basic discourse constituent units (DCUs). Branches are labelled with *discourse relations* that specify the functional relation of the constituent with the rest of the discourse. Compare syntactic labels like subject, or direct object. Typical discourse relations are *narration*, *elaboration*, *causation* and *contrast*. Which discourse relation is applied, depends on several features of the utterance and the immediate context. For example, a so called *cue word* like 'but' indicates that there is a relation of contrast between the latest constituent and the previous one. This approach is similar to rhetorical structure theory (RST) that was originally developed for text analysis (Mann and Thompson 1988). Discourse relations correspond to the rhetorical relations, that indicate the relationship between various segments of the text. Discourse relations are also called coherence relations (Asher and Lascarides 1998a).

Discourse grammars have been particularly successful in the area of coordination and ellipsis (Prüst et al. 1994; Grover et al. 1994). Once there is a discourse structure that provides relations between DCUs, it is easier to find to what existing part of the tree an elliptic element must be connected to. The approach uses typed feature structures to represent attributes of objects, and assumes an extensive type hierarchy that defines what objects and are instantiations of more and less specific types (Carpenter 1992). The original idea is to use higher order unification, see also (Shieber et al. 1996). Parallel elements are discovered on the basis of surface level cues and intonation. For such parallel elements, the most specific common denominator (MSCD) is found. This is a type that is common to both. In the type hierarchy, it is the nearest parent that dominates both elements. In example (118)) below, 'insects' dominates both 'ants' and 'beetles' and emotional attitude dominates both 'hate' and 'like'. Now the meaning of the target utterance can be reconstructed by unifying the MSCD with the representation of the target. Note that the parallel anaphor 'she' is resolved to Hannah.

(118) Hannah like ants. She hates beetles.

$$
\begin{array}{lll}
\text{source:} & \mathit{like} & \begin{bmatrix} \text{agent:} & \mathit{hannah} \\ \text{patient:} & \mathit{ant} \end{bmatrix} \\
\text{target:} & \mathit{hate} & \begin{bmatrix} \text{agent:} & \mathit{female} \\ \text{patient:} & \mathit{beetle} \end{bmatrix} \\
\text{MSCD} & \mathit{emot\_att} & \begin{bmatrix} \text{agent:} & \mathit{female} \\ \text{patient:} & \mathit{insect} \end{bmatrix} \\
\text{unified:} & \mathit{hate} & \begin{bmatrix} \text{agent:} & \mathit{hannah} \\ \text{patient:} & \mathit{beetle} \end{bmatrix}
\end{array}
$$

Please note that issues provide a semantics to the MSCD structure! This little discourse discussed the emotional attitudes of Hannah with respect to insects. So that issue is what structures the discourse. What we ignored in chapter 3 were the representational aspects of such an account, and the type hierarchy. Grover et al. (1994) apply default unification to account for similar data. In default unification, more specific information in a recent utterance overrides less specific information in the corresponding slots. In this way the effect of taking the MSCD and then unifying with the target, can be approximated in one step.

The same idea can be applied in dialogue as well. Witness the following exchange with the current SCHISMA prototype. Figure 5.1 shows what the dialogue structure for this example would look like. This type of 'en'-continuation is very common in the SCHISMA corpus.

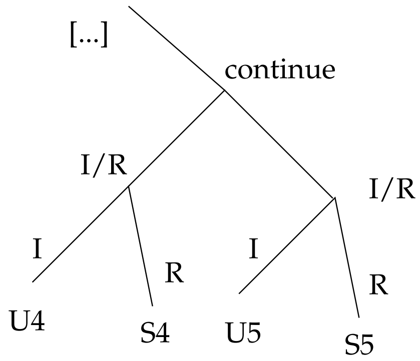


Figure 5.1: Dialogue structure for example (119).

(119) [...]
U4: hoe laat begint fame?
What time does fame start?
S4: De voorstelling "Fame"begint 18 februari om 20.00.
The performance "Fame" starts Februari 18 at 8 pm.
U5: en ernst jansz?
en ernst jansz?
S5: De voorstelling "De overkant" begint 18 februari om 20.00.
The performance "De Overkant" starts Februari 18 at 8 pm.

In U1 the user asks for the starting time of the performance "Fame". This performance thereby becomes the topic. The issue is the starting time of the performance. In S1 the system answers the question. A question is form of initiative. So the coherence relation of S1 with respect to the dialogue, is that of answering. In general, each initiative-response pair, corresponds to a coherence relation (see below). In U2 we have an utterance "en ernst jansz", which needs to be connected to the dialogue structure at the most appropriate place, and by some coherence relation. Because of the cue word 'en' the coherence relation is: *continuation*. By looking for parallel elements to "ernst jansz", in the preceding utterance, we find the performance "Fame". Replacing this with new information, but keeping the rest of the old utterance, we get the intended meaning. The user continues the issue of starting time, but changes the topic to the performance by Ernst Jansz.

5.2.3 SDRT

A theory that deals with similar data is Segmented Discourse Representation Theory (SDRT) (Asher 1993; Asher and Lascarides 1998a; Asher 1998; Asher and Lascarides 1998b; Lascarides and Asher 1999). SDRT was originally developed to model narrative text structure and tense and aspect. Segments of the discourse are represented by regular DRSs. Just like for discourse grammars, segments are connected in a hierarchical structure by means of *coherence relations*. One of the motivations behind this framework is that discourse structure conveys information about the content too. Take the following standard example of narrative structure.

(120) a. John fell. Bill pushed him.
b. John fell. Then Bill pushed him.

In (120a) we assume that the second constituent must be relevant to the first constituent. The most relevant issue with respect to the first constituent is: what caused this event of John's falling? So the coherence relation is one of causation or explanation. By assumed relevance we infer that John fell because Bill pushed him. Therefore we infer that Bill's pushing was prior to John's falling. But in (120b) the cue word 'then' suggests that the discourse relation must be one of narration or continuation. The cue word makes explicit that the event of John's falling is prior to Bill's pushing. So the way constituents are attached to the dialogue structure implies important information about the content of a dialogue. Also note that the inference is defeasible. The initial inference in (120a) is defeated by more specific information in(120b): the cue word 'then'.

For dialogue, coherence relations correspond to typical exchange patterns. For inquiry dialogues Asher and Lascarides propose the following coherence relations: *indirect question answer pair* (IQAP), *not enough information*, *question elaboration* (Q-ELAB), *explanation*, *result* and *acknowledge*. How do we know by which particular coherence relation a constituent is to be attached? There are rules for inferring to what constituent a new segment should be attached and by what coherence relation. The rules are based on the task and activity type, the content of the constituent, the current topic, syntactic and prosodic information and specific cue phrases. Because the inference is typically defeasible, Asher and Lascarides apply a non-monotonic logic. These defeasible *coherence principles* play the same role in the theory as the discourse grammar rules alluded to above. A formula $\varphi \rightsquigarrow \psi$ means that if φ is the case, then *normally* ψ holds. Coherence principles are of the form (121), where $\langle \tau, s_1, s_2 \rangle$ means that discourse segment s_2 is to be attached to discourse segment s_1 by means of some coherence relation R, and where s_1 is part of the discourse representation structure τ and where 'some stuff' is some additional information about s_1 and s_2 that can be inferred from their representations. The specification of s_1 and s_2 involves aspects of syntax, semantics and prosody.

(121) $(\langle \tau, s_1, s_2 \rangle \wedge \text{some stuff}) \rightsquigarrow R(s_1, s_2)$

With respect to the coherence relations for dialogue, Asher and Lascarides make a link between the utterance and the purpose with which it was said. They call this purpose the SARG, for *speech act related goal*. It corresponds to what we have called the task-related aspect of the function of a dialogue act. Calculating coherence requires that the SARG of each utterance is inferred. Based on SARG one can infer a possible underlying plan or goal. Consider the coherence relations IQAP and Q-ELAB. The IQAP relation is concerned with what we have called the resolution of issues. Issues can be raised explicitly or implicitly. A so called QAP, a question answer pair, is a special case of an IQAP, an indirect question answer pair. Two utterance representations u_1 and u_2 are in relation IQAP(u_1, u_2) when u_1 somehow triggers a question or issue to become salient, and u_2 conveys information that is an answer or solution to that issue. The Q-ELAB relation is concerned with formulating queries that correspond to the preconditions of a goal. So, two utterances u_1 and u_2 are in relation Q-ELAB(u_1, u_2) when u_2 is a question of which the answers will help to achieve part of the plan suggested by the SARG of u_1. For each coherence relation, a number of constraints apply. Moreover, general constraints such as sincerity and cooperativity apply. Based on these, one can infer the effects on the dialogue representation structure and on the information of the dialogue participants.

5.2.4 Coherence Constraints

We discussed three approaches to interaction patterns: dialogue games, discourse grammars and SDRT. They share a number of common elements: a set of utterance types or moves, and a way to recognise and distinguish them. A set of grammar rules or defeasible dialogue principles that express what sequences of moves are well-formed. And a set of coherence relations that apply to well-formed exchanges, and further constrain appropriate exchanges on the basis of content and form.

It is a good topic for further research to see how insights from SDRT, discourse grammars and dialogue games can be combined. To implement these combined dialogue games we would choose either a default logic for the meta-principles and a standard semantic theory like DRT for the content, as in Asher's theory, or else a unification-based grammar with typed feature structures for the content. In a unification grammar, coherence constraints can be formulated as a kind of agreement relations over constituents. With respect to the content, we suggested that contributions must be consistent, relevant, informative and licensed. With respect to the task, the intention that underlies a constituent must contribute to a general plan to achieve the current goal.

The following example concerns *indirect speech acts* (Searle 1975). It is meant to show how such an integrated theory might operate. Although it is possible to infer the underlying meaning from the literal meaning via intentions and practical reasoning, we can do so right away, by convention.

(122) U: Kan ik bij u ook kaartjes reserveren voor vanavond?
Can I reserve tickets with you at all for tonight?
S: Ja, er zijn de volgende voorstellingen op dinsdag 14 oktober: ...
Yes, we have the following performances on Tuesday, 14th of October: ...

The user utterance in (122) is literally a yes/no question, asking if it is possible to reserve tickets. The system can answer the question by doing a database look-up to find out and displaying the result. Note that 'yes' is not enough. A positive answer to a yes/no question is often accompanied by a 'witness' of the positive results. Similarly a negative answer is often accompanied by an apology or explanation. But (122) can also be taken as a request. A precondition of a successful reservation is that there are tickets available. Often we find that people use preconditions of actions to request the action itself. A direct request limits the other participant's freedom. An indirect request leaves it to the other participant to decide to how to take the utterance. This reduces negative face (Brown and Levinson 1987). Because questions for preconditions are such common ways of making a request, we can also install a rule of the form: $(\text{ask}(a, b, ?\varphi) \wedge ?\varphi \models ?\text{pre}(\alpha)) \rightsquigarrow \text{request}(a, b, !\text{achieve}(b, \alpha)))$. Based on such a rule the system will start a reservation action. A reservation requires that a performance is selected. And performance selection requires that the list of available performances are known. Since this is not yet the case, the system replies with a list of available performances to choose from. So in both cases the response is a list of available performances. And in both cases the answer is coherent, even though it is motivated in a different way. In the first case it concerns a yes/no question – yes/no answer sequence with a witness. In the second case it concerns a request – accept – comply sequence. The difference comes afterwards. A system that understood the question literally will wait for the user to start another initiative. The system that understood that a reservation was requested, will take over the initiative and ask questions according to the reservation plan.

Parameter	Description
roles	initiator (I), responder (R)
illocutionary point	goal of the initiator in starting the game
goals-of-R	non-empty set of goals of the responder during the game
conventional conditions	set of state descriptions, including at least the following: 1. I is pursuing the illocutionary point as goal. 2. I believes that achieving the illocutionary point is feasible. 3. I believes that R's achieving the goals-of-R are feasible. 4. I has the right to achieve the illocutionary point. 5. I has the right to use the dialogue game. 6. The illocutionary point has not already been achieved. 7. R is willing to pursue the goals-of-R.

Figure 5.2: Parameters of a dialogue game (Mann, 1988)

5.3 Dialogue Game Parameters

Our conceptual framework is based on dialogue games in a wider sense. In the previous section we discussed several ways to represent the rules of a dialogue game. But the rules are only one part. What other parameters characterise a dialogue game? Figure 5.2 specifies the dialogue game parameters according to the pioneering account of Mann (1988), developed in the 70's. There are *roles*: the initiator and the responder. Then there is the *illocutionary point* of the game, which describes the goal of the initiator. Once the responder accepts to take part in the game, he is bound to pursue some goals as well: the *goals-of-R*. So once again, games, goals, intentions and commitments align. The *conventional conditions* of a game specify its applicability conditions; not for single dialogue acts, but for the complete game. These types of conditions correspond to the coherence constraints mentioned above.

Figure 5.3 lists some example dialogue game types. The list of game types is by no means complete, and does not constitute an analysis of the games occurring in a particular type of dialogues. Rather, it illustrates different kinds of games from different kinds of applications. We replaced Mann's notation for required information specifications Q with formulas $?\varphi$, and propositions P with formulas φ, to make the notation compatible with the formalism used in this thesis. Mann's action-type formulas A are replaced by our α, although later this chapter we will use directives $!\varphi$ for the content of dialogue acts like request, suggest and propose that are part of an action seeking game. Mann did not have a semantics for the P, Q and A notation at the time; now we do.

An information seeking game corresponds to a question-answer sequence. However, a formulation in these 'neutral' terms stresses, we believe rightly, that information seeking can be accomplished in different ways. Consider the possibility of a so called declarative question, like "I would like to know if ...". Information offering corresponds to an inform act or weak assertion, followed by a acknowledgement. A prolonged sequence of information seeking and information offering games about a single topic constitutes what we have called an inquiry game in chapter 3. The main difference lies in the initiative handling. For an information seeking game the inquirer leads; for an information offering game the expert is the initiator. The information probing game corresponds to a confirmation sequence, or a check. In different settings it could correspond to an exam

Game	Illocutionary Point	Goals-of-R	Conventional Conditions
information seeking	I knows $?\varphi$	I knows $?\varphi$	R knows $?\varphi$
information offering	R knows φ	R knows φ	I knows φ ; R's information and φ are consistent
information probing	I knows whether R knows $?\varphi$	R informs I of R's knowledge about $?\varphi$	I knows $?\varphi$
helping	I is able to perform α	I is able to perform α	R is able to cause I to be able to perform α; I has the right to perform α
dispute	R believes φ	R justifies that I might not believe φ	I believes φ; R does not believe φ
permission seeking	I knows that R grants the right that I performs α	R grants the right that I performs α or not, and I knows this	I wants to perform α; I does not have the right to perform α; R can grant the right to perform α
action seeking	R causes α to be performed	R causes α to be performed	R would not cause α to be performed in the normal cause of events

Figure 5.3: Examples of dialogue games (Mann, 1988; p515)

question. Here the point of the initiator is not to find out some information, but merely to check if the other participant agrees. Carletta et al. (1997) call this an *align* move. A helping game corresponds to a clarification question and subsequent explanation. The fact that clarifications are needed at all, shows that exchanges are not guaranteed to succeed. In some cases misunderstandings have to be cleared up. The dispute game illustrates that dialogue participants do not have to be cooperative nor in agreement. Obviously, the dispute game is important for activity types like debates or arguments, but it may well be important for inquiry and transaction too. The dispute sequence can help to resolve misunderstandings. Consider for example a client in the theatre, who wrongly thinks there is discount on Mondays and believes that she can pay less than the actual price. All the system can do in this case is justify the actual price, by explaining the general pricing convention. Finally, an action seeking game corresponds to what we have called a proposal, suggestion or request followed by an counter proposal, acceptance or rejection. Again, the 'neutral' formulation of action seeking suggests that there are various ways to accomplish this. In chapter 4 we briefly discussed the close relationship between statements of preference and requests.

A number of observations can be made about the way the dialogue games are defined. The first thing that strikes you is that the games are defined in terms of goals. And apart from condition 4 and 5 in figure 5.2 all the conventional conditions are similar to conditions found in accounts of partial planning Grosz and Kraus (1996). This is further evidence that dialogue games are a special case of recipes for joint action. In fact, the sorts of goals used in the specification are joint goals. For example, both the illocutionary point and the responder's goal of an information offering game are that the responder should get to know some information. Both participants have to take action to achieve this goal. The initiator should make an utterance, but the responder should attend to the utterance, and interpret and understand it in relation to existing information, if possible.

Second, it is clear that Mann tries to account for social relationships, as well as information exchange. The seeking and granting of permission is an example of this. In our negotiation examples, we have no need for permission and prohibition. However, it would not be difficult to add. We could treat permission as a modal operator much like possibility. The states of affairs that are permitted can thus be modelled as a set of worlds (Lewis 1979). However, the initial case is the opposite. Initially no state of affairs is allowed. Just like assertions eliminate worlds, permissions add worlds. As you may recall from the discussion of Von Wright's accounts of deontic notions in chapter 4, permission and prohibition can also be analysed as the extremes of some underlying preference order. This would explain what to do in case of a dilemma. Which violation of the prohibitions is worse or better than another? The granting of permission can also be the subject of a negotiation process. In general more restrictive rules are preferred by the authority that grants the permissions, whereas less restrictive rules are preferred by the subject seeking permission. The right to grant permission is related to some social relationship between participants. In some activity types, one of the dialogue participants is just an intermediary for some higher authority. For example, the cashier in the movie theatre does not grant the permission to smoke at the balcony. There is some rule about smoking and the cashier merely informs the customers about this rule. A special case arises when there is no initial social relationship between participants, and permissions are concerned with general decency and consideration of the other participants. For example, "Do you mind if I smoke?", asked in a restaurant. Here the right to grant permission is derived from the expected possible harm that may be done. These aspects of permission are again related to the notion of *face* (Brown and Levinson 1987).

Third, some of these parameters are very similar to the ones used in use case analysis, as we suggested in chapter 1. There we used actors instead of roles, and in addition to the goals of both users, we specified the success and failure conditions. The conventional conditions partly correspond to the context field that was used in the use case analysis to describe long term contextual restrictions, and partly to the trigger conditions, which describe the short time reason to start an interaction at all. This correspondence is not a bad thing at all. It shows that in the general development process, the role of use cases can be taken over by that of dialogue games, provided we come up with a theory of dialogue games that goes beyond the use of a metaphor. In particular, we would need a theory that is specifically targeted for analysis of interactive, mixed initiative dialogues. To describe the goals of the participants we use the machinery of chapter 4. To describe the interaction patterns, we can use finite state techniques for simple applications and either discourse grammar rules or coherence principles for more complicated cases.

Fourth, these games are rather domain independent. There are a number of games which are typical for a particular activity type but may occur elsewhere too. The power of these conventional interaction patterns is rather that they are applicable in new and unchartered situations. If we are looking for a library with tools to model and build dialogue systems for different applications, the level of these games are the right level.

Now we turn to dialogue acts and games that are typical for negotiation.

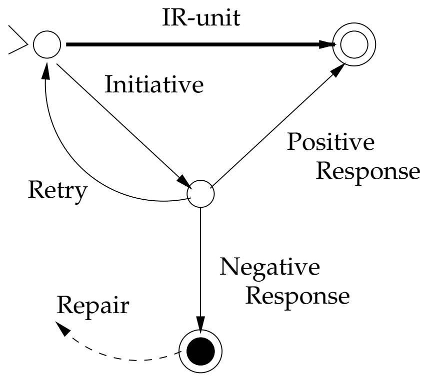


Figure 5.4: Initiative-response units

5.4 Negotiation Games

In chapter 4 we already discussed the general task structure of negotiation. We distinguished five negotiation phases, and described a model in terms of which the content of dialogue acts for negotiation can be captured. Now we discuss the interaction aspects which are typical for a phase in terms of dialogue acts, and the games that relate them.

5.4.1 Assumptions

We make a number of assumptions about the nature of dialogue acts in negotiation. They are inspired by the ADEPT negotiation protocol proposed by Jennings et al. (to appear), originally designed for application in business process management. The assumptions are adapted for human dialogue based on ideas by Di Eugenio et al. (1998). They studied dialogues between two subjects who were asked to select furniture for a room, while remaining within a certain limited budget. This produces negotiation, since every move has to be approved by the other, and because accepting a move has consequences for further moves. Other work along these lines is done by Walker (1996b) who studied acceptance and rejection in radio conversations about financial advice.

There are three kinds of dialogue acts. Initiatives, responses and neutral acts. Initiatives of a particular type require a response of a particular type. Responses on the other hand require that some initiative of the right type preceded them. Several people, most notably Traum (1997), have proposed to model the effect of an initiative as an obligation on the addressee to respond. However, the type of obligation involved in interaction patterns is of a low-level reactive type. Current state of the art agent technology favours hybrid architectures. In a hybrid architecture a low-level reactive mechanism such as a dialogue game can be given a semantics in terms of high-level deliberative notions, such as commitment and obligation. Dialogue game rules are a good way to express interactive commitments.

Initiatives and responses are often related by a triangle structure such as the one depicted in figure 5.4: the horizontal IR transition is composed of an initiative and a positive or negative response, or else a retry. For example, a proposal is an initiative, an acceptance is the corresponding positive response, a rejection is the negative response and a counter proposal is an example of a retry of the initiative. In case of a negative response some precondition for a successful completion of this exchange may have been violated. In that case a repair is needed of parameters that were fixed in a previous phase. For example, if a customer wrongly thought there is a discount on Mondays, the pricing rules which were agreed on implicitly in the previous phase, may have to be explained explicitly now.

The figure suggests that the response must occur after the initiative. This is not so. Many types of non-verbal feedback can occur simultaneous to the utterance they give feedback to (chapter 2). A combined initiative-response unit, called *IR-unit* or *exchange*, results in some information being added to the common ground: a response of the right type has a grounding effect. For example, a returned greeting produces mutual contact, a proposal followed by an acceptance produces a mutual agreement, and a question followed by an answer produces a mutual belief.

For simple dialogues, the initiative-response structure can be modelled by a stack. An initiative is pushed onto the stack; a response requires an appropriate previous initiative to be popped off the stack. This is a simplification. Initiatives do not necessarily have to be addressed in the reverse order in which they were asked, as would be expected by popping them off the stack. This is illustrated by the following example of Asher (1998). Instead we expect a *partial order* among initiatives, which resembles the QUD data-structure of Ginzburg (1995) sketched in chapter 3.

(123)	A: Where were you on the 15th?	I_1
	B: Why?	I_2
	A: Do you remember talking to anyone after the incident?	$R_2 \| I_3$
	B: I was at home.	R_1
	I didn't talk to anyone after the incident.	R_3
	A: What do you remember about the incident?	I_4

Interestingly, the content of most initiatives corresponds to an issue: it opens alternatives. The content of a response usually narrows down the number of alternatives. This nicely conforms to the suggestion of a 'grounding issue' too. A grounding issue temporarily opens up the possibility that the assertive was not correct. It is an implicit invitation to respond with an acknowledgement.

Dialogue games are sequences of IR-units like the one in figure 5.4. The 'glue' that combines such sequences is formed by coherence relations. Sequences of IR-units do not imply that the initiative always alternates regularly. Initiative shifts are allowed. For example when an 'information seeking game' is alternated with an 'information offering game', in terms of Mann (1988). Not all IR-units or exchanges need to have a binary triangle shape like this one. Another common type of exchange starts with a so called *pre-sequence* (Schegloff et al. 1977). It invites the other participant to join in actual exchange. For example: "Can I ask you something? – Yeah – I was wondering, could we come over this weekend?" where a pre-question precedes the real question. This pre-question probes one of the applicability conditions, namely that the responder is willing to respond or willing to comply. Or consider: "So, good-bye then – Yeah, bye! – Bye!" where a pre-sequence announces the closing sequence. In a similar way there are also post-sequences. More importantly, a response may function itself as an initiative to respond to. Hence the possibility of counter proposals.

In the description of a response act, the so called *scope* of the act should be clear from the context. For example, a question is an act of type ask$(a, b, ?\varphi)$. The act of answering this question is indicated by answer(b, a, ψ), where we require that ψ is pertinent to $?\varphi$. In this case, we call $?\varphi$ the scope of the response. The dialogue context must therefore contain a pointer to the latest move (cf. Ginzburg 1995). However, the scope of a response is not always the result of just the latest move. It can be a combination of several previous moves. And indeed, often it is not precisely the question that was asked, which is answered, but a different and related one. Recall example (92) where the platform number was added in response to a question about a departure time. For proposals, the scope is the 'strongest proposal on the table'. So the scope of a response is the current most salient active contribution of the right type.

Both initiatives and responses can be made implicitly. Initiatives may for instance be triggered by the situation. Moreover, a single utterance can perform several communicative functions at once. This is the mechanism used by of indirect dialogue acts. A question like "Do you know the time?" asks for a precondition for successful completion of the question-answer exchange that was intended. This only makes sense in case the speaker actually wants to start the question-answer exchange. Also acknowledgements can be implicit in another type of response. In general, a pertinent continuation of the dialogue counts as an acknowledgement of the previous move.

Recall from chapter 2 that we view the interaction process as a layered protocol. We assume there is contact between participants and some level of mutual attention. At the presentation level, the utterance is vocalised and attended; at the locutionary level, the utterance is formulated and identified with respect to structure, words and intonation. At the illocutionary level, the function and content of the utterance are conceptualised and comprehended. Finally, at the perlocutionary level, we find aspects related to task and activity. At each of these levels interaction takes place by means of triangle shaped protocols like the one in figure 5.4. Our working hypothesis is that a combined initiative and response at a higher level produces mutuality at that level, as well as at all the lower levels. For example, an answer whose content is judged pertinent indicates receipt and understanding of the question. Failing to respond at one level automatically brings up the level below. For example, failing to answer a question indicates that the hearer was not paying attention. Therefore, the hearer should signal at least receipt and understanding, even if he does not know the answer: "Yeah, uh, I don't know." A negative response on the other hand does not bring up the level below, but triggers the need for a repair at the current level, or it suggest a return to the previous dialogue phase. For example, an irrelevant answer or the rejection of a proposal suggest a clarification or a counter proposal.

This layered structure leads to a genuine problem with dialogue act recognition. There is ambiguity in the intended level of a response, which often gives rise to misunderstanding. For example, a positive response, 'Yes', may be interpreted at any level: as an indication of attention (yes, I am listening), as an indication of receipt (yes, I got that), as an indication of understanding (yes, I see what you mean), as an indication of agreement (yes, I agree) or acceptance (yes, all right) or as a indication of a commitment (yes, I will). The only way to distinguish these is by the relative prominence of the expression that is used, which is indicated by prosodic means, by surrounding expressions and verbal additions, and by the exact moment at which the agreement is uttered. For example, nodding simultaneously to an utterance only signals acknowledgement, but nodding right after a an explicit question, signals agreement. Because of this ambiguity *align acts* are needed that invite the other participant to give feedback. Consider "Isn't it?", "Right?" or in general a rising pitch contour instead of a 'declarative' hat-shaped pitch contour. More on this

Action	Description	Applicability
utter(a, b, φ)	make utterance	requires attend
attend(a, b)	indicate receipt	in response to any dialogue act
ask_clarification(a, b, φ)	indicate non-understanding of φ	requires clarify
clarify(a, b, φ)	give explanation or motivation φ of apparent source of non-understanding	in response to ask_clarification

Figure 5.5: Presentation level acts

layered ambiguity in section 5.6.

5.4.2 Dialogue Acts for Negotiation Phases

In chapter 4 we already discussed five negotiation phases: open, inform, propose, confirm and close. We present a rough characterisation of the dialogue acts that occur in each phase. Acts that are mentioned for one phase, can very well occur in other phases as well.

There is a natural ranking among acts according the communication layers that are affected, indicated by the notation '$\prec$'. It corresponds to the action ladders discussed in chapter 2. We have $act_1 \prec act_2$ when act_2 is a more specific instance of act_1. By combining these ranking relations we define a dialogue act hierarchy. Below we also use notation $act_1 \simeq act_2$ to indicate that the effects of act_1 and act_2 are the same.

Presentation Level

Figure 5.5 shows a first selection of presentation level acts. They may occur at any point in the dialogue. We could add other dialogue control acts, for interrupting and resuming, or for apologies. Exactly how to model the presentation level depends on the modality: speech, keyboard-based or multi-modal.

All responses are instances of an attend act to indicate that the responder is paying attention: attend $\prec$ response. Each initiative and some kinds of responses are instances of an utter act: utter $\prec$ initiative. Which kinds of responses count as utter acts? Utter acts are usually verbal and if they are not they always occupy the main communication channel. Attend acts are usually non-verbal or at least collateral: they do not occupy the main communication channel. This might be used as a definition: an act is an instance of an utter act when it occupies the main communication channel. So that means that: utter $\prec$ response, if non collateral.

The helping game of Mann reappears here as a sequence of a clarification question and a clarification answer. A clarification question may be indicated by a puzzled look. What counts as a clarification is extremely difficult to assess beforehand, because you do not know what will lead to a misunderstanding. Clarification must be context dependent. In spoken dialogue systems often it helps to present the current dialogue status, a motivation of the preceding system utterance and a suggestion about how to continue (Bernsen et al. 1998).

Action	Description	Applicability
greet(a, b)	establish contact	requires or in response to greeting
meet(a, b)	identify and establish social relationship	a and b have contact, requires or in response to meeting

Figure 5.6: Opening acts

Opening

During the *opening* phase participants exchange greetings to acknowledge each other as negotiation parties; they establish contact and find a common ground of social and linguistic conventions for interaction. Do we speak the same language? Can I trust this person? Do we attach the same preferences to attributes, or distinguish among objects the same way? The opening phase sets apart the bystanders from the dialogue participants (Clark 1996). It is established what social relationship there exist and how this affects the roles in the interaction.

We model two simple actions in figure 5.6: greeting and meeting. Greeting merely establishes contact; meeting establishes contact, and identifies the dialogue partner with respect to the the social relation and role relevant for the interaction: greet $\prec$ meet. In human interaction, greeting would be an exchange of "Hi" or "Good afternoon" with an accompanying smile. Meeting would involve shaking hands, examining each other and mentioning names. Greeting and meeting are examples of actions that are symmetric; they can act both as initiatives or as responses. In fact, both are joint actions, of which the participatory actions are symmetric. Other actions that could be part of the opening phase are informs or questions that are meant to give a motivation or background to the interaction. Think of a joke or anecdote to make the other participants feel at ease.

