A Method for Enterprise Ontology based Design of Enterprise Information Systems

To Ellen.

The whole of science is nothing more than
a refinement of everyday thinking.

Albert Einstein (1879-1955)

A Method for Enterprise Ontology based Design of Enterprise Information Systems

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. ir. K.C.A.M Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op 25 november 2013 om 12.30 uur

door

Johannes DE JONG
wiskundig ingenieur
geboren te Ridderkerk.

Dit proefschrift is goedgekeurd door de promotor:
Prof. Dr. ir. J.L.G. Dietz.

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. Dr. ir. J.L.G. Dietz,	Technische Universiteit Delft, promotor
Prof. Dr. E. Babkin,	Higher School of Economics, Nizhny Novgorod, Russia
Prof. Dr. ir. J. van den Berg,	Technische Universiteit Delft
Prof. Dr. ing. J.B.F. Mulder,	Antwerp Management School, België
Prof. Dr. H.A. Proper,	Radboud Universiteit Nijmegen
Prof. Dr. J. Tribolet,	Technische Universiteit Lisboa, Portugal
Prof. Dr. J. Verelst,	Universiteit Antwerpen, België
Prof. Dr. ir. F.M. Brazier,	Technische Universiteit Delft, reservelid

SIKS Dissertatiereeks nr. 2013-39
Het in dit proefschrift vermelde onderzoek is uitgevoerd onder de auspiciën van SIKS, de Nederlandse School voor Informatie- en KennisSystemen.

Verspreid door:
Mprise Group BV.
Postbus 598
3900 AN Veenendaal
jdjong@mprise.nl

ISBN: 978-90-5335-758-3

Druk: Ridderprint, Ridderkerk
Omslagontwerp: Danieck Meere, Meere Communicatie

Acknowledgements

During holidays, we make long bike rides. We usually drive through beautiful scenery. Sometimes we lose the road, sometimes it goes against the wind and rain, but in the end we enjoy the trip. We meet people and have interesting conversations with them. Although we enrich ourselves personally with new experiences, we are always glad to reach our destination. Then, a sense of accomplishment arises.

To me, writing a dissertation is similar to such a bike ride. Not like a bike ride during a holiday because being a manager of an IT company swallows a lot of time and energy. However, I have often experienced that working on a dissertation even may be a relaxing pastime, how strange this may sound. Working on this dissertation has given me, both personally and professionally, a constant feeling of intellectual growth.

Writing a dissertation also distinguishes from a bike ride by the absence of a road map. That means that one easily can end up in mazes. Especially during the first years of the study, it happened to me more than once. Then, it is necessary to meet people who encourage you to continue.

First of all, I thank my promote Jan Dietz for guiding me during my scientific ride through the enterprise engineering field and for the introduction into the world of conceptual modeling of organizations from the DEMO mindset. It has enriched my life. In addition to this, I also thank Hans Mulder who always stimulates and supports me on his typical inspiring way.

Performing a PhD project next to a busy job has a tremendous impact on time at the expense of my family. One thing is beyond any doubt: without the loving support, sacrifice and attention of my wife Ellen this PhD project never came to an end. To her I dedicate this book.

Joop de Jong, 2013

Summary

Problems and Objectives

With the purchase of a business software application (at this study called Enterprise Information System, EIS for short), the enterprise has opted for an operational way of working, without being very well informed about the relevant features of the package and without the supplier of the package being aware of exactly how the enterprise works. People realize this, but there is a generally accepted belief that potential problems due to the package not being aligned with the enterprise should be resolved during the implementation process. One customizes software or adapts business processes as alignment frictions occur. Clearly, the changes must be limited because the supplier must operate within a bandwidth in terms of time and cost.

The approach described here became commonplace the last decades. It can be characterized as an approach based on 'best practices'. This approach is widely accepted, partly because of the increased configuration capabilities of standard EISs.

From the early 1980s, the researcher consciously took part in this development in senior management positions at leading suppliers of standard ERP systems. He started developing standard software products and later went on to provide IT services to enterprises that had implemented large EISs. The researcher often wondered whether companies buy the software that they actually need (this question also lies at the basis of this study). By this is meant not only 'need at present' but also 'need in the near future'. That question was fueled by practical experiences that showed that (1) enterprises make only partial use of the options in the software they have purchased; (2) enterprises sometimes wonder whether their organization has not perhaps diverged from the EIS they once implemented, and (3) enterprises sometimes see their EIS as a barrier to implementing changes.

How should the relationship between an organization and an EIS be understood? In this summary, we shall not discuss this topic in great depth, it is clear that an EIS should support the company's employees in performing their tasks. An EIS is usually so overtly present in the organization that the way in which processes are operationalized is to a large extent determined by the construction of the EIS. As an example, a forklift could be considered as a tool in the hands of an employee within the organization meant to perform a particular production act. An EIS is of a very different order: it should be considered as a tool to operationalize a part of the business organization. It implicitly determines how people work together. It supports their information needs and 'remembers' new created facts to make them available as information later. The importance of this observation should not be underestimated. After all, the enterprise's operational performance is determined by the construction of the enterprise!

Taking this into account, we like to ask the questions that form the basis of this study, namely: how do we develop an EIS that fits in an organization and how do we decide whether a particular EIS includes what the organization needs? These questions assume, on one hand, that we can decide what the organization needs and, on the other hand, that we are able to conclude to what extent an existent EIS supports the organization. This study shows that this is indeed possible.

Research Approach

This study has been carried out by making use of the Design Science Research methodology. According to this methodology, an intervention type is developed for a class of problematic contexts in which a number of generative mechanisms are called on to achieve a desired outcome. In general, generative mechanisms explain why an intervention should lead to a desired outcome. The class of embedding standard EISs within organizations should be understood as the class of problematic contexts. We already mentioned that the embedding of a standard EIS within an organization does not always bring what the stakeholders of the organization expect. As a result, the organization will not perform optimally. Therefore, an intervention type needs to be defined by which an EIS can be developed that meets the ultimate needs of the organization. This intervention type is the subject of our study. It provides a set of rules based on a specification framework. The intervention type is called the Procedure for Implementation Design - Framework, the PID-Framework for short. The generative mechanisms on which the PID-Framework is grounded are the ψ-theory, the τ-theory, and the φ-theory. These theories are discussed in the second chapter of this dissertation. The ψ-theory regards the organization as a social system whereof the elements and the influence bonds between the elements stand for actor roles and transactions, respectively. Business actors perform coordination acts and production acts using different abilities in order to create coordination facts and production facts. Business actors need these facts, or derivations of these facts, for deciding the acts to perform. The τ-theory describes the generic process of deriving an object system from a using system, for example, the derivation of the infological organization from the business organization. From the using system the function model of the object system is defined. Subsequently the implementation-independent construction model of the object system, also known as the ontological model, is designed from the function model of the object system. Many alternative implementation models could be produced from the ontological model. The τ-theory shows that an EIS comprises the technology that must be allocated to components of a particular implementation model. The φ-theory, which the ψ-theory also invokes, throws considerable light on the idea of a ‘fact’ in the infological and datalogical organization within an enterprise.

Three applications have been executed to validate the applicability of the Framework and to refine the Framework. The first application regards a case- study which is already written extensively in literature, namely The Foundation for Consumer Complaints Boards. This case-study is sufficiently suitable as a basis for the design of an EIS by using the PID Framework. The second application regards the assessment of an EIS that has been implemented in an organization that provides commercial trainings, and that has already been in

use for several years. The third application focuses on the development of an EIS for a typical Dutch line of business, namely exporting flower bulbs.

Research Results

Current enterprise engineering research places a strong emphasis on describing the essence of the enterprise in an ontological model of the business organization. Although both original facts and derived facts are specified in the ontological model, no attention is paid to the real derivation of facts and how they are shared between the actors. According to the terminology used within the enterprise engineering domain, this is known as the realization of the essential model by adding a model for the infological and datalogical organization. There are many reasons to call why this is vital for the design of an EIS.

Firstly, the ontological model of the infological organization provides an implementation-independent model for remembering, deriving and providing C-facts and P-facts. The semantics and syntax aspects of a fact are distinguished. This distinction is important because the semantic meaning of a fact can be captured in one of more documents. The ontological model of the datalogical organization provides an implementation-independent model for archiving, transforming and fetching documents.

Secondly, besides the other aspect organizations, the physical organization is distinguished. Although the physical organization is not elaborated in this study, some features of this organization are necessary to understand. The ontological model of the physical organization provides an implementation-independent model for storing, copying, destroying, transmitting and retrieving files. A file is an imprint of a document.

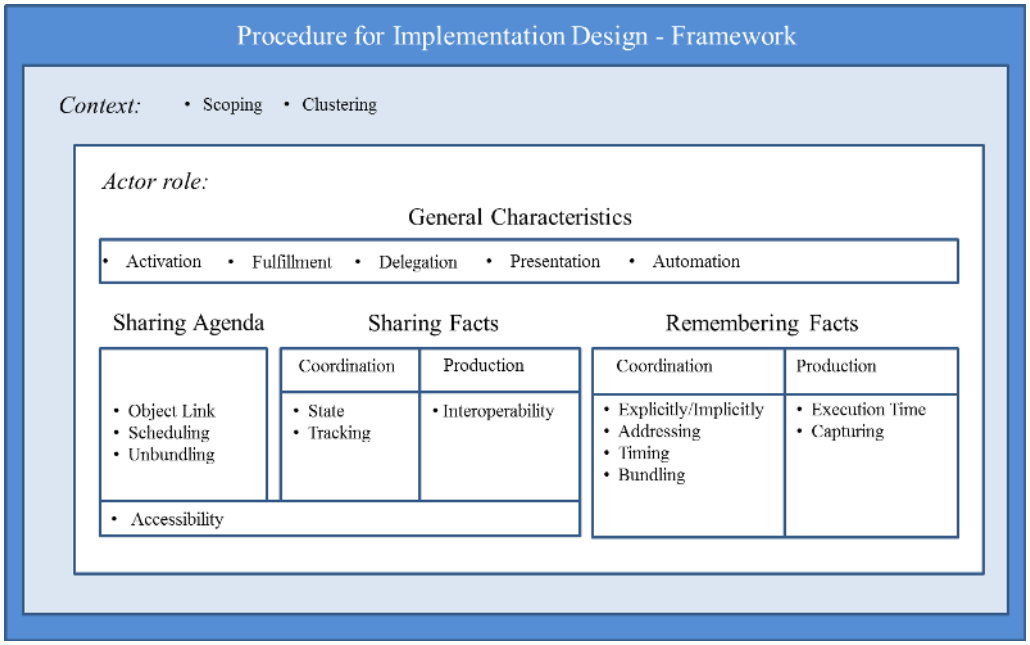
Thirdly, the enterprise is understood as a social system. In the pre-computer age, processing and using facts were a shared human activity. Currently, deriving and sharing facts are usually executed by a software system with the consequence that in many cases, it is completely unclear which actor bears responsibility for the quality of original facts. Clearness concerning this subject will be achieved by developing an EIS from an integral coherent ontological model of the business organization, the infological organization and the datalogical organization.

Fourthly, The availability of an ontological model of the infological and datalogical organization makes it possible to evaluate (at the level of actors) the extent to which actor roles should be fulfilled by human beings or agents (software technology).

In short, the availability of an extended ontological model of the enterprise is an essential requirement for the development or the selection of an EIS. This ontological model has to be considered as the starting point for the design of various scenarios for the EIS to be developed or selected.

This study does not discuss the criteria which may be used to distinguish the different implementation scenarios. It has been focused on the question of what conditions an implementation scenario must face if it reflects the underlying ontological model. Answers on questions as, to what extent actor roles are automated and what kind of technology is used for developing the software are understood as out of the scope of this study.

In summary, this study has resulted at a way of working to arrive at an EIS that is based on a coherent ontological model of the business organization, the infological organization, and the datalogical organization.



Many aspect groups are defined at the PID-Framework, each of which specifies a number of aspects. Every aspect is connected to one or more rules that should be considered as specific work instructions for the process of designing an EIS. These instructions are drawn up in such a way that they can be used not just to specify a new EIS from the ontological model of the organization but also to assess the applicability of an EIS for an enterprise.

In the third part of this study, three cases are discussed. On the basis of the PID-Framework, the aspects to be taken into account at the design of an implementation scenario are illustrated for a number of actor roles. These aspects are mainly derived from both the scientific literature and from several decades of practical experience of the researcher in the development and implementation of EIS's. In each case study, these aspects are validated by an intensive discussion with those who are directly involved. In the first case as well as in the third case-study, both engineers and users of the EIS were involved. The second case is different from the others by the presence of an EIS that was already implemented. In this case, the validation of the aspects from the PID-Framework took place with the users of the EIS. They told that some aspects of the implementation design of the EIS had not been implemented in their current EIS. The users discovered that actor roles were not named explicitly, and that they were not or only to a small degree supported by the EIS, and that a significant part of coordination between actor roles was based on improvisation.

Contents

Abbreviations

AM	Action Model
BCT	Bank Contents Table
C-fact	Coordination fact
CM	Construction Model
COTS Software	Commercial Off-The-Shelf Software
C-world	Coordination world
DEMO	Design & Engineering Methodology for Organizations
EIS	Enterprise Information System
ERP	Enterprise Resource Planning
FM	Fact Model
IAM	Interaction Model
ISM	Interstriction Model
PID-Framework	Procedure for Implementation Design - Framework
P-fact	Production fact
PM	Process Model
P-world	Production world
TRT	Transaction Result Table

PART I

Introduction

1 Introduction

1.1 Business Process Control

Interest in the active control of business processes is a relatively new phenomenon. The insight that enterprises could save a lot of money through small improvements in the area of effectiveness and efficiency only really took hold at the beginning of the twentieth century. Frederick Winslow Taylor, born in 1856, is widely acknowledged as a pioneer of this movement. In his classic work *Scientific Management* [Taylor 1911] Taylor argues against the notion that managers do not have to worry about motivating their employees and that merely reminding your staff of their duties should be sufficient. Taylor points out that as an employer, you can only seek to increase the efficiency of the work if you make sure that your employees are provided with proper training and good working conditions. Taylor worked in America, and in order to make his argument seem plausible he had to produce something spectacular. He therefore chose one of the most simple, unskilled types of work: moving pig iron. He revealed that the output per worker could be quadrupled by carefully studying the work, choosing the right man, adapting the tools, and effective instruction [Taylor 1903; Bennett 1990].

Around the same time, the Frenchman Henry Fayol, born in 1841, highlighted the responsibility of management in enterprises. Management is a profession, and managers therefore need to receive proper training. Managers are responsible for the performance of the organization. Fayol was central to the establishment of a theory of Management Studies [Fayol 1920]. In an interview that appeared in *La Chronique Sociale de France* of January 1925, Fayol said in response to what he considered to be the best way for an organization to be audited and to determine what improvements are needed: “The best method is to study what I have called the ‘control apparatus’. If the said control apparatus is as it should be, it will be possible to obtain accurate information about the current situation of the enterprise and its overall growth. One will then also be able to establish immediately whether prediction and planning, organization, management, coordination, and control are being taken care of properly, i.e., whether the organization is being properly managed. If gaps are found in the control apparatus, these will be the result of weaknesses in the organization or deficiencies in the functioning of the enterprise.” Fayol defines the control apparatus as “a system of observation that encompasses the present, the past, and the future, wherein the contributions of experienced members of staff, along with information from external sources, create the best opportunity for the board to estimate the possible consequences of their decisions.” It is clear from this that Fayol considers management to be responsible for the proper functioning of an organization, which is very much in line with the ideas behind Total Quality Management [Deming 1986; Deming 2000].

In Germany, Max Weber, born in 1864, was concerned for a long time with the phenomenon of power and leadership and in this context also the legitimacy of authority. According to Weber, legitimate authority is grounded in rationality, which is in turn based on the

belief at the legality of patterns of normative rules and the right, within the framework of those rules, of certain people has been assigned a role of authority to give orders. Such rational authority can also be found in bureaucracy [Fayol 1920].

The foundation had been laid. In that time, manpower was still relatively cheap. Industrial companies invested in inventories of raw materials, intermediate and finished goods needed to meet customer demand and keep production going. Because calculating the exact amounts of materials needed for production was far too labor-intensive for a company with many different and complex end products, various methods were developed to ensure sufficient inventory with relatively little effort [Woodward 1958; Woodward 1965]. Examples include inventory replenishment systems, where the inventory of an item is returned to a predetermined level once consumption has been detected, and order point inventory systems, where replenishing is based on assumptions about the consumption and delivery times of items. However, they are both only approximations in the sense that they fail to take the real demand for items into account. Actually, this approach only began to be considered when computers became available for commercial purposes in the 1960s. 'Total systems' also date from this time [Harvey 1968; Mohr 1971]. After all, computers can calculate anything as long as you give them enough data to work with. Orlicky [Orlicky, Plossl et al. 1972; Orlicky 1975] published an authoritative book that describes how computers are able to calculate which intermediate goods, and procurement items are needed at any given time in the production process in order to enable the execution of a given production plan. Every time the production plan changes, e.g. because a new order is placed, the consequences for the materials supply can be recalculated. This theory is known as 'Material Requirements Planning (MRP I)'. A further refinement of the theory led to what is currently referred to as 'Enterprise Resource Planning' (ERP). ERP systems have been widely used by organizations wishing to work with integrated information systems at the hope of increasing their market agility [Keller 1999; Grabski and Leech 2007]. In short, the foundation laid by Taylor, Fayol, and Weber for the active control of business processes was the starting point for the trend of approaching the design of organizations in an integral way [Dietz and Hoogervorst 2012]. This pursuit of effectiveness and efficiency was an important driver for the use of information technology in organizations, and indirectly also for the development of scientific knowledge about developing enterprise information systems.

1.2 Information Process Control

1.2.1 Generations of Information Science

Following Hirschheim [Hirschheim, Klein et al. 1995], he distinguishes seven different generations in the development of information science. The first generation, in which there was a noticeable structure for the first time, began in the early 1970s. The methods of this time focused on the phasing of information system development projects. A well-known methodology from that time is System Development Methodology [Hice, Turner et al. 1970]. It was only the next generation, in around the 1980s that focused on analyzing user specifications and incorporating them in a well written design. Well-known publications in this generation include Yourdon et al. [Yourdon and Constantine 1979], Codd [Codd 1969;

Codd 1970], Chen [Chen 1976], Lundeberg et al. [Lundeberg, Goldkuhl et al. 1981] and Nijssen et al. [Nijssen and Halpin 1989]. A distinction is made between physical and logical data structures [Langefors 1977]. Unfortunately, these models are usually not noted for their clarity and transparency. The next (third) generation introduced new, interactive methods. From the mid-1980s, prototyping and Rapid Application Development [Martin and Finkelstein 1981] became key concepts. The user became co-creator of the ultimate solution. But this also entailed risks. Mulder [Mulder 2006] points to the risk that the development of more and more functionality is seen as a solution, while this is actually merely treating symptoms caused by the lack of precise specifications. He also frequently sees that this kind of application development leads to the re-automation of existing manual systems and other systems. The fourth generation, in the early 1990s, put more emphasis on the socio-organizational aspects of changes. Hirschheim [Hirschheim, Klein et al. 1995] speaks of social relativism. The management formulates the corporate strategy, the employees understand the nature of the work, and within this framework information system development is carried out by the facilitator, computer experts, and users. This social relativism is reflected in methods such as ETHICS [Mumford 1985] and Soft Systems Methodology [Checkland 1988]. A number of experimental methods are known for the fifth and sixth generations, but no real-life examples. Therefore, these methods are disregarded. From a completely different perspective, Porter introduced the theory of the Value Chain [Porter 1985; Porter and Millar 1985]. Various researchers [Hammer 1990; Scott Morton 1991; Davenport 1993; Donovan 1994] have elaborated on it. Methods in the seventh generation of information science are developing into methods for organizational change. The Language Action Perspective [Austin 1962; Searle 1969] is required to play a dominant role here too [Goldkuhl and Lyytinen 1982; Dietz 1990; Dietz and Widdershoven 1991; Dietz 2004]. It provides assistance with designing and redesigning information systems and organizations [Keen 1991]. The key to the desired integration lies in elevating communication to the central concept for understanding organizations and their operation. The still relatively young approach of Language Action Perspective is in a position to achieve an integrated design comprising the business functions, business processes, organizational structure, and information systems. The focus is on communication as a concept for understanding and modeling systems and organizations. Speech Act Theory considers communication not only as a means to transfer information, but also as a kind of action, with which new facts can be created. Searle [Searle 1969] has attempted to categorize speech acts. He identifies five categories of speech acts. The philosopher Jurgen Habermas [Habermas 1981] sought to remedy particular weaknesses in Speech Act Theory with his Theory of Communicative Action. Central to the philosophy of Habermas are the validity claims of speech acts, which are now called communicative actions. According to Habermas, the claim to validity made by a person performing a communicative act, and the ability of the addressee to challenge it, constitute the operating principle of the coordination between people. There are three types of validity claims: the claim to truth, the claim to justice, and the claim to sincerity. Each type of claim has its own world from which its validity is derived. The claim to truth derives its validity from the state of the shared objective world of the speaker and the addressee. The claim to justice derives its validity from the state of the shared social world, and the claim to sincerity derives its validity from the state of the subjective world of the speaker. See also [Koningsveld and Mertens 1986; Reijswoud van

1996; Reijswoud van, Mulder et al. 1999; Dietz 2004; Finlayson 2005] for a more detailed treatment of this theory.