In spoken dialogue systems, the introduction is particularly important. It suggests how users can and should behave towards the system. Should they treat it as a computer or rather as a skilled language user? These questions are settled not only through the content of the message, but also by the personality and quality of the voice. For example, playing the Star Trek theme at the introduction of a British Airways' spoken dialogue system for enquiries, a theme which also appeared in their advertising campaign, made users feel 'part of the future'. In fact, it was a way of making them realise they were talking to a computer. This greatly reduced the number of problematic dialogues, while increasing user satisfaction rates. Apparently, users were more willing to adapt to the system than they would have been otherwise (Norman Fraser, p.c.).

Information

In the *information* phase, non-negotiable parameters of the activity type are agreed. Unlike acts from the proposing phase, the typical acts for information exchange do not generate a commitment to a future action. In the particular view we take here, an assertion is stronger than an inform act, although both are assertives. An assertion hopes to persuade somebody; acceptance is not more likely than rejection. An inform act assumes that the information was not very controversial, and therefore already expects the acceptance. This distinction is similar to the one between a normal yes/no question and a check. A check

Action	Description	Applicability
inform(a, b, φ)	state fact φ	requires acknowledgement, φ uncontroversial
acknowledge(a, b)	indicate receipt and understanding	in response to inform, suggest
assert(a, b, φ)	state opinion φ	requires assent
assent(a, b)	indicate receipt, understanding and agreement	in response to assert, check
deny(a, b)	indicate receipt, understanding and disagreement	in response to assert, check
correct(a, b, φ)	deny information in focus, and replace with φ	in response to assert, inform; requires assent
ask($a, b, ?\varphi$)	request information pertinent to $?\varphi$	requires answer
answer(a, b, φ)	provide information pertinent to $?\varphi$	in response to ask
check($a, b, ?\varphi$)	request agreement	provided $\varphi \in L_{\mathcal{P}}$; requires assent

Figure 5.7: Information acts

expects no content in its response; just a positive or negative reply. For this reason also the strengths of the responses are ordered. An assent indicates receipt, understanding and agreement; an acknowledgement merely indicates receipt and understanding. Note that a denial, a negative response, also signals receipt and understanding. In chapter 4 we also considered a denial with a semantic content. This type can be defined in terms of assertions: deny(a, b, φ) $\equiv$ assert($a, b, \neg\varphi$). A combination of a denial and a new assertion forms a correction: correct(a, b, φ) $\simeq$ deny(a, b);assert(a, b, φ). The scope of the correction is determined by the context. The combination of a question with an answer corresponds to the information seeking game of Mann. An answer can be any kind of assertive act, so it can be an inform or an assertion. Finally, the check act is added to allow for the align moves of the MAPTASK corpus and, in combination with an assent, corresponds to an information probing game.

So we have the following ranking: inform $\prec$ assert, acknowledge $\prec$ assent and acknowledge $\prec$ deny. The effect of a question followed by an answer is that it is clear from the dialogue history that both participants now know the answer, and know that they both know it. An assertion followed by an assent, or a check followed by an assent have the same grounding effect. So we get the following equivalences. A systematic investigation of such principles is a topic for further research.

(124) $M, \langle w, v\rangle, g \models$assert($a, b, \varphi$); assent($a, b$) iff
$M, \langle w, v\rangle, g \models$ask($a, b, ?\psi$); answer($a, b, \varphi$) and φ pertinent wrt $?\psi$ iff
$M, \langle w, v\rangle, g \models$check($a, b, ?\varphi$); assent($a, b$) iff
$\sigma_{a,v} = \sigma_{a,w}[\varphi]$ and $\sigma_{b,v} = \sigma_{b,w}[\varphi]$

Proposing

This part of a negotiation dialogue contains the negotiation proper. Consider the acts in figure 5.8. The propose and request dialogue acts generate a (partial) commitment to future action, provided they are accepted. A mere acknowledgement indicates that

Action	Description	Applicability
request$(a, b, !\varphi)$	get b to do $!\varphi$	requires acceptance
propose$(a, b, !\varphi)$	offer to do $!\varphi$	requires acceptance
suggest$(a, b, !\varphi)$	bring possibility of $!\varphi$ to attention	requires acknowledgement
counter propose$(a, b, !\varphi)$	reject current offer and replace with $!\varphi$	in response to request, suggest or proposal; requires acceptance
accept(a, b)	indicate receipt, understanding and agreement with current offer	in response to request or propose
reject(a, b)	indicate receipt, understanding and disagreement with current offer	in response to request or propose

Figure 5.8: Proposing acts

both parties know that there is an offer to be considered. The current 'offers on the table' structure a dialogue, just like the current issues. A suggestion is a bit weaker; it can only introduce a new offer to be considered, provided it is acknowledged. The type of content of these acts has been discussed before. They contain a content which corresponds to a directive update $!\varphi$, an update with the commitment to reach the (possibly joint) goal φ. Depending on the time-scale at which the action should take place a commitment is followed by putting actions onto the *agenda* of the actor. The agenda is the data-structure for managing the scheduling and resources of an agent's current actions. So propose, suggest and request are three ways of initiating an 'action seeking' game. An acceptance is just like an assent: it indicates receipt, understanding and agreement, but now for (partial) commitments. Similarly for the relationship between rejections and denials. So we have the following rankings: acknowledge $\prec$ accept, acknowledge $\prec$ reject.

A counter proposal is strictly speaking not needed in the theory; its function can be expressed as a rejection, with a subsequent new proposal. Similarly for counter requests or counter suggestions. Counter proposals are single utterances, which have this double function. Therefore they shift the initiative. So it holds that counter propose$(a, b, !\varphi)$ $\simeq$(reject(a, b); propose$(a, b, !\varphi)$). Furthermore, we require that the scope of the rejection, the proposal 'on the table' and the counter proposal are related. They must concern the same transaction.

In actual dialogues, counter proposals are not necessarily marked by an explicit 'no'. We need information about what counts as a stronger proposal to assess when a proposal should be classified as a counter proposal. What matters is the content of the latest proposal, who made it, and its scope. More about these kinds of decisions in section 5.6.

There is a strong connection between proposals, suggestions and requests on the one hand, and preference statements on the other. In case we have reason to assume cooperativity, for example because we are in a buying-and-selling setting and the interests of buyer and seller align, we can use the following default principles.

(125)
$$\begin{array}{llll} \text{assert}(a, b, \text{good}_a\varphi) & \rightsquigarrow & \text{request}(a, b, !\varphi), & \text{if } b \in \text{actors}(\text{bring\_about}(\varphi)) \\ \text{assert}(a, b, \text{good}_a\varphi) & \rightsquigarrow & \text{propose}(a, b, !\varphi), & \text{if } a \in \text{actors}(\text{bring\_about}(\varphi)) \\ \text{inform}(a, b, \forall x.\text{good}_x\varphi) & \rightsquigarrow & \text{suggest}(a, b, !\varphi) & \end{array}$$

Requests and proposals are stronger; they correspond to an assertion of a preference statement and therefore need a full acceptance or assent for success. Suggestions are weaker. Like an inform, they only require an acknowledgement. Moreover, they do not indicate who should carry out the action. Often the preference statement is depersonalised. "It would be good for everyone if ...".

A suggestion brings a future action to the attention. Bringing things to the attention is one of the purposes of asking questions and raising issues in general. So we would expect that questions can function as suggestions too. The following principle corresponds to something like "Who wants some cookies?" which is really a suggestion to take some.

(126) $\text{ask}(a, b, ?x.\text{good}_x\varphi) \rightsquigarrow \text{suggest}(a, b, !\varphi)$

And of course the same kind of reasoning holds for proposals and requests. Note that these principles correspond to the principles in (125) above, because a combined question-answer pair has the same effect as a combined assert-assent sequence. In both cases the information is grounded.

(127)
$$\begin{array}{lll} \text{ask}(a, b, ?\text{good}_b\varphi) & \rightsquigarrow & \text{propose}(a, b, !\varphi), \quad \text{if } b \in \text{actors}(\text{bring\_about}(\varphi)) \\ \text{ask}(a, b, ?\text{good}_b\varphi) & \rightsquigarrow & \text{request}(a, b, !\varphi), \quad \text{if } a \in \text{actors}(\text{bring\_about}(\varphi)) \end{array}$$

Confirmation

After acceptance of the various proposals that are related to a single transaction, the full consequences of the agreements made earlier must be confirmed. This is a last chance to correct misunderstandings. A confirmation means a check of the earlier agreements, as well as a check of possibly implicit consequences of the agreements. In case it can not be assumed tat both participants are aware of these consequences, they must be stated explicitly. For example, the European community directives on distance sales contains a general requirement that information about the transaction must include identity of the supplier, the main characteristics of the product or service, price and delivery costs as well as a reference to the 'right of withdrawal', that ensures that a consumer may cancel the transaction within in a limited time span.

An ask_confirm act is a special case of a check. Like a check it presupposes that the other participant agrees, otherwise the negotiation would not have ended up here. Confirm and disconfirm are analogous to acceptance and rejection. But whereas acceptance or acknowledgement can be implicit in subsequent acts, a confirmation requires an explicit positive or negative response. The other main difference is its scope. A confirmation concerns the complete transaction which is supposed to be closed under consequences; an acceptance only concerns the latest offer on the table. We have the following rankings: check $\prec$ ask_confirm, accept $\prec$ confirm, reject $\prec$ disconfirm.

Confirmation actions serve as a basis for common knowledge that the transaction is concluded. For example, the transfer of money tokens across a counter forms an infallible basis for common knowledge that the money changed owner; for payment by credit card or cheques this is already less obvious. Here the signature does double duty. It indicates agreement on the transaction as well as on the money transfer. For electronic transactions a good representation of money transfer has yet to be found. What is crucial here is that such a representation should not only be reliable or secure in a technical sense, but also in a social sense. Spending money must be marked as something important.

Action	Description	Applicability
ask_confirm($a, b, !\varphi$)	ask commitment on complete offer	a and b have contact; requires confirm
confirm(a, b)	indicate commitment on complete offer	a and b have contact; there is a complete offer; in response to ask_confirm
disconfirm(a, b)	indicate absence of commitment on complete offer	a and b have contact; there is a complete offer; in response to ask_confirm

Figure 5.9: Confirmation acts

Action	Description	Applicability
thank(a, b)	strengthen relationship	b did a a service; optionally followed by ack_thanks
ack_thanks(a, b)	acknowledge thanks	in response to thanks
bye(a, b)	close contact successfully	a and b have contact, requires or in response to bye
break(a, b)	close contact unsuccessfully	

Figure 5.10: Closing acts

Closure

After the confirmation phase or after an unsuccessful interruption of one of the other preceding phases, the interaction must be closed. The *closure* phase is meant to release other participants from the commitment to maintain contact and possibly to re-establish social relations that may have been harmed by an unsuccessful interaction. Since it is rude to withdraw before others, withdrawal must be synchronised. Often this happens by pre-sequences, discussed before. Here we mention only a simply bye-bye sequence.

Another exchange that occurs often in the final stages of a transaction is a "thanks – you're welcome" exchange. Thanks express that the speaker is grateful for the other doing some service. Thanks are a mechanism to re-balance face. If somebody does something for you, your negative face increases. Thanks may be used to counter the imbalance. All you can do is express that you are 'obliged', or you can bake a cake or bring chocolates. Depending on the strength of the thanks and on the scale of service that was done, thanks can be followed by an acknowledgement. Something like: "you're welcome", "my pleasure" or "no thanks", which expresses that there was no imbalance at all.

Ending an interaction without greeting is modelled by a break action. It is never well-formed. It is extremely rude, because you remove the opportunity for the other to say anything back. Moreover, greetings simply mark the boundaries of a conversation. Incidently, even when talking to computer, users of spoken dialogue systems still often thank the computer and say good bye. This is evidence that sequences of dialogue acts are often just triggered by the circumstances.

Communicative Function	Appropriateness Conditions
wh-question	*S* wants to know which elements of *X* satisfy *p*. *S* suspects that *H* knows which elements of *X* satisfy *p*. *S* suspects that there is an elements of *X* that satisfies *p*.
yn-question	*S* wants to know whether *p*. *S* suspects that *H* knows whether *p*.
check	*S* wants to know whether *p*. *S* suspects that *H* knows whether *p*. *S* suspects that *p*.
alts-question	wants to know whether x_1 or ... or x_n satisfy *p*. *S* suspects that *H* knows which elements of x_1, ..., x_n satisfy *p*. *S* suspects that either x_1 or ... or x_n satisfies *p*.
wh-answer (/weak)	*S* knows that *H* wants to know which elements of *X* satisfy *p*. *S* knows(/suspects) that *V* are the elements of *X* that satisfy *p*.
yn-answer (/weak)	*S* knows that *H* wants to know whether *p*. *S* knows(/suspects) that *p*.
confirm (/weak)	*S* knows that *H* wants to know whether *p*. *S* knows(/suspects) that *p*. *S* knows(/suspects) that *H* suspects that *p*.
disconfirm (/weak)	*S* knows that *H* wants to know whether *p*. *S* knows(/suspects) that *p*. *S* knows(/suspects) that *H* suspects that not-*p*.
inform (/weak)	*S* wants *H* to know that *p*. *S* knows(/suspects) that *p*.
agreement (/weak)	*S* wants *H* to know that *p*. *S* knows(/suspects) that *p*. *S* knows(/suspects) that *H* suspects that *p*.
disagreement (/weak)	*S* wants *H* to know that *p*. *S* knows(/suspects) that *p*. *S* knows(/suspects) that *H* suspects that not-*p*.
correction	*S* wants *H* to know that *p*. *S* knows that *p*. *S* knows that *H* knows that not-*p*.

Figure 5.11: Communicative functions for information dialogues (Rats, 1996)

5.5 Comparisons

In this section we compare the set of dialogue acts developed in the previous section, with other proposals from the literature.

5.5.1 Rats

Figure 5.11 shows a set of communicative functions developed in Tilburg by Rats, Bunt and others (Rats 1996). The roles *S* and *H* refer to the speaker and hearer respectively. Notation *X* indicates a set of objects of a particular type and *p* indicates a proposition. These communicative functions are specific for the type of human-human airport information dialogues studied by Rats. Most dialogues centre around a flight which is usually identified by a flight number. Users enquire about departure or arrival times, or whether that flight has landed. So there is hardly any negotiation in these dialogues; all confirmation concerns the grounding of information. However, a large number of dialogue control acts does occur. The telephone line is of a bad quality, people speak unclearly or the service provider needs time to look up some flight details.

This set of communicative functions remains much closer to natural language classifications, than our set. The distinction between wh-questions, alts-questions, yes/no questions and checks for example, depends on the content of an act. An alts-question is a restricted wh-question. Most of the distinctions can be directly translated into surface level features for recognition of the communicative function. A yes/no question has different intonation and word order characteristics as a wh-question.

There is a difference between strong and weak communicative functions. A communicative function is called weak, if the underlying precondition is based on a assumption, not on knowledge. For example, a disagreement is weak when the speaker suspects that the hearer is of different opinion. A disagreement is strong when the speaker knows that the hearer believes otherwise. Obviously, the weak and strong forms must connect to different ways of expressing the dialogue acts. For example, 'no' expresses a strong disagreement, whereas a weak disagreement would be expressed by "Do you think so?". This distinction is partly related to our distinction between acknowledgement and assent or acceptance, but it lacks the connection to grounding.

The list is not complete. Typical dialogue control acts, like winning time, apologising and opening and closing are not listed. Obviously 'action seeking' acts are left out as well. There is some overlap too. Typically, yn-answers and wh-answers are special cases of an inform action. Likewise, a check is a special case of a yn-question. There is some redundancy between agreement and disagreement on the one hand, and confirm and disconfirm on the other. As we have seen, the distinction depends on the role they play in relation to surrounding acts. The crucial difference between a confirmation and an agreement (assent in our terminology) is that a confirmation is explicitly asked for and must be made explicitly. Under that definition, agreements get the same role as acceptances or acknowledgements in our theory.

5.5.2 Verbmobil

The VERBMOBIL project needed yet another list of dialogue acts (figure 5.12). These acts are developed for negotiation dialogues about a time and place for a meeting (Alexandersson et al. 1995). There are four levels of analysis: the *dialogue level*, which we we would call the task level or activity level. The activity is appointment scheduling. Below that we find the *phase level*, distinguishing among a greeting phase, a negotiation phase and a closing phase. This conforms to our intuitions, except that we distinguish a separate information phase. We could say that it is 'conventionally skipped'. The next level of analysis is that of *turns*, which comprise several utterances by the same speaker. Finally, at the *utterance level* utterances are analysed as dialogue acts, that sometimes do (e.g. suggest_date) and sometimes do not have a content (e.g clarify_query).

We do not agree with the turn level. There is a need for an intermediate level between dialogue acts and phases, but this level can be much better characterised by exchanges. Turns are a by-product of initiative handling. The turn structure cuts across the functional

introduce	introduce_name	
	introduce_position	
	introduce_react	
init	init_date	
	init_location	
	init_duration	
motivate	motivate_appointment	
request_suggest	request_suggest_date	
	request_suggest_location	
	request_suggest_duration	
suggest	suggest_exclude	suggest_exclude_date
		suggest_exclude_location
		suggest_exclude_duration
	suggest_include	suggest_include_date
		suggest_include_location
		suggest_include_duration
accept	accept_date	
	accept_location	
	accept_duration	
reject	reject_date	
	reject_location	
	reject_duration	
confirm		
feedback		
give_reason		
garbage		
clarify	clarify_query	
	clarify_answer	
	clarify_state	
request_comment	request_comment_date	
	request_comment_location	
	request_comment_duration	
deliberate	deliberate_explicit	
	deliberate_implicit	
digress	deviate_scenario	
	refer_tosetting	
thank	thank_init	
	thank_react	
greet		
bye		

Figure 5.12: Dialogue acts for VERBMOBIL (Alexandersson et al, 1995)

role of exchanges. Witness the following example (Alexandersson 1996, p. 152-153)[2].

(128) A1a: ja prima, (feedback)
Well
A1b: dann lassen Sie uns doch noch einen ⟨!einn⟩ Termin ausmachen. (init_date)
Let us make a date
A1c: wann wär's Ihnen denn recht? (request_suggest_date)
When does it suit you?
B1: also ich dachte noch in dee nächsten Woche, auf jeden Fall noch im April. (suggest_support_date)
Well I thought in the next week, in any case in April.
A2a: ja am Dienstag den sechsten April hätt' ich noch einen Termin frei allerdings nur nachmittags. (accept_date)
Yes on Tuesday the sixth of April is possible for me, but only in the afternoon.
A2b: geht es da bei Ihnen ⟨!ihnn⟩ auch? (request_comment_date)
Does that suit you?

Now consider example (128). The first turn of participant A takes three utterances. First an introductory phrase which is annotated as a feedback act. Then a suggestion to negotiate a date, followed by a wh-question which requests the other participant to make a suggestion. The first two utterances clearly belong in an opening phase. They set the stage for the rest of the negotiation. Together with B's reply the last utterance of the turn forms a initiative-response unit. However, B's reply contains a suggestion that still needs to get some support. So the reply serves as an initiative again, for a response in A2a, which in fact accepts the suggestion. This is also the way it is annotated. As Alexandersson points out the utterance does double duty. Not only does it accept the suggestion for 'April next week', but it also refines the suggestion into the sixth of April. This refinement is in need of support, which is indicated by the request in A2b. However, according to the dialogue grammar rules of the VERBMOBIL project, an acceptance can not be followed by a request for support. And this makes sense.

The example illustrates a number of difficulties. First, many phrases and utterances can have different functions, depending on the place where they occur. 'Ja prima' can be used to indicate positive feedback, but in this case it merely marks the end of a previous phase in the conversation, and the beginning of a negotiation. Second, the turn structure cuts across the initiative-response structure, which is more useful in describing the function of utterances. Third, an initiative-response structure that is too simple would fail, because of the partiality of suggestions or proposals. A refinement of the current offer counts both as an acceptance and as a further proposal. Fourth, utterances may have two or more functions at the same time, thus making it difficult to assess their effect on the surrounding dialogue acts. All of these difficulties apply in equal measure to our account.

5.5.3 Discourse Research Initiative

The discourse research initiative (DRI) is a group of researchers who have agreed on the following classification (Allen and Core 1997). The classifications have been tested and

[2]The translations are not produced by the VERBMOBIL system but by the researcher; the utterance numbers are adjusted by me.

Forward looking function		
Statement	Assert	
	Reassert	
	Other-statement	
Influencing-addressee-future-action	Open option	
	Action-directive	
Info-request		
Committing-speaker-future-action	Offer	
	Commit	
Conventional	Opening	
	Closing	
Explicit-performative		
Exclamation		
Other-forward-function		

Backward looking function		
Agreement	Accept	
	Accept-part	
	Maybe	
	Reject-part	
	Reject	
	Hold	
Understanding	Signal-non-understanding	
	Signal-understanding	Acknowledge
		Repeat-rephrase
		Completion
	Correct-misspeaking	
Answer		
Information-relation		

Figure 5.13: Dialogue Act Markup in Several Layers (Allen and Core, 1997)

developed with the MAPTASK, TRAINS and VERBMOBIL corpora in mind. One could say that it represents the state of the art in dialogue act classification schemes (figure 5.13).

The proposal has the following interesting features. Classifications are organised in several layers, called levels here. At the lowest level, there are categories for the *communicative status* of an utterance, which is to indicate whether it was intelligible and whether the speaker completed it. At this level aspects like mumbling, hesitations or self-corrections can be annotated. Then there are tags at an *information level* that concern the semantic content of utterances. Distinctions are made between utterances that carry information about the task, about task management, about communication management and others. Information about the task is the normal case. Task management concerns the status of the task, for instance "Are we done?" or "What next?". Communication management concerns what we termed dialogue control: greetings, good-byes, "Wait a minute!", "Sorry?" and all kinds of grounding acts. The category others contains unclassifiable utterances like jokes, nonsense and small talk. Then there are two levels to indicate the communicative functions of an utterance, divided in a *forward looking function* section and a *backward looking function*, corresponding to aspects of an utterance that serve as an initiative, and as a response respectively. Roughly the same repertoire of acts is given as in our proposal, with some different choices as to the role of commitments and the place of suggestions (open-option). There are no explicit confirmation acts to confirm a complete transaction in this proposal. This makes sense because such confirmations are not a special category linguistically. The commit act, in the section committing-speaker-future-action, can perform this role in negotiation dialogues.

The annotation scheme comes with a decision tree to help annotators. As a matter of fact, annotators are allowed to look ahead in the dialogue to see how the utterance was actually taken up. Much of the actual ambiguity is dealt with by people choosing to take an utterance is certain way. If the corpus is large enough, it could provide predictions about the preferred way to deal with ambiguous utterances. Again, a single utterance may have both kinds of functions simultaneously. Many of the backward looking acts are grounding acts. The DRI account is based on the theory of grounding developed by (Traum and Hinkelman 1992). Obviously, this account is more detailed and empirically more informed than our proposal.

5.5.4 Discussion

That concludes our survey of competing theories of dialogue acts. This survey is by no means complete. Other applications typically require other sets of dialogue acts, with other specific details that matter in that application domain. Also the personal preferences of the developers of a theory should not be underestimated. Many differences in annotation schemes have to do with a different terminology and background of the researchers. The first conclusion is therefore, that it seems useless to try and standardise a set of dialogue acts. For every purpose and activity, a different set is developed. And this is as it should be. As we claimed earlier, we believe in an activity based theory of natural language processing.

Still, it seems that there are certain 'trends' in these models, which suggest a systematic way of constructing dialogue acts from some basic primitive parameters. Many of these trends are already clear from the names of the DRI acts. So what we called a request is called a 'influencing addressee future action' here. We would expect that a core of dialogue acts is universal. More specific types can be added as new distinctions are relevant. Witness the specific types of acts for dates and locations in the VERBMOBIL set.

We hypothesise that the following choices will be important for a systematic account of dialogue acts for negotiation dialogues. Obviously, some combinations of parameters do not make sense. For the combinations that do make sense, the choices suggested here provide a decision tree.

(129)	function:	interaction or task related
		initiative or response or both
		implicit or explicit
		assertive or interrogative or directive or control
	content:	proposition or issue or commitment or none
		factual or preferential or procedural
		binary or scale
		positive or negative
		context dependent or not

5.6 Implicit Acceptance and Rejection

As we remarked in several places, some dialogue acts can be expressed implicitly, as part of another exchange. The most obvious example is an acknowledgement, which is implicit in any kind of pertinent reply. But think of acceptance or assent. How does somebody indicate agreement? How can counter proposals and corrections be distinguished from normal contributions?

The mechanism by which one can recognise acceptances and rejections is studied by Walker (1996b). There is other literature which points in the same direction, in particular (Di Eugenio et al. 1998). As a matter of fact, Walker makes a distinction between acceptance and assent, or between denial and rejection, but not in her terminology. In the section below the terms acceptance and rejection therefore cover both kinds of responses. Based on empirical studies of financial advice giving dialogues, Walker tried to characterise what utterances count as an acceptance or rejection. Apart from studying explicit expressions for acceptance, such as "uh huh" or "yeah", and explicit expressions for rejection like "no", Walker studied the mechanism of implicit acceptance and rejection. The goal was to find out when an assertion, suggestion, request or proposal is implicitly accepted or rejected by subsequent utterances.

Walker notices that logical consistency of the content of an utterance with the information in the dialogue context is a necessary but not a sufficient condition to signal acceptance. On the other hand, inconsistency of an utterance is a sufficient condition for rejection, but not a necessary one. So apparently, something more is needed.

For implicit acceptance several different types of utterance can be used: repetitions, paraphrases and utterances that make some line of inference explicit. A class of utterances that can function as potential acceptance cues is defined: redundant utterances. In Walker's definition an utterance u_1 that expresses a content φ, is called *informationally redundant* in a dialogue context represented by σ, when there is another utterance u_2 that has already been uttered, with a content ψ which entails, presupposes or implicates φ. In the terminology of chapter 3, an utterance would be informationally redundant when its content is supported by the current information state, so if $\sigma \Vdash \varphi$. Actually, Walker's definition of redundant utterances is stronger; she requires an explicit antecedent utterance u_2 that supports the information. Roughly $\sigma[\psi] \Vdash \varphi$.

Walker shows that these types of informationally redundant utterances do occur, and occur often in the corpus. This very fact is a problem for the use of update semantics as a tool for understanding naturally occurring dialogue. Update semantics is build around the notion of redundancy; without it the notion of support would be undefined. The solution lies with the notion of awareness. Once we make a distinction between the explicit information in an information state, and the implicit information that the agent is not necessarily aware of, the function of these apparently redundant utterances becomes clear: they serve to remind the participants again of their content, or of the possibility of a certain line of inference.

A crucial notion to understand acceptance and rejection, is the so called *attitude locus*: the point in dialogue at which the other participant has the first opportunity to respond. In chapter 2 we called this the transition relevant place. The idea behind the existence of an attitude locus, apart from turn taking, is related to the following collaborative principle: participants must provide evidence of a detected discrepancy in belief, as soon as possible (Walker 1996b, p269). This principle is an instance of the general principle of joint salience that we discussed in chapter 2 in the context of the work of Clark (1996). Using empirical evidence of the distribution of acceptances, Walker shows that there is indeed an attitude locus. Sequential organisation (repetition) and prosodic features (de-accenting) are the main superficial cues that help to identify informationally redundant utterances.

5.6.1 Acceptance

Walker's view on acceptance can now be summarised as follows. Informationally redundant utterances that occur in the attitude locus of a proposal, function as an indicator of acceptance. However, these utterances merely assert receipt and understanding, while implicating acceptance. The inference of acceptance is a form of default inference, based on general coherence principles. So this observation fits within the general framework of Asher and Lascarides (1998a) discussed in section 5.2.3. According to Walker the mechanism behaves much like a scalar implicature. There is a scale of strength to the various ways of responding. This is what we modelled by the ranking constraints. According to the Gricean principle of quantity, one should make one's utterances as strong as possible. Since acceptance is stronger than acknowledgement, merely indicating acknowledgement implicates non-acceptance. For example "Yeah, I see what you mean" as a response to a proposal, does not indicate acceptance. A simple 'yes' on the other hand can be assumed to indicate an acceptance, unless there is evidence to the contrary. The same holds for acceptances that are implicit in other utterances. The occurrence of an informationally redundant utterance at the attitude locus asserts acknowledgement and implicates acceptance.

5.6.2 Rejection

There are two classes of rejections: rejection of assertions – what we called denials – and rejections of proposals, suggestions and requests. Scales and scalar implicatures play an important role here. This time scales are mainly based on the content of utterances. Apart from the explicit rejections, the first kind of rejection of assertions, is called *rejection by implicature* by Walker. Consider the following exchange. Note that B's utterance is perfectly consistent with A's utterance. So as we said inconsistency is not enough to detect rejections.

(130) A: Kim ate all of the cookies.
B: She ate some of them.

There is a scale of strength that ranks utterances according to the amount of information they carry. A's statement is stronger than B's statement. Given the previous utterance, B obviously violates the Gricean maxim to make one's utterances as informative as possible. The most probable reason is that B does not agree; B's utterances implicates a rejection of A's utterance. Now this implicature is a defeasible form of inference. The rejection can be overruled by an explicit "In fact, she ate all of them!".

So a weaker utterance than would have been possible implicates a rejection. Consider again the Carl Lewis examples of chapter 4.

(131) A: Lewis jumps 7.02 meters.
B: He jumped only 6.97 at the Olympics.

There is something else going on here. B only has limited information. Therefore he merely states the information he's got. Again, it implicates a rejection. Interestingly, B's utterance can also be an acceptance: an exclamation to indicate astonishment with the amount of progress made by Lewis. The only difference lies in the tone of voice (enthusiasm) and other non-verbal feedback.

In the second class called *epistemic rejections* by Walker, rejections are indicated by utterances that violate or question the applicability of general epistemic principles like consistency, sincerity, reliability and relevance. Consider responses like "That does not add up!", "I don't trust you.", "How do you know?" or "So what?". In general, all utterances that explicitly deny one of the applicability conditions or preconditions of a successful utterance, may count as a rejection. In particular, denying presuppositions of a certain utterance can have this effect. So in example (130) B might reject A's accusation with "There were no cookies".

A third interesting class concerns rejections to proposals. These are more difficult to discern, because they often involve reasoning about planning and action. The first group rejects the content of the proposal. Recall that proposals are ordered on a scale of strength. Our work on preference orders in chapter 4 was motivated by the hope that such orders may help to indicate acceptance and rejection. The simplest case is the basic negotiation case: haggling at the flea market. The most important scale is the relative price-quality relationship. Each utterance, or even non-utterance counts as a rejection of the other proposal, until the proposals reach in the middle.

(132) B1: How much for that jacket?
A1: That is a nice one. 70 Guilders
B2: It is a bit worn at the elbows.
A2: pause
B3: I'll give you 60.
A3: All right

Walker observes three major classes of rejections: (1) refusals, with or without stating a motivation, (2) mentioning the negative consequences of the proposed action, and (3) denying the preconditions of the proposed action. The following exchanges in the ice cream parlour illustrate these distinctions. The first is a simple refusal. Often such a refusal is accompanied by a kind of reason: "No, I am on a diet". Example (134) illustrates the denial of a precondition. This is a very common way of rejecting. The third example lists possible negative consequences.

(133)	A: Cream?	propose
	B: No thanks.	reject
(134)	B: I like vanilla and strawberry.	request
	A: Sorry, we don't have strawberry.	reject
(135)	A: Shall I make you an espresso?	propose
	B: Then I can't sleep.	reject

Analogous to the epistemic rejections which relied on principles of cooperative interaction, there is a class called *deliberation rejections* by Walker, which is based on the application of principles of rationality and reasoning about action. There two types of rejections in this class: (1) negative evaluations, in which the proposed actions is stated not to be desirable, and (2) conflicting intentions, which mentions that there is another action which is better for reaching the same goal. The following examples illustrate these classes. The first is a common rejection in the ticket reservation domain. The second is taken from the TRAINS corpus.

(136) S: Three tickets for Macbeth, normal price. That makes Dfl 105,-
U: Oh, that's expensive.

(137) A: I need to get the train at Avon to Bath.
B: Get the one at Corning. That's faster.

To summarise, both rejections and acceptances can be implicitly conveyed by subsequent utterances. In general, acknowledgement is indicated by a pertinent utterance at the attitude locus; the acceptance or rejection is implicated by means of a kind of scalar implicature. For acceptances the scale is based on our ranking of act types that relates to the action ladder. The trigger is a redundant utterance. For rejections, the underlying scales are more diverse. A typical source of scales is formed by the preference orders that underly desires, and therefore also the goals of agents. Denials of assertions are indicated by a weaker expression than could have been possible. By means of a scalar implicature we can infer that the responder does not agree. Another class of rejections denies the applicability of cooperative interaction principles. Denials of proposals work in roughly the same way. Again, by implicatures on the basis of a scale, often price related, a rejection can be assumed. A last class of rejections relies on principles of reasoning about action. The desirability or the usefulness of an action to reach a goal can be denied.

5.7 Conclusion

This chapter described how a sequence of dialogue acts becomes a coherent dialogue when seen as a sequence of different moves in a dialogue game. We mention Mann (1988) in particular, who describes games as conventional interaction patterns to reach joint goals. The game rules do not only concern the sequences of moves, but also the content of the moves, the roles of the players and their social relationships as well as the private and public goals of the players. For this reason, a theory based on these dialogue games in a wider sense can give a theoretic basis to the use cases applied in chapter 1. We also mentioned an empirical approach; certain patterns of dialogue acts are more frequent in a corpus than others. This gives information about the rules that describe dialogue exchanges for a particular type of dialogue. Still the recognition of dialogue act types on the basis of surface information remains difficult.

In the discourse grammar approach there are grammar rules which describe well-formed sequences of discourse segments. The resulting parse-tree functions as a dialogue representation structure. In a unification grammar framework coherence constraints can be represented just like agreement relations for a normal grammar. An account that shares similar intuitions, Asher's SDRT, uses discourse representation theory to represent the content of discourse or dialogue segments. The various DRSs are connected in a hierarchical structure by coherence relations. Each coherence relation corresponds to a number of coherence principles: defeasible rules of inference that indicate how to attach new contributions. Now crucially, for dialogue some coherence relations correspond to exchanges or initiative-response units. An utterance can be coherently attached to the dialogue, when there is an exchange such that the utterance fits into this exchange, and the exchange contributes to the task of the ongoing dialogue.

There are many ways to define a set of dialogue acts for a particular type of interaction. A comparison of theories reveals a systematics for constructing acts on the basis of a number of parameters: is it an initiative or a response? How does it relate its content to the context? Is it an assertive an interrogative or directive? Is the content positive or negative; can it be put on a scale? Well-formed sequences of dialogue acts are a basis for a common ground among the dialogue participants that the combined content of the acts is publicly known. When we realise that all acts need some kind of grounding feedback, but that feedback is often collateral or implicit in the subsequent utterances, we can understand many types of misunderstanding from a lack of feedback.

This chapter was supposed to deal with coherence. When does an utterance 'fit' the context? We have given only part of the answer here: it must fit into an appropriate interaction pattern. It is interesting to compare it with other accounts of coherence. A central principle is that those aspects of an utterance which are already implicit in the dialogue context, need not be mentioned again. This strengthens coherence, and reduces processing effort.

The amount of coherence in text has been associated with the application of pronouns and other anaphoric expressions, in particular in the natural language generation literature. So coherence is related to the amount of coreference (Hobbs 1979). Anaphoric expressions typically indicate how new information should fit old information. When we look at verb-phrase anaphora and ellipsis, we find that a parallel semantic and syntactic structure, coherence relations and the topic-comment and focus-ground structures helps to identify antecedents for resolving anaphoric links. With respect to focus and ground, we can ask whether the focus of a utterance is relatively new or whether it is contrastive. This is related to the fact that conversation often proceeds in oppositions: "on the one hand, on the other hand". Contrast is an example of a coherence relation. The amount of coherence has been related to the way new utterances fit the topic-comment structure. If there is a particular topic, does a new utterance constitute a topic shift, or a continuation? In general dialogues with a continuing *topic chain* are perceived as coherent. If the dialogue shows many topic shifts, it is perceived as incoherent. The notion of relevance is obviously related to coherence. As we explained in chapter 3, the content of an utterance can be relevant, when it resolves a contextual issue. But, raising issues is relevant when it concerns the dialogue topic, which often related to the task.

Finally, there is a line of research that relates coherence to the usefulness of an utterance to the current goal or task. Under this view an utterance is coherent when it constitutes a step forward in a plan to solve that goal. This idea of coherence is implicit in most plan recognition approaches, e.g. (Litman and Allen 1987; Carberry 1990), and has been applied to misunderstanding by Ardissono et al. (1998).

Chapter 6

Applications

In this chapter we provide some evidence for our claims about usability in dialogue systems. We will discuss three cases in depth: usability experiments with the PADIS directory system, the development and evaluation of a voice operated voting machine called STEMSTEM, and the design of an utterance generator component for a virtuel agent with an animated face.

6.1 Introduction

In this chapter we give some background to our claims about usability in dialogue systems. In chapter 1 we proposed to analyse usability of dialogue systems as a combined notion, that involves both the task the dialogue system is designed to assist the user with, and the dialogue interaction itself. The task related properties are effectiveness and efficiency. Dialogue related properties essentially come down to coherence, although coherence also involves the relation of the function of utterances to the task. In addition, we saw that transparency is important: the function of each utterance, the reason it is asked at that point, must become transparent by the dialogue design. These considerations might be compared with the Neo-Gricean maxims of Bernsen et al. (1998). In the previous chapters we gave a theoretical foundation for the notions occurring in design recommendations. To make the theory more concrete, we will now discuss three different cases of dialogue system development, in depth.

The first case concerns usability experiments with the PADIS automatic telephone directory system, developed by Philips, Aachen (Kellner et al. 1996). As part of the graduation work of Gies Bouwman, we conducted usability experiments, focussed on the confirmation strategy of the dialogue manager (Bouwman 1998; Bouwman and Hulstijn 1998). Duration and the occurrence of misunderstanding leading to 'dead-ends', were found to be the key factors that kept users from using the system. On the basis of the first round of experiments, the confirmation strategy was adjusted, to avoid misunderstandings. Prompts were also redesigned. Reliability measures provided by the speech-recognition component were used to limit the number and length of verification questions. Tasks with a higher risk or cost were distinguished from task with a lower risk. Tasks with a higher risk, need to be verified more accurately. A new round of experiments was conducted to evaluate the change. It turned out that the changes made the system faster and reduced the number of misunderstandings. The perceived speech recognition rate went down, although the actual recognition was more or less the same. This may indicate that the changes made the system more transparent; rightly shifting the perceived cause of the problems from mis-apprehension, to mis-hearing the utterances. This case indeed shows that usability involves aspects of the task, effectiveness and efficiency, as well as aspects of the dialogue interaction. The importance of misunderstandings as the main indicator for usability, is born out. Finally, it shows that in addition to efficiency and effectiveness, transparency is important, even at the cost of some reliability.

The second case concerns the prototype of a voice operated voting machine, called STEMSTEM (Vote by Voice), developed by students at the University of Twente as part of a so called design project (Ebbers et al. 1999). STEMSTEM allows users to vote by using their voice. In a simple three stage dialogue design users select a political party, select a candidate and confirm their choice. The system was especially designed for visually impaired users, who have great difficulty with the current Dutch voting machines. This case is used to illustrate the importance of user-centred development: the involvement of a particular group of end users in the design and evaluation of a dialogue system. It shows that even with simple, commercially available speech recognition techniques, huge usability gains can be made. In another respect the system was less successful. Obviously, voting procedures are regulated by law. Under the current law the prototype would not be allowed because the three-step dialogue strategy would possibly bias the user's vote.

The third case zooms in on a particular aspect of a dialogue system: the utterance generation component (Hulstijn and van Hessen 1998). The question is, how one should control the form, content and function of the dialogue system's utterances, given a text-to-speech system and a general dialogue system architecture. Natural language generation, much more than interpretation, is forced to look at pragmatic aspects of language use. In this section we present some simple and practical solutions. We use a set of utterance templates that can be selected and controlled by a limited number of parameters. In addition, we discuss how to add control parameters to the utterance generation system, to manage the lip movement and facial expressions of an animated face (van den Berk 1998).

6.2 Dialogue Re-design with Reliability Measures

6.2.1 Introduction

The speech recognition of spoken dialogue systems is increasingly reliable. Yet, complete recognition will remain impossible; so occasionally the user will have to correct the system. In order to facilitate easy correction, and enhance user satisfaction in general, different strategies can be used. For example, one can choose an *explicit verification* strategy, always asking the user directly if certain information was correctly understood. Such a strategy is reliable, but it tends to lengthen the dialogue. Instead, an *implicit verification* strategy can be used, formulating the next question in such a way that it conveys what information was understood. In case of a misunderstanding, the user is expected to protest. Implicit verification does speed up the dialogue, but it is less reliable (Danieli et al. 1997).

We report on a re-design of the dialogue strategy of PADIS, an automatic telephone directory developed at Philips Research, Aachen (Kellner et al. 1996). The Philips Automatic Directory Information System (PADIS) allows so called *voice-dialling* or *call completion*: by only uttering a name, the user will be directly connected over the telephone to the person he or she wants. PADIS now serves as a testbed for HDDL, the dialogue description language developed at the Philips Research, Aachen. PADIS is a spoken dialogue system that, besides call completion, can retrieve e-mail addresses and room or telephone numbers of all the laboratory employees. Employees are identified by name, title, gender or working group. Because of the possibility of speech recognition errors, such information has to be specified in a dialogue. The following dialogue is an example of a typical PADIS dialogue (Kellner et al. 1996). System utterances are indicated by an S, user utterances by a U. System utterance S2 is an example of explicit verification. Utterance S4 is an example of implicit verification: it asks for new information (*which Mary*) while implicitly confirming that an e-mail address is wanted.

(138) S1: PADIS. what can I do for you?
U1: I'd like to talk to Mike
S2: So, you would like to talk to Mike Anderson?
U2: Yes, please.
S3: The extension 506 is busy right now. Can I do anything else for you?
U3: Oh yes. Can you give me the e-mail address of Mary, please?
S4: Which Mary do you want the e-mail address of?
U4: Uh, she is in the speech recognition group.
S5: So you want the e-mail address of Mary White from the speech recognition group?
U5: Exactly
S6: The e-mail address is...

After the release for Philips laboratory personnel in March 1996, the number of telephone calls per day appeared to be far below expectations. One of the main reasons, indicated by many users, was the low usability of the system in relation to the regular three paged telephone list. Until now, the development had been driven by technological possibilities rather than user demands. A user-oriented experiment was set up to evaluate the current system and to improve the dialogue strategy according to user needs. Also a new technique to determine the reliability score of the speech recognition module was incorporated to achieve this goal.

User satisfaction has been frequently used as an external indicator of *usability* (Walker et al. 1997). Usability is the ultimate goal for a system designer: usability implies that the system is useful and it is easy to use. User satisfaction also involves these two aspects. It is possible to study user satisfaction using extensive user evaluation experiments. A representative group of subjects is asked to perform a number of assignments with the system. Both their results on the assignments, judged from the log-files and responses to a subsequent questionnaire can then be used to assess user satisfaction. It is a challenge to find a set of objective metrics that are somehow correlated with user satisfaction. Such metrics may be directly and automatically available from the log-files during development and will therefore reduce the cost of development. We decided to run a subjective evaluation in parallel to objective analysis of the log-files. Because of the limited resources at our disposal, we split the subject group in two: the bigger part consisted of people who could only spare a few minutes to fill in a questionnaire. The second group also completed a number of assignments with the system.

The interaction of subjects with the system was logged. The log-files were analysed, using objective evaluation techniques for each of the assignments. We measured the number of turns (dialogue duration), the number of correction turns ("Nein, ..."), the number of misunderstandings, judged by the number of times the system asked for correction ("bitte korrigieren Sie mich") and whether or not the dialogue ended successfully.

Right after the assignments, a telephone interview was held as a subjective evaluation. Based on the results of both subjective and objective evaluation, design goals were set for a re-design of the dialogue strategy. After the system was improved a new round of experiments was held on the same group of subjects, with similar assignments.

We make the following claims:

(i) Dialogue duration is one of the most important factors to determine usability of a spoken dialogue system.

(ii) In some cases, the speed advantage of suppressing verification outweighs the disadvantage of occasional dialogue failure resulting from it.

In section 6.2.2 we present the first experiment and motivate the assignments. In section 6.2.3 we present the post test interview and its results. Section 6.2.4 will illustrate the main problems with some dialogues we collected from the log-files. Section 6.2.5 then proposes a solution, that could be realized by the dialogue strategy re-design. The results of the experiments with the new system are presented in section 6.2.6 followed by the conclusions in section 6.2.7.

6.2.2 Experiment I

Motivation

A set of assignments was designed for user evaluation. There were two important aspects. On the one hand we wanted the users to get acquainted with the system and to form an opinion about it's behaviour. In particular we were interested in the *cognitive model* that users develop during interaction. On the other hand, the goal was to observe and compare the task comprehension in different user groups on the same task. In particular, we distinguished on professional background and level of experience with speech understanding systems. So, we wanted to see whether different *user models* could be distinguished. We did not use any task-assignment design guidelines since we could not find any for spoken dialogue systems.

Assignments

The goal of the experiment was to investigate the different strategies that users apply to select a person in the PADIS telephone directory. Therefore all assignments have to do with finding information about a certain person, where the person can be specified in various ways. Some assignments were designed to test for the users attitude towards speed as an indicator of usability and to find out how users would handle misunderstanding and corrections. In order to avoid priming effects pictures and tables were used, cf. (Dybkjaer et al. 1996). About 25 subjects finally took part in the assignment test. They were free to do it whenever they had time.

1.1 How was it to speak to PADIS?

1.2 What is your impression of this system? (specific problems/ advantages)

1.3 How would you judge your own way of speaking, in relation to your normal way of speaking? (command-language, slower language, more clear language, etc.).

2.1 How easy was it to solve the task? (hard – easy)

2.2 How would you judge the error-rate? (low – very high)

2.3 How easy was it to make corrections? (hard – easy)

2.4 Would you prefer PADIS over a telephone list, and when? (never – always)

2.5 Would you use such systems in the future? (surely – surely not)

2.6 How well were you prepared to use this system? (enough – too little)

2.7 How do you like the speech output of PADIS? (too fast – too slow)

2.8 How do you like the length of the system's utterances? (too short – too long)

2.9 How do you like the length of the total dialogue? (too long – all right)

2.10 Would you like a confirmation that PADIS has understood you, and when? (always – never)

2.11 Did you know that you could still correct misunderstandings? (yes – no)

2.12 Would you think that a more strict dialogue handling, as in 'Only give the last name!', is helpful? (yes – no)

2.13 Would you like further help in using the system (yes – no), and if so how? (longer explanation, automatic explanation in case of problems, on-line help).

3 Judgements on Likert Scales of *flexibility, boredom, frustration, efficiency, desirability, reliability, complexity, friendliness, predictability* and *acceptability*.

4.1 How could the system be improved?

4.2 What did you like about the system?

4.3 What didn't you like about the system?.

Figure 6.1: English glossary of the questions in the post-test interview (Bouwman 1998).

6.2.3 Post-test interview

The last assignment was to have the system dial the number of the evaluator. The structure and contents of the subsequent interview were heavily based on Dybkjaer et al. (1996). Interviews were structured by a questionnaire in German, attached to the assignments (figure 6.1). Users would answer the questions and comment to the evaluator over the telephone. The questionnaire contained three questions about solving the tasks, the so-called 'open comments on system and self': 1.1 – 1.3. Then there were closed questions about the tasks and the system: 2.1 – 2.13. Then we asked users to judge the current system on ten Likert scales, for the aspects of *flexibility, boredom, frustration, efficiency, desirability, reliability, complexity, friendliness, predictability* and *acceptability*. The interview ended with three open questions about the whole system (4.1 – 4.3). We tried to invoke the users, many of which were scientists, to participate in the design process. Thanks to both the questionnaire and the assignment list, we already knew how users perceived their level of experience.

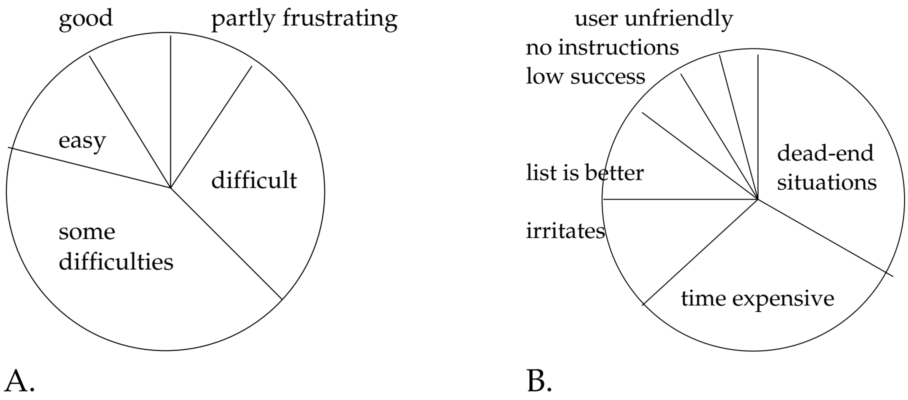


Figure 6.2: Scores on the question A. *How was it to talk to* PADIS*?* in evaluation I. and B. *What didn't you like about the system?* in evaluation I.

Interpretation of Results

Most users found it difficult to solve the assignments, or had at least experienced some difficulties. Figure 6.2.A displays the results of the question how it was to speak to the system. Judging from the answers in the interview, the main problem was the long duration of some telephone calls (time expensive). Another frequent problem had to do with the occurrence of misunderstanding resulting in so called 'dead-end' situations (section 6.2.4). Please note that these problems may be interrelated: long calls might result from misunderstanding. Figure 6.2.B displays what users did not like about the system.

With respect to the duration of the dialogues, users indicated that the speed of the spoken utterances themselves was all right, that the length of the utterances could be somewhat shorter and that most dialogues took far too much time. We concluded that the number of turns per dialogue was too high.

Furthermore, the users with intermediate or high technological insight wouldn't object against less verification of previously uttered parameters, especially when a dialogue failure had low costs: in case of direct phone connection they appreciate more extended information about whether the system understood them right, and connects them to the right person. The subjects had mixed feelings about the possibility of more help-functionality. Some thought it would make the system even slower then it was already. Others liked the idea of keyword-activated help, in order to know how to get out of a dead-end situation. From the log-files however, we learned that users don't ask for help spontaneously. Although the subjects complained about the non-cooperativeness of the system, they didn't think it was rigid. This was true in particular of people with a high level of experience with similar systems. In addition, these people were also more optimistic about acceptability of the system.

Users consider it an advantage that the system can easily be kept up-to-date, and that it is always reachable from everywhere.

6.2.4 Log-file analysis

From the log-files we collected a subset of 'problem dialogues' where the user acted normal and cooperative, while the system showed unexpected or unclear behaviour. Turn by turn we separated the regular from the irregular interactions. From the latter set we extracted an abstract classification of undesired properties of the system. The long duration of problem dialogues turned out to be caused by (i) dead-ends and misunderstanding, (ii) lengthy prompts and (iii) verification exchanges.

A good example of misunderstanding is dialogue session (1326), where the user tries to obtain the telephone number and e-mail address of a colleague. English translations are given in italics. The phrases in `type cast` indicate what the system understood[1]. Information in parentheses was understood by the speech recognition module, but ignored by the dialogue manager.

(139) (1326)

S1: PADIS, was kann ich für Sie tun?
PADIS, *what can I do for you?*

U1: Die Telefonnummer von Herrn Hattrup in der Gruppe EL.
The telephone number of Mr Hattrup in the EL group.
`Die Telefonnummer von Herrn (Hattrup:name) in der Gruppe GL:group.`

S2: Von wem aus der GL möchten Sie die Telefonnummer haben?
Of whom in the 'GL' would you like to have the number?

U2: Der Gruppe EL.
The EL group.
`Gruppe_TEAM:name - Belief inconsistent; removing old belief`

S3: Sie möchten also die Telefonnummer von der Gruppe TEAM?
So you would like the telephone number of the TEAM group?

U3: NEIN
no
`nein`

S4: Bitte korrigieren Sie mich!
Please, correct me!

U4: Telefonnummer von Herrn Hattrup
The telephone number of Mr Hattrup
`Telefonnummer von Herrn Hattrup:name`

S5: Herr Hattrup. Die Telefonnummer lautet 367. Soll ich die Telefonnummer noch einmal wiederholen?
Mr Hattrup. The telephone number is 367. Shall I repeat the telephone number once more?

U5: Nein, aber die Email-Adresse.
No, but the e-mail address.
`nein Xaver:fname die Email-Adresse`

S6: Sie möchten also die Email-Adresse von Xaver Riederer?
So you would like the e-mail address of Xaver Riederer?

U6: Warum das denn? Nein, von Herrn Hattrup.
Why is that then? No, of Mr Hattrup.
`(von) nein danke Hattrup:name`

S7: Herr Hattrup, ich verbinde...
Mr Hattrup, I connect you...

[1]The system uses tags to mark special types of input such as names (`name`) and groups (`group`).

One of the strong advantages of the system is that it allows voice-dialling: the ability to get a direct connection after just one turn. For instance session (1455). The majority of all calls is of this type. In order to allow voice-dialling, the system assumes that the user wants to speak with somebody.

(140) (1455)

S1: PADIS, was kann ich für Sie tun?
PADIS, *what can I do for you?*

U1: Herr Gies Bouwman, bitte.
Mr Gies Bouwman, please.
`Herr Gies Bouwman:name bitte`

S2: Herr Bouwman, ich verbinde ...
Mr Bouwman, I connect you...

Session (1151) is another typical example of a dead-end situation. The misunderstanding in this particular dialogue resulted from a combination of the *implicit verification* strategy with the assumed default of voice dialling. The user wants to know in which room (*Raum*) Mr Schreinemacher works. The system is not able to deduce a room number request from the word 'room' only, and assumes that the user wants to use voice dialling for Mrs (*Frau*) Schreinemacher. In the subsequent turn, the user utters his correction too slowly and is therefore interrupted by the system (S3). In U3 again the system can not deduce that a room number is wanted. After that, it cannot be corrected anymore. So for instance in U6 and U7 the speech recognition understands correctly that a room number is wanted (*Zimmernummer*), but this information is put aside by the rigorous behaviour of the dialogue manager.

(141) (1151)

S1: PADIS, was kann ich für Sie tun?
PADIS, *what can I do for you?*

U1: in welchem Raum arbeitet Herr Schreinemacher
In which room does Mr Schreinemacher work?
`nein ich brauche (den) Frau Schreinemacher:name`

S2: Sie möchten also Frau Schreinemacher sprechen?
So you would like to speak to Mrs Schreinemacher?

U2: nein ich möchte den ...
No, I would like the ...
`nein, ich möchte (den)`

S3: Bitte korrigieren Sie mich!
Please, correct me!

U3: ich möchte den Raum von Herrn Schreinemacher
I would like the room of Mr Schreinemacher
`ich möchte (den Raum von) Herrn Schreinemacher:name`

S4: Sie möchten also Herrn Schreinemacher sprechen?
So you would like to speak to Mr Schreinemacher?

U4: nein ich möchte die Zimmernummer von Herrn Schreinemacher
no I would like the room number of Mr Schreinemacher
`nein ich möchte (den Zimmernummer von) Herrn Schreinemacher:name`

S5: Bitte korrigieren Sie mich!
Please, correct me!

U5: Zimmernummer
room number
`(Zimmernummer)`

S6: Entschuldigung, ich habe Sie nicht verstanden. Sie sagten, daß Sie nicht Herrn Schreinemacher sprechen wollten. Korrigieren Sie mich bitte!
Excuse me, I didn't understand you. You said that you did not want to speak to Mr Schreinemacher. Please correct me!

U6: Zimmer, Herr Schreinemacher
room, Mr Schreinemacher
`(klar) Herrn Schreinemacher:name`

S7: Ich habe Sie immer nog nicht verstanden. ...
I still didn't understand ...

Furthermore, session (1151) contains examples of lengthy prompts and unnecessary verification turns, where the user already knows what the system is going to say, but has no way to shortcut the system's long verification turns.

In these examples the system could have taken a risk by questioning the database with uncertain data; humans have a higher ability to detect ambiguities or misunderstandings and this should be exploited in the system design. Cases like these have a low 'error cost'; erroneously retrieved information can be easily corrected. We wanted to find out when potentially superfluous clarification and verification can be avoided, thus sacrificing reliability for speed. After all, verification is selective; it should only be initiated if there is reason to suspect that the overall performance and accuracy of the dialogue system will be improved (Smith and Hipp 1994). In our system, there is more to gain than to lose. In case of misrecognition, a verification and subsequent challenge/correction takes as much turns as an information presentation and a user-initiated (partial) reformulation of the request. The profit comes from the dialogues where recognition, although unreliable, is correct and is used to retrieve the desired information immediately. If high costs are involved, we have to act more carefully, and hold on to the old strategy.

6.2.5 Strategy Re-design

The strategy is based on the belief status of parameters. During the dialogue the system needs certain parameters from the user to query the database with. Depending on the status of a parameter, the dialogue proceeds in a different way. Up to now parameters could have one of three statuses.

(142)
1. Unknown *U*
2. Known *K*
3. Verified *V*

The strategy uses implicit verification. It asks for *U*-parameters, while implicitly mentioning *K*-parameters, obtained from the previous user utterance. When users do not explicitly correct *K*-parameters, they are considered to be verified (*V*). When there are no *U*-parameters left, the user is prompted to confirm all remaining *K*-parameters. So

Transition	Condition
$U \longrightarrow U$	No parameter of this type was understood
$U \longrightarrow K(u)$	A parameter was understood, unreliably
$U \longrightarrow K(r)$	A parameter was understood, reliably
$K(u) \longrightarrow K(r)$	An implicitly verified unreliable parameter wasn't challenged
$K(u) \longrightarrow V$	An unreliable parameter was confirmed after explicit verification
$K(r) \longrightarrow K(u)$	A reliable parameter was explicitly denied and replaced by an unreliable one
$K(r) \longrightarrow K(r)$	No change or reliable replacement
$K(r) \longrightarrow V$	Explicit confirmation

Figure 6.3: Four-valued Status Transitions

the last turn is always an instance of explicit verification. The database is queried with all V-parameters. This strategy is analogous to the C_nQ-strategy of Cozannet and Siroux (1994).

Previous to the experiments reported here, a change was made to the dialogue strategy to incorporate *reliability measures*. An algorithm was added to the speech recogniser to measure the confidence or reliability of newly recognised parameters. The strategy applied this confidence score to suppress a final explicit verification, in case a unique database entry could be found using only the V- and 'reliable' K-parameters. However, even under this adapted strategy, the problems still occured. When users missed their chance to correct implicitly verified $K-$parameters misunderstanding resulted. Therefore, the *Known* status was split into two subclasses, namely reliably and unreliably known parameters. So now we use a four-valued belief status for parameters.

(143)
1. Unknown U
2. Known (unreliably) $K(u)$
3. Known (reliably) $K(r)$
4. Verified V

New parameters are assigned to one of these statuses, according to the reliability score of their recognition. $K(r)$-parameters are treated similar to V-parameters, except that they can still be corrected by the user. For stability reasons this can only be done in an explicit way, for instance by speaking clearly, possibly leading to reliable recognition or by explicitly challenging the current belief and overriding it with a new value. Implicitly verified and confirmed $K(u)$-parameters are assigned the $K(r)$-status and are therefore still open to changes. Only explicitly confirmed verifications result in a transition to the V-status. For the four-valued belief status, the most important transitions are summarised in table 6.3.

A separation between actions with a low and a high cost was also made. Information retrieval, like finding somebody's telephone number, has a low cost. It will always be done as quick as possible, even if there are unreliable parameters involved. This in contradiction to direct phone connections, that have a high cost. Direct connections can therefore only be made with V or $K(r)$ parameters, just like in the former strategy.