The concept of the 'communicative act' as the key to understanding organizations transcends the concepts of the 'form' and 'content' of information. These are concepts that are familiar from Information Systems Engineering [Martin and Finkelstein 1981]. There is a growing perception that the concepts of 'form' and 'content' are not enough to understand the relationship between the organization and IT. A new concept is needed, which is the concept of 'intention'. If the communicative act functions as a key to understanding organizations, then the concept of 'intention' clarifies and explains the organizational concepts 'cooperation', 'responsibility', and 'authority' [Winograd and Flores 1986; Dietz 1990; Dietz and Widdershoven 1991].

1.2.2 Adoption of an Industrial Concept

In parallel with the growth of information science through many generations as outlined above, another phenomenon has been seen since the early 1980s. It is connected to the fact that enterprises sometimes have a lot in general, especially when they run in the same industry. It turns out that these enterprises need the support of information systems that have many in commonality. This raised the question of whether the development of information systems could be much more efficient if commonality were taken into account. A number the of forward-looking IT companies took up the challenge in the early 1980s and began developing and marketing so-called Commercial Off-The-Shelf (COTS) information systems [Scheer and Habermann 2000]. Holland and Light noticed this trend early on: "Companies are radically changing their information technology strategies by purchasing prepackaged software instead of developing IT systems in-house" [Holland and Light 1999].

The development of standard enterprise information systems, as COTS products are also called, has really taken off. Although efficiency is an important driving force in information system development, this was not the only mainspring for visionary entrepreneurs. They argued that history had shown that if the IT sector wished to become more professional, this could not be done by adhering to the service model. It would only be possible by adopting an industrial concept. Real progress could only be made by thinking in terms of a single standardized database and sharing implementation practices between companies. It was during this time that the so-called 'BOPS' companies were founded: **B**aan (1978), **O**racle (1977), **P**eopleSoft (1985), and **S**AP (1972). They were the four world leaders in the field of COTS software around the year 2000 [Koedijk and Verstelle 1999]. Frick et al. [Frick and Schubert 2009]: "The further development of software by the supplier is driven internally by the will to improve but also from external pressure to keep pace with the competition. Developers try to differentiate themselves from their competitors in every sector by offering innovative system architectures, new technologies, comprehensive business models and assertiveness in the market. This competitive pressure leads to the situation where systems do not continue in their old form but are continually altered and newly designed."

The author entered Baan in the early 1980s and was fully involved in this debate as he had ultimate responsibility for the development of standard enterprise information systems. BOPS companies saw the information technology sector as a sector that operated like craftsmen. Every information system delivered was seen as unique. From their viewpoint, most IT companies were reluctant to make in-depth investments in research and develop-

ment. They fail a long-term view of the market and their own place in that market. Their core business consists of detaching their own employees to their customers. According to Jan Baan, the founder of one of the BOPS companies [Baan 2005], one can recognize every advanced development by its supporting process type. In his opinion, the industrial level of most IT companies does not exceed that of the typical village carpenter's workshop. The BOPS companies, on the other hand, claim that they have implemented industrial processes by which software components are engineered. These components are merged into EISs. BOPS companies argued that the cost per customer can be much lower than if everyone were to write his own software, or have his own software written. Standard software is immediately available, whereas so-called custom built software would still need to be written, and the use of standard software avoids reinventing the wheel. Theoretically, data integration and best practices also provide opportunities to improve business operations.

Organizations that adopt ERP expect, among other things, to be able to eliminate inconsistent and duplicate data and to redesign their processes. They also expect to reduce the lead time of their production, make fewer mistakes in processing orders and thus increase customer satisfaction [Shang and P. 2002; Adam and O'Doherty 2003; Duplaga and Astani 2003]. Gluchowski et al. [Gluchowski, Gabriel et al. 2008] mention three classic advantages of package software compared with company-specific developed software: (1) low cost despite high procurement costs, (2) time saving and (3) long term security (see also [Davenport 1998; Shang and P. 2002; Bernroider 2008; Sneller 2010]).

Nevertheless, research shows [Koning de 2004] that the functional objectives of ERP implementation projects are not or not fully realized in most cases. Empirical research shows that whether an ERP implementation is successful or not (where success is defined as the achievement of project objectives) depends to a large degree on the fit between the ERP package and the business process to be supported by the package. This confirms what Sumner reports based on a literature review on specific risk factors and ERP projects [Sumner 2000]. Besides lack of skills and expertise in the implementation-team and problems with respect to the planning and the integration of the technical solution, he recognized an inadequate adaptation of business processes, a lack of an organization-wide approach to data integration, and an insufficient adaptation of the organization to the concept of the standard EIS as risk factors. That adjustments of business processes are needed has been amply demonstrated [K. and Hillegersberg 2000; Hong and Kim 2002; Grabski and Leech 2007; Bernroider 2008; Kallunki, Laitinen et al. 2011].

In this context it should also be noted that many enterprises that purchase an EIS fail to understand the meaning of the EIS for their organization. In a number of meetings with business people and managers, the author asked whether they could tell him the differences between buying an EIS and, for example, a truck. The majority saw no differences between the procurement of both goods. Another group believed that there were differences but did not know exactly what they were. Only a few individuals could explain that a business application should be understood as the technology needed to make business and information processes operational. People hardly seemed to realize that the responsibility for selecting the technology, and appropriately connecting it up to the business and information processes, lies entirely within the enterprise.

With this, we have arrived at the heart of the problem, namely the disconnect between the disciplines related to thinking about the construction of business processes and those dealing with the use of IT within the organization.

1.3 Business-IT Alignment

Research into Business-IT alignment has yielded many scientific publications. Henderson and Venkatraman [Henderson and Venkatraman 1993] argue that the lack of alignment between business operations and IT causes many IT investments to fail to pay off optimally. Strassmann [Strassmann 1997] establishes the link between the alignment and the contribution of IT to an organization. The overviews [Melville, Kraemer et al. 2004; Chan and Reich 2007] in this field sketch a picture of the quest that has since been performed by a large number of authors. It lacks practical handles and is therefore, in large measure irrelevant [Sauer and Burn 1997], it is unclear how to reach [Bryson and Currie 1995] and it is even inconvenient and harmful [Ciborra 1997]. It depends on the use of the system [Godwin 1992], namely the extent to which it is actually used and the extent to which the user is satisfied. Briggs, De Vreede & Nunamaker introduce the Technology Transition Model [Briggs, Vreede de et al. 2003]. They describe aspects that determine the added value of IT: economic, satisfaction, physical and cognitive, political (dominant position), social (personal relationships). They take into account the frequency with which these aspects occur and the costs and benefits of the transition process from the old to the new circumstances. The value added can be expressed in many aspects, but also depends on the view one chooses that of the shareholder that of an employee, or that of a customer. Cronk and Fitzgerald [Cronk and Fitzgerald 1999] indicate that the added value not only depends on the stakeholder, but it also depends from the business and the type of system.

Finally, it is not possible to speak about alignment in a general sense. Because the emphasis is very much on the process of aligning the business with IT, the question of what should exactly be the result of this alignment fades somewhat into the background. It is a difficult task to measure accurately the added value of IT for the business of the organization. However, many studies have shown that the perceptions of managers regarding the contribution of IT to the business are often useful indicators for showing the real situation [Tallon, Kraemer et al. 2000].

It is clear that the implementation of alignment between business and IT should lead to the joint performance of business people and IT people of added value that should ultimately be reflected in an effective and efficient operation. Dietz [Dietz 2006] indicates that the effectiveness and efficiency of operations are the result of the design of the organization. In this context, he speaks of the construction of the organization. Much research related to the business-IT alignment domain relates to the revenues of the organization and the contribution of IT in it, but according to Dietz, alignment between business and IT should be reduced to a question of operation of the organization.

Despite this research, organizational studies are still unable to address the various aspects of an organizational design (structure, business processes, information systems, etc.) in a sufficiently integrated manner [Mulder 2006]. Dietz et al. [Dietz and Mulder 1998; Dietz and Hoogervorst 2012] found that the methods developed during the first six generations of information science, see section 1.2.1, are not suitable for the development of information systems. “At present there is a gap between approaches for modeling business processes and those for modeling information systems. Due to this gap, the translation of the business processes into an information system (and vice versa) and consequently the alignment of business and IT has become difficult.” [Reijswoud van, Mulder et al. 1999]

This is the fundamental problem posed by this study. There is a strong need for a methodology for creating a supporting information system that is based on the design of the construction of an organization. Barjis [Barjis 2008] notes that the practice of the last two decades shows the failure of software systems over and over again, due to poor modeling. These systems fail not because of technical flaws, but because they do not adequately support the underlying business processes. He refers to a survey by the Standish Group [Standish 2004] conducted among IT executives indicating that only 29% of software projects succeeded, while 53% were challenged and 18% completely failed. As pointed out by these IT executives, the primary reason for software projects being challenged or failing is poor conceptual modeling (requirements' definition). Surveys conducted every 2–3 years since 1994 by Standish Group shows consistently that the primary reason for software projects being challenged or failing is poor conceptual modeling. According to Barjis this has all instigated the start of a new wave in conceptual and process modeling that has been seen as a requirement for successful software intensive systems design [Carr 2003; Smith and Fingar 2003; Smith and Fingar 2003]. This has led to the emergence of a large number of methodologies for modeling and analyzing business processes. Many of these methodologies present rich design environments (tools, graphical editors, library support), but lack theoretical rigor that should lead to accurate conceptual models and abstractions. The fact that they do not have a clear theoretical underpinning makes it difficult to justify the models created with them.

Dietz [Dietz 2006] argues that a method to design a supporting information system from the organization design must meet the C4E quality criteria, i.e., it must be coherent, comprehensive, consistent, concise, and essential. These notions are conceived by the author as follows. By *coherent* is meant that the working process within this method [Seligmann, Wijers et al. 1989] must constitute a reasonable and truly integral whole. By *comprehensive* is meant that it must include all relevant aspects needed to move from a model of the business organization to a model of the information organization and then on to a model of an implementable information system. By *consistent* is meant that the approach must be free of irregularities, and by *concise* is meant that no unnecessary trappings should be incorporated in the process and that the method should be defined as concisely as possible. Finally, by *essential* is meant that the method must be universally applicable, that is to say that it generalizes from the particular functionality of the information system to be designed.

The research of this study ties in with the seventh generation in information science [Hirschheim, Klein et al. 1995], which considers organizations to be social systems in which people communicate, where the dialogue that really matters is formalized in transaction kinds, and where people who perform similar transactions are grouped into roles. This approach stimulates the developer of information systems to think in terms of the needs of actors. This information science approach shows significant overlap with the organizational science approach [Reijswoud van, Mulder et al. 1999]. These new insights about the organization do not have direct consequences for the relation between the organization and IT. According to Te'eni [Te'eni 2001], the software systems we develop should play a role in enabling effective communication within an organizational or business setting. In order to achieve this, the underlying model should draw a balance between relationship and ac-

tion, cognition and affect, message and medium. Dietz [Dietz 2003] and Maij et al. [Maij, Toussant et al. 2002] derive a set of use cases from the essential business process that is considered as a starting point for the development of an information system. Mallens, Dietz and Hommes [Mallens, Dietz et al. 2001] define an 'information system' as a system that belongs to the category of rational systems. They talk about rational individuals who perform rational acts. According to Mulder and Dietz [Mulder and Dietz 2002], "Information systems belong to the category of rational or conceptual systems. The components of a rational system collect, distribute, include, and derive knowledge of facts about some world. They act upon each other by emitting commands to each other to perform these rational operations. The components run in a rather mechanical way, i.e. a command is a cause for some effect, a stimulus to which there is a particular well-defined answer. This mechanical understanding of how a system works fits perfectly well for rational systems, thus for information systems. The mistake many IT people make is that they consider a business system as a kind of information system, and consequently apply the (rational) data model and information process model to understand business systems. They fail to recognize and appreciate then that organizations are essentially social systems, not rational ones." Mulder [Mulder 2006], finally, writes about actors as social components who are embedded in information systems and infrastructure systems. Mulder and Dietz, once again [Mulder and Dietz 2002]: "In both informative and performative communication, but most significantly in performative communication, the subjects are engaged in mutual commitments. With the notion of commitment we are at the heart of the category of social systems, of which organizations are a special kind. A social system is a system of which the elements are social individuals (human beings) who enter into and comply with commitments. This is the working principle of social systems, thus also of organizations." It is clear that what is meant here is that it is not the nature of the productive activity that determines the system category, but rather the concept of 'transaction', it being understood that the claim to justice is the dominant claim in business, while for intellectual activities this is the claim to truth [Habermas 1981].

The trend is apparent: in scientific research, information science and organizational science are converging. The concept of 'communicative action' is the key to understand organizations. From this concept, the coherence between business processes and information processes is elaborated in this study. Based on an integrated model of business organization and information organization that is constructed in an implementation-independent manner, the author will present a method for arriving at the design of an EIS for a particular scope of interest.

1.4 Research Method and Design Proposition

The proposed study is situated within the field of design science research. After discussing the concept of 'design science research', the design proposition will be formulated and the underlying principal research questions explained. Lastly, the research strategy that will be followed for this proposition is elaborated.

1.4.1 The Research Method

Recently there has been an increased interest in the design science paradigm and its potential for improving the relevance and application potential of the research base [Aken van 2004; Aken van 2005]. Denyer [Denyer, Tranfield et al. 2008] states that design science research privileges prescriptive knowledge, i.e. knowledge linking interventions to outcomes, and grapples with the vexing question faced daily by managers of 'how should things be?' This does not mean the actual application of scientific knowledge to solve a particular managerial problem - this is the domain of the workers in practice - but the development of scientific knowledge to solve a class of managerial problems. That does not involve recipes, but the development of field-tested and grounded technological rules to be used as design exemplars of managerial problem-solving [Aken van 2004].

In this context, based on the idea of Simon [Simon 1996] about fundamental differences between (natural) science and the 'sciences of the artificial', March and Hevner [March and Smith 1995; Hevner, March et al. 2004] speak about the difference between behavioral sciences and design sciences. The mission of an explanatory or behavioral science is a quest for truth by developing knowledge aimed at the classical triplet of description, explanation and prediction. It seeks to develop and justify theories (i.e. principles and laws) that explain or predict organizational and human phenomena surrounding the analysis, design, implementation, management, and use of information systems [Hevner, March et al. 2004].

Design sciences include engineering, law and, according to Simon, also management. The mission of a design science is to develop knowledge for the designing and achieving artifacts, i.e. to solve construction problems, or to be used in the improvement of the performance of existing entities, i.e. to solve improvement problems. Hevner [Hevner, March et al. 2004] states that design science addresses research through the construction and evaluation of artifacts designed to meet the identified business need. In other words, the ultimate objective of research in these sciences is to develop appropriate and reliable knowledge to be used in creating solutions to problems. Hevner [Hevner, March et al. 2004] states that the goal of design science is utility, i.e., the construction and evaluation of generic means–ends relations [Winter 2008]. Research on the basis of the paradigm of the design sciences is characterized as follows by Van Aken [Aken van 2004]:

- research questions being driven by an interest in field problems;
- an emphasis on the production of prescriptive knowledge, linking it to artifacts, providing the key to solving field problems;
- the justification of research products mainly based on pragmatic validity (do the actions based on this knowledge produce the intended outcomes?).

This kind of research can be conducted in any discipline, but usually it is only considered as 'mainstream' research in design science, i.e. in a discipline concerned with 'how' as well as 'what' questions. The proposed research in this thesis is based on the Information Systems Research Framework (cf. Fig. 1.1) [Hevner, March et al. 2004; Hevner 2007]. The framework is shown in three columns.

The environment is described in the leftmost column, which includes the goals, tasks, problems, and opportunities that define business needs as they are perceived by people within the organization. The middle column describes how design science research is conducted in two complementary phases. Behavioral science addresses research through the development and justification of theories that explain or predict phenomena related to the identified business need. In contrast, design science addresses research through the construction and evaluation of artifacts designed to meet the identified business need. The right-hand column contains the knowledge base. It provides the raw materials from and through which Information Systems research is accomplished. The knowledge base is constituted of foundations and methodologies. Methodologies provide guidelines used in the justify/evaluate phase. Rigor is achieved by appropriately applying existing foundations and methodologies.

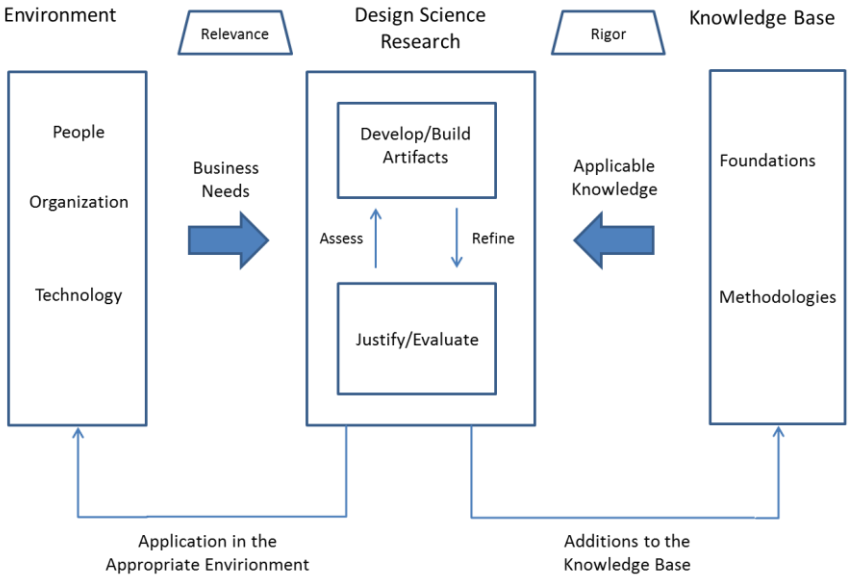


Fig. 1.1 Information Systems Research Framework [Hevner, March et al. 2004]

1.4.2 The Design Proposition

Before explaining the development and evaluation of the artifact based on Figure 1.1, the design proposition is presented first with a number of sub-questions. Take into account that the theoretical framework of DEMO is assumed to be known, see for further explanation section 3.1 and 3.2.

The artifact being designed in the context of this study can be described as follows:

A specification framework is created that enables the specification of an implementation model for the information provision in an organization. This framework is based on the realization of the essential model of the organization.

The following three sub-questions can be derived from this design proposition.

1. *What is understood by the realization of the essential model of an organization that constitutes the basis for the design of an implementable model of the information provision in the organization?*

Enterprise Engineering provides theories and methods for building construction models of an organization. The highest level construction model of an organization is called the essential model of an organization (per definition [Dietz 2008]). An essential model defines the essence of an organization in an implementation independent way. It includes also the information requirements for each actor role for which an original production act is defined. The essential model is the starting point for designing the ontological models of the business organization, the infological organization and the datalogical organization. These three organizations are understood as three aspect systems of the organization [Dietz 2006]. Facts are created in the business organization. They are derived in the infological organization and are transformed in the datalogical organization. The three mentioned ontological models constitute an integral coherent model which is considered as the lowest level construction model from which an implementable model for information provision can be designed. The process of constitution the integral model is called realization.

2. *What does an implementation model of information provision based on the realization of the essential model of the organization consist of?*

Once the integral, implementation-independent model of the organization is available, the question arises of how to move from this model to a model that can be implemented. We shall see that actor roles could be fulfilled either by subjects or by agents. A subject is understood as a human being and an agent is understood as an automated actor. It consists of a piece of software that executes acts, which are described by the actor role. Agents are found particularly frequently in infological organizations. The situation is different in business organizations because original facts simply cannot be created by agents. In the implementation model the interrelationships between subjects and agents must be elaborated.

3. *What guidelines are needed for the design of an implementation model of the organization, on which the design of an enterprise information system could be based?*

As was stated under the previous research question, a method must be developed to create an implementation model of the organization on the basis of the integral, implementation-independent model of the organization. For that reason, a specification framework should be defined. That framework makes it possible to design the implementation model in a methodical way by using the specifications that are provided by the framework.

1.5 Research Strategy

Figure 1.2 shows the Information Systems Research Framework drawn on the paper "A Three Cycle View of Design Science Research" [Hevner 2007]. Three research cycles are shown.
The *relevance cycle* provides a bridge between the contextual environment of the research project and the design science activities. An application domain consists of the people, organizational systems, and technical systems that interact to work towards a goal. Design science research often begins by identifying and representing opportunities and problems in an actual application environment. However Iivari [Iivari 2007] points out that some design science research is about potentiality: the identification of new opportunities to improve practice before any problem is recognized. So, the relevance cycle begins in an application context that not only gives the requirements for the research as inputs (e.g., the opportunity/problem to be addressed) but also defines acceptance criteria for the final evaluation of the research results. The business alignment problem as described in section 1.3 provides the criteria that must be satisfied by an acceptable solution of the specified problem. The results of the field testing, which are given in Chapter 6, 7, and 8, will determine whether further iterations of the relevance cycle are needed in this design science research project. A comparison of the problem formulation and criteria formulated in section 1.3 with the results of Chapters 6, 7, and 8, is presented in Chapter 9.

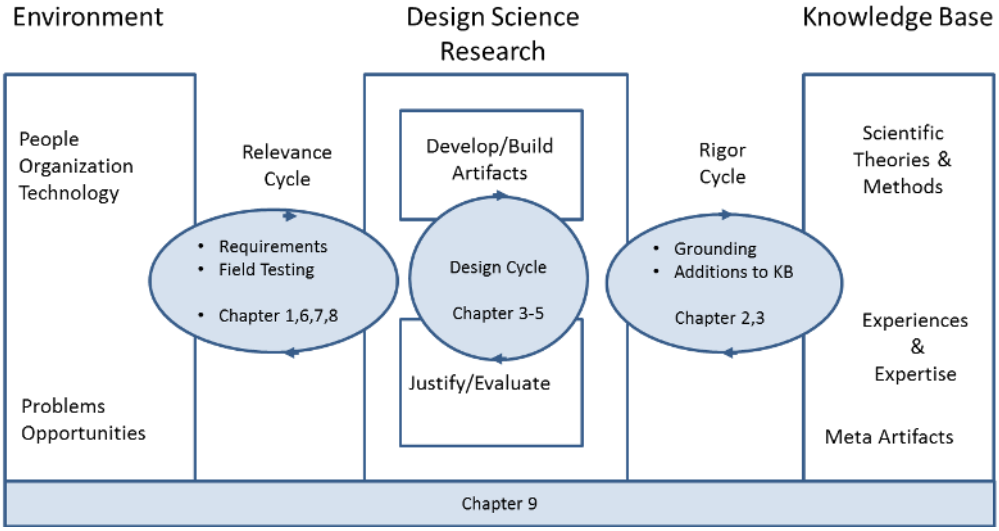


Fig. 1.2 Information Systems Research Framework (adapted from [Hevner 2007]

The *rigor cycle* connects the design science activities with the knowledge base of scientific foundations, experience, and expertise that informs the research project. The rigor cycle provides the research project with prior knowledge to ensure it is innovative. In Chapter 2, a number of theories and a method in which the design of the artifact is grounded are described as prior knowledge. It ensures that the designs produced are research contributions and not routine designs based upon the use of well-known processes. In order to design the artifact, it was necessary to begin by thinking about the constructional aspects of the infological organization of an enterprise. Chapter 3 includes a detailed analysis of the

results of this study. The results must be regarded as extensions to the original theories and methods. Therefore, they may be added to the current knowledge base.

The central *design cycle* iterates between the core activities of building and evaluating the design artifacts and processes of the research study. This cycle of research activities iterates between the construction of the artifact, its evaluation, and subsequent feedback to enable further refinements to the design. The design cycle is discussed in Chapters 3 to 5. Simon [Simon 1996] describes the nature of this cycle as generating design alternatives and evaluating the alternatives against requirements until a satisfactory design is achieved. As Iivari [Iivari 2007] states in his essay, "The essence of Information Systems as design science lies in the scientific evaluation of artifacts." Therefore, artifacts must be rigorously and thoroughly tested in laboratory and experimental situations before releasing the artifact for field testing along the relevance cycle. This calls for multiple iterations of the design cycle in design science research before contributions are output into the relevance cycle and the rigor cycle.

Yin [Yin 2003] lists five different possible research strategies: experiment, survey, archival analysis, history, and case study. He proposes first of all basing one's choice of a research strategy on the well-known series or types of questions: 'who', 'what', 'how', and 'why'. Taking account of the artifact being designed in the context of this study—namely a specification framework that is used in the designing process of an implementation model of an organization—the question types 'who' and 'what' are not applicable. The research strategy must therefore be based on the question types 'how' and 'why'. Of the five strategies in the classification of Yin, the more explanatory research strategies such as history, experiment, and case study remain possible strategies. According to Yin, there exists a further distinction between history, case study, and experiment, namely, the extent of the investigator's control over and access to real behavioral events. If there is virtually no access or control, histories are the preferred strategy. History can therefore be rejected as a research strategy here. So, only the experiment and the case study remain. The difference between experiment and case study concerns the possibility of manipulating relevant behaviors. The case study is preferred for examining contemporary events that cannot be manipulated. An experiment is performed if an investigator can manipulate behavior directly, precisely, and systematically. Experiments can be carried out either in a laboratory setting or in a field setting. To be able to make an informed decision between the two strategies, we must look back at how the artifact that is to be validated was formed. The artifact is the result of a design process. The specification framework was established on the basis of a number of scientifically grounded principles on the one hand, and the demands of the business organization on the other hand. Validation thus concerns the question of whether the framework is usable in practice for the purpose for which it was designed, namely, for the design of an implementable model of information provision. We can conclude that the introduction of an artifact in an operational environment immediately and precisely manipulates behavior. The research strategy that should be used for this research is the experimental method. This conclusion is entirely in line with what Hevner [Hevner, March et al. 2004] states. He too refers to the experimental method. He then points to the fact that the use of this framework introduces a completely new approach that has not been used in practice so far. It is not possible to observe existing cases. He notes that, the experimental method can be used to examine the created artifact in a controlled environment. In this study, we use the name 'application' instead of 'experiment'.

Concluding, three steps provide the evidence for the design of the artifact. In the first step evidence from literature is determined. See the rigor cycle (cf. Fig. 1.2). In the second step arguments based on several papers that have been discussed in various workshops of the Enterprise Engineering Working Conference [Jong de 2009; Jong de and Dietz 2010; Jong de 2011] are discussed, and in the third step three applications are elaborated. They are discussed in Chapters 6, 7 and 8.

1.6 Structure of the Thesis

In Chapter 2, the theories and methodologies in which the dissertation research is grounded are examined. In this context, Hevner [Hevner, March et al. 2004] talks about the knowledge base within the Information Systems Research Framework (cf. Fig 1.1 and Fig. 1.2). The analysis is not exhaustive; rather, it focuses on the theories that will be referred frequently.

In Chapter 3, the concept of the 'realization of the organization' is discussed. Within the enterprise engineering domain, Dietz [Dietz 2006] developed a method for revealing the essence of organization. The essence of an organization is described by different models. The original facts that are created and the (derived) facts that are asked by business actors are identified within those models. However, the constructions of the infological and the datalogical organizations are not dealt with. In this chapter, a modeling method is presented for revealing and displaying the white box model of the infological and datalogical organization.

In Chapter 4, the question about designing an implementable model of the infological organization is discussed. In addition, scientific research that has been and that is still ongoing about designing implementation models based on the essential model of an organization is presented. The main distinguishing feature of this thesis research, if compared with other studies, is that both the business organization and the infological organization are seen as social systems. Social systems are centered on human beings. Therefore, a business organization as well as an infological organization is centered on human beings.

In Chapter 5, the specification framework is positioned; it enables practitioners to formulate the implementation model of the information organization based on the integral, implementation-independent model of the organization. The framework includes a set of rules that has to be used in designing an implementation model that corresponds with an EIS. The framework has to be discerned from an architecture framework. Architecture is defined as "a consistent and coherent set of design principles that embody general requirements, where these general requirements hold for a class of systems" [Dietz 2008]. A rule is defined as "a prescribed guide for conduct or action" [Merriam-Webster 2012]. That means that a rule gives directions for designing an implementation scenario. That is completely different from a requirement that has been raised to the status of a design principle.

In Chapter 6, the first application at the Conciliation Board for Consumers is described. The Conciliation Board for Consumers is a non-profit organization with the aim to negoti-

ate quickly, less costly and easy solutions for disagreement between consumers and suppliers. The case study has been described in detail in several publications. The specification framework has been applied to design a new EIS.

In Chapter 7, the second application which has been described at a particular unit at Mprise is described. This unit provides a standard training program to the customers of Mprise. The unit has implemented a supporting EIS. Based on the specification framework the EIS has been analyzed in depth.

In Chapter 8, a third application is described. The experiment has been made for a group of distributors of flower bulbs. Based on an implementation independent construction model, a part of the implementation model has been designed.

In Chapter 9, we reflect on the applications and determine to what extent the objectives of this study were achieved, based on the research questions.

PART II

Theoretical Foundations

2 Grounding Theories

2.1 Introduction

In this chapter, more specifically in sections 2.2, 2.3, and 2.4, we will briefly elaborate on the main theories that form the basis for this study. The Information Systems Research Framework of Hevner [Hevner, March et al. 2004] places these theories in the knowledge base (cf. Fig. 1.1). Dietz, in his farewell lecture at Delft University of Technology [Dietz 2009], named these theories using letters of the Greek alphabet. In this study I will continue to use the same letters. The first theory discussed is the φ theory (the Greek letter phi is pronounced like FI, which are the initial letters of Fact and Information). We then continue by discussing the τ theory (the Greek letter tau is pronounced like TAO, which are the initial letters of Technology, Architecture, and Ontology), and lastly the ψ theory is discussed (the Greek letter psi consists of the initial letters of Performance in Social Interaction). These theories form the grounding of a research school in the field of enterprise engineering that was set up by Dietz [Dietz 1990; Dietz and Widdershoven 1991; Dietz 2001; Dietz 2006; Dietz 2009; Dietz, Albani et al. 2010] in the early 1990s.

2.2 Summary of the φ theory

The φ theory is a theory about the conceptualization of factual knowledge. The theory is rooted in semiotics, in ontology, and in logic [Dietz 2009]. Within the discipline of semiotics, the semiotic triangle is represented in Figure 2.1 [Morris 1938; Stamper 1973; Stamper 1996].

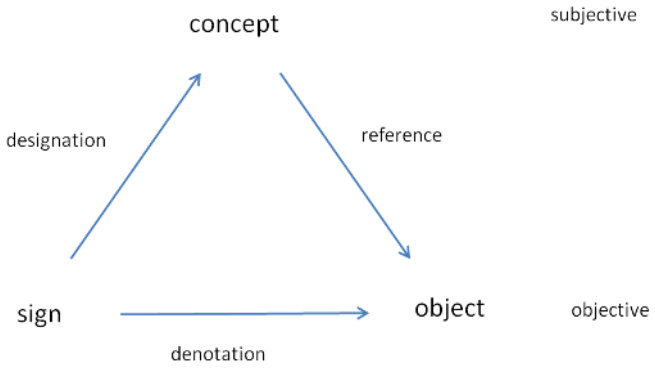


Fig. 2.1 The semiotic triangle [Dietz 2006]

It shows how people use symbols as representations of objects in order to be able to communicate about these objects without the objects being present. The elementary no-

tions within the diagram are ‘sign’, ‘object’, and ‘concept’. A thought (concept) is in the mind of a subject (i.e. a human being), the term or symbol (sign) designates the concept and the real thing (object) is the object referred to by the concept and denoted by the sign.

The first thing we would like to mention regarding the object is that the object should be considered an identifiable thing. It is for example a person, a car, a sales agreement, a rental contract, etc. Although the object is not always something you can see (how could you see a sales agreement?), the object is still present and by definition a thing in the objective world. It can be detected through its properties. Quite apart from the presence in the objective world, an image of the object exists in the mind of the observer. Most communication between human being refers to common images of objects. An image of an object that is kept in mind is called a ‘concept’. Accordingly, the concept is a thing in the subjective world. The question then arises of how to communicate with others about a concept that is part of the subjective world. The answer to this question is that people are inclined to have a concept that reflects the concept that other people have of the same object. Dietz speaks in this context of the “human social condition” [Dietz 2006]. He concludes that “the notion of subjectivity is a highly inter-subjective notion of subjectivity”. Communication between people about a concept is impossible if no ‘signs’ are used. Signs are sounds, words, diagrams, drawings or other symbols that must be understood as the form of a concept. The form of a concept must be separated from its physical substrate. The physical substrate in itself is not so important. It is not part of the sign, but just its carrier. For example, the sign ‘thesis’ is one and the same sign, whether typed into a Microsoft Office Word document with the aid of a PC or written with a pen on a piece of paper; it is only shown in two different ways, or substrates [Liu 2000; Stamper, Liu et al. 2000].

A question that now arises is about the relationship between object and information. Information refers to an object. Information can be seen as the inseparable duality of a sign and the concept it designates. The sign is called the form of the information and the concept is called the content of the information. In this thesis the term ‘document’ is used to refer to the form of an information item.

The second thing to mention regarding the object is that the term ‘object’ is very broadly defined. It covers not only physical objects (a tree, machinery, a clock, a piece of furniture, etc.) but also abstract objects (a lease contract, a project, a loan, a marriage, etc.). In this context, we need to address the philosophical position that is considered in this study. Searle [Searle 1995] makes a distinction between the position of the objectivist, the subjectivist and the constructivist. These positions are not elaborated on. In this study the position of a constructivist is taken. Dietz [Dietz 2006]: “The constructivists agreed with the subjectivists that there is no absolute objective reality (as the objectivists believe), but they believe that there is instead a kind of semi-objective reality that they call an inter-subjective reality. It is built and continuously adapted through the negotiating and achieving social consensus among subjects”. An example of this reality is a leasing agreement. A leasing agreement is an abstract object which is the result of negotiations between two subjects. The agreement exists on the moment that *both* parties consider it as an agreement.

The semiotic triangle does not only apply to individual concepts but also to generic ones. Dietz [Dietz 2009]: “A generic concept is called a *type* and a generic object is called a *class*. A class is the *extension* of a type, and a type is the *intension* of a class. The notion of type can best be understood as prescription of form, where form is some collection of (relevant) properties. If an object *conforms* to the prescription of form of a type, then its refer-

encing concept is said to be an *instance* of the type. Next, an object is said to be a *member* of the class that is the extension of the type to which it conforms (cf. Fig. 2.2)".

This can be illustrated by the example of a lease transaction. There is a class of closed lease agreements that are (semi-objectively) observable for everyone. These lease agreements have features in common, which are shown in the set of relevant properties. The intension of this class of lease agreements is the type 'lease contract'. A particular, semi-objectively observable lease agreement, denoted as the object in Fig. 2.2., conforms to the type 'lease contract'. A particular contract, a concept, is called an instantiation of the type 'lease contract'.

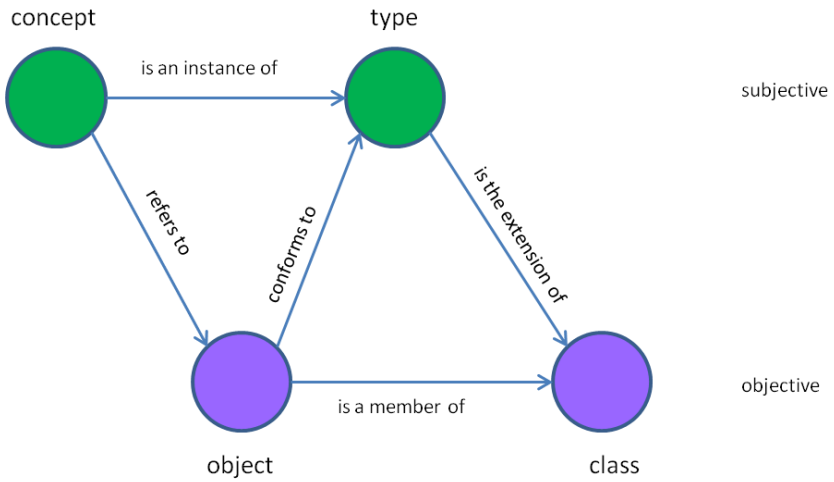


Fig. 2.2 The ontological parallelogram [Dietz 2006]

The representation of the concept, denoted as the 'sign' in Fig. 2.1, is not shown in Fig. 2.2 since the sign is not relevant from the ontological perspective. Because, ontology is about the essence of things, not about how we name them. The ontological parallelogram will be the basis for developing the ontology of a world. At any moment a world is in a particular state, which is defined as a set of objects; these objects are said to be current during the time that the state prevails. The world of an organization can be understood as an example of the mentioned world.

This study concerns the operation of organizations. The operation of an organization leads to a transition of the 'world of the organization'. A transition is defined as a state change of the world at a particular point in time. Dietz [Dietz 2006] argues in this context that "the state of the object world reflects the effects of the production acts that are performed by the elements of the system". Dietz considers the organization as a system. That is based on the system definition of Bunge. [Bunge 1979] Dietz expands the definition of Bunge with the addition that a system has a definite production: "By this we mean what is brought about by the elements of the system once they are activated through their structural links".

Organizations are complex. A general approach to examine a complex system is to build a model of this system. For our definition of a model, we refer to Apostel [Apostel 1960]: Any subject using a system A that is neither directly nor indirectly interacting with a sys-

tem B, to obtain information about the system B, is using A as a model for B. According to this definition, a system has to be considered a model if the system is used as a model. There are three different categories of systems that can be distinguished; namely concrete systems, symbolic systems, and conceptual systems. The relations between these categories are shown in Figure 2.3. The conceptual system is considered a conceptual model of the concrete system by means of conceptualization, and the conceptual system is considered a conceptual model of the symbolic system by means of interpretation. Conversely, the concrete system is considered a concrete model of the conceptual system through implementation of the conceptual system. The symbolic system is considered a symbolic model of the conceptual system after formulation of the conceptual system.

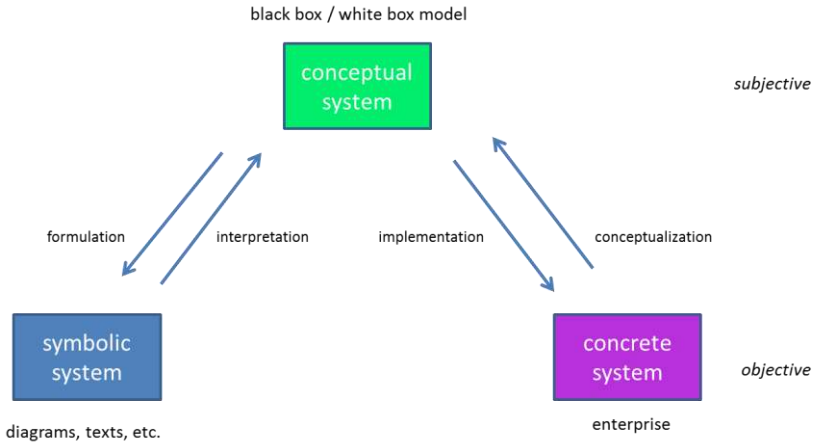


Fig. 2.3 The model triangle [Dietz 2006]

Dietz explains that there are two different types of conceptual models of a concrete system, namely a black box model and a white box model. A black box model is defined as a conceptual model of the function of a concrete system. A white box model is defined as a conceptual model of the construction of concrete system. A white box model is adequate for the purpose of building or changing a concrete system. It is therefore the dominant type of model in all engineering sciences and therefore also in this study. Both concepts, black box modeling and white box modeling, are elaborated in the next section.