Besides this, we changed the formulation of certain prompts, to show the user more clearly what the current belief status is and how he or she can manipulate it. The accompanying strategy is illustrated in figure 6.4. At every point, rule 0 applies.

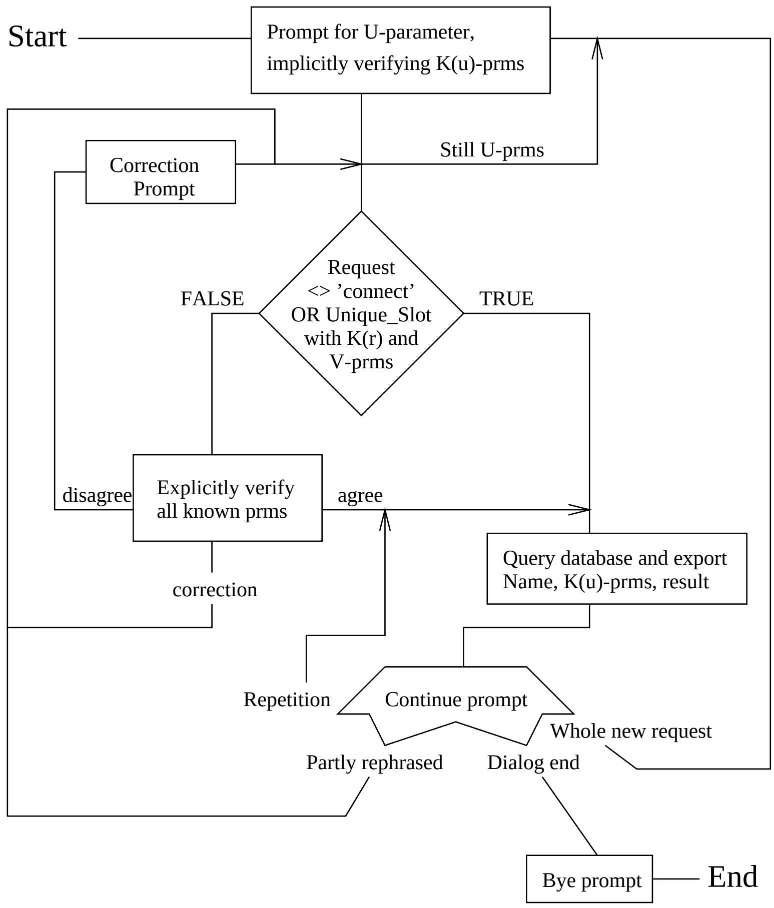


Figure 6.4: New strategy (Bouwman, 1998)

0. Goto 1, unless a request and a unique entry can be found in the database.
1. Ask for a relevant parameter that has the U-status, implicitly confirming $K(u)$-parameters.
2. If the request parameter doesn't equal *direct connection* or a unique database entry can be found using only $K(r)$- and V-parameters, then goto 4.
3. Explicitly verify all $K(u)$- and $K(r)$-parameters, focussing on $K(u)$-parameters.
4. Query the database. Report the result, preceded by the person's name and $K(u)$-parameters.
5. Ask for repetition, but listen to a possible new request.

Of course, in case of disagreement or correction, we jump back to the appropriate places. We also reformulated the final *continue prompt*, that should invoke spontaneous initiative of the user to control the continuation of the dialogue. In the new strategy, the continuation prompt allows four functions. The user may wish (i) to have the last query-results repeated, (ii) to rephrase only a part of the request, (iii) to abort the dialogue or, (iv) to start a whole new request. The continuation prompt now reads *Can I do something else for you?*. Depending on different keywords or input, the system will repeat, answer a rephrased request, quit or start a new request. Furthermore we added a 'dirty bit' for voice-dialling. When the user does not use voice-dialling in the first turn, the dirty bit

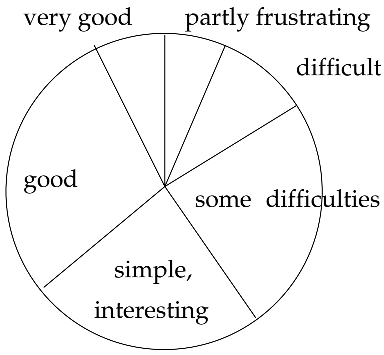


Figure 6.5: Scores on the question *How was it to talk to* PADIS*?* in evaluation II.

is set and he/she has to make a connection request explicitly. This improvement was made to prevent high cost actions like turn S7 in session 1326. Finally, some dangerous words and entries were removed from the vocabulary. For instance, group names were removed as entries in the database; they could now only be used as selection criteria. Another example is the function word 'Wiederholung' (repetition) that was often confused with a name in the database: 'Wiederhold'. To reach Mr Wiederhold, one now has to use either 'Rainer Wiederhold' or 'Herr Wiederhold'. We have no general way to solve this interference between function words and content words.

6.2.6 Experiment II

The second round of experiments was conducted along the same lines as experiment I. Some of the subjects did not participate, due to external reasons. 83 % of the original number took part. The assignments were set up in a similar way to the first set. However, one of the assignments turned out to be much harder than in the first round. It contained a word group, 'Philips Dialog Systems', that could only be recognised by the system when pronounced with a certain speech prosody. This might have affected the attitude of subjects and therefore their overall performance; speech recognition is quite sensitive to attitude related prosody, for instance irritation. Also the questionnaire of the post-test interview was similar. In addition to the earlier open and closed questions, it now contained a question asking users how they would deal with the speed-reliability trade-off.

On the whole, the second round of experiments confirmed our re-design choices. Judging from the log-files, 'dead-end' misunderstanding no longer occurred. Dialogue duration was cut considerably. Also, people did not seem to mind information retrieval based on misrecognition; they simply overruled the system with the correct value in a second try. The suppression of the repetition prompt also had a good effect. In assignment 7, where users had to pretend that they misheard the system's answer, they came spontaneously with replies like 'repetition' or 'again'.

From the post-test interviews we learned that users would like to use the system again in future. When comparing the *How was it?* question in figure 6.5, with the original results in figure 6.2 we see that the part of users that indicated 'good' has increased considerably. So overall satisfaction increased. Figure 6.6 depicts the results of the judgements of users on the Likert scales. They indicated that the system had become simpler, more flexible and

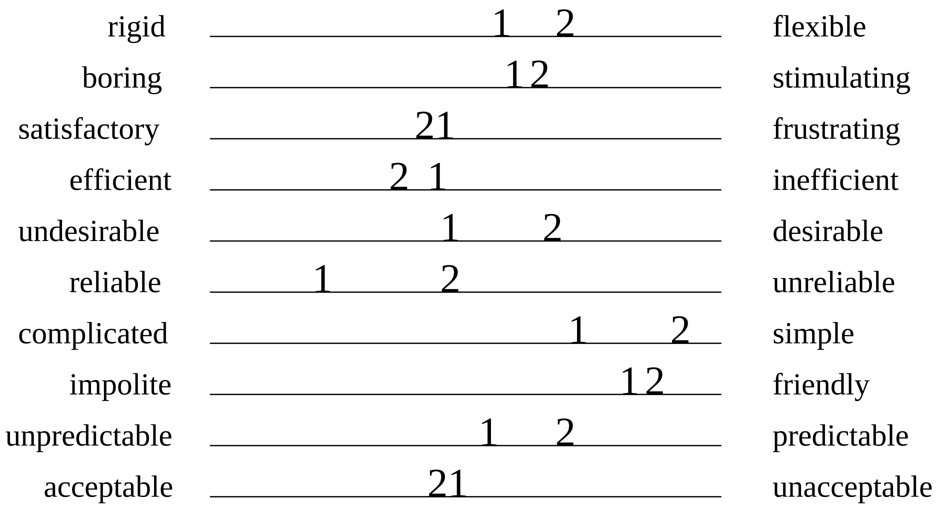


Figure 6.6: Scores on the question *What do you think of the current system? Was it, ...?* '1' indicates evaluation I and '2' evaluation II.

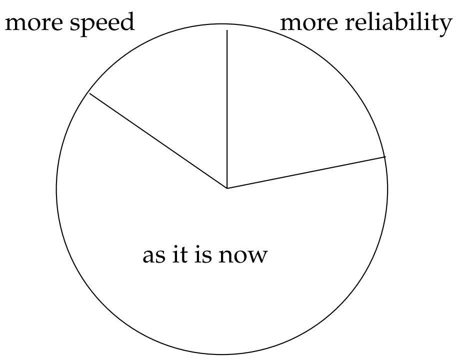


Figure 6.7: Scores on the question *Would you change the trade-off speed-reliability in* PADIS*?*

predictable and desirable. One remark is necessary: it seems that the system's reliability has decreased tremendously. However this is the result of a change in the wording of the question. The original question was interpreted by the subjects as a question about reliability of the information in the database; in the second evaluation it was understood as the reliability of the system as a whole.

The improvement on these aspects may indicate that the system gained transparency, which had a positive effect on the usability. Transparency is understood as the systems ability to be clear about the status of it's beliefs. The average user was content with our choice for the trade-off balance between reliability and 'speed' or dialogue duration. These findings are depicted in figure 6.7. Both the log-file analysis and the subjective evaluation, suggest that users were better able to pinpoint the causes of dialogue problems, which were often the consequence of speech recognition errors. In fact, users objected that speech recognition quality had decreased. This was not the case, because the recognition module remained unchanged. It might be that the new strategy accentuates mistakes in speech recognition. However, most users adapted to the appropriate repair strategy.

6.2.7 Conclusions

We agree with Cozannet and Siroux (1994), that there are two main principles for a dialogue strategy: *dialogue duration* and *reliability*. These principles form a trade-off: how much reliability should be sacrificed for a shortened dialogue duration? The subjective part of the evaluation validates the claims stated in section 6.2.1.

(i) dialogue duration is indeed one of the most important factors to determine usability of a spoken dialogue system, at least for applications like PADIS. In the first evaluation users indicated that the original system was too slow to add value to their daily tasks. The measurements taken after the re-design were all related to shortening the dialogues. The result was that users expressed their appreciation about increased usability.

(ii) in some cases, the advantage in duration of suppressing verification outweighs the disadvantage of occasional dialogue failure resulting from it. From the subjective evaluation one must conclude that a significant majority of the users appreciates the shorter dialogues at the cost of having to correct the system in some occasions.

The successful re-design illustrates that the two main design factors, duration and reliability do not have to be incompatible. When both the *cost* of a dialogue move and *reliability measures* of the speech recognition are taken into account, a good balance can be found. For information with a low cost, duration is more important then reliability. Analysis of the log-files after the second round of experiments showed that for simple information retrieval, the speed advantage indeed outweighed the decreased reliability. Users were able to correct the system and obtain the correct information in case of misrecognition. Even when the system needs correction, it takes the same number of turns as in the original strategy. For information with a high cost, reliability could be maintained. A direct connection requires explicitly verified information. After the initial turn, in which voice dialling still remains possible by default, direct connections need to be requested explicitly.

Moreover, after the re-design users correctly attributed misunderstandings to poor speech recognition. The perceived recognition rate decreased. Although this may have to do with the more difficult assignments in experiment II, we think it also shows that the new strategy accentuates recognition errors. In this way users obtain a more truthful 'cognitive model' of the system. The system may not hear very well, but it seems to understand the user better.

Acknowledgements

The research described in this section was conducted by Gies Bouwman at Philips Research Laboratories Aachen, Germany. The text in this section is almost identical to Bouwman and Hulstijn (1998). We are very grateful to Philips Research for the opportunity to experiment with PADIS. In particular, we would like to thank Bernhard Rüber for his support and numerous suggestions. Please note that PADIS was never designed for user experiments with dialogue strategies; it was merely intended as a voice-dialling system. Therefore its performance in dialogue may have stayed below that of a completely developed product.

6.2.8 Postscript: Adaptive Strategies

What has happened to the use of reliability measures in spoken dialogue systems since the experiment was conducted? Philips and other research institutions have continued to incorporate reliability measures into their dialogue strategies. There are a number of interesting developments that have come out of this kind of research.

Positive and Negative Feedback

More and different evaluation experiments were done on the trade-off between the explicit and implicit verification strategies. In particular, Krahmer et al. (1999) have done evaluation experiments to test the grounding hypothesis that underlies implicit verification. Recall that grounding proceeds in two phases: a presentation phase and an acceptance phase (Clark and Schaefer 1989). The signals in the acceptance phase can be either positive, urging the speaker to continue, or negative, urging the speaker to retract and try again. The assumption is that negative cues are relatively more marked, since an apparent communication problem is more salient. This conforms to the principle of minimal effort (Clark 1996), since a breakdown in communication would cost a lot of effort for both participants to repair. Krahmer et al. studied a corpus of 120 dialogues taken from evaluation sessions on two versions of the telephone-based public transport information system (Weegels 1999). They compared the types and amounts of negative and positive responses to both explicit and implicit verification prompts, in the case of a dialogue problem or misunderstanding and in case of unproblematic dialogue. With respect to the different cues they found the following results:

Length: utterances following an explicit verification are longer than those following an implicit verification both in case of a problem and not. A good threshold is 8 words or more.
No response: empty turns do occur and only after an (apparently problematic) implicit verification, but this happens not very frequently.
Word order: marked word orders (e.g. topicalisation) are found especially in utterances after a problematic implicit verification.
Yes/no: after an explicit verification a positive confirmation (e.g. 'yes') is found mostly in case no problem arose. The absence of a positive answer thus indicates a problem. After implicit verification it is more difficult. A negative confirmation does indicate a problem, but this does not occur frequently. More often problems are signalled in other ways.
Repeated, corrected and new slots: these cues are related to the content of the utterances. After an explicit verification prompt, both repetitions and corrections typically occur in problem dialogues. New information items occur in both and are therefore not a good cue. After an implicit verification prompt, again repetition and correction indicate a problem.

Which of these features are good cues for detecting problem dialogues? Krahmer et al. calculated the recall and precision of the predictions for problem dialogues on the basis of different cues. Following an explicit verification, the absence of a positive confirmation (yes) is the best cue for spotting an error. A negative confirmation (no) has much less recall. Following an implicit verification, the most informative cue is the number of corrected information items. The conditions length, disconfirmation (no) and marked word order also have a high precision, but a low recall, because they occur infrequently. With respect to positive cues, the best indicator of the absence of a problem is the positive confirmation for the explicit verification strategy, and the absence of corrected slots for the implicit verification strategy. Krahmer et al. also discuss combinations of cues to improve the predictions.

This research is important because it shows conclusively that a dialogue system that can detect these negative cues and positive cues, could thereby detect communication problems and adapt its verification strategy to the circumstances. Initially it will assume good communication and use an implicit verification strategy, until it detects (repeated) negative cues and switches to the safer explicit strategy. Another application would be to switch to a different speech recognition module, trained on hyper-articulate speech, after the detection of problems. Usually corrections are pronounced with a higher pitch range, more pauses and more volume. For regular speech recognition modules recognition performance usually goes down on such articulate speech. Detecting problem dialogues makes it possible to overcome this.

Adaptive Strategies

The distinction between implicit and explicit verification strategies is one possible distinction, in a space of many possible ways of formulating prompts. Based on a hierarchical slot structure Veldhuijzen van Zanten (1998) designed a fully adaptive dialogue strategy. System questions may range from relatively open questions, such as "What can I do for you?" or "When would you like to travel?", to quite specific verification questions as "Do want to travel on September 30th?".

The choice for a question type depends on the particular slots that still need to be filled in the hierarchy. This depends on the partiality of information and on the relative difficulty the user is having with the dialogue. In the examples above the slot that needs to be filled is obviously a date. In principle, open questions have a high potential for information gain. If they succeed, they greatly reduce partiality. But open questions do not give much guidance to the user. Specific questions on the other hand do not capture much information at the same time, but do give the user more guidance and therefore reduce the risk of a misunderstanding. In general, a system should start with open, unspecific questions, narrowing down to specific questions later, or when forced by communication problems.

This proposal is particularly interesting because it relates to ideas about information exchange developed in chapter 3. In a way, Veldhuijzen van Zanten's theory can be seen as an implementation of the ideas embodied in update semantics: adding information is eliminating possibilities. Depending on the currently open issues, a different system initiative is selected. Another example of adaptive strategies is given by Litman et al. (1999).

6.3 Vote by Voice

Until a few years ago, voting was done with a red pencil. Nowadays, in most constituencies in the Netherlands voting is done by a machine. The machine is made to look like the traditional ballot paper. It consists of a large panel, on which a matrix of buttons can be found. Each button corresponds to a candidate. Pressing the button makes the name appear on a small screen. By pressing another button, a new selection can be made. Pressing a special confirmation button finally confirms the choices made. Just like the traditional ballot paper the names on the panel are organised in columns that correspond to a *ballot list* (kieslijst). The same arrangement can be found on the information leaflet with candidates that every voter receives along with the ballot call. In practice, each ballot list corresponds to a political party. But legally, people vote for a candidate; not for a party. Therefore, the law provides a number of restrictions on the way political parties are presented to the voter.

It was known that these types of machines present usability problems for certain categories of voters. In general, elderly people have difficulties keeping up with technological progress. People in wheel chairs, for example, have difficulties reaching the control panel and pressing the buttons. With special arrangements, such as a different type of table and personal assistance, these problems can sometimes be overcome. For visually handicapped people however, there is no way to adjust to the present systems. Not only for people who are completely blind, but also for people with diminished eyesight the lay-out and small print of the buttons are too problematic. That means that visually handicapped people are forced to either not vote at all, or else to authorise somebody else to vote for them. They feel this as a genuine problem; it restricts them in practising their political rights. Since there is no alternative to them, for these target users the added value of a voice operated voting machines would be tremendous. In addition, automatic telephone-based voting procedures apparently have a large future in the marketing and entertainment world.

The STEMSTEM student design project was set up to find out if state of the art speech technology was good enough to build a reliable voice-controlled voting machine (Ebbers et al. 1999). The purpose was to design, implement and test a computer program with which visually handicapped could vote, using commercially available speech recognition technology and text-to-speech components. Fluent Dutch Text-to-Speech (Dirksen 1997), developed by Arthur Dirksen at OTS Utrecht, was used as the text-to-speech generator. The Philips Speechpearl toolkit for discontinuous speech was used to develop the speech recognition module. Unlike the Speechmania toolkit for continuous speech, Speechpearl requires that every possible user utterance must be stored in the lexicon as single unit. This makes Speechpearl unsuitable for dialogue systems with a mixed initiative interaction strategy, because this requires an analysis of the structure of utterances by means of keyword spotting or parsing. However, for the activity of voting a system-directed interaction mode is preferred. Because the task is so simple and because reliability of the vote is crucial, there are hardly any efficiency or usability gains to be expected in using a mixed initiative design. On the other hand, a system directed design gives users more guidance, which is beneficial for first time users. The system can instruct people what to do at every stage of the process.

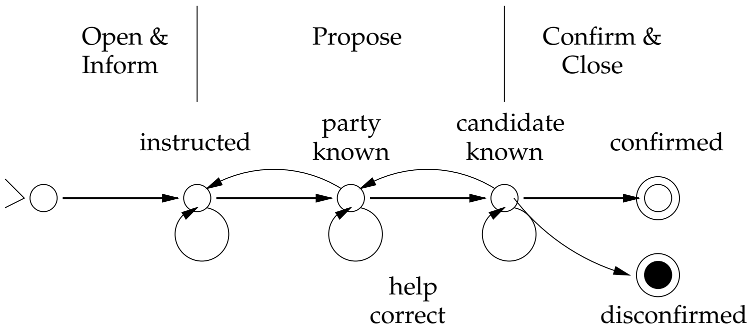


Figure 6.8: The task structure of voting

6.3.1 Analysis

The activity of voting can be seen as very basic transaction, as described in chapter 4. It is depicted in figure 6.8. After a combined opening and information stage in which the user is instructed, the user makes a proposal for a vote in two stages. First a party is selected, then a candidate is selected. Each of these stages is performed by an exchange consisting of a system initiative, followed by a user response. In case the response is positive, the system proceeds to the next stage. Otherwise, the system repeats the initiative. The content of the transaction is restricted: it concerns a single vote for a candidate. Acceptance of a proposal thus merely concerns the recognition: did the system recognise the right party or candidate, or not? In case a recognition error was made, the user can still make corrections. The user can also ask for help at any stage. These corrections and requests for help are depicted by the reflexive arcs at the bottom of the picture. The confirmation stage of a vote is rather elaborate. After the build-in implicit confirmation of the recognised material, which implies an acceptance when the user does not protest, there is an additional explicit confirmation sequence at the end. In case the confirmation fails the voting transaction should be aborted. Voting must be absolutely secure. There must be no chance of registering a wrong vote without the user's consent.

The system is meant for visually handicapped people, with few or no prior experience with either speech technology, or computers. The system should be modular and reusable. The system must be adjustable for each new election on the basis of a candidate list. The task of the system is to assist the user in selecting a candidate, confirming the selection and registering the selected candidate.

6.3.2 Design

These requirements lead to a design consisting of three modules. The main voting module registers the votes and calls the other modules. The dialogue module reads a dialogue specification file and initialises a dialogue of the specified type. The communication module acts as a bridge between the other components. It calls the speech recognition and text-to-speech components. The system was trained and tested using the official candidate list of the general election of May 1998 (Tweede Kamerverkiezingen).

With respect to the dialogue control structure, the following issues were taken into account.

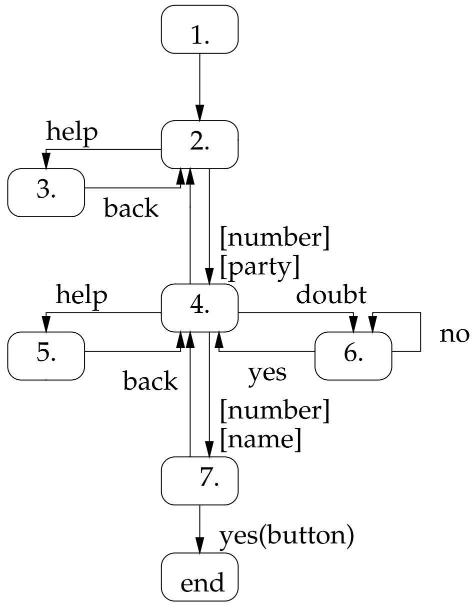


Figure 6.9: Initial dialogue design

(144)
- the dialogue should be as short as possible.
- to avoid unnatural verification questions, implicit verification is used
- the user should respond in separate words, not complete sentences.
- at every stage, the user should be able to ask for assistance, including a list of possible continuations.
- at every stage, the user should be able to go back to the previous question.
- before registration of the vote, the user is explicitly asked to confirm his or her choice, by pressing a button
- the user is not assumed to have any experience with computers or speech technology

This analysis lead to an initial dialogue design, represented by a simple finite state automaton in figure 6.9. Each node corresponds to a dialogue state. At each particular state, a particular system prompt is uttered. The numbers in the nodes correspond to the system prompts of figure 6.10. Different types of user response lead to different new dialogue states. A user response or the absence of a user response is represented by an arc leading out of the node.

In the first step, the user is welcomed and the system's behaviour is explained. The arrow from the second to the fourth state, indicates that the computer has indeed understood a party description, which is either a number, a party name or an abbreviation (e.g. 'PvdA' instead of 'Partij van de Arbeid'). In case the computer has not understood, the user is asked to repeat it, and to speak clearly. This leads to a potentially endless loop. Nevertheless, we assumed that users would be able to instruct the system eventually. In the third state a list of all the parties with their numbers and their most used names (abbreviations) is read aloud. This provides a list of candidates for the users to choose from. In the fourth state we find an example of implicit verification. In case the right party is not recognised, the user can correct the computer by saying 'Back'. Otherwise, the user names the candidate name and the voting procedure continues. The arrow between 4 and 5 indicates that the user said 'Help', said nothing or said something incomprehensible. In

1. Good morning, you are now going to vote for the general elections. Please try to speak clearly and quietly and try to respond in a single word instead of a complete sentence. At any moment you can as ask for assistance by saying ‘Help’. If you want to return to the previous question, you can do that by saying ‘Back’.
2. For which party do you want to vote?
3. You can select the name or list number of one of the following parties. Say ‘Stop’ to interrupt the enumeration.
   List 1 - Partij van de Arbeid
   List 2 - CDA
   List 3 - VVD
   etc.
4. For which candidate of [party] would you like to vote? Just mention the last name or the number of the candidate.
5. You will now get an enumeration of the candidates of [party]. First choose ‘Complete list’, or ‘List of men’ or ‘List of women’. Say ‘Stop’ to interrupt the enumeration.
6. There are more candidates with this last name. Do you mean [number][name]?
7. Do you want to vote for [number][name] of [party]? Press the button to confirm your vote.

Figure 6.10: System prompts belonging to the initial dialogue design (translated from Dutch).

that case, the user gets a complete list of candidates. Enumeration can be interrupted by ‘Stop’. In that case the system continues in state 4. The arrow from 4 to 6 indicates that the computer found several possible candidates. There are a number of candidates with the same last name on the list. In this case the system asks explicitly which candidate is selected. After the 7th state the selection is to be confirmed by the user, otherwise the system returns to 4.

6.3.3 Implementation

The dialogue structure is coded in a dialogue specification file, written in a special purpose dialogue description language. It describes the content of the prompts and the possible responses for each dialogue element, at a high conceptual level. For this reason it takes relatively little effort to adjust the dialogue design. At the beginning of each session the dialogue description is compiled.

In the dialogue specification file the design is ordered in so called *dialogue elements*. A dialogue element consists of three parts: the specification of a system prompt, the specification of the context for processing of the expected user response, and the selection of a next dialogue element. The system prompt can be either a statement or a question, and is either coded as text, as a phonetic description or a audio file. With respect to the response processing, it is indicated if the user may interrupt the system. For a statement, the selection of the next element is always the same. For a question, the selection has to be made on the basis of the user’s response. Here, a specific sub-lexicon can be indicated that will contain the expected user response. A variable is defined to store the user’s responses. A list of mappings from response types, to next dialogue elements, specifies the selection procedure. A special mapping for ‘error’ responses is also provided.

These dialogue elements used by the students in their design project, essentially correspond to the *initiative-response units* or *exchanges* discussed in chapter 5. For system-directed designs, and for simple mixed initiative designs, IR-units form a very natural way of translating a dialogue design into actual code. The system's behaviour is designed as a cycle that repeats the steps of reading out the prompt, recording, recognising and interpreting the user's response and finally selects a new prompt to read. In each cycle the same program modules can be called; which particular prompt should be read, and which context should be used in the recognition module, is indicated by the IR-units. Selection of the next IR-unit is regulated by a set of grammar rules or dialogue principles. In this way, a design based on IR-units can still account for relatively complex designs, where designing a finite state machine out of the blue becomes too cumbersome.

By specifying a sub-lexicon the speech recognition can be made context dependent. If the user is asked for a number, interpretation as numbers can be favoured. However, prompts must be designed carefully, so as to elicit a desired user response. For example, in Dutch there are many ways of answering positively to a yes/no question: “ja, ja hoor, uhuh, prima, fijn, ok, in orde, mooi, correct, ja, ...” Ideally, each of these alternatives should be included in the lexicon. A specific prompt may elicit a specific echoing response. To the confirmation question “Is het correct dat ...?” (Is it correct that ...?) a majority of users answered “Dat is correct.” or simply “Correct!”, which happened not to be in the lexicon.

How does the system deal with uncertainty and verification? SpeechPearl offers the possibility of both implicit and explicit verification. This too can be specified in each dialogue element separately. Just like in the case of PADIS the risk of an error influences the choice. For the closure of a transaction an explicit confirmation is always recommended. Confirmation by pressing a button reduces the advantages of the voice operated system. Again, physically handicapped people may have problems using it. On the other hand, the modality switch from speech to direct manipulation makes users more aware of the importance of the confirmation. This improves reliability.

6.3.4 Testing and Evaluation

Internal Test

The system was tested extensively. First, an internal test was conducted to find major problems in the software and to see if the program was ready to be taken outside. Test cases were conducted for initialisation, voice quality and understandability. Normal voting behaviour was tried for each candidate name by the designers of the system. In these tests they varied the use of first names, last names, abbreviations and numbers. Other tests involved correcting the system, voting blank, using a non-existing party name, using a non-existing candidate name, using a name that appears twice, testing the help facility and testing for unclear or fast speech. The tests revealed some initialisation errors. In particular, the possibility of a blank vote was initially overlooked. Help functioned as expected. Voting by number always went correctly. Voting by last name only turned out to be difficult and often needed several attempts to get it right. In particular, the name ‘De Jong’ gave problems, possibly because it interferes with the function word ‘stop’. In the internal test, 48 % of the votes succeeded the first time, 15 % succeeded after some tries and 11 % was incorrect.

1. Vote for somebody of your own choice.
2. Vote for somebody of your own choice, by naming the party and the candidate number.
3. Vote for a candidate form the following list, by only naming the last name of the candidate:

SGP: 23	A.P De Jong
SGP: 28	Tj De Jong
CDA: 41	Piet Jansen
CDA: 54	Arend Jansen
Senioren 2000: 5	Geert Bos
Senioren 2000: 18	Gerard Bus
D66: 24	Bob van den Bos
D66: 29	Willem Bos

4. Ask for help, if you have not already done so.

Figure 6.11: Test cases for external tests

Intermediate External Test

Second, an intermediate external test was conducted. The intermediate test followed the test case design for external tests (figure 6.11). Subjects were carefully instructed about the purpose of the test, about the voting procedure, about the function words 'terug' (back) and 'help' and were instructed to speak slowly and quietly, and in single words. For each subject the gender, age, occurrence of a regional accent and the experience level of the subject with computers and speech technology were recorded. Each trial included four test cases: (1) a vote for a person selected by the subject; this was to reassure subjects and make them get used to the set-up and the microphone, (2) a vote by party and candidate number, (3) a vote from a selected list of candidates with similar of equal last names; this test was to generate some misunderstandings and corrections and (4) asking for help, which was included to make sure all users had tried this.

The testing took place at a meeting in Hengelo of the Bartiméus foundation, which is an association that provides care facilities for handicapped people, as well as tools and appliances and information about dealing with handicaps. All five subjects were visually handicapped. There were three men between 40 and 60 years old and two women, between 20 and 30 years. Two subjects had experience with both computers and speech technology. One subject had a strong regional accent. The test configuration was a regular PC, with regular loudspeakers and the microphone adjusted in a headset. Users had to press the space-bar while speaking to 'open' the microphone.