2.3 Summary of the τ theory

For an extensive description of the τ theory we refer to Dietz [Dietz 2006; Dietz 2008]. The τ theory is a theory about the development of artifacts. Since an enterprise can also be considered as an artifact, the de τ theory should be viewed as a theory for developing organizations. By developing systems, as organizations, we understand the bringing about of a new system or of changes to an existing system. The system development process comprises all activities that have to be performed in order to arrive at an implemented system. In this section we will present the nature of this process, which will lead to the conception of a generic model of the system development process (cf. Fig. 2.4). Any development process concerns two systems, which we call the using system and the object system. By the object system (OS) is understood the system which is going to be developed. By the

using system (US) is understood the system that is going to use the services (the functionality) of the object system. As an explanatory example of the US, we take the sales department of an enterprise. One of the services that the US takes from the OS is the monthly turnover of the enterprise. As an explanatory example of the OS we take a sales information system. One of the monthly services it produces is the sum of a number of numerical values. This sum is interpreted by the US as the monthly turnover.

We will divide the complete system development process in three major sub processes or phases: design, engineering, and implementation.

Let us discuss the design phase first. Two major activities can be distinguished in the design of the OS. The first one is called 'function design' and the second one is called 'construction design'. Function design starts from the construction of the US and ends with the function of the OS. This means that the specified function of the OS does not contain any information about or even any clue to the construction of the OS. In other words, the function of the OS must be specified fully and only in terms of the construction of the US. Construction design is quite different from the function design. It starts with the specified function of the OS and it ends with the construction of the OS. Construction designers have to bridge the mental gap between function and construction. This is nothing less than establishing a correspondence between systems of different kinds: the system kind of the US and the system kind of the OS. The result of the activity function design is a functional model of the OS, which by nature is a black box model. It contains all functional specifications for the OS to be built. They include the specifications for the interface through which the services of the OS are provided to, or acquired by, the US, next to the specifications of the services themselves.

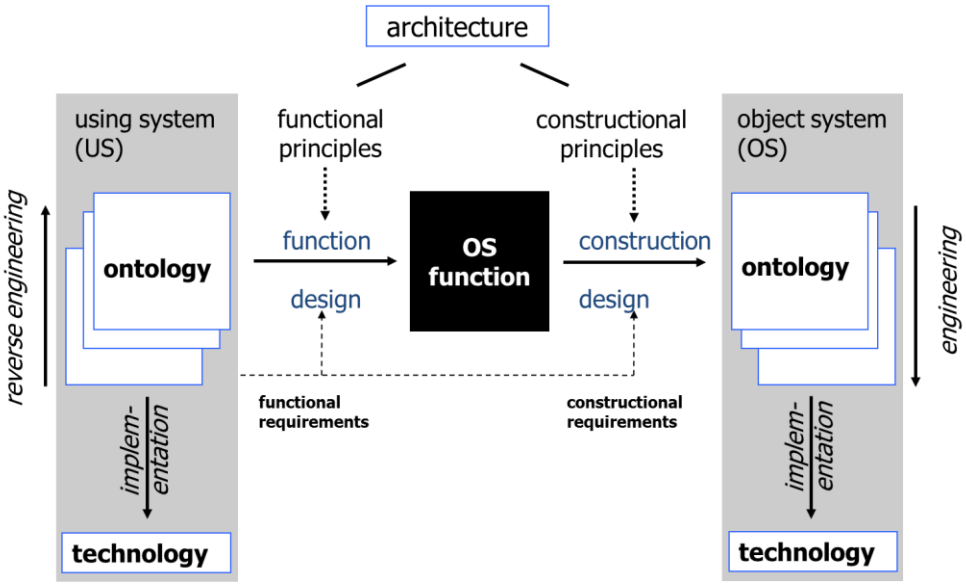


Fig. 2.4 Generic System Development Process [Dietz 2006; Dietz 2008]

The primary input for the activity function design is the set of functional requirements, provided by the US. They need not be similar to the functional specifications, as contained

in the functional model of the OS, for several reasons. Firstly, requirements may be unfounded. Secondly, both the functional and, particularly, the constructional specifications must be implemented feasibly, given the available technology and budget.

The entire design of a system is understood as a process of alternating analysis and synthesis steps. An analysis step is one in which the problem becomes better understood; a synthesis step is one in which the solution becomes clearer. In this context Christopher Alexander should be mentioned. He wrote one of the most consistent and influential contributions to the methodology of design [Purgathofer 2006]. In his 'Notes on the Synthesis of Form' [Alexander 1964], he states that the actual process of designing is not one (large) function design step, followed by one (large) construction design step, but rather a sequence of (small) alternating analysis and synthesis steps. In an analysis step, a better understanding is achieved of the requirements of the user system (including those enforced by the functional principles). This improved understanding results in extending or improving the black-box model of the object system. In a synthesis step, a better understanding is achieved of how the system could be created. This results in extending or improving the specifications regarding the construction and operation of the object system.

So, in general, designing includes negotiation, in order that the end result is a balanced compromise between (reasonable) requirements and (feasible) specifications. This holds, mutatis mutandis, also for the construction design phase. The constructional requirements which are provided by the US have to be understood as the primary input for this phase. They are usually called non-functional requirements. Constructional requirements regard the performance characteristics with which the OS is going to provide its services.

Subsequently, the phases engineering and implementing are discussed. Both ontology and technology play an important role now. Then ontology or the ontological model of a system is a model of its construction that is completely independent of the way in which it is implemented. By definition, it is the highest –level constructional model of a system.

The engineering of a system is the activity in which a series of constructional models are produced. Every model is fully derivable from the previous one and the available specifications. Contrary to designing, engineering is rather a matter of craftsmanship than of creativity. Engineering starts from the ontological model and ends with the implementation model. For that reason, engineering is also called implementation designing. By implementing a system is understood the assignment of technological means to the constructional elements in the implementation model. Once implemented correctly, the system can be put into operation.

The concept of enterprise architecture has been introduced to avoid giving designers and engineers excessive design freedom [Hoogervorst 2004; xAF 2004; Hoogervorst and Dietz 2005; Dietz and Hoogervorst 2007; Op 't Land and Proper 2007; Dietz 2008; Hoogervorst and Dietz 2008; Op 't Land 2008]. After all, unbounded design freedom has the consequence that systems thus developed would typically be suboptimal in use, maintenance, and costs. In the development process, the interests, concerns, and objectives of all stakeholders have to be taken into account. They restrict the design freedom of a class of object systems. This has been the starting point for the developers of the Extensible Architecture Framework, who define architecture conceptually as a normative restriction of the design freedom and operationally as a consistent and coherent set of design principles that embody general requirements, where these general requirements hold for a class of systems

[xAF 2004]. These requirements consist of policies, guidelines, standards, and preconditions [Davenport, Hammer et al. 1989; Tapscott and Caston 1993]. They can be divided into functional and constructional requirements. Constructional principles have to be taken into account when developing both the ontological model and the implementation models of an OS.

An example of an instantiation of the GSDP is the process for developing an organization. The organization wants to contribute to the goals of the stakeholders, otherwise the organization has no *raison d'être*. The contribution to the stakeholders' goals can be found in the function model of the OS. Designing the construction model of the organization begins with the function model and ends with the highest level implementation independent construction model of the organization. From this ontological model the implementation model can be engineered through the implementation design process. The operational organization is then brought into being by assigning technology to components in the implementation model.

2.4 Summary of the ψ theory

The ψ theory [Dietz 2006] consists of four axioms, viz. the operation axiom, the transaction axiom, the composition axiom, and the distinction axiom, as well as the organization theorem. In this section, these axioms and the organization theorem are elaborated briefly.

The *operation axiom* states that the operation of the organization is constituted by the activities of actors, which are elementary chunks of authority and responsibility fulfilled by human beings. Actors perform two kinds of acts: production acts, or P-acts for short, and coordination acts, or C-acts for short. These acts have definite results, namely production facts and C-facts, respectively. By performing P-acts, actors help create goods or services or information or data that are delivered to other actors. A P-act is either material or immaterial. Examples of material acts are the manufacturing and storage of goods and transportation. Examples of immaterial acts are the judgment by a court condemning someone, granting an insurance claim and selling goods. By performing C-acts, actors enter into and comply with commitments towards each other regarding the performance of P-acts. A C-act is defined by its proposition and its intention. The proposition consists of a P-fact, e.g. 'Purchase order #200 is delivered' and a delivery date. The intention represents the purpose of the performer; examples of intentions are 'request', 'promise', and 'decline'. The consequence of performing a C-act is that both the performer and the addressee of the act become involved in a commitment regarding the P-act being referred to.

The *transaction axiom* states that coordination acts are performed as steps in universal patterns. These patterns, also called transactions, always involve two actor roles, i.e. two chunks of authority and responsibility. They are aimed at achieving a particular result, the P-fact. Figure 2.5 shows the standard transaction pattern. A transaction evolves in three phases: the order phase (O-phase for short), the execution phase (E-phase for short), and the result phase (R-phase for short). One of the two partaking actor roles is called the initiator, the other the executor of the transaction. In the order phase, the initiator and the executor seek to reach consensus about the P-fact that the executor is going to create as well as the intended time of the creation. In the execution phase, the executor creates this P-fact. In

the result phase, the initiator and the executor seek to reach consensus about the P-fact that is actually produced as well as the actual time of creation (both of which may differ from the request). Only if this agreement is reached will the P-fact become existent. The request->promise->execute->state->accept path in Figure 2.5 is called the basic pattern; it is the course that is considered when the initiator and the executor consent every time. However, they may also dissent. There are two states from which this may happen, namely the 'requested' and 'stated' states. Instead of promising, an actor may respond to a request by declining it, and instead of accepting, one may respond to a statement by rejecting it. These responses put the process in the 'declined' and 'rejected' states respectively. These states are indicated by a double disk in Figure 2.5, meaning that they are discussion states. If a transaction ends up in a discussion state, the two actors must 'sit together', discuss the situation at hand and negotiate how to get out of it. The possible outcomes are a renewed request or statement (probably with a modified proposition) or a failure (quit or stop).

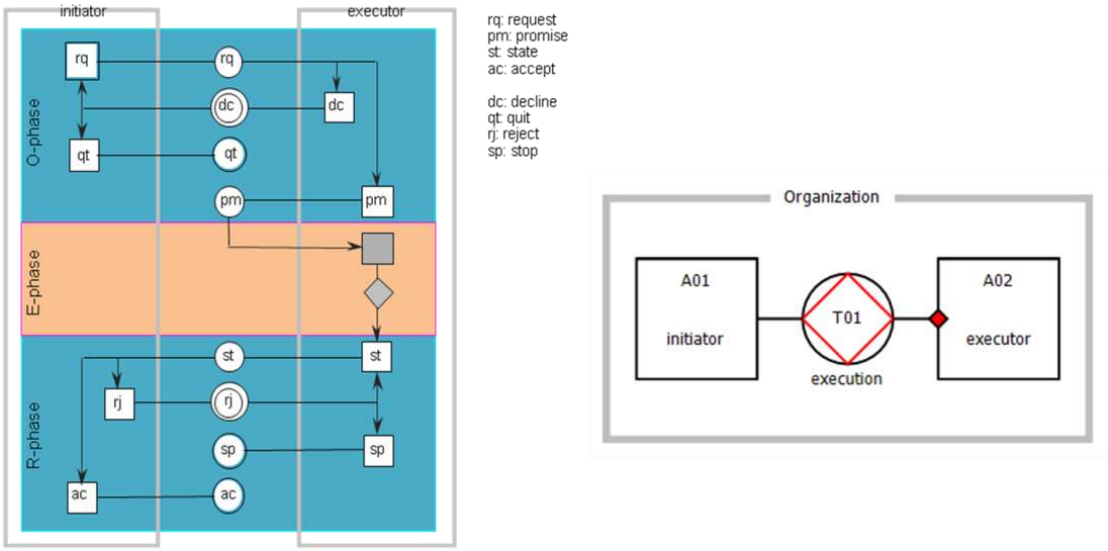


Fig. 2.5 The standard transaction pattern elaborated (left) [Dietz 2006] and in DEMO notation (right)

The *composition axiom* states that every transaction is either enclosed in some other transaction, a customer transaction of the organization under consideration, or a self-activation transaction. The question dealt with by this axiom concerns the interrelationship of the transactions and their resulting production facts. Let us consider the assembly of a bike as an example. The construction of a bike is not one atomic act but an assembly of a number of (atomic) parts. One atomic part cannot be assembled. A bike can be conceived of as a tree structure of parts, both atomic parts and subassemblies. The creation of each part is the outcome of a successful transaction (e.g. manufacturing or purchasing). The logical sequence of the component structure puts restrictions on the sequence in which all these transactions are executed.

The *distinction axiom* states that there are three distinct human capabilities playing a role in the operation of actors, called performa, informa, and forma. These capabilities are recognized in both kinds of acts that actors perform.

Let us first look at the P-act. The forma capability is the human capability to carry out document-related actions, such as archiving, transforming, fetching, etc. These are all ac-

tions in which the content of the documents is of no value. Actors which use the forma capability to perform P-acts are called datalogical actors or D-actors for short. The informa capability is the human capability to carry out intellectual actions, such as reasoning, computing, remembering, and recalling knowledge, etc. These are all actions by which the content of data or documents, irrespective of the form, is of value. Actors which use the informa capability to perform P-acts are called infological actors, or I-actors for short. The performa ability is the human capability to carry out new, original actions, such as decisions, judgments, etc. The performa ability is considered as the essential human capability to do business, of any kind. Actors which use the performa ability to perform P-acts are called ontological actors or O-actors for short.

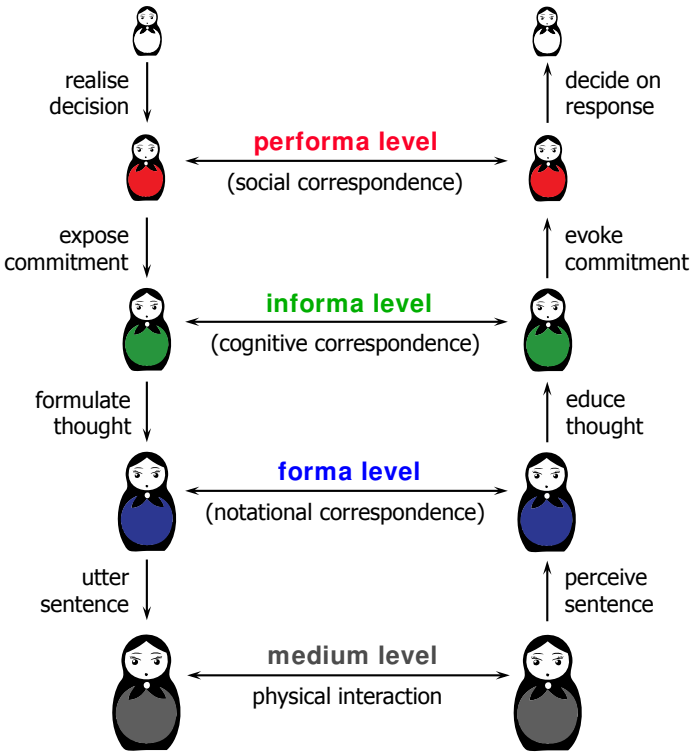


Fig. 2.6 ψ theory: coordination [Dietz 2006; Dietz 2013]

Let us next look at the C-act (cf. Fig. 2.6). By performing C-acts, actors enter into and comply with commitments towards each other with respect to the performance of P-acts. The effect of performing a C-act is that both the performer and the addressee of the act become involved in a commitment concerning the P-act being referred. That commitment is a result of a performative exchange that can only be made by actors in their performa ability. However, the only way for the performer of the C-act to reveal its commitment and to make it knowable to the addressee, is to make use of its informa ability, followed by the inducement in the mind of the addressee of an equivalent thought, by means of its informa ability. The intellectual understanding between the two actors comes into existence by means of an informative exchange. Expressing a thought can only be done by formulating it in a document in some language, and at the same time uttering it in some form, such as speaking or writing. Significational understanding between the two actors only comes into

existence by means of a formative exchange by actors in their forma ability. That means that the performer has to use a language that is known to the addressee. So both the performer and the addressee shift between abilities several times during a C-act. Concluding, O-actors, I-actors and D-actors only differ with regard to the kind of production act.

The last part of the ψ theory that we would like to explain is the *organization theorem*. It states that the organization of an enterprise is a social system that is constituted as the layered integration of three homogeneous systems: the B-organization, the I-organization, and the D-organization. The D-actors in the D-organization support the I-actors of the I-organization, while the I-actors in de I-organization support the B-actors of the B-organization (cf. Fig. 2.7).

Although in non-published papers a B-actor could be an O-actor, an I-actor or a D-actor [Dietz 2013], we consider in this study that the B-organization only contains O-actors. All three systems are called aspect systems of the overall organization of the enterprise. A system can be considered from two different perspectives, namely from the function perspective or from the construction perspective, see section 2.3. Therefore, it is the function of a supporting system, e.g. the I-organization that supports the construction of the supported system, e.g. the B-organization. Integration between the three organizations is established through the cohesive unification of human beings. Let us elaborate this point in more detail, taking the I-organization as the starting point. From the functional perspective the I-organization provides information services to the B-organization, i.e. to O-actors. However, how does an O-actor actually receive information from an I-actor? The answer is given by the distinction axiom. The subject who fulfills the O-actor role is able to take on his/her informa shape for initiating an infological transaction with an I-actor in order to obtain the requested information [Dietz 2006; Dietz 2008].

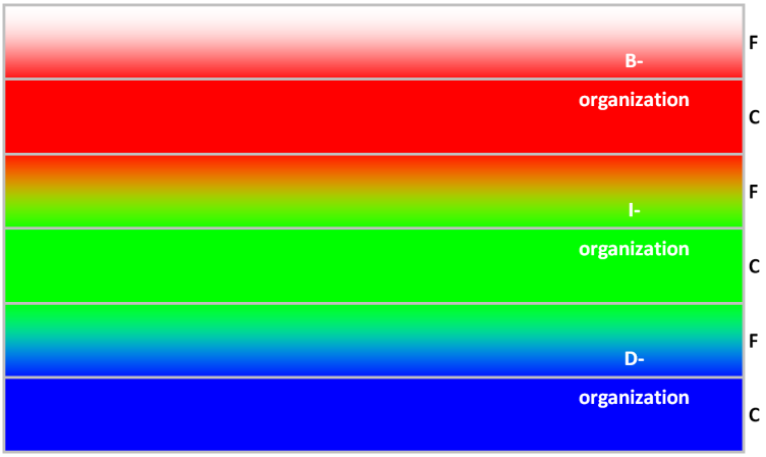


Fig. 2.7 The layered integration of the organization of an enterprise [Dietz 2006]

3 Realizing the Organization

3.1 Introduction

In section 2.2 we already indicated that a system can be considered from the function perspective, as well as from the construction perspective. In the existing literature both perspectives are used when describing an enterprise information system. The function perspective on an information system usually covers the entire lifecycle of the information provision system. Choo [Choo 2008] writes: "the management of information processes is seen as a strategic advantage that affords four kinds of benefits to an organization: (i) a reduction of costs, (ii) a reduction of uncertainty or risks, (iii) added value to existing products and services and (iv) the creation of new value through the introduction of new information-based products and services". Other researchers in the field of information management promote a construction view of information management 1990s [Davenport 1993; McGee and Prusak 1993; Choo 2002; Detlor 2004; Robertson 2005; Karim and Hussein 2008; Detlor 2010]. For example, Wilson [Wilson 2003] defines information management as "the application of management principles to the acquisition, organization, control, dissemination and use of information relevant to the effective operation of organizations of all kinds." Similarly, Choo [Choo 2002] defines information management as the management of processes that acquire, create, organize, distribute, and use information. It is assumed that the operation of an organization can only be explained by the construction of the organization.

Our research is grounded on the knowledge base [Hevner, March et al. 2004] that has been found within the domain of enterprise engineering as defined in [Dietz and Hoogervorst 1213].

In section 3.2 the usual way of modeling the essence of an organization within the enterprise engineering domain is discussed. The methodology used is known as the DEMO methodology[1] which has been developed by Dietz [Dietz 2006]. The organization of an enterprise is understood as a social system that is constituted as the layered integration of three aspect organizations: the B-organization, the I-organization, and the D-organization. As DEMO is focused on modelling the essence of an organization, current DEMO models relate to the construction of the B-organization in an enterprise. Developing the ontological models of the I-organization and the D-organization is called the realization of the organization. This chapter focuses on the realization of the organization. Section 3.3 deals with the ontological model of the B-organization which is distinguished from the essential model in some details. This model is used in section 3.4 as the starting point for the development of the ontological model of the I-organization. Subsequently, based on the model of the I-organization the ontological model of the D-organization is elaborated in section 3.5.

[1] DEMO stands for Design & Engineering Methodology for Organizations

This chapter ends with some remarks about the File Management organization of the Pizzeria in section 3.6 and some conclusions in section 3.7.

The use of the term 'organization' with regard to an ontological model requires clarification. By 'organization' is meant a 'scope of interest'. It can coincide with the boundary of an enterprise, but it can also regard a part of an enterprise, or can exceed the boundary of an enterprise.

3.2 The organization

Dietz distinguishes four different aspect models for the ontological model of an enterprise, which is called its essential model [Dietz 2006]. The first aspect model, the Construction Model (CM), specifies the transaction kinds identified and the associated actor roles, as well as the information links between the actor roles and the information banks; in short, the CM specifies the construction of the organization. It is the most concise model. The CM is split into two parts: the active part, the Interaction Model (IAM), and the passive part, the Interstriction Model (ISM). The IAM of an organization consists of the transaction kinds and the recognized actor roles that participate as initiator or executor. The transaction kinds with the associated result kinds are defined in the Transaction Result Table (TRT). This table is understood as a part of the IAM. The ISM on the other hand shows the passive system structure i.e. the information links between actor roles and banks. The contents of the banks are defined by the Bank Contents Table (BCT) which is part of the ISM. The second aspect model is called the Process Model (PM*)*. The PM specifies the state space and the transition space of the coordination world (C-world). Every transaction kind in the CM includes the basic pattern, standard pattern and the cancellation pattern. In addition, the PM contains causal and conditional relationships between transactions. A set of causally related transactions is called a business process. The third aspect model, the Action Model (AM), specifies the action rules that serve as guidelines for the actors in dealing with their agendas. The AM contains one or more action rules for every agendum kind. The AM is the most detailed and comprehensive aspect model. At the ontological level of abstraction, there is nothing below the AM. The fourth aspect model, the Fact Model (FM), specifies the state space and the transition space of the production world (P-world): the object classes and fact kinds, the results kinds, and the ontological coexistence rules. The FM may be viewed as the detailing of one part of CM, namely the contents of the information banks (especially the production banks).

Figure 3.1 exhibits the CM of a simple Pizzeria where only pizzas are made and delivered on the basis of orders received at the counter. The prices of the different pizza kinds are adjusted periodically by the business process that consists of O-T04 and O-T05. The transaction results are specified as follows:

- O-R02: [order] has been completed
- O-R03: [order] has been prepared
- O-R04: price assignment in [period] has been controlled
- O-R05: [price_assignment] has been done
- O-R06: [order] has been paid

The Pizzeria uses a discount stack for determining the price to be paid. This stack is only applied to orders from companies. For example, if a customer places an order for pizzas, and he has already ordered 10 pizzas this year, a discount rate of five percent will be applied to each new order. For every additional 10 ordered pizzas, the discount rate increases. The stack for determining the price is listed in the external production bank PB90.

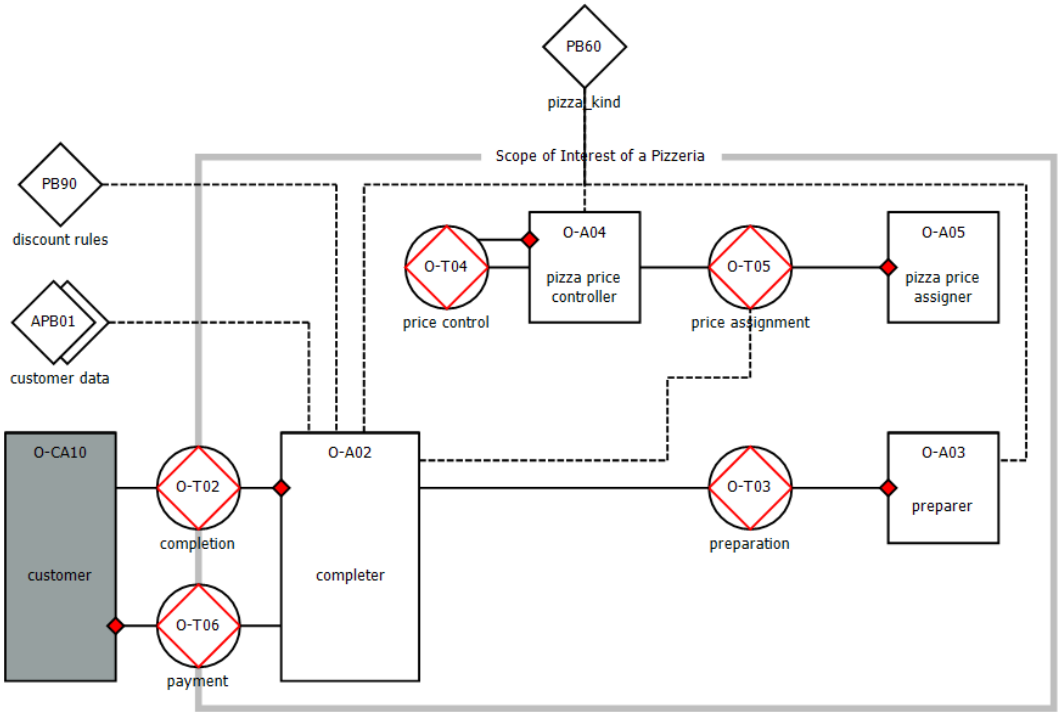


Fig. 3.1 Construction Model of Pizzeria

Customer O-CA10 asks O-A02 to complete a pizza order. In order to perform O-T02, O-A02 needs the result of the enclosed O-T03 from the preparer O-A03. Both actors produce original production acts (this is indicated by the red color in the transaction symbol). All P-facts which are created by actor Ai are stored in the bank Ti. This bank also contains the C-facts in transactions Ti. By means of information links, one models that an actor role requires access to a bank. The information links are represented by dashed lines. Figure 3.1 exhibits a model that shows that O-A02 uses facts from fact banks that correspond to the transactions O-T02, O-T03 (there is also an information link below each solid line in the construction model) and O-T05. Besides, P-facts are used from the external fact banks PB60, PB90 and APB01. O-A02 also uses C-facts from the fact banks corresponding to O-T02 and O-T03.

Figure 3.2 shows the AM of the Pizzeria. An action rule is a guideline for an actor for responding to an agendum. The execution of an action rule results into the performance of one or more coordination acts. The decision about which coordination act has to be performed is dependent on the state of the C-world and the P-world. The decision is taken in the response part of an action rule. The assess part is divided in three sections, corresponding with the three validity claims of Habermas [Habermas 1981; Dietz 2006]: the claim to

justice, the claim to sincerity, and the claim to truth. Assessing justice comes down to validating the authorization of the performer. In our case it means that a preparer acknowledges the authority of a completer to be an initiator of O-T03. Assessing sincerity means validating the sincerity of the performer of the coordination act. The question is, for example, can the completer trust that a particular customer is sincere in his request for a pizza. Lastly, assessing truth means validating the truth of the P-fact. It is valid if the fact exists, or if creating the fact leads to a lawful new state of the production world. For the Pizzeria this is guaranteed as long as the requested pizza kind is known.

Actor O-A02:

when	completion **for** new Order is requested **with** **the** pizza_kind **of** Order **is** a Pizza_kind **the** #ordered_pizzas **of** Order **is** a #Ordered_pizzas **the** customer **of** Order **is** a Customer	O-T02/rq
assess	*justice:* **the** Performer **of the** request **is the** customer **of** Order *sincerity:* <no specific condition> *truth:* <existence of Pizza_kind> <Price of Pizza_kind is available>	
if **then** **else**	*complying with request is considered justifiable* promise completion **for** Order decline completion **for** Order	

when	completion **for** Order is promised	O-T02/pm
assess	*justice:* **the** Performer **of the** promise **is the** completer **of** Order *sincerity:* <no specific condition> *truth:* <no specific condition>	
if **then**	*complying with promise is considered justifiable* request preparation **for** Order	

when	preparation **for** Order is stated	O-T03/st
assess	*justice:* **the** Performer **of the** state **is the** preparer **of** Order *sincerity:* <no specific condition> *truth:* <preparation of pizza is well done>	
if **then** **else**	*complying with statement is considered justifiable* accept preparation **for** Order request payment **for** Order **with** the Amount_to_pay **is an** Order_amount reject preparation **for** Order	

when payment **for** Order is stated **with** **the** Amount_ paid **is an** Amount		O-T06/st
assess	*justice:* **the** Performer **of the** state **is the** customer **of** Order *sincerity:* <no specific condition> *truth:* Amount_paid **is equal to the** Amount_to_pay	

if	*complying with statement is considered justifiable*	
then	accept	payment **for** Order
	execute	completion **for** Order
	state	completion **for** Order
else	reject	payment **for** Order

Actor O-A03:

when	preparation **for** Order is requested	O-T03/rq
assess	*justice:*	**the** Performer **of the** request **is the** completer **of** Order
	sincerity:	<no specific condition>
	truth:	< recipe of Pizza_kind is available >
if	*complying with request is considered justifiable*	
then	promise	preparation **for** Order
else	decline	preparation **for** Order

when	preparation **for** Order is promised	O-T03/pm
assess	*justice:*	**the** Performer **of the** promise **is the** preparer **of** Order
	sincerity:	<no specific condition>
	truth:	<no specific condition>
if	*complying with promise is considered justifiable*	
then	execute	preparation **for** Order
	state	preparation **for** Order

Actor O-A04:

when	price control **for** Period is requested	O-T04/rq
assess	*justice:*	**the** Performer **of the** request **is the** price controller **of** Period
	sincerity:	<no specific condition>
	truth:	<no specific condition>
if	*complying with request is considered justifiable*	
then	request	price control **for** next period
	with the requested_creation_time **is equal to** now + 1 period	
	promise	price control **for** Period
else	decline	price control **for** Period

when	price control **for** Period is promised	O-T04/pm
assess	*justice:*	**the** Performer **of the** promise **is the** price controller **of** Period
	sincerity:	<no specific condition>
	truth:	<no specific condition>
if	*complying with request is considered justifiable*	
then	**for** each Pizza_kind in Pizza kinds	

request price assignment **for** new Price_assignment

when price assignment **for** Price_assignment is stated	O-T05/st
assess *justice:* **the** Performer **of the** state **is the** price assigner **of** Price_assignment *sincerity:* <no specific condition> *truth:* <no specific condition>	
if *complying with statement is considered justifiable* **then** accept price assignment **for** Price_assignment	

when price assignment **for** Price_assignment is accepted	O-T05/ac
assess *justice:* **the** Performer **of the** accept **is the** price controller **of** Period *sincerity:* <no specific condition> *truth:* <Price_assignment is the last Price_assignment to accept>	
if *complying with acceptance is considered justifiable* **then** execute price control **for** Period state price control **for** Period	

when price control **for** Period is stated	O-T04/st
assess *justice:* **the** Performer **of the** state **is the** price controller **of** Period *sincerity:* <no specific condition> *truth:* <no specific condition>	
if *complying with statement is considered justifiable* **then** accept price control **for** Period	

Actor O-A05:

when price assignment **for** Price_assignment is requested	O-T05/rq
assess *justice:* **the** Performer **of the** request **is the** price controller **of** Period *sincerity:* <no specific condition> *truth:* <no specific condition>	
if *complying with request is considered justifiable* **then** promise price assignment **for** Price_assignment **else** decline price assignment **for** Price_assignment	

when price assignment **for** Price_assignment is promised	O-T05/pm
assess *justice:* **the** Performer **of the** promise **is the** price assigner **of** Price_assignment *sincerity:* <no specific condition> *truth:* <no specific condition>	

if	*complying with promise is considered justifiable*
then	execute price assignment **for** Price_assignment
	state price assignment **for** Price_assignment

Fig. 3.2 Action Model of Pizzeria

Figure 3.3 shows the PM of the Pizzeria. It shows the tree structures of both identified business processes and the exact way in which a transaction kind is enclosed in another one. The disk of the transaction symbol is stretched horizontally, such that looks like a sausage. One must imagine that there is an invisible and non-proportional time line from left to right. For example, the promise of O-T02 is performed after its request. Coordination acts and facts are represented respectively by small boxes and disks on the border of the transaction symbol. The production act is represented by a small grey box on the edge of the production symbol. Waiting links are represented by dashed lines.

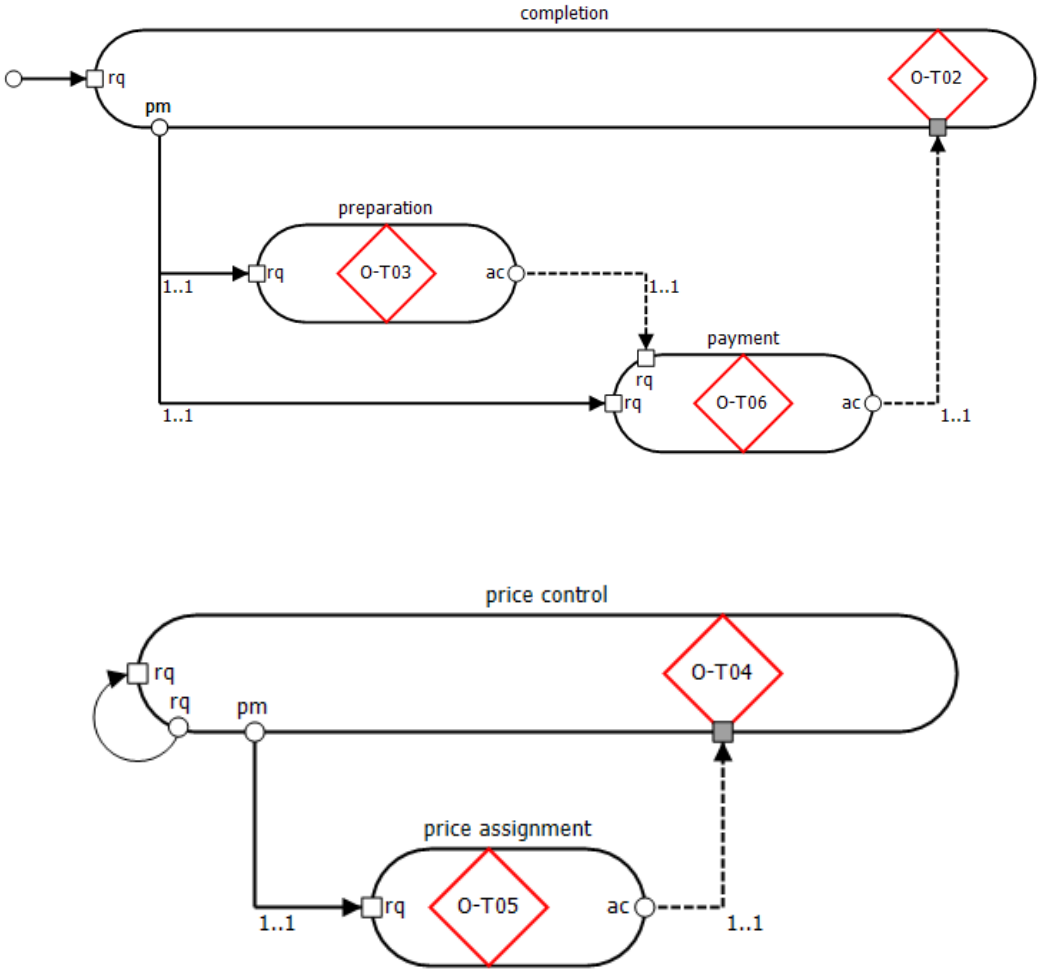


Fig. 3.3 Process Model of Pizzeria

Figure 3.4 shows the FM of the Pizzeria. The ‘roundangles’ represent object classes. The object classes ORDER, PERIOD, PIZZA-KIND, RECIPE and PRICE_ASSIGNMENT are

primal classes, also called categories. That means that they cannot be defined on the basis of other fact kinds. The other object classes are PREPARED ORDER and COMPLETED ORDER. The color grey of the category RECIPE, CUSTOMER, and PIZZA_KIND indicates that these categories are external. The instances of a dependent fact kind, also called a property type, are existentially dependent on the existence of the objects in the object class. A property type is called an attribute type if the range is a value scale. The result kinds '[order] has been prepared' and '[order] has been completed are (existentially) independent unary fact kinds. They are de results of transactions O-T03 en O-T02 respectively. Figure 3.4 exhibits that any order can become a prepared order, but only prepared orders can become completed orders.

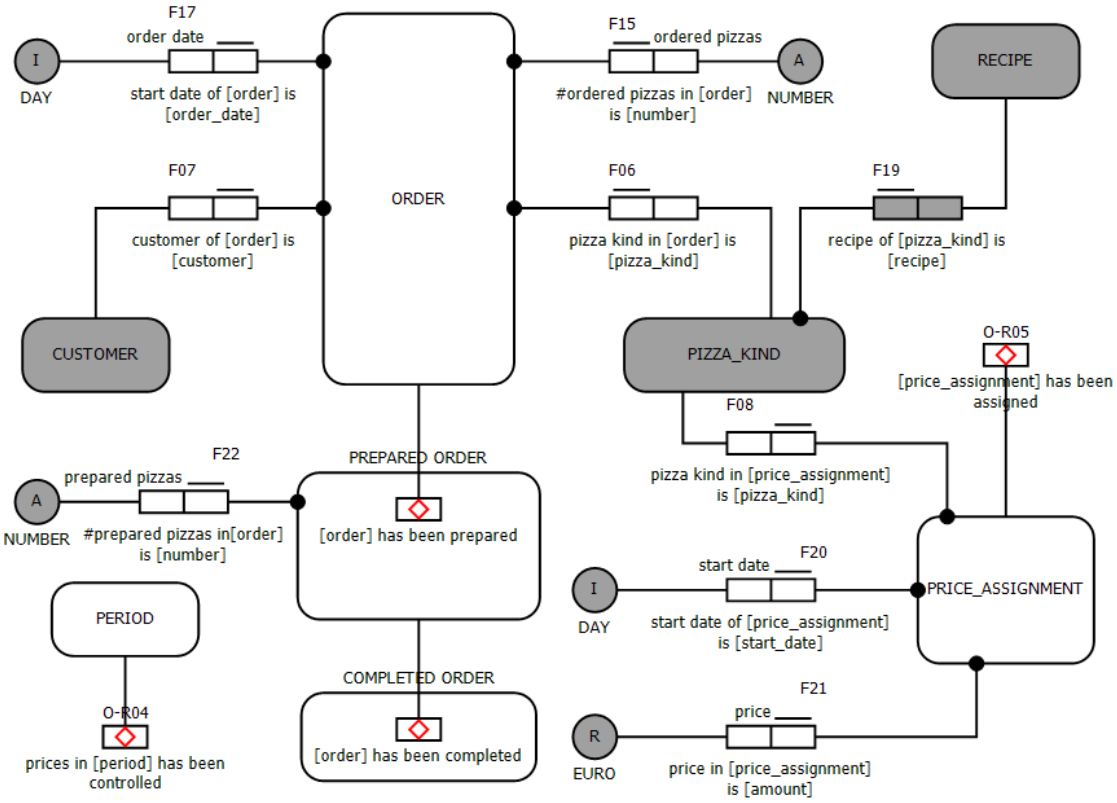


Fig. 3.4 Fact Model of Pizzeria

In the remainder of this chapter, Figure 3.1 to Figure 3.4 will be frequently referred to. These models constitute together the essential model of de Pizzeria. In the next sections we derive the integral set of ontological models of the corresponding B-organization, I-organization and D-organization. This integral set forms the basis for the development of an EIS.

3.3 The B-organization

From the CM in Figure 3.1 the interaction part of the CM of the B-organization is derived (cf. Fig. 3.5). This model differs from the CM in Figure 3.1 that it does not contain any

information links. Information links will be represented in the integrated model of the B- and I-organization as infological transactions. This will be elaborated in section 3.4. The PM in Figure 3.3 and the FM in Figure 3.4 are similar to the PM and the FM of the B-organization.