The results showed that over all, the system could be used to vote reliably. A number of practical points were noted. Female subjects still had large difficulties getting the name 'De Jong' recognised. Male subjects had no particular problems on this name. In general, misrecognition may have to do with differences in speech rate or differences in intonation patterns between subjects. Some subjects speak slowly and clearly, as instructed. Others speak too quickly. Regional accents did not seem to make much difference. Subjects without experience with speech technology are confused when the computer repeats the wrong name after a misrecognition of a function word. It turned out that the meaning of the function words was chosen wrongly. A correction, 'nee' (no), was seen by the system as a denial of the implicit verification, and thus as command to go back to the previous stage, just like 'terug' (back). So the system interpreted 'no' as a correction of the party name, whereas often the user meant to correct the candidate name. The function word 'help' (help) usually worked, except for a user who spoke very softly, because she was shy. The function word 'terug' (back) was sometimes misrecognised, because it can be pronounced in different ways. Users often forget to press the speech button while speaking; in this way the system ignores their speech, which is confusing. Also, users often forget to finally confirm their vote by pressing the confirmation button. It seems that the modality switch is not designed well. Users often did not know when the system was finished speaking and when it was their turn to speak. Careful prompt design or an indicator like a beep might help here. Voting for candidates with foreign names poses a problem. Obviously, for foreign names the speech generation lexicon and the speech recognition lexicons must explicitly overrule the Dutch pronunciation of these names generated by the toolkits. Even though different possible pronunciations were hand-coded in the lexicon, subjects turned out to be quite inventive in the way they choose to pronounce foreign names.

Despite these problems, all subjects were very satisfied with the system. They did not expect this type of performance; especially the quality of the speech synthesiser surprised them. Subjects did not like to press the space bar when speaking. On average, a voting dialogue took about 70 seconds. A vote was never concluded in less than 50 seconds.

6.3.5 Redesign

After these tests a number of improvements to the dialogue structure was made. The result is depicted in figure 6.12. The corresponding prompts can be found in figure 6.13. The following changes were made. The welcome prompt was made simpler. The possibility of a blank vote was added. In case of a blank vote a completely separate path is followed. A number of errors or misunderstanding that may occur are explicitly handled. In particular, the computer should act when the user has not said anything, or when the computer has not recognised what the user said. This is handled in blocks 8, 11, and 13. The maximum delay before the computer takes action can be adjusted. Finally, the voting procedure can be interrupted by the command 'Stop', at any moment. This is done so that, in case the voting is a total disaster, the user may get some assistance and start again. In this case no vote is registered.

External Test

On the basis of the internal and the intermediate tests, the dialogue design was adopted. The improved dialogue design is shown in figures 6.12 and 6.13. This design was tested again. Unfortunately, the set-up of pressing the space bar while speaking could not be adjusted. The objective of the second external test was to assess the usability of the voting system. Could the system actually be used by target users, to cast their votes reliably?

The testing was done with fourteen subjects, four women and nine men, in the age range between 20 and 60. All subjects were visually handicapped or blind. The majority of the subjects was from Twente. Some had strong regional accents. One subject was born in Turkey and had a strong foreign accent. Six people had experience with computers, but little experience with speech technology. The test configuration was a laptop computer with separate loudspeakers, and a microphone adjusted in a head-set. The recorded and the recognised speech of the users was logged. Subjects were instructed about speech

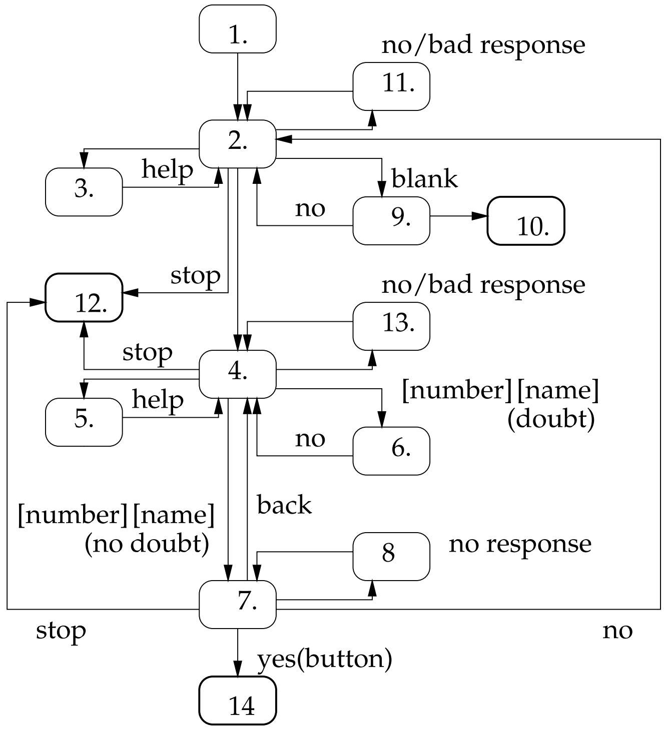


Figure 6.12: Improved dialogue design

recognition and about the program. This time they were not told that they could use function words 'back' and 'help'. The test cases were similar to the ones used in the intermediate test, with the exception of voting by number. Again, we tried to initiate some misunderstandings by asking subjects to vote for candidates with difficult, foreign or non-unique names. This may have influenced the success rate, because it is known from elections that most people vote for the party leader.

The results are as follows. In total 61 votes were made. Eight of these were votes for the wrong candidate. That is 13 % percent. Thirteen votes, or 21 %, involved misrecognitions of the system which needed some correction. The remainder of forty votes or 66 % was recognised without problems. The average time to vote was 62 seconds; invalid votes not counted. The fastest vote took 32 seconds; the slowest 251 seconds.

The following issues were remarkable. In general, subjects did not know what to do in case of a misrecognition. In the introductory prompt the system explains the existence and use the function words: 'back', 'no' and 'help'. Nevertheless, users hardly ever use them unless they are being reminded of the possibility by the experimenters. Apparently users do not pay attention to instructions or can not apply them when needed. This is not strange. People that use a voice operated computer for the first time receive a lot of information at once. Similar problems with instruction texts were found in user studies of the telephone based public transport information system (Weegels 1999). In any case, function words should be designed to be natural in the circumstances in which they have to be used. So instead of using 'back', one should use 'no' as a keyword for corrections. The results showed that the program almost lived up to expectations. In general, subjects were able to vote. However, in some cases, the program did manage to register a wrong vote. This shows that the final confirmation mechanism needs to be revised.

1. Welcome. You are now going to vote for the general elections. Please try to speak clearly and quietly and try to respond in a single word instead of a complete sentence. At any moment you can as ask for help by saying 'Help'. If you want to return to the previous question, you can do that by saying 'Back'.
2. To which party belongs the candidate that you want to vote for?
3. To the party question you should respond with the name of a political party. You may mention the list number as well. Please pronounce the party name clearly and quietly and keep the button pressed while speaking.
4. For which candidate of [party] would you like to vote?
5. To the candidate question you should respond with the name of a person: only the last name or first the first name or initials followed by 5the last name. Say 'Back' if you want to respond to the party question first. Please speak clearly and quietly and keep the button pressed while speaking.
6. You mean [candidate 1] or [candidate 2]?
7. Is it correct that you want to vote for [candidate] of [party]?
8. Say 'yes' if you do want to vote for that person. Otherwise say 'No'. You may still indicate whom you want to vote for.
9. You voted blank. Is that correct?
10. Your blank vote has been registered.
11. Your response to the party question has not been recognised. Please try again. Instead of a party name you may also say 'Help' for clarification.
12. The voting dialogue has terminated.
13. Your response to the candidate question has not been recognised. Please try again. Instead of a name you may also say 'Help' for clarification.
14. Your vote has been registered.

Figure 6.13: System prompts belonging to the improved dialogue design.

6.3.6 Conclusion

What can we conclude from this application of speech technology? First of all, it can be concluded that the technology is good enough to be applied in simple tasks. Users did manage to vote; with another round of testing and redesigning the prompts and the function words, the system might have reached an acceptable quality level. The results of the improvements after the intermediate testing round show once again that it is crucial that design is seen as an iterative process, and that expected target users are involved in the design from the beginning. It also shows that details do matter. For example, getting all the different possible pronunciations of foreign candidates into the lexicon improves general performance considerably.

A second conclusion concerns the high user satisfaction rates, for a system that is still rather clumsy and difficult to handle. When there are no real alternatives, the added value of an imperfect system may still be huge. Users are willing and quite able to adapt to a system, if it enables them to do their task, in this case, without the help of others. This is consistent with results of the Dutch public transport information system, which show that the success rates of the system when used at night are much higher, when the alternative human operator is not available (Sturm et al. 1999).

Thirdly, the successful application of IR-units in a finite state design shows that for simple applications a simple design method is sufficient. The theory of initiatives with expected responses can be easily projected onto a dialogue description language which guarantees a flexible and adjustable implementation.

Fourthly, the voting application shows that speech and language technology are applied as a part of society. Voting has obvious legal and social consequences. It may be that the law that currently prohibits a three-stage voting process is too much biased towards the traditional paper procedure. According to our task analysis the activity of voting can be seen as a three-step transaction process anyway, consisting of the selection of a party, of a candidate, and a confirmation. In this respect we might say that the law is obsolete.

6.4 Utterance Generation with an Animated Face

The VMC (Virtual Music Centre) is mentioned throughout this thesis as an example of a virtual environment where users and artificial agents can interact (Nijholt et al. 1999; Nijholt 1999; Nijholt and Hulstijn to appear). On the basis of the actual blueprints a virtual copy of the music centre was constructed which now serves as a testbed for new ideas on human-machine interaction. In chapter 2 we discussed theoretical issues related to situated and multi-modal interaction. In this section we want to address some of the more practical issues in the design of an embodied dialogue agent. We designed a spoken version of the SCHISMA dialogue system (van der Hoeven et al. 1995). The design is currently being implemented; similar design considerations apply to the implementation of a navigation agent, which is designed to help users navigate through the virtual space (Klumper to appear). The agent for theatre information and reservation is called Karin, and is situated in the VMC behind an information desk.

The first part of this section describes how to give Karin a voice. We discuss the design of an utterance generation module for Karin (Hulstijn and van Hessen 1998). Most of the original ideas are from van Deemter et al. (1994), van Deemter and Odijk (1997), van Deemter (1998), Theune (1997) and Dirksen (1992). The utterance generation module works with prosodicly annotated utterance templates. An appropriate template for a given dialogue act is selected by the following parameters: *utterance type*, *body* of the template, *given* information, *wanted* and *new* information. Templates respect rules of accenting and deaccenting.

The second part of the section deals with parameters for controlling of Karin's animated face, developed by van den Berk (1998). The spoken utterance generation module can be extended with parameters for controlling mood, attention and facial gestures. We use the mechanism of the utterance generation module to calculate the intonation centre of an utterance and synchronise it with the culmination point of a facial gesture.

6.4.1 Utterance Generation

We assume the system is mixed initiative. During inquiry the user has the initiative; the system answers the user's questions. This makes sense because the user is most competent in what she wants to ask. When the user has indicated that he or she wants a transaction – a reservation in our case – the system takes initiative. The system is most competent in what is needed for a successful reservation. The system checks if all parameters are known. If not, the system will ask for them. The inquiry and transaction functions are intertwined.

The system consists of three modules in a pipeline architecture. User utterances are processed by a *speech recognition* module including a disambiguation module and a limited parser. When a keyboard-based dialogue system is adapted for speech, one of the most difficult changes concerns the *verification* mechanism (section 6.2). Verification prompts are distinguished by a rising intonation, indicating uncertainty. The module that decides what to do next, is called the *dialogue manager*. It maintains two data-structures: a representation of the *context* and a representation of the *plan*, the current domain-related action that the system is trying to accomplish. The dialogue manager interprets the parser output against the context. It updates the context, resolves presuppositions and anaphora and applies inference rules that constitute domain knowledge. Based on the parser output, the context, the plan and the database, the dialogue manager selects a response action. Actions are composed of basic *database actions* and basic *dialogue acts*. Planning and action selection are based on a set of principles, called *dialogue rules*. Finally, each dialogue act is put into words by a general purpose *utterance generation* module. It determines the utterance-structure, wording, and prosody of each system response. General rules govern the translation of dialogue acts into the parameters that control the style of prompts. A different set of rules produces a system with a different personality. In an efficient, curt system, pronouns and implicit objects are preferred over definite NPs to express the given items, and short answers are preferred over longer answers that echo part of the question.

The utterance generation module uses a list of utterance templates. Templates contain gaps to be filled with *information items*: attribute-value pairs labelled with syntactic, lexical and phonetic features. Templates are selected on the basis of five parameters: *utterance type*, the *body* of the template and possible empty lists of information items that are marked as *given*, *wanted* and *new*. The utterance type and body determine the word-order and main intonation contour. The given, wanted and new slots, as well as special features, affect the actual wording and prosody.

Utterance Type	Surface features	Intonation Contour
whq	finite verb on 2nd position, wh-word on 1st	hat
decl	finite verb on 2nd position, no wh-word	hat
ynq	finite verb on 1st position, subject on 2nd	rising
imp	finite verb on 1st position, no subject	hat
verif	verification, as decl	rising
text	longer text to be read	depends on punct
short	short answer (PP,NP,ADV)	hat
meta	e.g. thanks, greetings, yes, no,	hat

Figure 6.14: Basic Utterance Types

Utterance type The *utterance type* roughly corresponds to the mood of a sentence: whether it is a declarative, wh-question, yes/no question or imperative. However, many utterances do not consist of complete sentences. Often they involve short answers, commands or remarks, consisting of single VPs, NPs or PPs. Figure 6.14 shows a list of utterance types. It is a modified version of the utterance types developed for classification of utterances in the SCHISMA corpus (Andernach 1996). The first column in figure 6.14 gives the main syntactic features; often the position of the subject and finite verb. The last column gives the main intonation contour. The regular declarative intonation is the *hat* shape: rising during the phrase and falling at the end. The regular interrogative intonation is characterised by a sharp *rising* at the end. When applied to an otherwise declarative sentence, a rising intonation indicates uncertainty. This is applied in verification prompts. For the reading of longer texts, the system assumes a *reading voice*. Within text, the intonation depends on punctuation marks. Other utterance types may be added for application in multi-modal interaction, such as a *table* format for listing performances on the screen.

Body This slot selects the *body* of the template: the content that is not especially marked. It corresponds to the *tail* in the *link-focus-tail* trichotomy of Vallduví (1990). Note that the 'link' corresponds to the topic of the conversation, and his 'focus' corresponds to our 'new'. In most templates the body is expressed by the main verb. Usually it is deaccented, but important cue words like *niet* (not) or *maar* (but) get some extra stress.

Given The *given* slot contains information that is presented as given in the conversation. This is usually reflected by the use of pronouns, definite articles or demonstratives. For most templates the given elements occupy the 'topic' position: the first position before the verb, or the position just after the verb when some new element is topicalised. With respect to intonation, given items are deaccented. Very often the 'given' items refer to the topic of the conversation (Rats 1996), the object the conversation is currently about. But not always. The topic-comment distinction does not align with the given-new or focus-ground distinctions (Vallduví 1990). In fact, in most utterances the global topic, the performance or the reservation, is not mentioned at all. It is implicit. In the templates the choice between pronouns, definites and implicit objects is marked by features. The decision is made according to the dialogue rules that govern the translation of dialogue acts into the five utterance parameters.

New This parameter contains information items which are presented as either *new* or *contrasted* with respect to the dialogue context. These items roughly correspond to the focus of an utterance. The notion of focus is not well defined for questions and imperatives. Therefore we prefer the older given-new distinction. In most templates new elements are placed towards the end of the utterance. The new field is also used for items that need contrastive stress, as in (147c) that suggests that 'Macbeth' is sold out, but no other performances.

Wanted This *wanted* slot contains the type of information that is wanted from the user. This is indicated by the choice of wh-word, e.g. *why* for reasons, or by the NP in the wh-phrase, e.g. *which genre* for genres. Examples (145 – 147) show the mechanism of template selection. Only some of the features are shown. The parameters are listed as: `utterance type; body; given; wanted; new`. Features and values are given between brackets.

(145) a. `meta:[att:sorry]; no;;;`
Nee, sorry.
No, sorry
b. `verif;;;date:tomorrow;`
Morgen?.
Tomorrow?

(146) a. `whq; want; you:[polite:+]; thing;`
Wat wilt u?
What would you like?
b. `whq; want; you:[polite:+]; performance[prep:to]; date:tomorrow`
Naar welke voorstelling wilt u morgen?
To which performance would you like to go tomorrow?

(147) a. `decl:[att:sorry]; sold_out; it;;`
Sorry, maar het is uitverkocht.
Sorry, but it is sold out.
b. `decl; sold_out; performance:[impl:+, title:'Macbeth'];;`
'Macbeth' is uitverkocht.
'Macbeth' is sold out.
c. `decl; sold_out; ;; performance:[def:+, title:'Macbeth']`
De voorstelling 'MACBETH' is uitverkocht.
The performance 'MACBETH' is sold out.

6.4.2 Prosody and Text-to-speech

For pronouncing the utterance templates we use the *Fluent Dutch* text-to-speech system (Dirksen 1997). Fluent Dutch runs on top of the MBROLA diphone synthesiser (Dutoit 1997). It uses a Dutch voice, developed at the Utrecht institute of linguistics (OTS). Fluent Dutch operates at three levels: the *grapheme* level, the *phoneme* level and a low level representation of *phones* where the length and pitch of sounds is represented. For many words, the phonetic description is taken from lexical resources of *Van Dale* dictionaries. Other

Operator	Utterance manipulation
Question	Change the usual hat-shaped intonation contour into a rising contour. This is used for verification prompts and yes/no questions.
Quote	Set a string of words apart from the rest of the utterance by small pauses and a lifted pitch level. This is used for names and titles.
Accent	Calculate the intonation centre for the accented phrase. The intonation centre is at the vowel that would receive the highest pitch when pronounced in a declarative way. Add 10 % to the pitch level. Lengthen the vowel by 50 %. This is used for new items and important cue words.
Deaccent	Calculate the intonation centre for the deaccented phrase. Level the pitch to the average of the two neighbouring intonation centres. This is used for given items.

Figure 6.15: Generalisations for prompt manipulation

phonetic information is derived by heuristic rules. We design prompts at the grapheme level. It is possible to manipulate prosody by adding punctuation at the grapheme level, by adding prosodic annotations at the phoneme level or by directly manipulating the phone level. General operators for prompt manipulation are listed in figure 6.15.

We can also adjust other speech parameters, such as the speaking rate and the general pitch level. This influences the general impression of the voice. For reading reviews and other text fragments the system assumes a *reading voice*, characterised by a lower speaking rate and a reduced pitch range. At the end of the transaction the system concludes with a cheerful wish: *Veel plezier met <title>!* (Have fun with <title>!). It sounds cheerful, which is achieved by a higher speaking rate, a slightly lifted pitch level and a large pitch range. We expect to make the user feel committed by adding this extra exchange of greetings at the end of the transaction. For the user this is a last opportunity to change her mind. It is also meant to show that the system 'cares'; it will hopefully add to the personality of the system. Obviously such a claim needs to be verified by user centred evaluation studies.

6.4.3 Synchronisation with an Animated Face

We developed an animated talking face to be used in the VMC virtual environment (van den Berk 1998). The face has a set of basic facial features. Figure 6.16 shows the face in an angry and in surprised or uncertain state. Now consider the synchronisation between different interaction modes, like speech and gesture, that would be needed in the utterance generation component for an integrated multi-modal dialogue system with an artificial face. How can we integrate the earlier prosodic parameters, with the parameters for controlling facial expression? For example, when should we have the face smile? When to raise its eyebrows?

With respect to lips and mouth, the animated face can synchronise the movements of its lips on the text-to-speech server. The basics of the phoneme to viseme conversion are as follows. There are classes of phonemes that look similar, when pronounced. Such a class defines a *viseme*, a visible speech unit. Visemes are obviously connected to phonemes but are not exactly simultaneous. Each of these classes is assigned a visual representa-

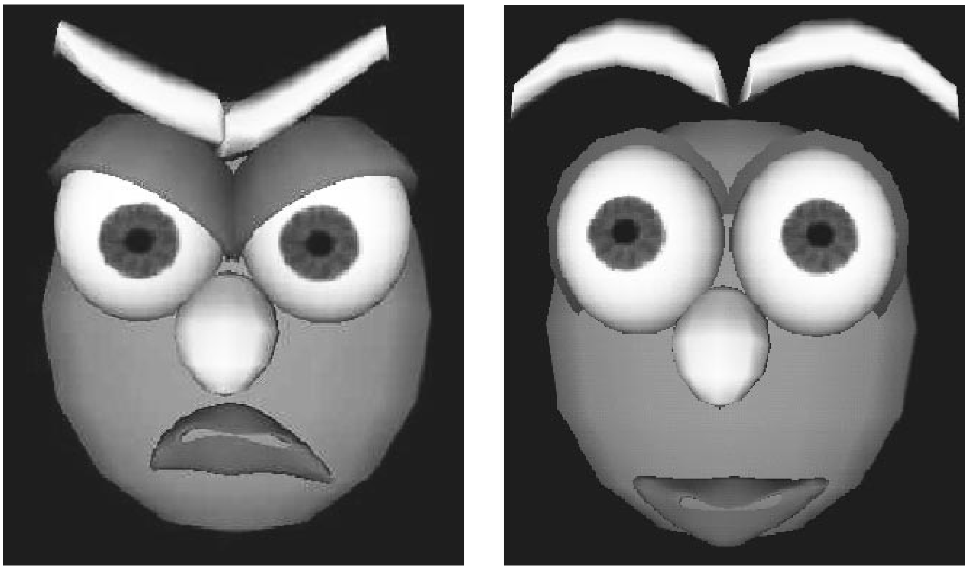


Figure 6.16: The face in an angry and a surpised state

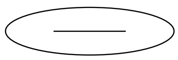
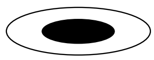
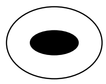
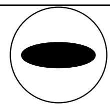
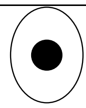
									
p	**p**ot	D	**d**ak	t	**t**ak	S	**sj**aal	f	**f**eest
b	**b**ak	K	**k**oe	tj	**tj**arda	H	**h**uis	v	**v**eel
m	**m**ok	G	**g**oal	dj	**dj**oerd	j	**j**as	w	**w**el
		n	**n**at	z	**z**on	J	aa**i**	W	eeu**W**
		nj	ra**nj**a	s	**s**la	y	f**uu**t		
		i	p**i**t	Z	ra**g**e	2	k**eu**s		
		E	p**e**t	G	**g**even	Oe	fr**eu**le		
		Y	p**u**t	x	**ch**aos	Y:	k**eur**		
		A	p**a**d	N	ba**ng**	O:	k**oo**r		
		O	p**o**t	l	**l**at	E~	t**im**bre		
		@	g**e**val	L	kaa**l**	A~	ch**an**son		
		i	n**ie**t	a	t**aa**k	U~	parf**um**		
		R	paa**r**	e	k**ee**t	O~	b**on**b**on**		
		r	**r**aak	o	p**oo**k				
		u	b**oe**k	I:	k**ee**r				
				Ei	t**ij**d				
				9y	h**ui**s				
				Au	k**ou**d				

Figure 6.17: Phoneme to viseme conversion for Dutch

tion. We apply the matrix in figure 6.17, which is a Dutch approximation of the model of Parke developed for the DECface facial animation program (Parke and Waters 1996). As a working hypothesis, we currently ignore dynamic effects. For instance, to make an /s/ the mouth already has to start moving long before the onset of the sound.

Apart from the lips, the virtual face has a number of dynamic control parameters (van den Berk 1998, p 39). These are listed in figure 6.18. The *eyes* can gaze at a certain direction. This can be used to direct attention towards an area. The *eyelids* may be opened and closed, for blinking. The *eyebrows* can be lifted to indicate surprise or lowered for distress. The shape of the *mouth* can be manipulated into a smile or and angry expression. The *colour* of the face can be deepened, to suggest a blush that indicates shyness or embarrassment. The *orientation* of the head can be manipulated, leaning forward and backward or tilting left and right. This may produce important facial gestures like nodding and shaking one's head. It can also be used to indicate attention; leaning forward means being interested, leaning backward means loosing interest. In general, a character's facial animation should not be still. The head should wiggle a bit and its eyes should wonder. This is called *idle behaviour*. Many existing 'talking heads' look artificial because of their stillness. Moreover, not moving can also be taken as a sign. For instance, Clermont et al. (1998) found that a fixed stare indicates a misunderstanding in the dialogue. The *frequency* of idle movements is an indicator of the livelihood of the character; it serves as a type of volume, to the existing emotion. So, many random movements of the head, combined with smiles and attentive eyes, indicate a very happy personality; stillness, a neutral mouth shape and looking away, indicate a withdrawn and unhappy personality. But an angry face, combined with a blush and a lot of movement, indicate increased anger. Jerky movements with wondering eyes indicate nervousness. Since our agent is supposed to be professionally friendly, she will be generally smiling and will have a moderate movement frequency.

Each of these basic features can be combined into facial *gestures* that can be used to signal something. Gestures like nodding, shaking and shrugging can be used separately, but often utterances are combined with gestures or utterance related facial expressions. The timing of the gesture or the expression must be aligned with the utterance. We use the following general heuristic for alignment of gestures.

Like any event, an utterance and a gesture have an *entry* and an *exit* point. Moreover, an utterance can be broken down into phrases; each phrase has a so-called *intonation centre*, the moment where the pitch contour is highest. Since pitch accents are related to informativeness, we can assume that the accent lands on the most prominent expression. Usually the accent lands towards the end of an utterance. Similarly, each gesture has a *culmination point*. For instance for pointing, the moment that the index finger is fully extended. The visual animator extrapolates a nice curve from the entry point to the culmination and again to the exit point. Our current working hypothesis is that gestures synchronise with utterances, or precede them. So we link the gesture's entry and exit points to the entry and exit points of the utterance and make sure that the culmination point occurs before or on the intonation centre.

So how do we control this wealth of features? We propose a blackboard architecture. A *blackboard* architecture is often used when different modules potentially influence the processing of the other modules (Engelmore and Morgan 1988). Speech recognition may for example benefit from pragmatic expectations from the dialogue manager. A blackboard architecture makes use of a common storage space: the blackboard. Each of the

feature	manipulation	meaning
eyes:	gaze direction	idle behaviour, attention, indexing
eyebrows:	lift, lower	surprise, distress, angry
lips:	form visemes stretch, round	talk smile, laugh, neutral, angry, kiss
mouth shape:	stretch, round	smile, neutral, angry
colour:	blush	shyness, embarrassment
head:	orientation idle behaviour movement frequency	nodding, shaking head, attention neutral emotional 'volume'
shoulders:	shrug	indifference

Figure 6.18: Facial features

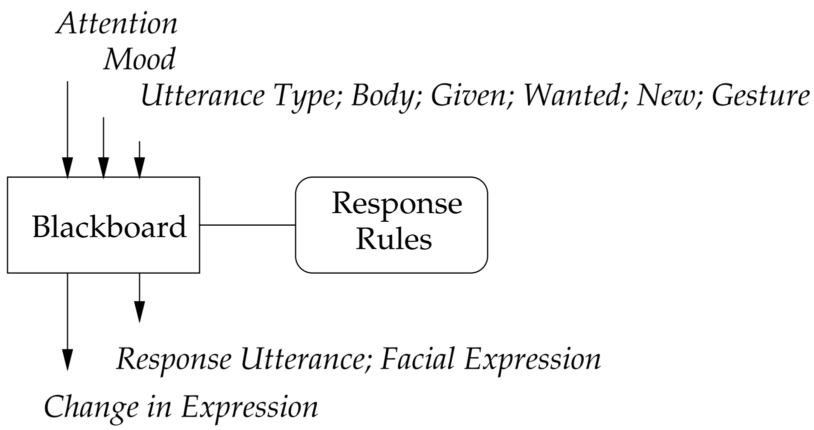


Figure 6.19: Blackboard Architecture for Controlling Facial Features

modules can read and write on the blackboard, independent of the others. Modules react to new information as it comes available on the blackboard. So a pipeline architecture as discussed above can be seen as a special case of a blackboard.

A good way of implementing the blackboard knowledge structure is by a typed feature structure (Carpenter 1992). Reading and writing on the blackboard happen by unification of feature structures. Often defaults can also be handled. Using a special type of default unification, one value may override another. Each of the modules controls a specific part of the typed feature structure, that again corresponds to a compartment of the blackboard. Compare the *sign* of a lexicalised feature grammar that has features for different linguistic aspects: phonology, syntax, semantics and pragmatics. Communication between these different aspects is well-defined by re-entrance: the structure sharing of parts of a typed feature structure.

To integrate animation and utterance generation a blackboard is used. The input and output parameters for the rules are indicated in figure 6.19. van Deemter (1998) defines rules in a similar way to account for the interaction of different semantic parameters on intonation patterns. Each specific combination of input parameters triggers a rule which produces a response action, which is controlled by the output parameters. A response can take the form of a gesture or utterance, or of a more permanent change of expression. Roughly, there are two types of facial behaviour that need to be modelled.

Firstly, we need to model permanent features like the facial expression, gazing direction and general movement characteristics, both when speaking and when idle. These can be controlled by two parameters: *mood* and *attention*. The *mood* parameter indicates the general attitude of the personality in the conversation. It is a state that extends over a longer period. Is the agent happy, sad, angry or uncertain? The *attention* parameter controls the eyes and gazing direction. We believe that one of the benefits of a talking face is that turn taking and attention management in dialogues will be made easier. The gazing direction of the eyes and the head position are crucial for this. Experiments with different video conferencing environments have shown that gaze information is more important to a smooth conversation, than a simple television type image of the talking person, when it is unclear what he or she is looking at (Vertegaal 1998). Usually mood and attention are fixed for a given personality. Temporary changes in emotion and attention may result from previous utterances or from the conversation in general. For instance, anger at an insult or increased interest after a misunderstanding.