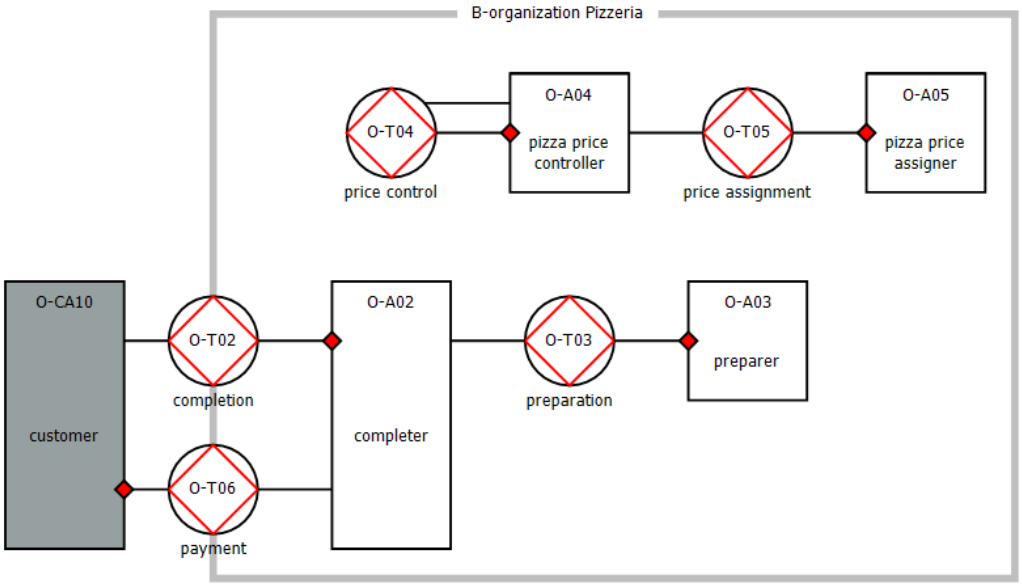


Fig. 3.5 Interaction model B-organization of Pizzeria

Roughly, there is only a slight difference between the AM of the B-organization (cf. Fig. 3.6) and the AM of the essential model that is exhibited in Figure 3.2. Although in the assess parts of the action rules of the essential model the objects for infological production acts are defined, a reference to remembering the ontological P-facts is missing in the AM. Therefore, to the AM of the B-organization (cf. Fig. 3.6) the agendum 'when preparation of [order] is accepted' is added. This agendum indicates that the P-fact of O-A03 has been accepted by the initiator of the transaction. From that point in time the P-fact of O-A03 comes into existence.

Actor O-A03:

	when preparation **for** Order <u>is requested</u>	O-T03/rq
assess	*justice:* **the** Performer **of the** <u>request</u> **is the** completer **of** Order *sincerity:* <no specific condition> *truth:* < recipe of Pizza_kind is available >	
if	*complying with request is considered justifiable* **then** <u>promise</u> preparation **for** Order **else** <u>decline</u> preparation **for** Order	

	when preparation **for** Order is promised	O-T03/pm
	assess *justice:* **the** Performer **of the** promise **is the** preparer **for** Order *sincerity:* <no specific condition> *truth:* <no specific condition>	
	if *complying with promise is considered justifiable* **then** execute preparation **for** Order state preparation **for** Order	

	when preparation **for** Order is accepted	O-T03/ac
	assess *justice:* **the** Performer **of the** accept **is the** completer **for** Order *sincerity:* <no specific condition> *truth:* <no specific condition>	
	if *complying with acceptance is considered justifiable* **then** < P-fact must be remembered >	

Fig. 3.6 Actions rules for O-A03 in B-organization

3.4 The I-organization

3.4.1 GSDP applied to the development of an I-organization

The development process of the I-organization that supports a B-organization is considered as an instance of the GSDP (cf. Fig. 3.7).

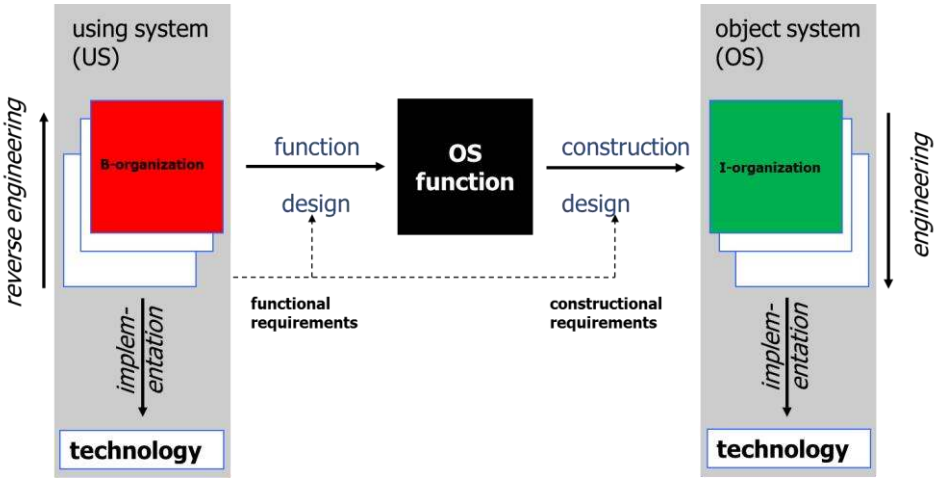


Fig. 3.7 Developing the I-organization of a B-organization [Dietz 2008]

The information needs are found in the AM of the B-organization. Dietz [Dietz 2006]: "Those who have made the effort to understand and to use the AM will discover that it provides a full account of the essential operational decisions in the enterprise". O-actors require two main functions from the I-organization, namely (1) remembering facts and (2) sharing facts. Both main functions relate to C-facts and P-facts. They are elaborated in sec-

tion 3.4.2.From a particular B-organization, concrete information services are designated for each of the main functions. The total set of information services constitutes the function model of the I-organization. From the function model the highest construction model of the I-organization is designed. This model is implementation-independent and must for this purpose be considered as the ontological model of the I-organization. It clarifies the internal construction and operation of the I-organization.

3.4.2 Determining Information Services

The question to face in this section is: how to determine a complete set of information services to support the business processes in the B-organization?

The direct relationship between the B-organization and the I-organization lies in the AM of the B-organization. See section 3.3. The AM contains service inquiries which belong to three different service types. One of these service types is related to the first main function of the I-organization, namely remembering original facts. The remaining two service types concern the second main function of the I-organization, namely sharing facts. We discuss the three different service types one by one:

1. *Remembering facts.* Coordination acts and production acts result into C-facts and P-facts, respectively. They are remembered so that they can be recalled later on if desired.
2. *Sharing agenda.* A C-fact may be an agendum for some actor. In dealing with an agendum, one or more action rules are executed by the actor. The collective agenda together waiting for processing is considered by the actor as its 'to-do list'.
3. *Sharing facts.* Assessing justice (claim to justice) comes down to validating the authorization of the performer. Assessing sincerity (claim to sincerity) means validating the sincerity of the performer of the coordination act. Assessing the truth (claim to truth) means validating the truth that the P-fact can be created. Both C-facts and P-facts may be needed in order to decide whether a specific action rule has to be executed. P-facts are distinguished in dependent P-facts and independent P-facts. Dependent P-facts correspond to fact kinds in the FM and independent P-facts correspond with result kinds in the FM.

We will discuss the three service types in detail.

Ad 1. Remembering facts. Both P-facts and C-facts have to be remembered in the I-organization. Although they are conceptually different kinds of facts, there is a relationship between C-facts and P-facts [Dietz 2006]. A C-fact assigns a status to the corresponding P-fact. So, the status of a P-fact can be 'requested', 'promised', 'declined', 'stated', 'accepted' or 'rejected'. That means, for example, that a P-fact with the status 'requested' can be differ from a P-fact with the status 'stated'. A P-fact comes into existence once it has the status 'accepted'. See Figure 3.6. That differs from a C-fact. A C-fact comes into existence immediately after its creation. With remembering the P-fact of an actor is meant the remembering of the dependent and independent P-facts which are contained in the BCT for the actor. The term result kind, which is used in the BCT, is a synonym for the independent P-fact kind. The BCT from the example of the Pizzeria is exhibited in Figure 3.8.

Object class, Fact kind, or Result kind	P-Bank
COMPLETION_ORDER	PB02
#ordered pizzas in [order] is [number]	PB02
pizza kind in [order] is [pizza_kind]	PB02
customer of [order] is [customer]	PB02
start date of [order] is [order_date]	PB02
[order] has been completed	PB02
#prepared pizzas in [order] is [number]	PB03
[order] has been prepared	PB03
PERIOD	PB04
prices in [period] has been controlled	PB04
PRICE_ASSIGNMENT	PB05
pizza kind in [price_assigment] is [pizza_kind]	PB05
start date of [price_assignment] is [start_date]	PB05
price [price_assignment] is [amount]	PB05
[price_assignment] has been assigned	PB05
PIZZA_KIND	PB60
recipe of [pizza_kind] is [recipe]	PB60
CUSTOMER	ABP01

Fig. 3.8 Bank Contents Table of Pizzeria

Ad 2. Sharing agenda. After obtaining an agendum, the actor interprets it and executes subsequently one or more action rules. Two situations have to be distinguished (cf. Fig. 3.2). Firstly, the agendum 'requested' of O-A03 is related to an object ORDER. Then, the instance of ORDER comes available for O-A03 by gaining the agendum. Secondly, it may happen that an agendum is not related to an object. This would be the case if the request is performed by an actor that starts a business process. Then, the initiator creates the object during the coordination act and gives some facts along with the coordination act. They are created by the initiator at that time as dependent P-facts of the object. The initiator is an external actor (outside the organization, see O-CA10 in Fig. 3.5) or a self-activating actor (within the organization, see O-A04 in Fig. 3.5).

Ad 3. Sharing facts. The justice conditions are facts that are contained in transaction banks of assignment transactions. These banks contain the assignments of subjects to actor roles. They are present in each organization. The sincerity conditions are kept out of scope in this study. C-facts and dependent and independent P-facts are obtained by an actor from fact banks which are inside and outside the organization boundary. These P-facts contain the state 'accepted'. Otherwise they do not exist. During the coordination act 'request' between O-CA10 and O-A02, in the Pizzeria example, an object ORDER is created. The pizza_kind, the customer, and the number of ordered pizzas are given along with the coordination act from O-CA10 to O-A02 (cf. Fig. 3.2). Extra customer data is contained in APB01. The pizza_price is contained in PB05, the order_amount needs to be calculated

using the discount rules which are contained in PB90, and the current date is assumed to be given (cf. Fig. 3.1, Fig. 3.8).

All relevant dependent facts have to be obtained before the independent P-fact of O-A02 on this object comes into existence. Before it is created the preparer O-A03 is requested for the creation of its P-fact (cf. Fig. 3.3). O-A03 determines the prepared number of pizzas (cf. Fig. 3.8) and creates its independent P-fact. Then the dependent P-fact 'prepared pizzas' is also created. The other mentioned dependent P-facts come into existence together with the independent P-fact of O-A02 (cf. Fig. 3.4).

The information services to be provided by the I-organization of the Pizzeria can be determined on the basis of the AM of the B-organization.

Actor	Service type	Service	Description
A02	1	1	Remember C-fact
		2	Remember P-fact
	2	1	Share agendum
	3	1	Share authorization of performer
	4	1	Share customer
		2	Share number of pizzas
		3	Share ordered pizza_kind
		4	Share price of pizza_kind
		5	Share order_amount
		6	Share existence of pizza_kind
		7	Share indication whether the prepared pizza is OK
		8	Share indication whether the paid amount is equal to the order amount
A03	1	1	Remember C-fact
		2	Remember P-fact
	2	1	Share agendum
	3	1	Share indication whether the performer is authorized
	4	1	Share availability recipe of pizza_kind
A04	1	1	Remember C-fact
		2	Remember P-fact
	2	1	Share agendum
	3	1	Share indication whether the performer is authorized
	4	1	Share available pizza_kind
A05	1	1	Remember C-fact
		2	Remember P-fact
	2	1	Share agendum
	3	1	Share indication whether the performer is authorized

Fig. 3.9 Function model of the I-organization of Pizzeria

All information services with respect to the mentioned service types for the actors O-A02 thru O-A05 have been listed in Figure 3.9[2]. The second column refers to the service type that belongs to the concerning information service.

[2] Subjects who fulfill actor roles in an operational organization may need another type of information that is not specified in the ontological model of the organization. It concerns information that supports weaknesses in the competencies of a specific actor. An example of this type of information is an instruction manual used by the actor for executing its production act. It can be understood as a human technology service, offered by an organization that provides human technology services.

3.4.3 Modeling the I-organization

The ontological model of the I-organization contains the infological actor roles which support the O-actor roles in the B-organization. The ψ-theory states that a subject that fulfills actor roles always executes its production acts using one of three abilities, viz. the performa, informa or forma ability, and that a subject is able to transform from any specific ability into any other ability. This means that although subjects who fulfill O-actor roles create facts in the B-organization, they are also able to perform some other activities in their O-actor role in another shape. For instance, O-actors initiate infological transactions but are also able to interpret documents, to formulate facts, to perform logical operations within action rules in their informa ability.

In the next part of this section the way of modeling will be discussed for each service type separately.

Service Type 1: Fact remembering
C-facts and P-facts are created by O-actors and are remembered by I-actors. By 'remembering a fact' is meant the infological production act in order to recall the concerning fact. By remembering C-facts and P-facts separately, a mild type of redundancy is accepted in the ontological model of the I-organization. After all, a C-fact regarding an 'accept' also includes the P-fact that is to be remembered. We believe that by allowing this form of redundancy, the comprehensibility of the models increases.

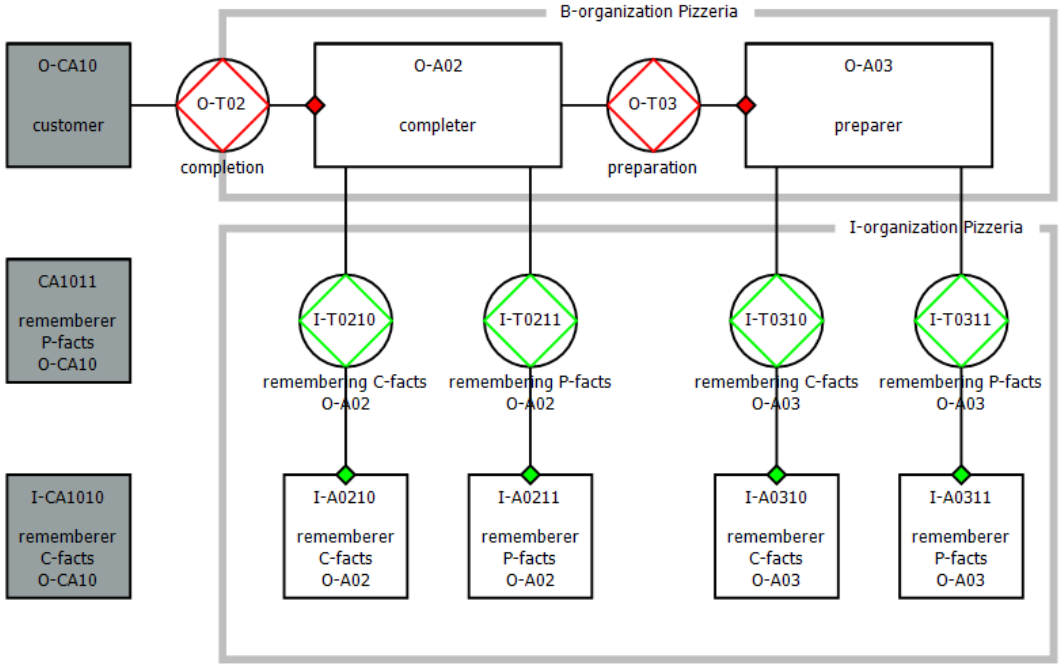


Fig. 3.10 Fact remembering of O-A02 and O-A03

The way of modeling of this service type is illustrated by the Pizzeria case. See section 3.2.2. Generally, an O-actor, called O-Ai, inside the boundary of the B-organization creates

new facts, i.e. C-facts and P-facts. These facts are remembered by the I-actor I-Aij[0,1] ('i' stands for actor rol number, 'j' stands for the number of the service type, '0' stands for C-fact and '1' stands for P-fact) .

Figure 3.10 exhibits the IAM of the I-organization. It shows that the customer O-CA10 is positioned as an external actor. O-CA10 is drawn as a composite actor. Its operations are unknown. The composite actors I-CA1010 en I-CA1011 remember all C-facts and P-facts, respectively, which are created by O-CA10.

The green color in the diagram of Figure 3.10 represents an infological production act. Figure 3.11 exhibits a part of the AM of the B-organization in detail, especially elaborated for remembering. Figure 3.11 is an extended version of Figure 3.6. It shows that every C-act is remembered directly after its creation. After the acceptance of the P-fact of O-A03, the P-fact is remembered too.

Actor O-A03:

when preparation **for** Order is requested	O-T03/rq
assess *justice:* **the** Performer **of the** request **is the** completer **of** Order *sincerity:* <no specific condition> *truth:* < recipe of Pizza_kind is available >	
if *complying with request is considered justifiable* **then** promise preparation **for** Order request remembering C-facts **for** preparation for Order is promised **when** remembering C-facts **for** preparation for Order is promised is stated accept remembering C-facts **for** preparation for Order is promised	

when preparation **for** Order is promised	O-T03/pm
assess *justice:* **the** Performer **of the** promise **is the** preparer **for** Order *sincerity:* <no specific condition> *truth:* <no specific condition>	
if *complying with promise is considered justifiable* **then** execute preparation **for** Order state preparation **for** Order request remembering C-facts **for** preparation for Order is stated **when** remembering C-facts **for** preparation for Order is stated is stated accept remembering C-facts **for** preparation for Order is stated	

when preparation **for** Order is accepted	O-T03/ac
assess *justice:* **the** Performer **of the** accept **is the** completer **for** Order *sincerity:* <no specific condition> *truth:* <no specific condition>	
if *complying with promise is considered justifiable* **then** **for each** (in)dependent P-fact of O-A03 request remembering P-facts **for** P-fact **when** remembering P-facts **for** P-fact is stated accept remembering P-facts **for** P-fact	

Fig. 3.11 Actions rules for O-A03 in B-organization

Service type 2: Agendum selection
Actor Ai constantly loops through the actor cycle, in which it deals with its agenda. By an agendum is meant a C-fact with a proposed time, created by Aj, to which the actor Ai is committed to respond. Using the terminology of the B-organization and I-organization, O-Aj creates C-facts. These facts are remembered by I-Aj10 (cf. Fig. 3.10) and are shared by I-A0020 (sharer agenda). So, if O-Ai wonders whether there is an agendum ready for it then it should ask I-A0020 to select an agendum. This is illustrated by actor O-A02 and O-A03 in Figure 3.12.

For example, O-A03 asks I-A0020 (using his informa ability) to select an agendum for it. I-A0020 initiates I-T0230 and I-T0330 in order to get an agendum from the set of C-facts created by O-A02 and O-A03 and remembered by I-A0210 and I-A0310 respectively. A search for an agendum is successful if the addressee of the selected C-fact matches with the subject that fulfills the actor role O-A03. If the actor gains an agendum, usually an object is linked to the agendum (cf. Fig. 3.2). This is, however, not the case if the actor starts a process. Figure 3.13 exhibits a part of a detailed AM of the B-organization of the Pizzeria. This part shows the action rules of O-A02 particularly. O-A02 starts the completion process. In the model, an object ORDER looks like to be created by O-A02. In reality, the object is created in the C-act 'request' which is performed by O-CA10. After creation the order object, the facts pizza_kind, number of ordered pizzas and the customer pass along with the C-fact 'request' from O-CA10 to O-CA02. This is important, because otherwise O-A02 is unable, for instance, to get detailed customer data, which is contained in APB01 (cf. Fig. 3.1), and to start up the completion process. Thus, the object with the three mentioned dependent facts are created actually by the performer of the coordination act.
Within a process, an object passes along with the C-fact from the initiating O-actor to the executing O-actor and vice versa. So, the P-fact of the executing actor is sent to the initiator along with the 'state'. Looking to the diagram in Figure 3.12, the 'state'' which is sent from O-A03 to O-A02 contains the P-fact of O-A03. Actually, O-A02 initiates I-T0020 and I-A0020 initiates I-T0330. Subsequently, the 'state' of A03 with its P-fact is recalled by I-A0330 and send by its 'state' to I-A0020 and the P-fact of I-A0020 is shared with its 'state' with O-A02.