Secondly, we need to model utterance related attitudes. Since we cannot monitor the user's utterances in real-time, at the moment this is limited to system utterances only. Think of smiling at a joke, raising eyebrows at a question or a pointing gesture to accompany a deictic expression. Conventional gestures can be modelled as a special instance of response actions. Nodding or shrugging are coded like any other utterance, except that they can be silent. Utterance related features are controlled by the existing utterance parameters, extended with a new parameter, *gesture*, that contains a label for (a combination of) facial gestures to be synchronised with the utterance template. Because we know all utterance templates in advance, the synchronisation can be manually adjusted if needed. The extend of the gesture and its final shape also depend on the general emotional state and attention level.

We also hope to introduce some variation in the exact choice of movement. Variation is important. For instance, it is natural to combine "yes" with a nod, but when every affirmative is combined with the same nod it looks mechanical. Another example is the raising of eyebrows. In an early version of the system the eyebrows were directly controlled by pitch level. Thus, the agent would nicely express uncertainty on a question, which has a rising intonation. But of course, pitch is also used for accenting. So the agent wrongly expressed surprise at expressions that were stressed. Synchronising the apparently random movements with fixed speech from templates is difficult. We have chosen to align the culmination points of the movement with the intonation centres of the phrases uttered. But the exact frequency and extent of the movements will be randomly distributed, partly based on mood and attention.

6.4.4 Why Embodied Agents?

There is a development in research from spoken dialogue systems, via multi-modal systems based on interactive maps and diagrams, towards full embodied conversational agents, e.g. (Cassell to appear). One of the reasons to add a face to a dialogue agent, is that it allows designers to show the emotions of the system. The agent may look concerned, happy or unhappy. We expect that this advantage is most important for systems that attract leisure users. We envisage a merge between interactive dialogue systems on the one hand and computer game technology on the other. The player that is most versatile in controlling a conversation, wins the game. How to express emotions and how to try to affect the users' emotions is the topic of the emerging field of *affective computing*. A particular example of this may be the generation of *computational humour*. For an introduction into computer generated humour see the collection (Hulstijn and Nijholt 1999). Given rich enough semantic resources it is possible to make a system that generates simple puns (Binsted 1996). One needs a number of pun or riddle templates, of the kind "What you get when you breed an X with a Y?" To fill the templates, a knowledge base is required with semantic information, often in the form of stereotypes of people and events. In addition, one needs a selection function that selects the funniest combinations of concepts X and Y to be combined in a template. In evaluation studies Binsted found that the computer generated puns were recognised as jokes, although not as very funny ones. Interestingly, the puns were rated more funny by children than by adults. One could say that computer humour is still in its infancy.

However, embodied agents have some added value beyond affective capabilities. Sengers (1999) argues that a *comprehensible* artificial agent should at all times be able to show what it is doing and why. One of the ways in which this can be done is by the body posture and gazing behaviour. Embodied agents have more ways of expressing the motivation behind their behaviour. Sengers developed a story with two cartoon-like characters that embody artificial agents. In one condition the agents would simply enact the story; in the other condition, they would look at each other and hesitate at the beginning of a new episode, and indicate by shrugs and by the movement frequency whether they were excited or not. In this way, the story became much more comprehensible. In their movement, the agents expressed some of the motivations behind their behaviour.

The same holds in principle for facial expressions. Facial expressions are not only used for showing emotions. Eye-gaze behaviour, body posture and facial gestures like nodding facilitate the communication process. We believe that the benefit of an animated face is mostly due to its potential for non-verbal feedback. Cassell and Thórisson (1999) show that feedback concerned with the process of communication, is rated higher than emotional feedback, which is concerned with possible emotional problems related to the content of the interaction.

This concludes the overview of some of the design issues in development of an animated face. We are experimenting with various applications that may wear the face. Apart from Karin, the information agent, we think of an automated story teller. Special SGML tags could be used to put director's hints into the text of a story. At the episode of the wolf, for example, the storyteller could lower its voice and focus on the listener, to make the episode scary. This is an active area of research. For an overview of our ideas and hopes for the future we refer to Nijholt and Hulstijn (to appear).

Chapter 7

Conclusions

This chapter concludes the thesis. There are two main themes that connect the various chapters. One combines aspects of chapter 2, 4 and 5. It concerns the relation between interaction patterns and theories for joint action and planning. We argue that dialogue games are recipes for joint action. The other theme is the official topic of the thesis. It concerns the usability modelling of chapter 1 and 6 and the content related coherence properties defined in chapter 3. We characterise consistency, informativeness, relevance and licensing, and discuss several aspects of coherence and transparency.

7.1 Dialogue Games are Recipes for Joint Action

Communication is driven by people solving coordination problems. On the other hand, communication processes themselves are combinations of *joint actions* between speaker and hearer (Clark and Schaefer 1989). Joint actions are carefully coordinated. The effect of coordination is the establishment and maintenance of a *common ground* among participants (Lewis 1969). The common ground contains mutual information of all aspects relevant to the interaction. First, there is a shared background of social, cultural and linguistic conventions. Second, there is mutual awareness of the physical and social situation. Third, there is public information on the recent interaction history, including recent contributions by the participants and on the current status of the interaction. The common ground both produces and facilitates coordination. Roughly, there are three ways to coordinate: by *convention*, by *agreement* or by *salience* (Clark 1996). Communication processes can be seen as joint actions that are scheduled in parallel *at* and *between* different linguistic levels. The synchronisation aspects of communicative processes can be modelled by layered protocols expressed in a process algebra.

At the *presentation level* speaker and hearer coordinate on mutual attention and on the delivery of the utterance. This constitutes a series of joint actions of vocalising-attending subsequent utterances. A similar remark can be made about gestures, which are deliberate movements, that have to be attended to.

At the *locutionary level* participants coordinate on the wording, structure and prosody of an utterance or on the shape and size of a gesture. Such form-related aspects convey information about the semantic content and communicative function of the utterance. For example, form-related aspects like parallelism and intonation indicate the scope of a focus sensitive operator, such as ‘only’ or ‘not’.

At the *illocutionary level* participants establish the semantic content, and communicative function of an utterance, in relation to the context of surrounding utterances. At this level an utterance can be described as a *dialogue act*. A dialogue act is fully characterised by a *semantic content* and a *communicative function* which has task-related aspect and an interaction-related aspect. The semantic content of an act models the information it conveys. In an update semantics, the meaning of an utterance is equated with the difference it makes to the information in the dialogue context. Information is modelled as a set of possible worlds which is structured by a partition, or equivalently, an equivalence relation to model the current 'questions under discussion' or *issues* (Ginzburg 1995; Hulstijn 1997). Different types of acts affect different aspects of the information in a dialogue context. We distinguish *assertives*, that add factual information, *interrogatives* that raise issues and thereby structure the information (Groenendijk 1999), and *directives*, that affect the commitments being made by agents and thereby partly determine future actions. The communicative function is often related to the the underlying task. An initiative raises an issue. It is used to invite a response that would resolve the issue. The response may be needed for the task. However, there are also dialogue control acts such as greeting and acknowledgement whose function is merely related to the interaction process (Allwood et al. 1992). An acknowledgement signals receipt and understanding of an utterance; it helps to establish a basis, an interaction history, upon which a common ground can be build. Many utterances both have a task and an interaction-related communicative function. For example, a pertinent response also functions as an acknowledgement.

At the *perlocutionary level* participants coordinate on participatory actions to accomplish a joint goal, which is partly determined by the social activity. For different activity types different conventions apply. Activity types are determined by the roles of the participants, their private and public goals, the physical and social settings and a number of phases that indicate the progress of the activity. In a transaction there is an information phase in which information is exchanged about the transaction, a negotiation phase with proposals and counterproposals, followed by a confirmation phase in which the transaction is closed. Like any interaction, a transaction has an opening phase in which mutual contact is established, and a closing phase in which the contact is closed.

7.1.1 Plans, Goals and Commitment

Like any action, a dialogue act can be characterised by its preconditions and intended effect, sometimes extended with applicability conditions and failure conditions. From basic dialogue acts plans can be constructed. Plans do not have to be re-calculated every time. The agent may select a pre-compiled plan from a plan-library. The exact plan can not always be known in advance. In dialogue, you do not know how the other participants will respond. Dialogue is opportunistic, so it pays to have only a partial specification of the actions to undertake. Previously compiled partial specifications of future action are called *recipes*. They contain roles to be filled by agents and for each role a task (Grosz and Kraus 1996). Once a plan to achieve a particular goal is adopted, the agent will typically persist in trying to achieve the goal until it is reached, becomes unreachable or becomes undesirable. This aspect is called *commitment* (Cohen and Levesque 1990). It stabilises behaviour, and makes the agent more reliable. It is crucial for cooperation. Commitments made to other agents are the 'glue' that keeps joint actions together. There is a social obligation to keep your commitments, but also common decency plays a role.

What are the characteristics of a joint action? Grosz and Kraus (1996) describe the following set of constraints. In particular, (i) agents must do means-end reasoning involving joint actions, (ii) there must be decision procedures for selecting a recipe and for assigning agents to roles and thus to tasks, (iii) there must be communication procedures for reaching agreement, (iv) to assess their abilities, agents must compute the context in which they plan to carry out an action, and (v) the intended effects of sub-actions must be consistent. Note that constraints (ii) and (iii) refer to conventions for interaction.

In the account of commitments of Wooldridge and Jennings (1999) a joint commitment of a group of agents A is characterised by an *immediate goal*, a long-term *motivation goal*, a set of *preconditions* and a set of *conventions* ρ that defines the interaction protocol. Commitments are goals that are meant to be kept, but only relative to a motivation and as long as it is practical. The *conventions* indicate, among other things, the circumstances under which a commitment can be abandoned. A convention is modelled by production rules. Each time the trigger condition is true, the agent will adopt the corresponding goal. In effect, these rules constitute what we have called a recipe above, although Wooldridge and Jennings do not deal with partiality. The convention parameter ρ can be filled in different ways. For example, it may contain sanctions upon not keeping a commitment. Usually it contains a rule to notify the other agents, once you have completed a participatory action. What are sensible ways to fill in the ρ parameter? What are good recipes for the kinds of joint action that we find in communication?

7.1.2 Dialogue Games

We do not need to analyse each utterance in terms of the underlying intentions. Looking at a corpus, greetings simply mark the beginning of an interaction; acknowledgements just follow assertions. Such stereotypical sequences of dialogue acts are called *interaction patterns*. It is possible to analyse such patterns by plan recognition and generating a co-operative response (e.g. Litman and Allen 1987). However, many replies seem not to be consciously deliberated by the agent, but conventionally triggered by the circumstances. Interaction patterns can also be fruitfully studied and applied at the surface level.

A useful metaphor for studying interaction patterns is that of a *dialogue game*. Each participant plays a role and expects the others to play their respective roles. The rules and the state of the game, the game board if you like, together determine which moves are allowed. But each move in turn determines the state of the game. The dialogue game board functions as a semantics: each utterance can be interpreted as change to the game board. There is also the aspect of winning the game. A move is useful when it contributes to the underlying task. So the notion of a dialogue game 'in a wider sense' involves all aspects of coherence. The form indicates how to recognise moves and which moves are possible at all. The content determines how the move relates to the dialogue game board. The function of the move has both an interaction related aspect, is the move allowed at this point, and a task related aspect, how does the move contribute to winning the game?

Based on corpus research Carletta et al. (1997) have detected frequent interaction patterns, analysed as conversational games. Moves in a game can be either initiatives or responses. Typically, each initiative must be followed by an appropriate response, although there may be other sub-games too. For example, a clarification sequence may precede the answer to a question. Initiative-response units have been successfully used in the design of spoken dialogue systems, e.g. (Bilange 1991; Jönsson 1997).

Discourse grammars (Prüst et al. 1994) are used to study ellipsis and parallel elements in adjacent utterances. Discourse grammar rules define constraints for well-formed sequences of utterances. The resulting parse tree may function as a discourse representation. At the leaves we find the content of an utterance; the branches are labelled by coherence relations. This approach has also been applied to dialogue. Asher and Lascarides (1998a) propose coherence relations and coherence principles for dialogue, which are partially based on initiative-response units, and partially on contributions to the task.

The dialogue game metaphor has been applied in different ways. Interestingly, early work by Levin and Moore (1978) and particularly Mann (1988) lists the following parameters for a game definition. (i) There are roles, in this case initiator and responder. (ii) The game has an illocutionary point, the goal of the initiator in starting the game. (iii) The responder has goals upon accepting the game, and (vi) there are a number of constraints: the initiator and responder must pursue their respective goals, goals must be believed to be feasible, the illocutionary point must not already be achieved, the initiator must have the right to initiate the game, and the responder is willing to pursue its goals upon accepting the game. These parameters and constraints are very similar to the ones for joint action listed above. The goals upon accepting a game are nothing but commitments of initiator and responder.

If we can integrate these three ways of using the dialogue game metaphor, the coherence constraints of the discourse grammar rules and coherence principles, the empirical data on frequently occurring interaction patterns, and the link to joint action by means of the commitments of initiator and responder upon accepting a game, then we have a good general framework to study the notion of coherence. Such an account of dialogue games can serve as a theoretical background to use case analysis tools, discussed in chapter 1. There we argued that the current tools for use case analysis do not support the right specification mechanisms for mixed-initiative dialogue. Tools that support aspects of the dialogue game metaphor discussed here, would be better suited.

In particular, we suggest unification-based grammar rules and typed feature structures or otherwise a default logic combined with DRT to account for the interaction related aspects. For the task, we would need an environment for joint planning and action, with scheduling algorithms and a partial representation of recipes.

That concludes the comparison of accounts of joint action with interaction patterns. What it shows is that the alleged opposition between two approaches to the study of natural language dialogue, namely interaction patterns analysed as dialogue games versus plans and goals (e.g. Jönsson 1993, Ch3) is false; the two approaches are complementary. Once the constraints of joint planning and action are taken seriously, a need for conventions to regulate the interaction arises. These conventions are represented by recipes. The smallest recipes for joint action, are precisely the interaction patterns described by dialogue game rules. Dialogue games thus serve as 'compiled-out' recipes for joint action. On the other hand, plans and goals may function as a semantics for dialogue game rules. They motivate the illocutionary point of initiating a game and explain various aspects of cooperativity in dialogue. This is compatible with a general trend towards hybrid agent architectures, which increasingly combine high-level deliberative features with low-level reactive features.

7.2 Usability Modelling

In chapter 1 we introduced usability as the key quality factor for dialogue system design. Because of the acclaimed advantages of natural language interfaces, this is where most of the added value for dialogue systems is to be expected. We formulated a hypothesis about the relationship between the quality factor usability, and a number of dialogue properties, in particular effectiveness, efficiency and coherence. We argued first that these dialogue properties affect usability, and second that they might be verified for a dialogue design, given a sufficiently formal model of the task and application domain. Thirdly, we hoped to develop dialogue design principles on the basis of these dialogue properties.

In the course of the thesis we developed the machinery to answer the first part of the claim. In chapter 3, we developed a theory that relates coherence properties at the content level to information in a dialogue context. The theory combines update semantics (Veltman 1996) with a semantics of questions and answers (Groenendijk and Stokhof 1996). An information state is a set of possible worlds, structured by an equivalence relation that indicates whether worlds are indistinguishable with respect to their answers to the current questions under discussion, or issues. The semantic import of an utterance is modelled by an update of the information state. Assertive updates eliminate worlds from the set; interrogative updates refine the issue structure. Now the content of an utterance is consistent with the context when its update does not result in an empty set of worlds. It is informative, when the update actually eliminates some possibilities, or removes some indistinguishabilities. An utterance is considered relevant, in a very technical sense, when its content resolves at least one issue in the context. This means that some alternative answer is eliminated. Finally, an utterance is considered licensed, when it resolves only current issues. It should not be over-informative. The relevance of issues themselves is more difficult to assess. It is related to the topic of the conversation, or to the task.

In chapter 4 we developed a theory of task success, which is directly linked to effectiveness and efficiency, and also to coherence at the task level. Moreover, we gave an account of awareness in terms of issues. Raising an issue is relevant when it makes others aware of the possible alternatives. The desires of agents are modelled by preference orders. Based on an insight of von Wright (1963), we defined an update of the preference order of an agent, by means of the conjunction expansion principle. If an agent indicates that it likes φ better than ψ, we eliminate all pairs of worlds from the preference order that rank $\varphi \wedge \neg\psi$-worlds above $\psi \wedge \neg\varphi$-worlds. Moreover, the order is closed under a number of preference constraints. In a cooperative setting, preference statements are often used to express requests, suggestions or proposals. The effect of these directive types of dialogue acts is the establishment of a commitment to future action.

In chapter 5, we continued our study of coherence, adding coherence constraints for the interaction level, and indicating how various aspects of coherence may be integrated in a dialogue game. And although syntactic, lexical and prosodic aspects of utterances were never really the focus of this thesis, we did develop some ideas on the coherence of utterances in these respects, part of which were applied in the utterance generator discussed in section 6.4. One could argue that a design based on a consistent and systematic use of dialogue principles based on these properties, especially the principle of coherence, enhances the transparency of a dialogue design.

So we might say that the first claim is warranted: effectiveness, efficiency and coherence are related to usability. A transparent dialogue design facilitates a coherent dialogue, and therefore possibly enhances understandability, learnability and operability. On the other hand, this is all a matter of definition. Why don't we include other aspects among the usability properties, such as the notion of reliability discussed in chapter 6? For PADIS the trade-off was between reliability and duration. And also for voting the aspect of reliability is crucial. We classified reliability under effectiveness, because it affects the task. The idea to take the expected costs of a failure into account, can also be interpreted in this light. For some tasks, reliability is more valuable than for others. But such a move remains speculative. In other words, to present this list of dialogue properties as a complete theory of usability, bags the question. How would you evaluate such a theory? The evaluation would apply measures from the very theory to be tested: signs of consistency, signs of coherence, signs of transparency etc. The only real test would be a test that addresses *user satisfaction*. If we could establish a correlation between usability metrics based on the dialogue properties addressed above, and measures of user satisfaction, we could validate our theory of dialogue properties affecting usability. In that case we could do some comparisons to see which factors most contributed to the user satisfaction. And even then, such results would still be dependent on the kind of application that we tested; for leisure users for example, we would predict that efficiency is far less important.

The set of usability related dialogue properties was also meant as a guideline for design. Similar to the neo-Gricean maxims of Dybkjaer et al. (1998), Bernsen et al. (1998), for each of these properties there is a corresponding maxim: be effective, be efficient, and coherent! And the last recommendation is the most important one. Coherence is an umbrella term. A dialogue is coherent to the extend that each of the utterances 'fit' into their context. And as we have seen, this involves several different aspects. In particular it involves aspects of form, content, and the task and interaction-related communicative function.

7.2.1 Towards Principles for Design

We have not systematically investigated what principles for dialogue system design can be based on the properties characterised in this thesis. For every property there is corresponding maxim to be consistent, relevant and coherent. But although we have a theory that links abstract usability properties to (formal) linguistic theories of dialogue, we have yet to indicate how these insights can be transferred to dialogue design. As a first step, we take one dialogue design property, transparency, and show how it relates to coherence. A dialogue design is transparent to the extent that it does not allow the user's mental model to deviate from the system's capabilities.

In chapter 6 we reported on an evaluation experiment of a dialogue strategy re-design (Bouwman and Hulstijn 1998). Part of the re-design involved the speech recogniser's reliability measures. The strategy was changed to take the cost of failure into account. For tasks with a high cost of failure the implicit verification was changed into an explicit one. For tasks with a low cost of failure, the verification was suppressed altogether, in case the reliability measure exceeded a threshold. For such tasks the user can always correct the system. This speeded up the dialogues considerably. The evaluation after the re-design showed that general user satisfaction had increased. Another remarkable result was that the perceived speech recognition rate had gone down. So people thought the system did not recognise their utterances as well as before. However, according to objective performance measure the recognition rate had not decreased at all. This may be explained as follows. After re-design users were forced to correct the system in case of a misrecognition. Probably this made them more aware of speech recognition errors. On the other hand, it removed some unnecessary verification effort. One could argue that this made the system more *transparent*: the effort was shifted from the dialogue in general to actual cases of misrecognition. One could even argue that it affected the mental model. Whereas at first users might have thought that the system did not understand them, they now realised that the difficulties were due to misrecognition.

In general, a dialogue design tends to be transparent when it indicates what it is doing, and why. For example, a good SCHISMA prompt might read: "So you want to make a reservation. What is your name please?". Now the user at least knows why the name is requested. A system prompt adds to a transparent design when its form clearly indicates its content – $?x.name(x, you)$ – and its function. The function is here to initiate a question-answer unit, which is related to the task of making a reservation. So the aspects of coherence generate a kind of check-list for transparency.

An intriguing feature of the VMC application (Nijholt and Hulstijn to appear), is that it attracts *leisure users*. These users are not necessarily interested in completing a task. Designing and testing artificial agents for such leisure environments is a challenge. Task based evaluation properties, such as effectiveness or efficiency, become moot. For a leisure user it is the entertainment value that counts. An important aspect is the *immersion* of the user into the environment: the extend to which the user feels herself to be a part of the virtual space. Not only the passive quality of the graphics, but also the interactive aspects of an environment and the ability for users to actively accomplish something, will play an important role in immersion. Therefore we expect dialogue agents embedded in such a virtual space to become increasingly important. Sengers (1999) argues that the behaviour of artificial characters must be comprehensible. Even for leisure environments, characters should indicate how a move or act is motivated, in terms of a believable story or personality.

Bibliography

Aist, G. (1998). Expanding a time-sensitive conversational architecture for turn-taking to handle content-driven interruption. In Manell, R. H., and Robert-Ribes, J., editors, *Proceedings of the Fifth International Conference on Spoken Language Processing (ICSLP'98)*, paper number 928.

op den Akker, R., ter Doest, H., Moll, M., and Nijholt, A. (1995). Parsing in dialogue systems using typed feature structures. In *Proceedings of the International Workshop on Parsing Technologies*, 10–11. Prague/Karlovy Vary, Czech Republic.

Alchourrón, C. E., Gärdenfors, P., and Mackinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530.

Alexandersson, J. (1996). Some ideas for the automatic acquisition of dialogue structure. In Luperfoy, S., Nijholt, A., and Veldhuijzen van Zanten, G., editors, *Dialogue Management in Natural Language Dialoguye Systems*, TWLT11, 149–158. University of Twente, Enschede.

Alexandersson, J., Maier, E., and Reithinger, N. (1995). A robust and efficient three-layered dialog component for a speech-to-speech translation system. In *Proceedings of the 7th Conference of the European Chapter of the ACL (EACL-95)*. Dublin, Ireland.

Allen, J., and Core, M. (1997). Draft of DAMSL: Dialog act markup in several layers. Technical report, Discourse Research Initiative, Schloss Dagstuhl.

Allen, J., Kautz, H., Pelavin, R., and Tenenberg, J. (1991). *Reasoning about Plans*. USA: Morgan Kaufmann, San Mateo, CA.

Allen, J. F., and Perrault, C. R. (1980). Analyzing intention in dialogues. *Artificial Intelligence*, 15(3):143–178.

Allen, J. F., Schubert, L. K., Ferguson, G., Heeman, P., Hwang, C. H., Kato, T., Light, M., Martin, N. G., Millerand, B. W., Poesio, M., and Traum, D. R. (1995). The trains project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical AI*, 7:7–48.

Allwood, J. (1995). An activity based approach to pragmatics. Technical report, Gothenburg papers in Theoretical Linguistics 76, University of Gothenburg, Göteborg, Sweden.

Allwood, J. (1999). Capturing differences between social activities in spoken language. In van Leusen, N., van Rooy, R., and Zeevat, H., editors, *Formal Semantics and Pragmatics of Dialogue (Amstellogue'99)*.

Allwood, J., Nivre, J., and Ahlsén, E. (1992). On the semantics and pragmatics of linguistic feedback. *Journal of semantics*, 9:1–26.

Andernach, T. (1996). A machine learning approach to the classification of dialogue utterances. In *Proceedings of the Second International Conference on New Methods in Natural Language Processing (NeMLaP-2)*, 98–109. Bilkent, Turkey.

Anderson, A., Bader, M., Bard, E., Boyle, E., Doherty, G., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C., and Thompson, H. (1991). The hcrc map task corpus. *Language and Speech*, 34(4):351–366.

Anderson, A., and Belnap, N. D. (1975). *Entailment: the Logic of Relevance and Necessity (I)*. Princeton University Press.

Androutsopoulos, I., Ritchie, G. D., and Thanish, P. (1995). Natural language interfaces to databases - an introduction. *Natural Language Engineering*, 1(1):29–81.

Ardissono, L., Boella, G., and Damiano, R. (1998). A plan-based model of misunderstandings in cooperative dialogue. *Human Computer Studies*, 48:649–679.

Asher, N. (1993). *Reference to abstract objects in discourse*. Kluwer, Dordrecht.

Asher, N. (1998). Varieties of discourse structure in dialogue. In Hulstijn, J., and Nijholt, A., editors, *Formal Semantics and Pragmatics of Dialogue (Twendial'98)*, TWLT13, 1–10. University of Twente, Enschede.

Asher, N., and Lascarides, A. (1997). Bridging. Technical report, HCRC, University of Edinburgh. Submitted to Journal of semantics.

Asher, N., and Lascarides, A. (1998a). Questions in dialogue. *Linguistics and Philosophy*, 21:237–309.

Asher, N., and Lascarides, A. (1998b). The semantics and pragmatics of presupposition. *Journal of semantics*, 15(3):239–.

Aust, H., and Oerder, M. (1995). Dialogue control in an automatic inquiry system. In Andernach, J., van de Burgt, S., and van der Hoeven, G., editors, *Corpus-Based Approaches to Dialogue Modelling*, TWLT9, 45–50. University of Twente, Enschede.

Austin, J. (1962). *How to do things with words*. Harvard University Press, Cambridge Mass.

Beaver, D. (1996). Presupposition. In van Benthem, J., and Ter Meulen, A., editors, *Handbook of Logic and Language*. Elsevier, Amsterdam.

Bell, J., and Huang, Z. (1997). Dynamic goal hierarchies. In Cavedon, L., Rao, A., and Wobecke, W., editors, *Intelligent Agent Systems: Theoretical and Practical Issues*, vol. 1209 of *Lecture Notes in Computer Science*, 88–103. Springer-Verlag, Berlin.

Bell, J., and Huang, Z. (1999). Dynamic obligation hierarchies. In McNamara, P., and H., P., editors, *Norms, Logics and Information Systems*, 231–246. IOS Press.

van den Berk, M. (1997). Het ontwerpen van een natuurlijke taal verwerkend systeem in de vorm van een gesproken dialoogvoering met behulp van spraakherkenning. Student Project Report, University of Twente, Enschede.

van den Berk, M. (1998). Visuele spraaksynthese. Master's thesis, University of Twente, Enschede.

Bernsen, N. O., Dybkjaer, H., and Dybkjaer, L. (1998). *Designing Interactive Speech Systems. From First Ideas to User Testing.* Springer-Verlag, Berlin.

Bilange, E. (1991). A task independent oral dialogue model. In *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics (EACL-91)*, 83–88. Berlin.

Binsted, K. (1996). *Machine Humour: An Implemented Model of Puns*. PhD thesis, University of Edinburgh.

Boër, S., and Lycan, W. (1986). *Knowing Who*. MIT Press, Cambridge Mass.

Bohlin, P., Cooper, R., Engdahl, E., and Larsson, S. (1999). Information states and dialogue move engines. In Alexandersson, J., editor, *Proceedings of the IJCAI'99 Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Stockholm*, 25–32.

Bolognesi, T., and Brinksma, E. (1987). Introduction to the ISO specification language LOTOS. *Computer Networks and ISDN Systems*, 14:25–59.

Bouwman, G. (1998). Spoken dialog system evaluation and user-centered redesign with reliability measures. Master's thesis, University of Twente, Enschede.

Bouwman, G., and Hulstijn, J. (1998). Dialogue (re-)design with reliability measures. In Rubio, A., et al., editors, *Proceedings of the 1st Int. Conf. on Language Resources and Evaluation (LREC)*, 191–198. Granada, Spain.

Bratman, M. E. (1987). *Intention, plans, and practical reason*. Harvard University Press, Cambridge Mass.

Bretier, P., and Sadek, D. (1997). A rational agent as the kernel of a cooperative spoken dialogue system: Implementing a logical theory of interaction. In Müller, J. P., Wooldridge, M. J., and R.Jennings, N., editors, *Intelligent Agents III: ECAI'96 Workshop on Agent Theories, Architectures and Languages (ATAL)*, vol. 1193 of *Lecture Notes in Computer Science*, 189–204. Springer-Verlag, Berlin.

Brown, P., and Levinson, S. C. (1987). *Politeness: some universals in language use*. Cambridge University Press, Cambridge. Second edition, First edition (1978).

Bunt, H. (1989). Information dialogues as communicative action in relation to partner modelling and information processing. In Taylor, M. M., Néel, F., and Bouwhuis, D., editors, *The structure of Multimodal Dialogue*, vol. I, 47–71. North Holland.

Bunt, H. (1996). Interaction management functions and context representation requirements. In Luperfoy, S., Nijholt, A., and Veldhuijzen van Zanten, G., editors, *Dialogue Management in Natural Language Dialoguye Systems*, 187–198.

Carberry, S. (1990). *Plan recognition in natural language dialogue*. MIT Press, Cambridge, Mass.

Card, S. K., Moran, T. P., and Newell, A. (1983). *The psychology of human-computer interaction*. Erlbaum, Hillsdale, N.J.

Carletta, J., Isard, A., Isard, S., Kowtko, J. C., Doherty-Sneddon, G., and Anderson, A. H. (1997). The reliability of a dialogue structure coding scheme. *Computational linguistics*, 23(1):13–32.

Carletta, J. C. (1996). Assessing the reliability of subjective codings. *Computational Linguistics*, 22(2):249–254.

Carlson, L. (1983). *Dialogue Games*. Reidel, Dordrecht.

Carpenter, B. (1992). *The Logic of Typed Feature Structures*. Cambridge University Press, Cambridge.

Cassell, J. (to appear). Not just another pretty face: Embodied conversational interface agents. *Communications of the ACM*.

Cassell, J., and Thórisson (1999). The power of a nod and a glance: Envelope vs. emotional feedback in animated conversational agents. *Applied Artificial Intelligence*, 13:519–538.

Chafe, W. L. (1976). Givenness, contrastiveness, definiteness, subjects and topics, and point of view. In Li, C., editor, *Subject and Topic*, 20–55. Academic Press, New York.

Chopra, S., and Parikh, R. (1999). An inconsistency tolerant model for belief representation and belief revision. In Dean, T., editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, 192–197.

Clark, H., and Marshall, C. R. (1981). Definite reference and mutual knowledge. In Joshi, A. K., Webber, B. L., and Sag, I. A., editors, *Elements of Discourse Understanding*, 10–63. Cambridge University Press, Cambridge.

Clark, H., and Schaefer, E. (1989). Contributing to discourse. *Cognitive Science*, 13:259–294.

Clark, H. H. (1996). *Using Language*. Cambridge University Press, Cambridge.

Clermont, T., Pomplun, M., Prestin, E., and Rieser, H. (1998). Eye-movement research and the investigation of dialogue structure. In Hulstijn, J., and Nijholt, A., editors, *Formal Semantics and Pragmatics of Dialogue (Twendial'98)*, TWLT13, 61–75. University of Twente, Enschede.

Cockburn, A. (1997). Using goal-based use cases - transitioning from theory to practice. *Journal of object-oriented programming*, 10(7):56–62.

Cohen, P. R., and Levesque, H. J. (1990). Intention is choice with commitment. *Artificial Intelligence*, 42:213–261.

Cohen, P. R., and Perrault, C. R. (1979). Elements of a plan-based theory of speech acts. *Cognitive Science*, 3:177–212.

Cozannet, A., and Siroux, J. (1994). Strategies for oral dialogue control. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'94)*, 963–966.

Danieli, M., Gerbino, E., and Moisa, L. (1997). Dialogue strategies for improving the naturalness of telephone human-machine communication. In *Proceedings of the ACL Workshop on Interactive Spoken Dialog Systems*, 114–120. Madrid.

Davidson, D. (1980). *Essays on Actions and Events*. Oxford Univerity Press, Oxford.

van Deemter, K. (1994). What's new? a semantic perspective on sentence accent. *Journal of Semantics*, 11:1–31.

van Deemter, K. (1998). A blackboard model of accenting. *Computer Speech and Language*, 12(3):143–164.

van Deemter, K., Krahmer, E., and Theune, M. (1999). Plan-based vs. template-based NLG: A false opposition? In *Proceedings KI'99*, 12–15. DFKI, Saarbrucken.

van Deemter, K., Landsbergen, J., Leermakers, R., and Odijk, J. (1994). Generation of spoken monologues by means of templates. In Boves, L., and Nijholt, A., editors, *Speech and Language Engineering*, TWLT8, 78–96. University of Twente, Enschede.

van Deemter, K., and Odijk, J. (1997). Context modeling and the generation of spoken discourse. *Speech Communication*, 21:101–120.

Dekker, P. (1993). *Transsentential Meditations – Ups and Downs in Dynamic semantics*. PhD thesis, Faculty of Philosophy, University of Amsterdam.

Dekker, P. (1997). On first order information exchange. In Jäger, G., and Benz, A., editors, *Proceedings of Mundial'97, Formal Semantics and Pragmatics of Dialogue*, 21–39. University of Munich.

Di Eugenio, B., Jordan, P. W., Thomason, Richard, H., and Moore, J. D. (1998). An empirical investigation of proposals in collaborative dialogues. In *Proceedings of COLING-ACL'98, Montreal, Canada*.

Dirksen, A. (1992). Accenting and deaccenting, a declarative approach. In *Proceedings of COLING'92*, 865–869. Nantes.

Dirksen, A. (1997). Fluent dutch text-to-speech. Technical report, OTS Utrecht / Fluency Speech Products.

Dutoit, T. (1997). High-quality text-to-speech synthesis: An overview. *Electrical and electronics engineering*, 17(1):25–36.

Dybkjaer, L., Bernsen, N., and Dybkjaer, H. (1998). A methodology for diagnostic evaluation of spoken human machine dialogue. *International journal of human-computer studies*, 48(5):605–626.

Dybkjaer, L., Bernsen, N. O., and Dybkjaer, H. (1996). Evaluation of spoken dialogue systems. In Luperfoy, Nijholt, and Veldhuijzen van Zanten, editors, *Dialogue Management in Natural Language Systems*, TWLT11, 219–233. University of Twente, Enschede.

Ebbers, R., Hofs, D., Kroezen, L., van Nieuwenhuizen, P., Zaaijer, A., and van Hessen, A. (1999). StemStem: Een sprekende en luisterende stemcomputer voor visueel gehandicapten. Student Project Report, University of Twente, Enschede.

Engelmore, R., and Morgan, T., editors (1988). *Blackboard systems*. Addison-Wesley, Wokingham.

Fagin, R., and Halpern, J. Y. (1988). Belief, awareness and limited reasoning. *Artificial Intelligence*, 34:39–76.

Fagin, R., Halpern, J. Y., Moses, Y., and Vardi, M. (1995). *Reasoning about Knowledge*. MIT Press, Cambridge, Mass.

Fikes, R., and Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3/4):189–208.

Fowler, M., and Scott, K. (1997). *UML Distilled: Applying the standard object modeling language*. Addison-Wesley, Reading, Mass.

Fraser, N., and Gilbert, G. (1991). Simulating speech systems. *Computer Speech and Language*, 5:81–99.

Gärdenfors, P. (1988). *Knowledge in Flux: Modelling the Dynamics of Epistemic states*. Bradford Books.

Gardent, C., Kohlhase, M., and van Leusen, N. (1996). Corrections and higher-order unification. In *Proceedings of 3. Konferenz zur Verarbeitung natuerlicher Sprache (KONVENS)*. Bielefeld, Germany.

Gazdar, G. (1979). *Pragmatics: Implicature, Presupposition and Logical Form*. New York: Academic Press.

Georgeff, M. P., and Lansky, A. L. (1987). Reactive reasoning and planning. In *Proceedings of AAAI-87*, 677–682.

Gerbrandy, J. (1997). Questions of identity. In Dekker, P., Stokhof, M., and Venema, Y., editors, *Proceedings of the 11th Amsterdam Colloquium*, 133–138. ILLC, University of Amsterdam.

Gerbrandy, J. (1999). *Bisimulations on Planet Kripke*. PhD thesis, ILLC, University of Amsterdam.

Geurts, B. (1994). *Presupposing*. PhD thesis, Universität Osnabrück.

Gibbon, D., Moore, R., and Winski, R., editors (1998). *Handbook of Standards and Resources for Spoken Language Systems*. Mouton de Gruyter, Berlin.

Ginzburg, J. (1991). Questions without answers, wh-phrases without scope. In *Situation Theory and its Applications, II*. CSLI, Stanford, CA.

Ginzburg, J. (1995). Resolving questions, I. *Linguistics and Philosophy*, 18:459–527.

Ginzburg, J. (1996). Interrogatives: Questions, facts and dialogue. In Lappin, S., editor, *The Handbook of Contemporary Semantic Theory*, chapter 15. Basil Blackwell, Oxford.

Ginzburg, J. (1998). Clarifying utterances. In Hulstijn, J., and Nijholt, A., editors, *Formal Semantics and Pragmatics of Dialogue (Twendial'98)*, TWLT13, 11–30. University of Twente, Enschede.

Goffman, E. (1981). *Forms of Talk*. University of Pennsylvania Press, Philadelphia.

Grice, H. P. (1957). Meaning. *Philosophical Review*, 66:377–388.

Grice, H. P. (1975). Logic and conversation. In Cole, P., and Morgan, J. L., editors, *Syntax and Semantics 3*, 41–58. Academic Press, New York.

Groenendijk, J. (1998). Questions in update semantics. In Hulstijn, J., and Nijholt, A., editors, *Formal Semantics and Pragmatics of Dialogue (Twendial'98)*, TWLT13, 125–138. University of Twente, Enschede.

Groenendijk, J. (1999). The logic of interrogation: classical version. In Matthews, T., and Strolovitch, D., editors, *Proceedings of the Ninth Conference on Semantics and Linguistics Theory (SALT-9), Santa Cruz*. CLC Publications. also ILLC research report PP-1999-19, University of Amsterdam.

Groenendijk, J., and Stokhof, M. (1984). *Studies on the Semantics of Questions and the Pragmatics of Answers*. PhD thesis, University of Amsterdam.

Groenendijk, J., and Stokhof, M. (1989). Type-shifting rules and the semantics of questions. In Chierchia, G., Partee, B. H., and Turner, R., editors, *Propertyies, Types and Meaning*, vol. II, 21–68. Kluwer.

Groenendijk, J., and Stokhof, M. (1991). Dynamic predicate logic. *Linguistics and Philosophy*, 14.

Groenendijk, J., and Stokhof, M. (1996). Questions. In van Benthem, J., and Ter Meulen, A., editors, *Handbook of Logic and Language*. North-Holland, Elsevier.

Groenendijk, J., Stokhof, M., and Veltman, F. (1996). Coreference and modality. In Lappin, S., editor, *Handbook of Contemporary Semantic Theory*, 179–213. Basil Blackwell, Oxford.

Grosz, B. J. (1977). *The Representation and Use of Focus in Dialogue Understanding*. PhD thesis, SRI International.

Grosz, B. J., Joshi, A., and Weinstein, S. (1995). Centering: A framework for modeling the local; coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Grosz, B. J., and Kraus, S. (1996). Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357.

Grosz, B. J., and Sidner, C. L. (1986). Attentions, intentions and the structure of discourse. *Computational Linguistics*, 12:175–204.

Grosz, B. J., and Sidner, C. L. (1990). Plans for discourse. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, 417–444. MIT Press, Cambridge, Mass.

Grover, C., Brew, C., Manandhar, S., Moens, M., and Schóter, A. (1994). Priority union and generalisation in discourse grammar. In Dorrepaal, J., editor, *Towards a declarative Theory of Discourse, LRE-61-062*, 57–104.

Hamblin, C. L. (1970). *Fallacies*. Methuen & Co, London.

Harel, D. (1984). Dynamic logic. In Gabbay, D., and Guenthner, F., editors, *Handbook of Philosophical Logic*, 497–604. Reidel, Dordrecht.

Heeman, P., and Hirst, G. (1995). Collaborating on referring expressions. *Compuational Linguistics*, 21(3):351–382.

Heim, I. (1982). *The Semantics of Definite and Indefinite Nounphrases*. PhD thesis, University of Massachusetts, Amherst.

Heim, I. (1983). On the projection problem for presuppositions. In Barlow, M., editor, *Proceedings of the Second West Coast Conference on Formal Linguistics, 2*, 137–178. Stanford University, Stanford, CA.

Heim, I. (1992). Presupposition projection and the semantics of attitude verbs. *Journal of Semantics*, 9(3):183–222.

Hendriks, H., and Dekker, P. (1996). Links without locations. In P, D., and Stokhof, M., editors, *Proceedings of 10th Amsterdam Colloquium*, 339–358.

Hinchey, M. G., and Jarvis, S. A. (1995). *Concurrent Systems: Formal Development in CSP*. McGraw-Hill, London.

Hoare, C. A. R. (1985). *Communicating Sequential Processes*. Prentice Hall, New York.

Hobbs, J. R. (1978). Resolving pronoun references. *Lingua*, 44:311–338.

Hobbs, J. R. (1979). Coherence and coreference. *Cognitive Science*, 3:67–90.

Hobbs, J. R. (1990). Artificial intelligence and collective intentionality: Comments on searle and grosz and sidner. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, 445–459. MIT Press, Cambridge, Mass.

van der Hoeven, G. F., Andernach, J., van de Burgt, S. P., Kruijff, G.-J. M., Nijholt, A., Schaake, J., and de Jong, F. (1995). SCHISMA: A natural language accessible theatre information and booking system. In *First International Workshop on Applications of Natural Language to Data Bases (NLDB'95)*, 271–285. Versailles, France.

Houghton, G., and Isard, D. (1987). Why to speak, what to say and how to say it. In Morris, P., editor, *Modelling Cognition*, 249–267. Wiley.

Huang, Z. (1994). *Logics for Agents with Bounded Rationality*. PhD thesis, ILLC, University of Amsterdam.

Huang, Z., and Masuch, M. (1995). The logic of permission and obligation in the framework of alx. 3: How to avoid the paradoxes of deontic logics. *Logique et analyse*, 38(149):55–74.

Huang, Z., Masuch, M., and Pólos, L. (1996). ALX, an action logic for agents with bounded rationality. *Artificial Intelligence*, 82:75–127.

Huls, C., Bos, E., and Claassen, W. (1995). Automatic referent resolution of deictic and anaphoric expressions. *Computational linguistics*, 21(1):59–80.

Hulstijn, J. (1995). Presupposition in update semantics. Master's thesis, Faculty of Mathematics and Computer Science, University of Amsterdam.

Hulstijn, J. (1996). Presupposition accommodation in a constructive update semantics. In Durieux, G., Daelemans, W., and Gillis, S., editors, *Papers from the Sixth CLIN Meeting*, 129–146. University of Antwerp, Antwerp.

Hulstijn, J. (1997). Structured information states: Raising and resolving issues. In Jäger, G., and Benz, A., editors, *Proceedings of Mundial'97, Formal Semantics and Pragmatics of Dialogue*, 99–117. University of Munich. also available as CTIT Technical Report 97-18, University of Twente, Enschede.

Hulstijn, J. (1998). Goals and information states. In Kraak, E., and Wasserman, R., editors, *Proceedings of Accolade'97*, 32–49. Dutch Graduate School in Logic (Ozsl) University of Amsterdam.

Hulstijn, J., and Nijholt, A., editors (1999). *Automatic Interpretation and Generation of Verbal Humor*, TWLT12. University of Twente, Enschede.

Hulstijn, J., Steetskamp, R., ter Doest, H., van de Burgt, S., and Nijholt, A. (1996). Topics in SCHISMA dialogues. In Luperfoy, S., Nijholt, A., and Veldhuijzen van Zanten, G., editors, *Dialogue Management in Natural Language Systems*, TWLT11, 89–100. University of Twente, Enschede.

Hulstijn, J., and van Hessen, A. (1998). Utterance generation for transaction dialogues. In Manell, R. H., and Robert-Ribes, J., editors, *Proceedings of the Fifth International Conference on Spoken Language Processing (ICSLP'98)*, paper number 776.

ISO 9126 (1991). Information technology - software product evaluation - quality characteristics and guidelines for their use. Technical report, International Organization for Standardization.

Jacobson, L., Christerson, M., Jonsson, P., and Vergaard, G. (1992). *Object-oriented Software Engineering, a Use Case Driven Approach*. Addison Wesley, Wokingham.

Jäger, G. (1996). Only updates - on the semantics of the focus particle *only*. In Dekker, P., and Stokhof, M., editors, *Proceedings of the 10th Amsterdam Colloquium*, 387–406. ILLC, University of Amsterdam.

Jennings, N. R., Norman, T., Faratin, P., O' Brien, P., and Odgers, B. (to appear). Autonomous agents for business processes. *Journal of Applied Artificial Intelligence*.

Jokinen, K. (1996). Cooperative response planning in cdm: Reasoning about communicative strategies. In Luperfoy, S., Nijholt, A., and Veldhuijzen van Zanten, G., editors, *Dialogue Management in Natural Language Dialoguye Systems*, 159–168.

Jönsson, A. (1993). *Dialogue Management for Natural Language Interfaces*. PhD thesis, Linköping University.

Jönsson, A. (1997). A model for habitable and efficient dialogue management in natural language interaction. *Natural Language Engineering*, 3(2/3):103–122.

Kamp, H., and Reyle, U. (1993). *From Discourse to Logic*, vol. 42 of *Studies in Linguistics and Philosophy*. Kluwer, Dordrecht.

Kaplan, S. J. (1982). Cooperative responses from a portable naural language system. *Artificial Intelligence*, 19:165–187.

Karttunen, L. (1973). Presuppositions of compound sentences. *Linguistic Inquiry*, IV(2):169–193.

Karttunen, L. (1974). Presupposition and linguistic context. *Theoretical Linguistics*, I:181–194.

Kay, M., Gawron, J. M., and Norvig, P. (1994). *Verbmobil : a translation system for face-to-face dialog*. No. 33 in CSLI lecture notes. CSLI, Stanford, CA.

Kellner, A., Rüber, B., and Seide, F. (1996). A voice-controlled automatic switchboard and directory information system. In *Proceedings of the IEEE Third Workshop on Interactive Voice Technology for Telecommunications Applications*, 117–120. Basking Ridge (NJ).

Klumper, J. (to appear). Dialoogfunctionaliteit voor de navigatieagent van het VMC. Master's thesis, University of Twente, Enschede.

Kowtko, J. C., Isard, S. D., and Doherty, G. M. (1992). Conversational games within dialogue. Technical Report HCRC/RP-31, Human Communication Research Centre, University of Edinburgh.

Krahmer, E. (1995). *Discourse and Presupposition*. PhD thesis, Katholieke Universiteit Brabant, Tilburg.

Krahmer, E., Swerts, M., Theune, M., and Weegels, M. (1999). Problem spotting in human-machine interaction. In Prószéky, G., Németh, G., and Mándli, J., editors, *Proceedings Eurospeech'99, Budapest*, 1423–1426.

van Kuppevelt, J. (1995). Discourse structure, topicality and questioning. *Journal of Linguistics*, 31:109–149.

Lascarides, A., and Asher, N. (1999). Cognitive states, discourse structure and the content of dialogue. In van Kuppevelt, J., van Leusen, N., van Rooy, R., and Zeevat, H., editors, *Proceedings of Workshop on Semantics and Pragmatics of Dialogue (Amstellogue'99)*. University of Amsterdam, Amsterdam.

van Leusen, N. (1997). The role of inference in the resolution of correction. Technical Report 93, CLAUS, DFKI, Saarbrücken.

Levelt, W. J. (1989). *Speaking: from Intention to Articulation*. MIT Press, Cambridge, Mass.

Levelt, W. J. M., and Kelter, S. (1982). Surface form and memory in question answwering. *Cognitive Psychology*, 14:78–106.

Levesque, H. L. (1984). A logic of implicit and explicit belief. In *Proceedings of AAAI-84, Austin, Texas*, 198–202.

Levin, J. A., and Moore, J. A. (1978). Dialogue-games: Metacommunication structures for natural language interaction. *Cognitive Science*, 1(4):395–420.

Lewis, D. (1969). *Convention: A Philosophical Study.* Harvard University Press, Cambridge.

Lewis, D. (1973). *Counterfactuals*. Basil Blackwell, Oxford.

Lewis, D. (1979). Scorekeeping in a language game. *Journal of Philosophical Logic*, 9:339–359.

Lie, D., Hulstijn, J., ter Doest, H., and Nijholt, A. (1997). Language analysis in SCHISMA. In *International Workshop on Parsing Technologies 1997 (IWPT '97)*, 243–244. MIT, Boston, USA.

Litman, D. J., , and Pan, S. (1999). Empirically evaluating an adaptable spoken dialogue system. In *Proceedings of the 7th International Conference on User Modeling (UM'99)*, 55–64.

Litman, D. L., and Allen, J. F. (1987). A plan recognition model for subdialogues in conversation. *Cognitive Science*, 11:163–200.

Mann, W. C. (1988). Dialogue games: Conventions of human interaction. *Argumentation*, 2:511–532.

Mann, W. C., and Thompson, S. A. (1988). Rhetorical structure theory: Towards a functional theory of text organization. *Text*, 8(3):243–281.

Mercer, R. E. (1992). Default logic: Towards a common logical semantics for presuppositions and entailments. *Journal of Semantics*, 9(3):223–250.

Meyer, J.-J. C., and Wieringa, R. J., editors (1993). *Deontic Logic in Computer Science: Normative System Specification*. Wiley and Sons, Chichester.

Milner, R. (1980). *A Calculus of Communicating Systems*. Lecture Notes in Computer Science. Springer-Verlag, Berlin.

Moore, J. D., and Paris, C. L. (1993). Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–694.

Moser, M., and Moore, J. D. (1996). Toward a synthesis of two accounts of discourse structure. *Computational Linguistics*, 22(3):409–420.

Müller, J. P. (1997). Control architectures for autonomous and interacting agents: A survey. In Cavedon, L., Rao, A., and Wobeck, W., editors, *Intelligent Agent Systems: Theoretical and Practical Issues*, vol. 1209 of *Lecture Notes in Computer Science*, 1–26. Springer-Verlag, Berlin.

von Neumann, J., and Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press, New Jersey.

Nielsen, J. (1993). *Usability engineering*. Academic Press, Boston.

Nijholt, A. (1999). The Twente virtual theatre environment: Agents and interactions. In Nijholt, A., Donk, O., and van Dijk, B., editors, *Interactions in Virtual Worlds*, TWLT15, 147–164. University of Twente, Enschede.

Nijholt, A., and Hulstijn, J. (to appear). Multi-modal interactions with agents in virtual worlds. In Kasabov, N., editor, *Future Directions for Intelligent Systems and Information Science*, Studies in Fuziness and Soft Computing. Physica-Verlag.

Nijholt, A., Hulstijn, J., and van Hessen, A. (1999). Speech and language interactions in a web theatre environment. In Dalsgaard, P., Lee, C.-H., Heisterkamp, P., and Cole, R., editors, *Proceedings ESCA tutorial on Interactive Dialogue in Multi-Modal Systems*, 129–132. Kloster Irsee, Germany.

van Noord, G. (1997). An efficient implementation of the head-corner parser. *Computational Linguistics*, 23(3):425–456.

Oviatt, S. (1995). Predicting spoken disfluencies during human-computer interaction. *Computer Speech and Language*, 9(1):19–36.

Oviatt, S. L., and Cohen, P. R. (1990). *Discourse structure and performance efficiency in interactive and noninteractive spoken modalities*. Menlo Park, CSLI/SRI.

Parke, F. I., and Waters, K. (1996). *Computer Facial Animation*. A.K. Peters Ltd, Natick, Mass.

Peirce, C. S. (1867). Upon logical comprehension and extension. In Fisch, M. H., et al., editors, *Writings of Charles S. Peirce : a chronological edition*, vol. I, 1857-1866, 70–86. Indiana University Press, Bloomington. 1982 edition.

Pinkal, M. (1995). *Logic and Lexicon: the semantics of the indefinite*. Kluwer, Dordrecht. translated from German, *Logik und Lexikon* (1985), De Gruyter, Berlin.

Poesio, M., and Traum, D. (1998). Towards an axiomatisation of dialogue acts. In Hulstijn, J., and Nijholt, A., editors, *Formal Semantics and Pragmatics of Dialogue (Twendial'98)*, TWLT13, 207–221. University of Twente, Enschede.

Polanyi, L. (1988). A formal model of the structure of discourse. *Journal of Pragmatics*, 12:601–638.

Polanyi, L., and Scha, R. (1984). A syntactic approach to discourse semantics. In *Proceedings of the 10th International Conference on Computational Linguistics (COLING10)*, 413–419. Stanford, CA.

Pollack, M. E. (1990). Plans as complex mental attitudes. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, 77–103. MIT Press, Cambridge, Mass.

Pressman, R. S. (1997). *Software Engineering, a practitioner's approach*. McGraw-Hill, european adaptation.

Prüst, H., Scha, R., and van den Berg, M. H. (1994). Discourse grammar and verb phrase anaphora. *Linguistics and Philosophy*, 17:261–327.

Rats, M. (1996). *Topic Management in Information Dialogues*. PhD thesis, Katholieke Universiteit Brabant, Tilburg.

Reiter, E., and Dale, R. (1997). Building applied natural language generation systems. *Journal of Natural Language Engineering*, 3:57–87.

de Rijke, M. (1993). *Extending Modal Logic*. PhD thesis, ILLC, University of Amsterdam.

Roberts, C. (1989). Modal subordination and pronominal anaphora in discourse. *Linguistics and Philosophy*, 12:683–721.

de Roeck, A., Ball, R., Brown, K., Fox, C., Groefsema, M., Obeid, N., and Turner, R. (1991). Helpful answers to modal and hypothetical questions. In *Proceedings of the EACL*, 257–262.

Rooth, M. (1992). A theory of focus interpretation. *Natural Language Semantics*, 1:75–116.

Rosenschein, J. S., and Zlotkin, G. (1994). Designing conventions for automated negotiation. *AI Magazine*, 15(3):29–46.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1991). *Object-oriented modeling and design*. Prentice Hall, New York.

Russell, S. J., and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall, New York.

Sacks, H., Schegloff, E., and Jefferson, G. (1974). A simplest systematics for the organisation of turn-taking for conversation. *Language*, 50:696–735.

van der Sandt, R. (1989). Presupposition and discourse structure. In *Semantics and Contextual Expression*, 26–94. Dordrecht: Foris.

van der Sandt, R. (1992). Presupposition projection as anaphora resolution. *Journal of Semantics*, 9:333–377.

Scha, R. J. H. (1983). *Logical Foundations for Question Answering*. PhD thesis, Philips Research Laboratories.

Schank, R. C., and Abelson, R. P. (1977). *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*. Erlbaum, New York.

Schegloff, E. A., Jefferson, G., and Sacks, H. (1977). The preference for self-correction in the organization of repair in conversation. *Language*, 53(2).

Schelling, T. (1960). *The Strategy of Conflict*. Harvard University Press, Cambridge, Mass.

van Schooten, B. (1999). Building a framework for designing interaction models: Overview of current research on interaction models in dialogue systems. Technical Report 99-04, CTIT, University of Twente, Enschede.

van Schooten, B., Donk, O., and Zwiers, J. (1999). Modelling interaction in virtual environments using process algebra. In Donk, O., and van Dijk, B., editors, *Interactions in Virtual Worlds*, TWLT15, 195–212. University of Twente, Enschede.

Schöter, A. (1996). Evidential bilattice logic and lexical inference. *Journal of Logic, Language and Information*, 5(1):65–105.

Searle, J. R. (1969). *Speech acts: an Essay in the Philosophy of Language*. Cambridge University Press, Cambridge.

Searle, J. R. (1975). Indirect speech acts. In Cole, P., and Morgan, J., editors, *Speech Acts*, Syntax and Semantics 3. Academic Press, New York.

Searle, J. R. (1983). *Intentionality: an Essay in the Philosophy of Mind*. Cambridge University Press, Cambridge.

Searle, J. R. (1990). Collective intentions and actions. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, 401–416. MIT Press, Cambridge, Mass.

Sengers, P. (1999). Designing comprehensible agents. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, 1227–1232. Stockholm.

Sgall, P., Hajiucová, and Panevová, J. (1986). *The Meaning of The Sentence in Its Semantic and Pragmatic Aspects*. Dordrecht: Reidel.

Shieber, S. M., Pereira, F. C., and Dalrymple, M. (1996). Interactions of scope and ellipsis. *Linguistics and philosophy*, 19(2):527–552.

Simon, H. A. (1955). A behavioral model of rational choice. *Qualterly Journal of Economics*, 69:99–118.

Singh, M. P. (1994). *Multiagent Systems: A Theoretical Framework for Intentions, Know-How, and Communications*, vol. 799 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin.

Smith, R. W., and Hipp, D. (1994). *Spoken Language Dialog Systems: A Practical Approach*. Oxford University Press, Oxford.

Sperber, D., and Wilson, D. (1986). *Relevance, Communication and Cognition*. Basil Blackwell, Oxford.

Stalnaker, R. (1979). Assertion. In Cole, P., editor, *Pragmatics*, Syntax and Semantics 9, 315–332. Academic Press, New York.

Stalnaker, R. C. (1984). *Inquiry*. MIT Press.

Stone, P., and Veloso, M. (to appear). Task decomposition, dynamic role assignment and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence*. to appear.

Sturm, J., Den Os, E., and Boves, L. (1999). Issues in spoken dialogue systems: Experiences with the dutch ARISE system. In *Proceedings ESCA Workshop on Interactive dialogue in multi-modal systems, Kloster Irsee, Germany*, 1–4.

Tanenbaum, A. S. (1995). *Distributed Operating Systems*. Prentice Hall, New York.

Tekinerdogan, B. (to appear). *Towards Automated Design of Adaptable Software Architectures.* PhD thesis, University of Twente.

Theune, M. (1997). Contrastive accent in a data-to-speech system. In *Proceedings of the 35th Annual meeting of the ACL/EACL, Madrid*, 519–521.

van der Torre, L. W. N., and Tan, Y. H. (1997). Obligations in update semantics. In *Proceedings of the Ninth Dutch Conference on Artificial Intelligence (NAIC'97)*, 249–258. University of Antwerp.

van der Torre, L. W. N., and Tan, Y. H. (1999). Rights, duties and commitments between agents. In Dean, T., editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, 1239–1244.

Traum, D. (1997). A reactive-deliberative model of dialogue agency. In Müller, J. P., Wooldridge, M. J., and Jennings, N. R., editors, *Intelligent Agents III: ECAI'96 Workshop on Agent Theories, Architectures and Languages (ATAL)*, vol. 1193 of *Lecture Notes in Computer Science*, 157–172. Springer-Verlag, Berlin.

Traum, D. R. (1994). *A Computational Theory of Grounding in Natural Language Conversation*. PhD thesis, University of Rochester, New York.

Traum, D. R., and Hinkelman, E. A. (1992). Conversation acts in task-oriented spoken dialogue. *Computational Intelligence*, 8(2):575–599.

Ullman, J. D. (1988). *Principles of database and knowledge-base systems*, vol. I: Classical database systems. Computer Science Press. 3rd edition, 1st edition 1982.

Vallduví, E. (1990). *The Informational Component*. PhD thesis, University of Pennsylvania.

Vallduví, E. (1994). The dynamics of information packaging. In Engdahl, E., editor, *Integrating Information Structure into Constraint Based and Categorial Approaches*, DYANA-2 Deliverable R1.3.B. ILLC, University of Amsterdam.

Vallduvi, E., and Engdahl, E. (1996). The linguistic realization of information packaging. *Linguistics*, 34(3):459–520.

Vanderveken, D. (1990). *Meaning and speech acts*. Cambridge University Press, Cambridge.