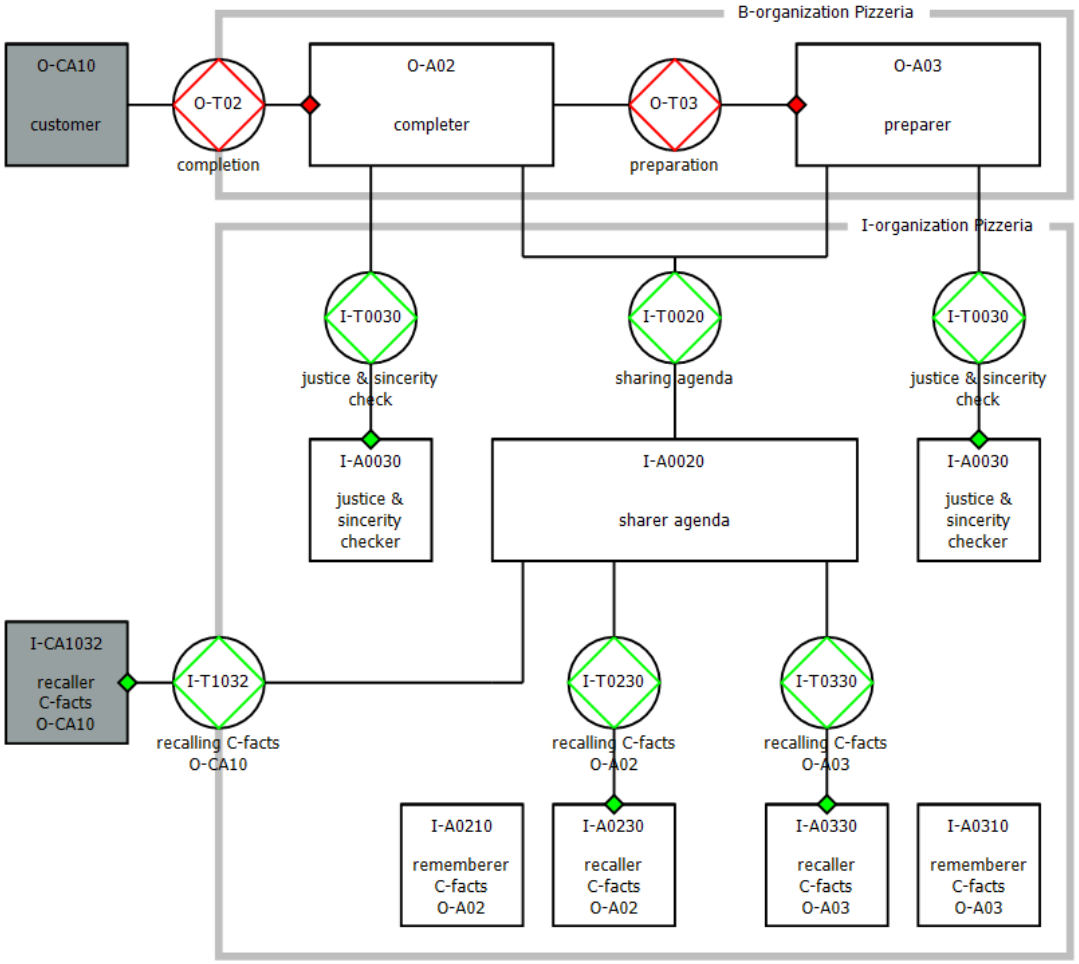


Fig. 3.12 Agendum selection of O-A02 and O-A03

We would like to emphasize that 'remembering' and 'sharing' of C-facts have only been modeled for the transactions in the B-organization. This was explicitly not made for the transactions in the I-organization and the D-organization in order to avoid the so-called Droste effect[3] [Wikipedia].

Service type 3: Fact sharing
The justice conditions which have to be controlled after receiving an agendum are facts that are contained in transaction banks of assignment transactions. These banks contain the assignments of subjects to actor roles. In the Pizzeria case these banks are not specified. Figure 3.12 exhibits a diagram that shows the actor I-A0030 which performs the check on justice and sincerity. The transaction I-T0030 is initiated by O-A02 or O-A03 just after a selected agendum is 'stated' to them. If I-T0030 does not react with a positive result, the actor is able to reject the agendum.

[3] The Droste effect – known as 'mise en abyme' in heraldry – is the effect of a picture appearing within itself, in a place where a similar picture would realistically be expected to appear the appearance is recursive: the smaller version contains an even smaller version of the picture, and so on.

The P-fact of O-A03 regards a physical product on which O-A03 does not capture facts at the ORDER (cf. Fig. 3.8). The acceptance of the P-fact by O-A02 will take place by means of a visual inspection. Actually, this can also be understood as an ontological transaction, but we have chosen, for reason of simplicity, for skipping this transaction kind in this example.

Actor O-A02:

when	completion **for** new Order is requested **with** **the** pizza_kind **of** Order **is** a Pizza_kind **the** #ordered_pizzas **of** Order **is** a #Ordered_pizzas **the** customer **of** Order **is** a Customer	O-T02/rq
assess	*justice:* **the** Performer **of the** request **is the** customer **of** Order *sincerity:* <no specific condition> *truth:* request sharing P-facts **for** existence of Pizza_kind **when** sharing P-facts **for** existence of Pizza_kind is stated accept sharing P-facts **for** existence of Pizza_kind request sharing P-facts **for** Price of Pizza_kind **when** sharing P-facts **for** Price of Pizza_kind is stated accept sharing P-facts **for** Price of Pizza_kind	
if **then**	*complying with request is considered justifiable* promise completion **for** Order request remembering C-facts **for** completion for Order is promised **when** remembering C-facts **for** completion for Order is promised is stated accept remembering C-facts **for** completion for Order is promised	

when	completion **for** Order is promised	O-T02/pm
assess	*justice:* **the** Performer **of the** promise **is the** completer **of** Order *sincerity:* <no specific condition> *truth:* <no specific condition>	
if **then**	*complying with promise is considered justifiable* request preparation **for** Order request remembering C-facts **for** preparation for Order is requested **when** remembering C-facts **for** preparation for Order is requested is stated accept remembering C-facts **for** preparation for Order is requested	

when	preparation **for** Order is stated	O-T03/st
assess	*justice:* **the** Performer **of the** state **is the** preparer of Order *sincerity:* <no specific condition> *truth:* <preparation of pizza is well done>	
if **then**	*complying with statement is considered justifiable* accept preparation **for** Order	

request remembering C-facts **for** preparation for Order is accepted

when remembering C-facts **for**
preparation for Order is accepted is stated
accept remembering C-facts **for**
preparation for Order is accepted

request payment **for** Order
with the Amount_to_pay **is an** Order_amount
request sharing order amount **for** Order_amount
with the price **for** Order_amount **is** Price
the #ordered_pizzas **for** Order_amount **is** #Ordered_pizzas
the customer **for** Order_amount **is** Customer

when sharing order amount **for** Order_amount is stated
accept sharing order amount **for** Order_amount

request remembering C-facts **for** payment for Order is requested

when remembering C-facts **for**
payment for Order is requested is stated
accept remembering C-facts **for**
payment for Order is requested

when payment **for** Order is stated O-T06/st
with **the** Amount_ paid **is an** Amount

assess *justice:* **the** Performer **of the** state **is the** customer **of** Order
sincerity: <no specific condition>
truth: Amount_paid **is equal to the** Amount_to_pay

if *complying with statement is considered justifiable*
then accept payment **for** Order
request remembering C-facts **for** payment for Order is accepted

when remembering C-facts **for**
payment for Order is accepted is stated
accept remembering C-facts **for**
payment for Order is accepted

execute completion **for** Order
state completion **for** Order
request remembering C-facts **for** payment for Order is stated

when remembering C-facts **for**
payment for Order is stated is stated
accept remembering C-facts **for**
payment for Order is stated

	when completion **for** Order is accepted	O-T02/ac
	assess *justice:* **the** Performer **of the** accept **is the** customer **for** Order *sincerity:* <no specific condition> *truth:* <no specific condition>	
	if *complying with promise is considered justifiable* **then** **for each** (in)dependent P-fact of O-A02 request remembering P-facts **for** P-fact **when** remembering P-facts **for** P-fact is stated accept remembering P-facts **for** P-fact	

Fig. 3.13 Actions rules for O-A02 in B-organization

The current price of pizza_kind is obtained from O-A05. (cf. Fig. 3.8). Furthermore, on basis of the discount stack in PB90 and the customer data in APB01 the amount payable is determined and the current date is assumed to be given. Figure 3.14 exhibits a part of the construction model of O-A02. It lacks the creation and remembering of the P-fact of O-A02. All (derived) facts are shared by I-A0031, otherwise the actor does not know which infological transaction to initiate for obtaining a specific fact.

It may not be assumed that the requested pizza kind is already known to the Pizzeria. That has to be ascertained by checking the external production bank PB60. This is done by requesting I-A602 for sharing the relevant pizza kind. If it turns out that the requested pizza kind is correct, the current price of the pizza kind has to be shared by I-A0531. Further, I-A10 calculates the order amount. It takes into account the discount rate which is shared by I-A902 and the customer kind (company or private) which is shared by I-A102. The discount rate is determined by the total amount of ordered pizzas during the current calendar year by the customer. In general, it holds that if an actor needs a derived fact that is derived from multiple facts, this derived fact must be produced by a specifically appointed I-actor.

I-actors do not create P-facts, they derive facts only. I-A10 derives its *raison d'être* from the fact that the O-actor itself cannot perform the infological production act that is required to bring together the facts by an infological production action from both sources into a derived fact.

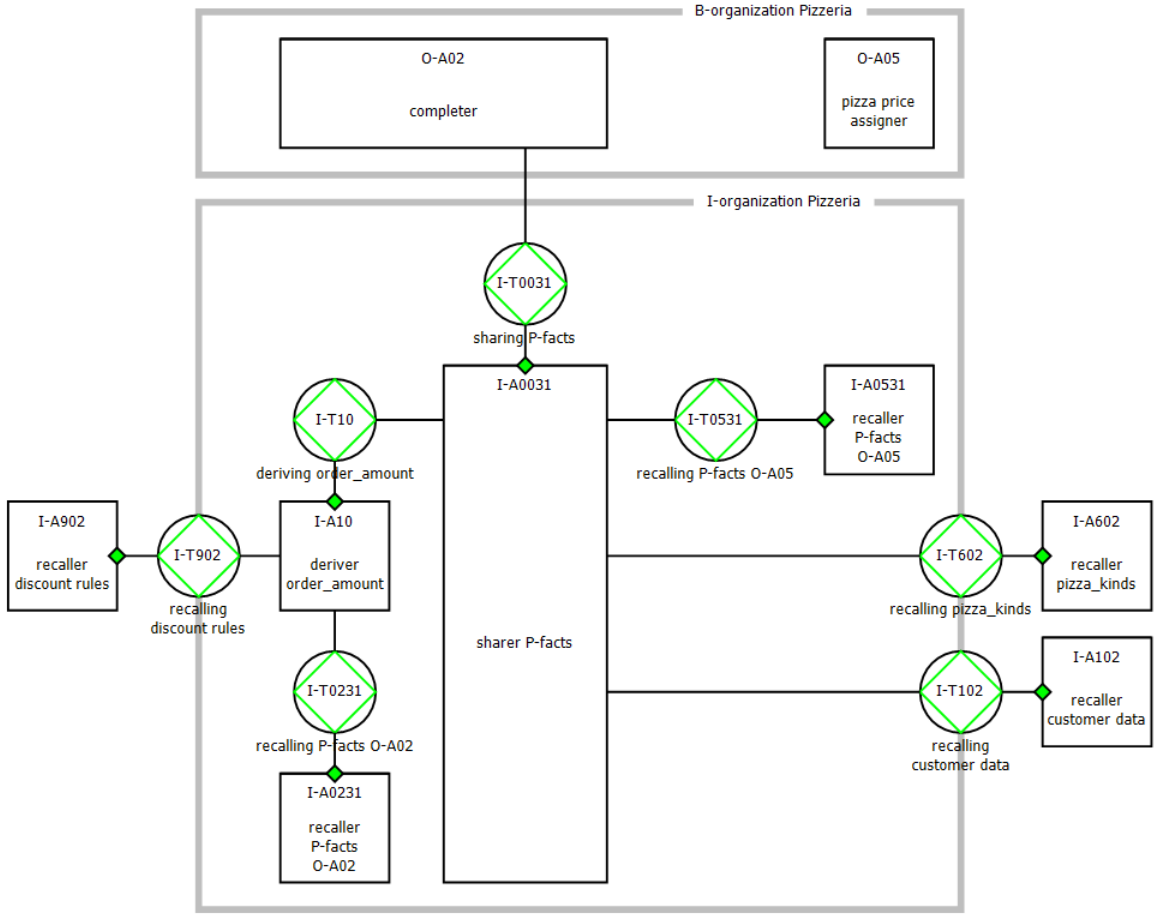


Fig. 3.14 Fact sharing for O-A02

3.5 The D-organization

3.5.1 GSDP applied to the development of a D-organization

After having looked at the cooperation between the O-actors and the I-actors and the cooperation between the various I-actors in the previous section, we focus now on the construction of the D-organization. First of all, it is important to emphasise once again that in the I-organization facts are understood in their semantic meaning. They are thoughts which only could be shared between O-actors and I-actors. D-actors do not pay attention to the semantic meaning of a fact. They pay attention to the form in which a fact is expressed. They act on the conceptualizations of physical documents. We call a conceptualization of a physical document a document and the physical document itself a file. The coordination and production acts on files, such as storing, copying, destroying, and transporting, are performed in the physical organization. We consider the physical organization as out of scope in this study.

The development process of the D-organization can be considered as an instance of the GSDP, see section 2.3. The I-organization has to be understood as the US, and the D-

organization as the OS (cf. Fig. 3.15). For the purpose of implementing 'remembering' and 'sharing', infological actors require two main functions from the D-organization, namely (1) archiving documents and (2) providing documents. These functions are elaborated in section 3.5.2.

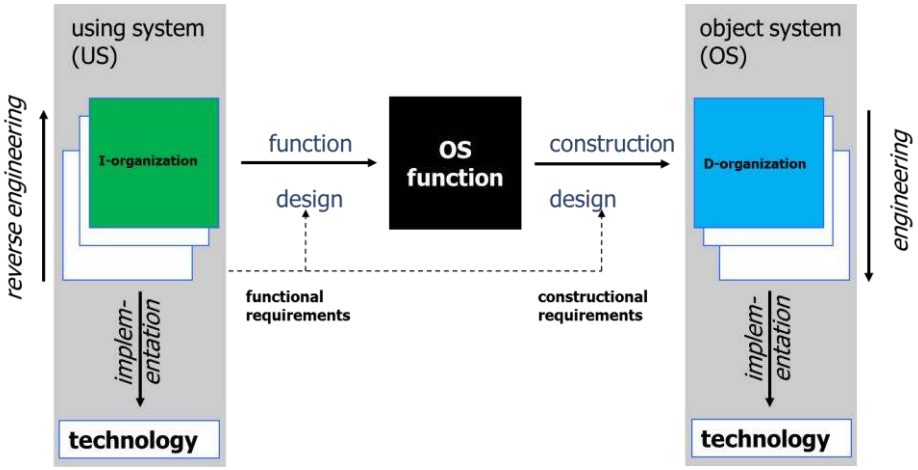


Fig. 3.15 Developing the D-organization from the I-organization [Dietz 2006; Dietz 2008]

From a particular infological model, concrete documental services can be designated for each of the main functions. The total set of documental services which belongs to the two main functions are considered as the function model of the D-organization. From the function model the ontological model of the D-organization is designed. The ontological model clarifies the internal construction and operation of the D-organization in terms of collaboration between its actors to deliver services to the I-organization.

The influence of architecture principles on the design process to achieve the construction of an implementation-independent model of the D-organization shall be disregarded here too. Our focus is primarily on the 'way of modeling' the D-organization.

3.5.2 Determining documental services

An infological actor invokes services of two different service types from the D-organization. One of these service types is related to the first main function of the I-organization, namely archiving documents. The other service type concerns the second main function of the D-organization, namely providing documents. We discuss the two different service types one by one:

1. *Archiving documents.* I-actors will enter into transactions with D-actors in order to take care of 'archiving' original or derived facts. This will often accompanied by a transformation of the form of the fact within the D-organization.
2. *Providing documents.* Facts are archived in order to provide them at a later point in time. Through this service type archived documents can be provided to I-actors, eventually after a form transformation.

It should be noted that within the D-organization the production acts must be defined in such a way that during the production act no loss of semantic meaning will take place.

Several documents can be merged into new documents; documents can be split up into separate documents and transformation of the form of the fact can be taken place. An example of the datalogical production act ‘transforming’ on a document is the conversion of all decimal numbers to binary numbers. The form of the fact changes but the semantic definition remains unchanged.

3.5.3 Modeling the D-organization

We will elaborate the two service types that are identified above.

Service type 1: Archiving documents

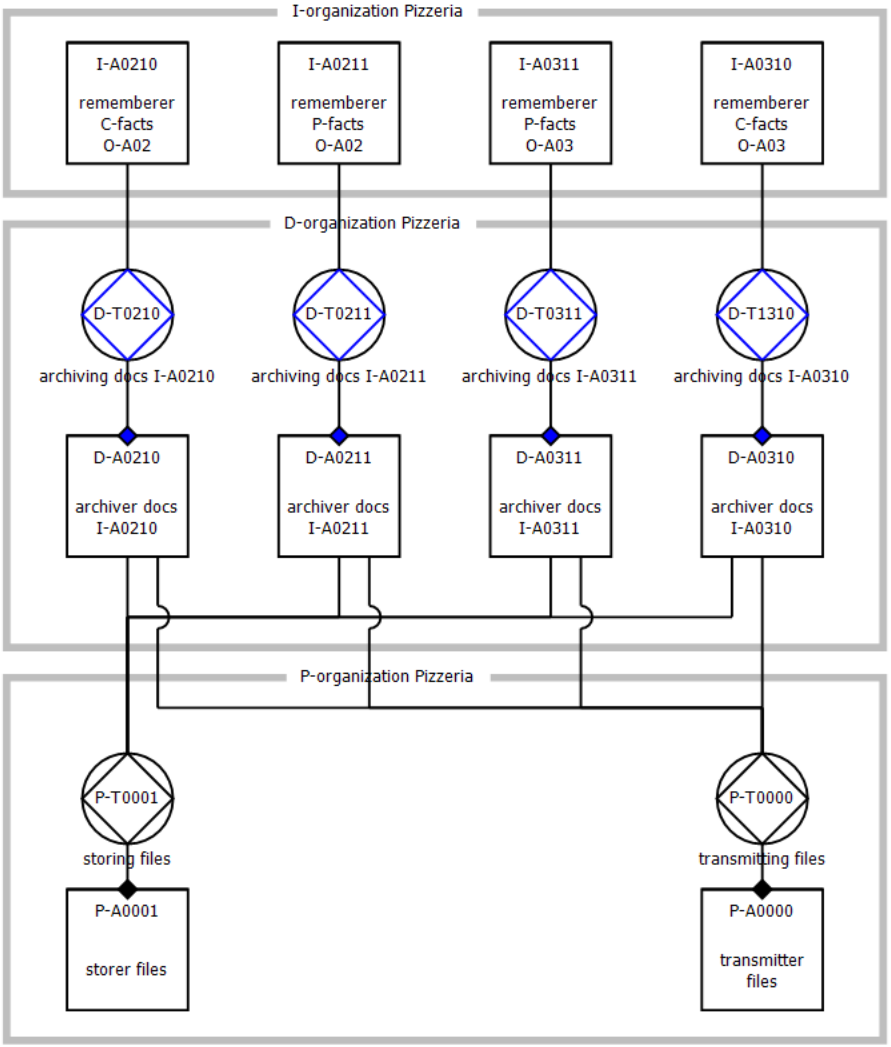


Fig. 3.16 Document archiving

Original facts and C-facts have to be remembered inside or outside the mind of the actor that created the relevant fact. The remembering means that the corresponding documents

have to be archived. Hence, the D-organization contains specific archiver roles to store documents in order to obtain them on a later point in time. Figure 3.16 exhibits a diagram which shows that each 'rememberer' I-Ai10 (C-facts) and 'rememberer' I-Ai11 (P-facts) use the actors D-Ai10 (C-facts) and D-A0i11 (P-facts) respectively, for archiving the corresponding documents. These actors enclose the ontological actors in the P(hysical)-organization of the Pizzeria for dealing with particular storing and transmission issues of the imprinted documents. Imprinted documents are physical objects which are also called files. The blue and the black color in the diagram of Figure 3.16 represent a datalogical production act on documents and an physical production act on files, respectively. Actually, the D-actor archiver initiates in its performa ability the transactions O-T101 and O-T102.