Veldhuijzen van Zanten, G. (1996). Pragmatic interpretation and dialogue management in spoken-language systems. In Luperfoy, S., Nijholt, A., and Veldhuijzen van Zanten, G., editors, *Dialogue Management in Natural Language Systems*, TWLT11, 81–88. University of Twente, Enschede.

Veldhuijzen van Zanten, G. (1998). Adaptive mixed-initiative dialogue management. In *4th IEEE International Workshop on Interactive Voice Technology for Telecommunications Applications (IVTTA'98)*, 65–70. Torino, Italy,. also as Technical Report 1488, IPO, Eindhoven.

Veltman, F. (1996). Defaults in update semantics. *Journal of philosophical logic*, 25(3):221–262. previous version in DYANA R2.5A, 1990.

Vertegaal, R. (1998). *Look who's talking to whom: mediating joint attention in multiparty communication and collaboration*. PhD thesis, University of Twente, Enschede.

Walker, M. A. (1996a). The effect of resource limits and task complexity on collaborative planning in dialogue. *Artificial Intelligence*, 85:181–243.

Walker, M. A. (1996b). Inferring acceptance and rejection in dialog by default rules of inference. *Language and Speech*, 39(2-3):265–304.

Walker, M. A. (1996c). Limited attention and discourse structure. *Computational Linguistics*, 22(2):255–263.

Walker, M. A., Litman, D. J., Kamm, C. A., and Abella, A. (1997). PARADISE: A framework for evaluating spoken dialogue agents. In *Proceedings of the 35th Annual meeting of the ACL/EACL*, 271–280. Madrid.

Wasserman, R. (1999). Resource bounded belief revision. *Erkenntnis*, 50(2-3):429–446.

Wasserman, R., and Hansson, S. O. (1999). Local change. Technical Report Report PP-1999-17, ILLC, University of Amsterdam.

Webster Inc., . (1983). *Webster's Ninth New Collegiate Dictionary*. Merriam-Webster Inc., Springfield USA.

Weegels, M. (1999). Usability evaluation of voice-operated information servives: A comparative study of VIOS and ARISE. Technical report, Dutch Priority Programme

Language and Speech Technology, IPO, Eindhoven.

Westerstrahl, D. (1985). Determiners and context sets. In van Benthem, J., and ter Meulen, A., editors, *Generalized Quantifiers in Natural Language*, 45–71. Foris Publications, Dordrecht.

Wittgenstein, L. (1953). *Philosophical Investigations*. Basil Blackwell, Oxford. 2nd edition (1972), translated from German by C. E. M. Anscombe.

Wooldridge, M. J., and Jennings, N. R. (1999). The cooperative problem-solving process. *Journal of Logic and Computation*, 9(4):563–592.

von Wright, G. H. (1963). *The Logic of Preference: an Essay*. Edinburgh University Press.

Zeevat, H. (1992). Presupposition and accomodation in update semantics. *Journal of Semantics*, 9:379–412.

Zeevat, H. (1997). How to dialogue? the proper role of common grounds. In Jäger, G., and Benz, A., editors, *Proceedings of Mundial'97, Formal Semantics and Pragmatics of Dialogue*, 195–214. University of Munich.

Samenvatting

Dit proefschrift heet *Dialogue models for Inquiry and Transaction*: Dialoogmodellen voor inlichtingen en transacties. Het beschrijft een aantal manieren waarop een dialoog kan worden gemodelleerd. De bedoeling is dat de eigenschappen die worden beschreven met deze modellen kunnen worden gebruikt bij het ontwerpen en testen van dialoogsystemen.

Een dialoogsysteem is een computer systeem dat in natuurlijke taal, dus bijvoorbeeld het Engels of Nederlands, communiceert met de gebruiker van het systeem over een bepaald onderwerp. Dat gaat via een toetsenbord of over de telefoon. In dat geval gaat het om een gesproken dialoogsysteem. In een enkel geval combineert het systeem verschillende 'modaliteiten' of vormen van invoer en uitvoer, bijvoorbeeld spraak met plaatjes of gebaren. In Nederland is openbaar vervoer reisinformatie de belangrijkste toepassing voor dialoogsystemen, maar er zijn ook systemen voor het reserveren van kaartjes, voor het plaatsen van orders op de beurs en voor het aanleren van een vreemde taal. Voor elk toepassingsdomein gelden andere sociale en taalkundige conventies. Ook de vooronderstelde achtergrondkennis verschilt. Om die reden moet je je in de informatica meestal beperken tot een bepaald toepassingsdomein. Het gaat in dit proefschrift vooral om dialogen voor het uitwisselen van informatie en het inwinnen van inlichtingen (inquiry) en voor het sluiten van overeenkomsten (transaction).

Wij hebben in Twente gekeken naar dialogen over schouwburgvoorstellingen en het reserveren van kaartjes. Het systeem dat we hebben gebouwd heet SCHISMA. Aan de hand van een aantal reserveringsscenario's, is er een *corpus* verzameld: een lijst van gesprekken van proefpersonen met een gesimuleerd systeem. Eerst wordt er vaak informatie uitgewisseld over de voorstellingen die er zijn. Het systeem geeft een overzicht of geeft informatie over een bepaalde voorstelling, zoals de prijs of de aanvangstijd. Een voorstelling kan worden geselecteerd in het programma van de schouwburg door het noemen van een titel, van de artiest of groepsnaam, van de datum of van het genre. Soms is een gebruiker tevreden met informatie alleen. Vaak echter zal de gebruiker kaartjes willen reserveren. Daarvoor moet ook weer informatie worden uitgewisseld over de belangrijkste eigenschappen van een reservering: de voorstelling, het aantal kaartjes en kaartjes met korting, de naam en het adres van de gebruiker, de totale prijs en een bevestiging van de reservering. Als er informatie ontbreekt, moet het systeem erom vragen.

De taakanalyse voor het reserveren van kaartjes is weergegeven in figuur 7.1. In de gespreksfase die hieraan voorafgaat is al een voorstelling geselecteerd; hierna volgt een afsluiting van de dialoog. Een dergelijke taakanalyse is een eerste stap in het ontwerp van een dialoogsysteem. Daarnaast moet er een model gemaakt worden van alle begrippen in het schouwburgdomein. Ieder begrip wordt gemodelleerd als een object met bepaalde eigenschappen of relaties tot andere objecten. Een dergelijk begripsmodel helpt bij het vinden van de dialoogstructuur. De objecten in het model en hun eigenschappen kunnen onderwerp van gesprek zijn (topic). Wanneer het huidige gespreksonderwerp de prijs is, is het begrip korting ook relevant. Een niet gerelateerd onderwerp, zoals een componist, is dan onwaarschijnlijk. Zo'n model helpt bij het begrijpen van verwijzingen en van weggelaten zinsdelen. Een uiting als "3 voor vanavond alstublieft" is alleen maar te begrijpen als je weet dat de achterliggende taak een reservering is, en dat daardoor het aantal kaartjes en de voorstelling waarvoor moet worden gereserveerd, relevante gespreksonderwerpen zijn.

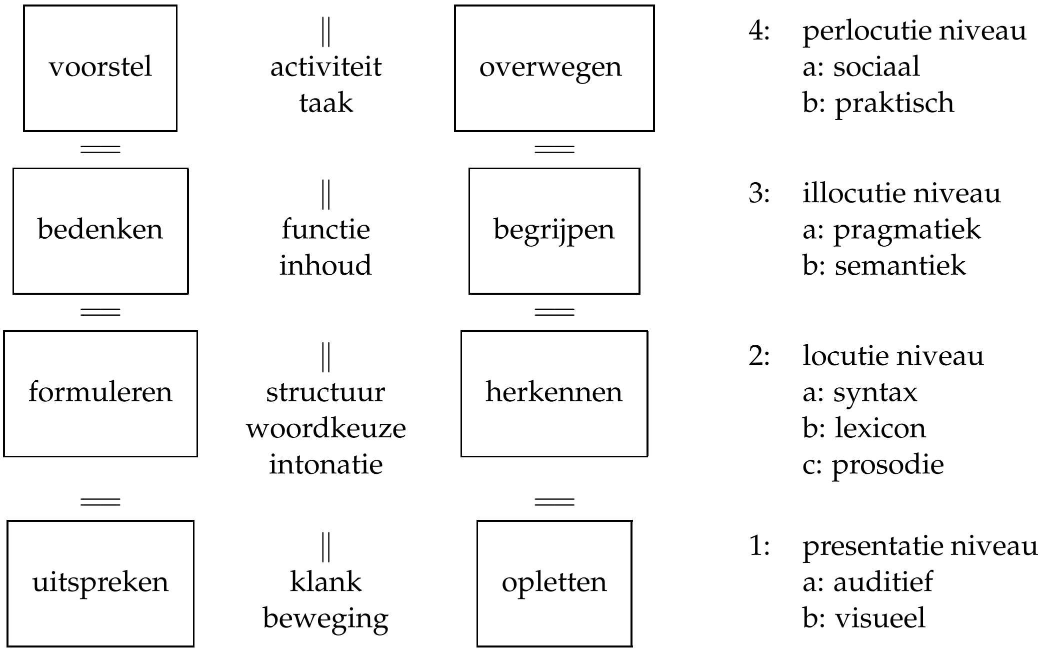
Reservering	
toepasbaarheids-voorwaarden	verzoek om reservering aangegeven door de gebruiker
geslaagdheids-voorwaarden	unieke voorstelling is geselecteerd aantal kaartjes is bekend aantal kaartjes met korting is bekend de naam en het adres van de gebruiker zijn bekend de plaatsen zijn beschikbaar de totale kosten zijn bekend bij de gebruiker overeenkomst is bevestigd door de gebruiker
bedoeld effect	gebruiker en systeem zijn het eens over de voorstelling, de kaartjes en de kosten plaatsen zijn gereserveerd voor de gebruiker de gebruiker is verplicht de kaartjes op te halen en te betalen
voorwaarden bij falen	er zijn geen plaatsen gereserveerd, de gebruiker is geen verplichtingen aangegaan
acties	stel de vragen die horen bij de bovenstaande voorwaarden, pas de reserveringsdatabase aan

Figuur 7.1: Reserveringen in het SCHISMA systeem

De domein- en taakanalyse moet worden omgezet in een ontwerp, en uiteindelijk in een implementatie van het systeem. Zo moet er een woordenlijst komen met alle verwachte woorden en hun betekenis. Grammatica regels beschrijven welke zinspatronen kunnen voorkomen en met welke betekenis. Er zijn interactieregels die bepalen hoe het systeem moet reageren op de gebruiker. Daarnaast is er een component die de gesprekscontext en de plannen van het systeem bijhoudt. De regels en principes die samen het gedrag van een systeem bepalen vormen het ontwerp.

Bruikbaarheid

Wat zijn de voordelen van natuurlijke taal ten opzichte van commando's of menu's? Taal is natuurlijk. Je hoeft geen speciale commando's of menustructuren te leren. Door het toepassen van spraak, wordt de toegankelijkheid van een systeem vergroot. Je hebt geen toetsenbord of scherm meer nodig. Dat is handig als je handen wilt vrijhouden in de auto, of als je blind bent. In hoofdstuk 6 beschrijven we een dialoogsysteem genaamd STEMSTEM, waarmee blinden kunnen stemmen voor de verkiezingen. Een belangrijk voordeel is de flexibiliteit van gesprekken; je bent zelf de baas. Tijdens een gesprek wordt een context opgebouwd. Je hoeft in principe niets te herhalen. En als het mis gaat, kun je de ander verbeteren of vragen om verduidelijking. Ondanks deze voordelen, zijn natuurlijke taalsystemen nauwelijks toegepast. Er zijn blijkbaar ook nadelen. Uitdrukkingen in natuurlijke taal zijn vaak ambigu, voor velerlei uitleg vatbaar. Voor een systeem is dat lastig, omdat het de achtergrondkennis mist om die ambiguïteit op te lossen. De flexibiliteit van een gesprek is een nadeel, als je niet weet wat er van je verwacht wordt, of wat het systeem kan. Dat lijdt dan tot vervelende misverstanden. Bovendien is het bouwen van een dialoogsysteem relatief duur. Je hebt er speciale taalkundige gegevens voor nodig, zoals een woordenboek, een grammatica en interactieregels. Daarvoor heb je experts nodig. Bovendien moet je relatief veel testen met proefpersonen, en dat kost tijd en geld.



Figuur 7.2: Coördinatie *op*, *tussen* en *binnen* verschillende communicatieniveaus

In hoofdstuk 1 raden we een zogenaamde iteratieve en op gebruikers gerichte ontwerpmethode aan. Het oorspronkelijke prototype moet voortdurend worden geëvalueerd en getest aan de hand van gegevens van gebruikers. Om te ontwerpen en testen, moet je de eigenschappen van een ontwerp kunnen relateren aan toetsbare eigenschappen van de uiteindelijke dialoog. Vanwege de verwachte voordelen van het gebruik van natuurlijke taal, zien we *bruikbaarheid* als de belangrijkste kwaliteitsfactor voor dialoogsystemen. We zijn dus opzoek naar eigenschappen die samenhangen met bruikbaarheid. In het algemeen is de bruikbaarheid van een systeem op te vatten als een combinatie van effectiviteit en efficiëntie, leerbaarheid, begrijpelijkheid en bedieningsgemak, met daarnaast de tevredenheid van gebruikers. Voor dialoogsystemen hebben die eigenschappen vooral te maken met de taak, waar effectiviteit en efficiëntie van afhangen, en met de samenhang of coherentie van de uitingen in de uiteindelijke een dialoog. De samenhang kan worden bevorderd door een dialoogontwerp dat transparant is. Een systeem is transparant als uit het gedrag van het systeem blijkt wat het aan het doen is, en waarom.

Coördinatie en coherentie

Een dialoog is een samenhangend geheel van uitingen. De samenhang van een uiting met de dialoogcontext ligt aan de vorm, de inhoud en de functie van de uiting ten opzichte van de rest van de dialoog. Hoe kunnen we al die aspecten combineren? Hoofdstuk 2 beschrijft een aantal modellen van *coördinatie*. Er is afstemming tussen de spreker en de hoorder op allerlei taalkundige niveaus (figuur 7.2).

Op het onderste niveau, het *presentatie niveau*, wordt de uiting gepresenteerd door de spreker; de hoorder moet actief blijven opletten. Spreker en hoorder houden contact met non-verbale terugkoppelingsmechanismen, zoals knikken en elkaar aankijken. Dit kan worden gemodelleerd als een reeks gezamenlijke uitspreek-oplet acties. Iets dergelijks geldt ook voor gebaren.

Op het niveau daarboven, het *locutie niveau*, gaat het om coördinatie met betrekking tot de formulering, de zinsstructuur en de intonatie van een uiting. Dergelijke vormaspecten maken duidelijk wat de inhoud is van de uiting en de communicatieve functie. Een vraagzin heeft bijvoorbeeld een andere woordvolgorde dan een bewerende zin. Opeenvolgende uitingen hebben vaak een parallelle structuur. Samen met de intonatie maakt deze vaak duidelijk waar in de zin de belangrijkste informatie staat. Kijk maar naar voorbeeld (148). Uit de structuur van de zin en de intonatie blijkt dat de kleur van de gewenste fiets, rood, moet worden verbeterd. De kleur is de *focus* van deze uiting. De rest van de uiting is de *ground*.

(148) Nee, ik wil geen *rode* fiets, ik wil een *blauwe*!

Op het *illocutie niveau* vinden we een gezamenlijke actie van bedenken en interpreteren. Iedere uiting kan worden gemodelleerd als een *dialooghandeling*. Deze is bepaald door de *betekenisinhoud*: de informatie die wordt weergegeven, en een *communicatieve functie*. Met een uiting doe je iets. Een uiting heeft een bepaalde functie. Een deel van de functie heeft te maken met het vervullen van de taak, een niveau hoger. Je moet bijvoorbeeld het aantal kaartjes te weten komen. Dat kan door het nemen van een *initiatief* in die richting: je stelt een vraag. Een ander deel van de functie van een uiting hangt af van de interactie. Als er net een initiatief geweest is, moet je vervolgen met een passende *respons*. In dit geval is dat een antwoord, of uitleggen dat je het niet weet.

Op het *perlocutie niveau* stemmen de deelnemers aan het gesprek hun acties op elkaar af, op zo'n manier dat die acties bijdragen aan een gezamenlijk doel. Het doel hangt onder andere af van de sociale activiteit. In dit geval is dat het kopen en verkopen van kaartjes. Dat is een voorbeeld van een transactie. Het type activiteit wordt bepaald door de rol van de deelnemers, van hun persoonlijke en gezamenlijke doelen, van de fysieke en sociale omgeving en van een aantal fases waaruit de voortgang van de activiteit moet blijken.

Inlichtingen

In hoofdstuk 3 gaat het om een geïdealiseerd model van informatie uitwisseling en het inwinnen van inlichtingen (inquiry). We combineren update semantiek (Veltman 1996) met een betekenismodel voor vragen en antwoorden (Groenendijk and Stokhof 1996). De informatie die al is uitgewisseld wordt voorgesteld door een *informatietoestand*: een verzameling van alle situaties die overeenkomen met de gegevens. We noemen dat mogelijke werelden. De gegevens zijn gestructureerd aan de hand van *kwesties*. Dat zijn de vragen die op dat moment van belang zijn. 'Kwestie' is een abstract begrip; het modelleert de betekenisinhoud van een vraag, net zoals een propositie de betekenisinhoud weergeeft van een bewerende zin. In het model worden kwesties voorgesteld door een equivalentierelatie. De werelden die hetzelfde zijn wat betreft hun antwoord op de vragen, zijn equivalent. Antwoorden komen overeen met equivalentieklassen. Update semantiek is dynamisch. De betekenis van een uiting is de verandering die de uiting teweegbrengt in de gesprekscontext. Bewerende uitingen voegen gegevens toe: ze strepen alle mogelijkheden weg die niet kloppen met de nieuwe gegevens. Vragende uitingen, of uitingen die op een andere manier een kwestie oproepen, voegen structuur toe. Dat gebeurt door paren van werelden die niet hetzelfde antwoord geven op de nieuwe vraag, weg te strepen uit de equivalentierelatie. Het onderscheid dat kan worden gemaakt op grond van kwesties, wordt daardoor preciezer.

In dit model kunnen we een viertal eigenschappen vastleggen, die bepalen wanneer de inhoud van een uiting coherent is met de informatie in de context. Een uiting is *consistent* met de context, wanneer het resultaat van de update niet de lege verzameling oplevert. Ze is *informatief* wanneer er daadwerkelijk (paren van) werelden worden weggestreept. Een uiting is *relevant*, althans in deze formele theorie, wanneer de inhoud een kwestie deels oplost. Meestal betekent dat dat er een equivalentieklasse afvalt. Tenslotte, een uiting is *toegestaan* wanneer de inhoud alleen maar kwesties oplost die op dat moment van belang zijn. Er mag dus geen irrelevante extra informatie worden gegeven. De relevantie van een kwestie is veel moeilijker vast te stellen. Of een kwestie van belang is, hangt af van het gespreksonderwerp en van de taak.

Een van de aardige dingen van kwesties is, dat ze op verschillende manieren kunnen worden opgeroepen. De formele betekenis van focus-ground constructies, zoals in voorbeeld (148), kan erin worden uitgedrukt. De focus van een zin wordt afgezet tegen een reeks alternatieven, de ground. Die alternatieven komen precies overeen met de equivalentieklassen van een kwestie, in dit geval, eentje voor elke kleur die de fiets kan hebben. De focus selecteert zo'n alternatief. In hoofdstuk 4 passen we kwesties toe op een karakterisering van het begrip *besef* (awareness). Door het oproepen van een kwestie gaan andere agents beseffen wat de mogelijkheden zijn. Op die manier kan een van de nadelen van de standaard kennistheorie, namelijk dat een agent wordt geacht alle consequenties van zijn kennis te kennen, worden verzacht. Hij moet ze wel kennen, desgevraagd, maar hij hoeft het zich niet te beseffen!

Transactie

Mensen beginnen een gesprek om een doel te bereiken. Of ze hebben een taak te vervullen. De eigenschappen effectiviteit en efficiëntie hebben te maken met de taak. Hoofdstuk 4 gaat in op het taakmodel van *transactie* dialogen. Een transactie bestaat uit een aantal fases: een fase van informatie uitwisseling, een fase van onderhandelen met het doen van voorstellen en tegenvoorstellen gevolgd door een bevestiging van de uiteindelijke uitkomst. Zoals bij iedere interactie, is er een openingsfase, waarin contact wordt gelegd en een afsluiting, waarin het contact netjes wordt verbroken. De voortgang van een transactie kan worden uitgelegd aan de hand van een onderhandelingsruimte. Deze bevat alle mogelijke uitkomsten en is gestructureerd door kwesties, zoals hierboven, en door de voorkeuren van de deelnemers. De wensen van een agent tijdens een gesprek worden gemodelleerd met een partiële preferentieordening. Als een agent aangeeft dat ie liever appels heeft dan peren, dan betekent dat volgens von Wright (1963), dat alle situaties met appels en zonder peren, geordend zijn boven situaties met peren en zonder appels. Over situaties met geen van beiden, of met appels èn peren, zegt het niets. Door het uiten van voorkeuren wordt de onderhandelingsruimte ingeperkt, totdat een goede uitkomst is bereikt.

Het uiten van een voorkeur kan in een coöperatieve omgeving opgevat worden als een verzoek, als een suggestie of een voorstel. Wanneer je een voorstel of suggestie aanvaardt, ga je een verplichting (commitment) aan. Verplichtingen zijn doelen waar je aan vast zit, totdat ze vervult zijn, of totdat het geen zin meer heeft. Een transactie is gelukt wanneer er overeenstemming is bereikt en er een wederzijdse verplichting is aangegaan. Een reservering geeft recht op een zitplaats.

Toepassingen

Sommige van deze inzichten zijn toegepast in kleinschalige projecten. Zo hebben we een systeem gemaakt, STEMSTEM, waarmee blinden zouden kunnen stemmen voor de Tweede Kamerverkiezingen. Stemmen is een eenvoudig soort transactie, waarbij de informatiefase grotendeels kan worden overgeslagen. Dit systeem staat beschreven in paragraaf 6.3. Ook hebben we gewerkt aan de component van het systeem die de uitingen moet formuleren en uitspreken. Het is hierbij vooral interessant om te zorgen dat de intonatie goed aansluit bij de context. Relatief nieuwe informatie krijgt nadruk; bekend veronderstelde informatie krijgt juist geen nadruk. Later is er ook een kunstmatig gezicht toegevoegd. We hebben een aantal basis gezichtsuitdrukkingen, die kunnen worden gecombineerd met de gesproken uitingen. Hier is het interessant om na te denken over de controle van het gezicht. Wanneer moet de agent onzeker kijken? Deze overwegingen staan in paragraaf 6.4.

In paragraaf 6.2 beschrijven we bruikbaarheidsexperimenten met een eenvoudig gesproken dialoogsysteem (Bouwman and Hulstijn 1998). Het systeem geeft op verzoek de telefoonnummers en kamernummers van medewerkers bij een bedrijf. Het kan je ook direct doorverbinden met een medewerker. De huidige spraakherkenning maakt veel fouten. Daarom moet de informatie van een uiting worden bevestigd. Dat kan expliciet, door te vragen aan de gebruiker of het goed verstaan is, of impliciet, door de te bevestigen informatie te verstoppen in de volgende uiting. Als het dan niet goed is wordt de gebruiker geacht te protesteren. Uit een vragenlijst en tests bleek dat het systeem in het begin langzaam werd gevonden en dat gesprekken soms verzandden in onmogelijke misverstanden. Het systeem werd aangepast, met behulp van betrouwbaarheidsmaten van de spraakherkenning. Direct doorverbinden mag eigenlijk niet misgaan; dan stoor je iemand. Dus daar werd de impliciete bevestiging vervangen door een expliciete. Bij het opvragen van nummers kan de gebruiker altijd nog het systeem verbeteren. Dus als de betrouwbaarheidsmaat hoog is, dan kan de extra bevestigingsvraag vervallen. De dialoog wordt daar veel korter van.

Uit de tweede testronde, na de aanpassing, bleek dat de tevredenheid van gebruikers was toegenomen. Opvallend was dat de waargenomen kwaliteit van de spraakherkenning was afgenomen. Mensen dachten dus dat het systeem hen minder goed verstond dan daarvoor. Uit de systeemcijfers bleek dat de herkenningsscore niet was afgenomen. Door de aanpassing werden gebruikers gedwongen het systeem te verbeteren in het geval van een spraakherkenningsfout. daardoor werden ze zich misschien meer bewust van spraakherkenningsfouten. Aan de andere kant verdween een deel van de bevestigingen. Dat maakte het systeem *transparanter*: de inspanning ging nu zitten in de echte herkenningsfouten; niet in de algemene dialoogvoering. Daardoor werd mogelijk het mentale model van een gebruiker beter afgestemd op de werkelijkheid. Eerst dachten gebruikers dat het systeem ze niet begreep, later bleek dat het systeem ze niet verstond.

Aist1998 op den Akker et al.1995 Alchourrón et al.1985 Alexandersson1996 Alexandersson et al.1995 Allen and Core1997 Allen et al.1991 Allen and Perrault1980 Allen et al.1995 Allwood1995 Allwood1999 Allwood et al.1992 Andernach1996 Anderson et al.1991 Anderson and Belnap1975 Androutsopoulos et al.1995 Ardissono et al.1998 Asher1993 Asher1998 Asher and Lascarides1997 Asher and Lascarides1998a Asher and Lascarides1998b Aust and Oerder1995 Austin1962 Beaver1996 Bell and Huang1997 Bell and Huang1999 van den Berk1997 van den Berk1998 Bernsen et al.1998 Bilange1991 Binsted1996 Boër and Lycan1986 Bohlin et al.1999 Bolognesi and Brinksma1987 Bouwman1998 Bouwman and Hulstijn1998 Bratman1987 Bretier and Sadek1997 Brown and Levinson1987 Bunt1989 Bunt1996 Carberry1990 Card et al.1983 Carletta et al.1997 Carletta1996 Carlson1983 Carpenter1992 Cassellto appear Cassell and Thórisson1999 Chafe1976 Chopra and Parikh1999 Clark and Marshall1981 Clark and Schaefer1989 Clark1996 Clermont et al.1998 Cockburn1997 Cohen and Levesque1990 Cohen and Perrault1979 Cozannet and Siroux1994 Danieli et al.1997 Davidson1980 van Deemter1994 van Deemter1998 van Deemter et al.1999 van Deemter et al.1994 van Deemter and Odijk1997 Dekker1993 Dekker1997 Di Eugenio et al.1998 Dirksen1992 Dirksen1997 Dutoit1997 Dybkjaer et al.1998 Dybkjaer et al.1996 Ebbers et al.1999 Engelmore and Morgan1988 Fagin and Halpern1988 Fagin et al.1995 Fikes and Nilsson1971 Fowler and Scott1997 Fraser and Gilbert1991 Gärdenfors1988 Gardent et al.1996 Gazdar1979 Georgeff and Lansky1987 Gerbrandy1997 Gerbrandy1999 Geurts1994 Gibbon et al.1998 Ginzburg1991 Ginzburg1995 Ginzburg1996 Ginzburg1998 Goffman1981 Grice1957 Grice1975 Groenendijk1998 Groenendijk1999 Groenendijk and Stokhof1984 Groenendijk and Stokhof1989 Groenendijk and Stokhof1991 Groenendijk and Stokhof1996 Groenendijk et al.1996 Grosz1977 Grosz et al.1995 Grosz and Kraus1996 Grosz and Sidner1986 Grosz and Sidner1990 Grover et al.1994 Hamblin1970 Harel1984 Heeman and Hirst1995 Heim1982 Heim1983 Heim1992 Hendriks and Dekker1996 Hinchey and Jarvis1995 Hoare1985 Hobbs1978 Hobbs1979 Hobbs1990 van der Hoeven et al.1995 Houghton and Isard1987 Huang1994 Huang and Masuch1995 Huang et al.1996 Huls et al.1995 Hulstijn1995 Hulstijn1996 Hulstijn1997 Hulstijn1998 Hulstijn and Nijholt1999 Hulstijn et al.1996 Hulstijn and van Hessen1998 ISO 91261991 Jacobson et al.1992 Jäger1996 Jennings et al.to appear Jokinen1996 Jönsson1993 Jönsson1997 Kamp and Reyle1993 Kaplan1982 Karttunen1973 Karttunen1974 Kay et al.1994 Kellner et al.1996 Klumperto appear Kowtko et al.1992 Krahmer1995 Krahmer et al.1999 van Kuppevelt1995 Lascarides and Asher1999 van Leusen1997 Levelt1989 Levelt and Kelter1982 Levesque1984 Levin and Moore1978 Lewis1969 Lewis1973 Lewis1979 Lie et al.1997 Litman et al.1999 Litman and Allen1987 Mann1988 Mann and Thompson1988 Mercer1992 Meyer and Wieringa1993 Milner1980 Moore and Paris1993 Moser and Moore1996 Müller1997 von Neumann and Morgenstern1944 Nielsen1993 Nijholt1999 Nijholt and Hulstijnto appear Nijholt et al.1999 van Noord1997 Oviatt1995 Oviatt and Cohen1990 Parke and Waters1996 Peirce1867 Pinkal1995 Poesio and Traum1998 Polanyi1988 Polanyi and Scha1984 Pollack1990 Pressman1997 Prüst et al.1994 Rats1996 Reiter and Dale1997 de Rijke1993 Roberts1989 de Roeck et al.1991 Rooth1992 Rosenschein and Zlotkin1994 Rumbaugh et al.1991 Russell and Norvig1995 Sacks et al.1974 van der Sandt1989 van der Sandt1992 Scha1983 Schank and Abelson1977 Schegloff et al.1977 Schelling1960 van Schooten1999 van Schooten et al.1999 Schöter1996 Searle1969 Searle1975 Searle1983 Searle1990 Sengers1999 Sgall et al.1986 Shieber et al.1996 Simon1955 Singh1994 Smith and Hipp1994 Sperber and Wilson1986 Stalnaker1979 Stalnaker1984 Stone and Velosoto appear Sturm et al.1999 Tanenbaum1995 Tekinerdoganto appear Theune1997 van der Torre and Tan1997 van der Torre and Tan1999 Traum1997 Traum1994 Traum and Hinkelman1992