Service type 2: Providing documents

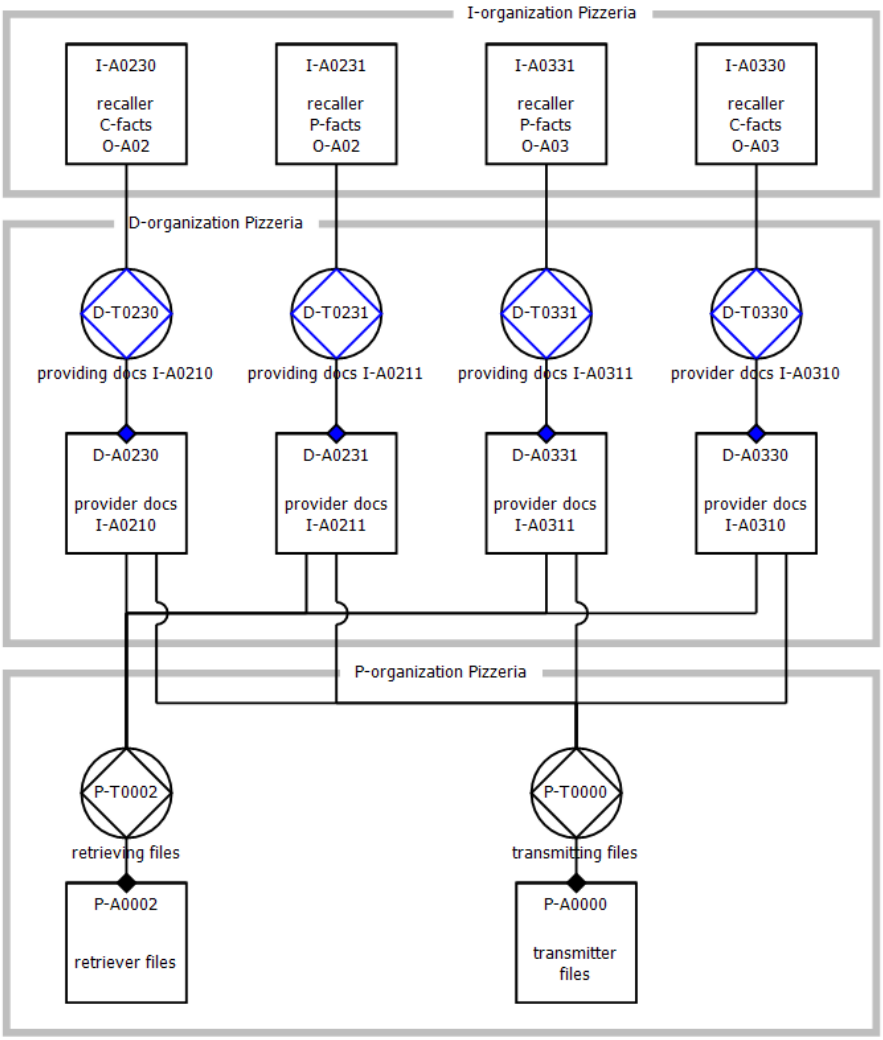


Fig. 3.17 Document provisioning

It should be possible to provide documents that have been archived so that they can be made available to actors that have requested them. The remembering of facts means that

the corresponding documents have to be archived. These facts could be recalled. Hence, the D-organization contains specific providing roles to retrieve the corresponding documents. Figure 3.17 exhibits a diagram which shows that each ‘recaller’ I-Ai30 (C-facts) and ‘recaller’ I-Ai31 (P-facts) use the ‘provider’ D-Ai30 (C-facts) and ‘provider’ D-Ai31 (P-facts) respectively, for providing the corresponding documents. These actors enclose the physical actors in the P-organization for dealing with particular retrieving issues of the stored files. A file has to be transmitted to the location that is accessible by the D-actor which has requested it.

Another example of a transaction kind within the D-organization is the form transformation. Sometimes, documents need to be converted to a different form without loss of semantic meaning.

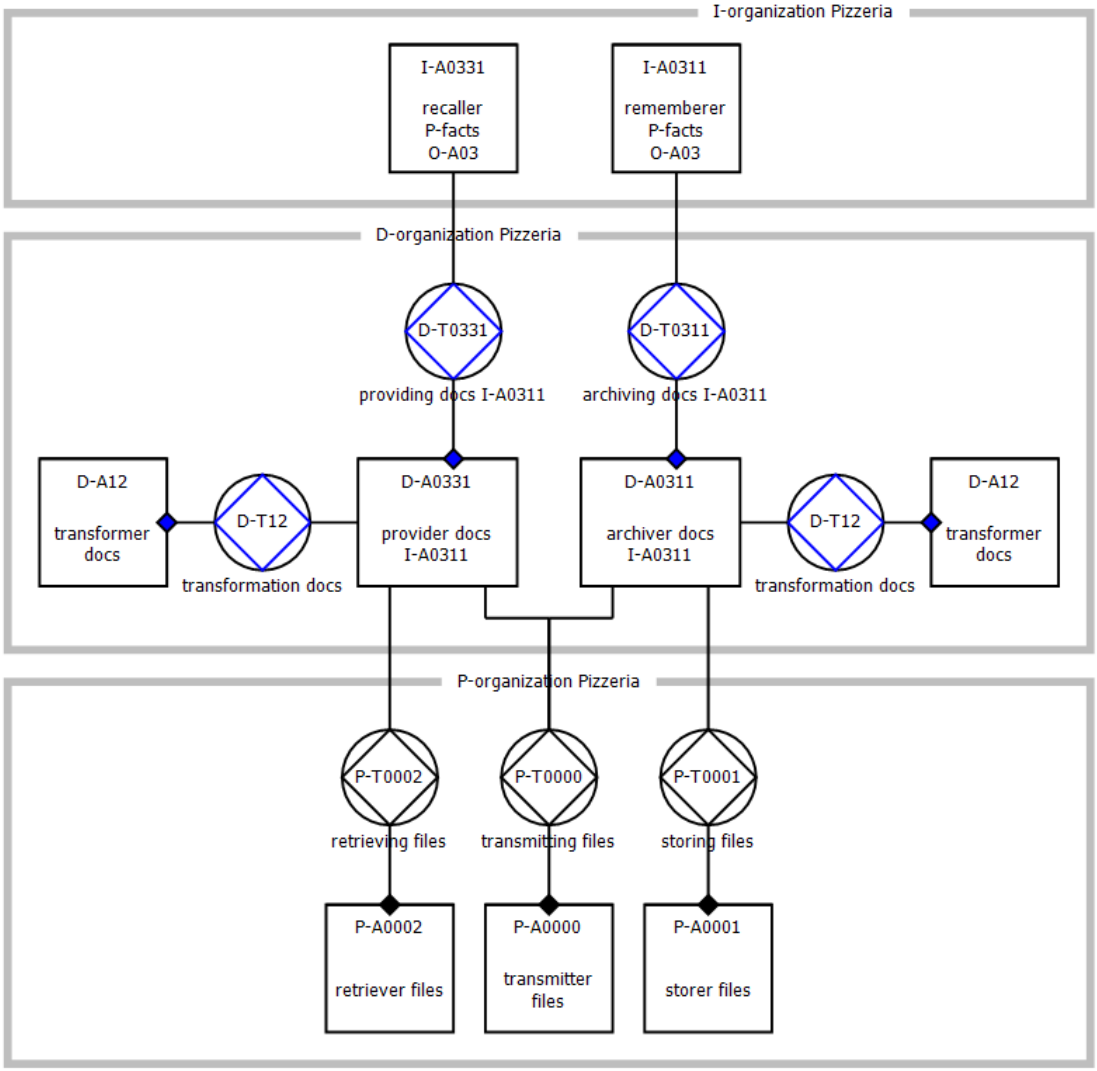


Fig. 3.18 Archiving and providing transformed documents

For example, O-Ai creates a fact. Imagine that this fact is captured in English text and that the document is archived and stored in a digital form. Suppose that O-Aj wants to use

this fact that corresponds with this document. I-Aj31 requests D-Aj31 to provide the document. D-Aj31 encloses retrieving and transmitting to get the corresponding file. The file is transmitted to a location that is accessible for D-Aj31. After receipt, D-Aj31 asks D-A12 for transforming the text into English text and provides it to I-Aj31. I-Aj31 put the file from the location to where it is transmitted.

Figure 3.18 exhibits a diagram which shows the transformer D-A12 that supports the providers D-A0331 and D-A0311.

3.6 The Physical Organization

As stated earlier, the handling of documents as physical objects ('things') takes place in the physical organization. Within the physical organization, the documents are stored, retrieved, transported and possibly removed or destroyed. Actually, for document handling an essential model of this organization can also be designed. We will not go into detail further. The physical organization is kept outside the scope of the study.

3.7 Conclusion

The I-organization and the D-organization have been elaborated from the construction perspective. The integral construction model is used for developing implementation scenarios in the next section.

For a clear understanding of this model of the organization a standard terminology for information management is used (cf. Fig. 3.19).

B-organization		I-organization		D-organization		P-organization	
F	C	F	C	F	C	F	C
	create *fact* use *fact*	remember *fact* share *fact*	derive *fact* recall *fact*	archive *document* provide *document*	transform *document*	store *file* retrieve *file* transmit *file*	copy *file* destroy *file*

Fig. 3.19 Terminology for Information Management [Dietz 2013]

Every fact has three different aspects. It has a semantic meaning, it has a form and it has a physical presence. A fact is created by an O-actor, is understood as semantic meaning (a thought) by an I-actor, is understood as a conceptual document by a D-actor, and is understood as a physical object by a physical actor. All of them have their particular kind of production acts: ontological, infological, datalogical and physical, respectively. The elaboration of the physical organization, however, is considered as out of scope in this study. The integral ontological model contains the connections between all kinds of actor roles. That

makes the integrated model suitable for being used as a reference model for designing implementation scenarios.

A particular implementation scenario is discussed in the next chapter, namely the implementation scenario that corresponds with an EIS.

4 Implementing the Organization

4.1 Introduction

In the previous chapter, the ontological models of the B-organization, I-organization, and D-organization were elaborated. The integral ontological model is the starting point for the design of an implementation model, also called engineering, of the organization. In this section,an integral implementation model is discussed. According to the φ-theory, the implementation model of the organization must be considered as a conceptual model of the operational organization. This conceptual system is represented in diagrams, procedure descriptions, work instructions, software code, etc. A part of the implementation model can be operationalized by computer systems. In this regard particular actor roles can be fulfilled by software programs. Then, we speak of automated actors, or alternatively of agents. We distinguish two types of actors is this section, namely human beings (subjects) and agents. A subject, of course, can also be supported by software tools. This study focuses on the actor roles that are (partly) covered by an EIS.

Section 4.2 contains the initiatives which have been taken so far to draw up guidelines for the development of implementation models to support actors in the B-organization. What is striking here is that although all these initiatives seek to tie in with the B-organization, there is hardly any discussion about the meaning of I-organization and D-organization as social systems which support the B-organization in the enterprise. Almost all initiatives understood information systems as rational systems which have to be related to the B-organization without any talking about their relationship with the I-organization and D-organization. Bunge [Bunge 1979] wrote already that the category of rational systems differs from the category of social systems.

In this study, we choose for another approach. According to the τ-theory the I-organization can be considered as an object system of the using system B-organization. In chapter 3 we have devised the ontological model of an I-organization. This ontological model is the starting model for engineering an implementation model of the information system. The operationalization of the implementation model is achieved by connecting technology to components of the implementation model. Then, the 'information system' comes into existence.

In section 4.3 we will discuss the notion 'implementation model' and its relationship with an EIS. Although the implementation of exceptions is of substantial importance [Guerreiro 2012], the discussion is limited to the success layer (basic pattern) of transactions. This study is primarily focused on the 'way of working' for engineering the implementation model of the organization. In section 4.4 and section 4.5 the implementation issues of the scope of an implementation scenario and the implementation issues of actor roles are discussed, respectively. In section 4.6, some elements of an implementation scenario are discussed on the basis of a real example of the ontological model of the Pizzeria. Finally, some concluding remarks are given in section 4.7.

4.2 State of current research

According to Dietz [Dietz 2006], all three homogeneous systems—the B-organization, I-organization and D-organization—belong to the category of social systems. Although the nature of the production acts in the I-organization and the D-organization is such that in many cases the actors in these aspect organizations can be automated, he also clearly shows that, in particular, production acts within the B-organization and coordination acts within all three organizations are reserved for human beings. After all, acts that are performed using the performa ability cannot be taken over by a machine. Since 2000, several studies have been published in which the information system is described as a rational system that works as an object system of the B-organization. These studies ignore that the information system itself is also a rationalization of a social system, namely of the I-organization and D-organization.

Mallens, et al. [Mallens, Dietz et al. 2001] define an information system based on the B-organization as a system in the category of rational systems. They call the elements of the system 'rational individuals', which perform rational actions like retrieving, recalling, providing knowledge, calculating, and making logical deductions. They discuss the application of a dedicated business process modeling technique prior to the systems design applying UML to model information systems. The result of this modeling step prior to using UML should be that the systems designer has a better starting point for making the right models that are complete and that have a basis in a structured business process description. Three problems are addressed: 1) The delimitation problem, i.e. how to delimit the scope of the information system that is to be developed; 2) The identification problem, i.e. how to be complete in identifying all relevant information system concepts within the scope; and 3) The specification problem, i.e. how to specify the identified information system concepts. For all three problems, it is discussed how modeling business processes prior to modeling information systems (using UML) could add value.

Maij et al. [Maij, Toussant et al. 2002] demonstrate that the DEMO models are suitable as a starting point for deriving system functionality by using the use case concept of UML. The case study demonstrates that in using this approach for the alignment of functional features of the ICT infrastructure to business processes, insight is gained into the mutual influence of ICT infrastructure and organization structure.

Shishkov et al. [Shishkov and Dietz 2003] study and analyze the strengths of DEMO in the derivation of use cases. This could be helpful for further activities directed towards finding out the most appropriate ways of identifying use cases from business processes.

According to Dietz [Dietz 2003], use cases are intended to capture the functional requirements of an information system, and the problem of identifying use cases has not yet been satisfactorily resolved. He presents an approach for deriving use cases from the business system models that are produced by applying DEMO. He shows that three attractive properties are possessed by use cases derived from these models: *essence* (real business items are identified and clearly distinguished from informational items), *atomicity* (one ends up with items that are units from a business point of view), and *completeness* (no business items are overlooked and the models do not contain irrelevant items).

Finally, Mulder [Mulder 2006] writes about actors from the I-organization and D-organization as rational components embedded in an information system and infrastructure system, respectively.

Besides this top-down approach in which the business system is seen as a social system and the information system as a rational system, the opposite approach has been taken in the University of Antwerp. It is precisely the connection with the business organization that the authors seek in the study of Normalized Systems by Mannaert and Verelst [Mannaert and Verelst 2009]. The rational information system should be connected with the social system of the business organization. The Normalized Systems theory focuses on achieving highly evolvable information systems or software, by investigating the modular structure of information systems in a manner that is programming language-independent. It is based on systems theory, on the concept of stability, and on thermodynamics, more specifically on the concept of entropy [Huysmans 2011]. In software engineering, stability has been defined as follows: "A design characteristic of software is stable if, when observed over two or more versions of the software, the differences in the metric associated with that characteristic are considered, in the context, to be small" [Kelly 2006]. More generally, stability is defined as the characteristic of being resistant to change propagations. Normalized Systems theory defines stability in terms of the absence of combinatorial effects. A combinatorial effect occurs when the impact of a change is dependent not only on the change itself, but also on the size of the information system or software [Mannaert and Verelst 2009]. Assuming that a system grows in size over time, this would lead to increasingly large impacts for a given change. This characteristic is considered instability in Normalized Systems theory, and therefore, Normalized Systems should be free of combinatorial effects [Mannaert and Verelst 2009]. Such Normalized Systems can therefore be considered to be evolvable. The elimination of combinatorial effects is also related to Manny Lehman's Law of Increasing Complexity. This law states that "as time goes by, the structure of software will degrade and become more complex as changes are applied to it, causing the impact of a given change to increase over time" [Lehman 1980; Lehman and Ramil 2001]. More specifically, as a combinatorial effect implies an increase in the number of impacts of a given change over time as the system grows, this phenomenon would contribute to Lehman's law applying to an increasing extent. Thus, Normalized Systems are defined as information systems exhibiting stability with respect to an anticipated set of changes and avoiding the occurrence of combinatorial effects [Mannaert and Verelst 2009].

Huysmans et al. [Huysmans, Bellens et al. 2010] introduce Enterprise Ontology as a promising approach that is relevant in this regard. In their paper, the authors combine Enterprise Ontology and Normalized Systems. To this end, they express the transaction pattern, a central construction of Enterprise Ontology, using the constructs of Normalized Systems. By aligning these constructs, they attempt to introduce traceability between the Enterprise Ontology level and the Normalized Systems level. A first method for deriving a Normalized System from the ontological model of the B-organization (provided by applying DEMO to an enterprise) was designed, tested and improved by Krouwel et al. [Krouwel and Op 't Land 2011].

The third approach that has emerged in the past decade is the focus on identifying and specifying business components. Albani et al. [Albani 2006; Albani and Dietz 2011] write that the identification of business components is a crucial factor. They seek to improve the identification of business components based on the ontological model of an enterprise satisfying well-defined quality criteria. Information systems need to be modeled at a high

level of abstraction that is also understood by business people, who are defining the requirements and using the systems in question. The use of business components for the development of a high-level information system is valuable since they "directly model and implement the business logic, rules and constraints that are typical, recurrent and comprehensive notions characterizing a domain or business area" [Barbier and Atkinson 2003] (all other components are considered either to deliver services to these business components or to offer some general functionality). The identification of business components is therefore the first step in the development of an information system, according to current standards. It is a very crucial one and, therefore, it should be performed to the highest possible level of quality. For the identification of the business components, they apply the *three-dimensional method for business components identification* (BCI-3D), which aims to group business tasks and their corresponding information objects into business components satisfying defined metrics. The metrics used—being minimal communication between, and maximum compactness of, business components—are the basic metrics for the component-based development of inter-enterprise applications.

Since the identification of business components is strongly dependent on the underlying business model, the BCI-3D method uses the object classes and fact kinds from the FM and the process steps from the PM, including their relationships. One can distinguish between three types of relationships needed for the identification of business components: (1) the relationship between single process steps; (2) the relationship between information objects; and (3) the relationship between process steps and information objects. A relation type distinguishes between subtypes expressing the significance of a relationship. E.g., the relationship between single process steps expresses—based on their cardinality constraints—how usually a process step is executed within a transaction and therefore how closely two process steps are related to each other in that business domain. The relationship between information objects defines how loosely or tightly the information objects are coupled, and the relationship between process steps and information objects defines whether an information object is used or created when executing the respective process step. All types of relationship are of great relevance in order to define which information object and process steps belong to which component. The relationships are modeled in the BCI-3D method using a weighted graph. The nodes represent either information objects or process steps, and the edges characterize the relationships between the nodes. Weights are used to define the different types and subtypes of relationships and to build the basis for assigning nodes and information objects to components. In order to provide optimal grouping while minimizing communication and ensuring compactness of components, an optimization problem needs to be solved; a genetic algorithm has been developed to do this.

Based on this method for identifying business components, i.e. coarse-grained modules of a service-oriented system, and the services required for interaction between business components, Terlouw [Terlouw and Maarse 2009; Terlouw 2011] developed a service specification framework in order to provide practitioners with a better understanding of how to deal with the modularity of enterprises and their supporting software systems. In her approach, service orientation is not seen as a technical paradigm but as a paradigm for structuring the entire enterprise. The designed service specification framework is based on the ψ-theory and the τ-theory. The ψ-theory provides a basis for formalizing the notion of service. The whole transaction pattern is used as a basis. The service is a pattern of coordination and production acts, performed by the executor of a transaction for the benefit of its initiator, in the order as stated in the complete, universal pattern of a transaction. A distinc-

tion is made between three types of services: ontological, infological, and datalogical. There is both a 'human' version and an 'IT' version of each of these three types of services.

Finally, a fourth approach has appeared through the research of Guerreiro and Van Kervel [Guerreiro 2012; Kervel van 2012; Kervel van, Dietz et al. 2012]. Van Kervel notes that current model-driven engineering approaches suffer from two major shortcomings. First, they are unable to deliver domain models that comprise all functional requirements. Second, the models to be produced during the system development process are not formally defined. He introduces the concept of ontology-based model-driven engineering and then notes that each step in the GSDP—namely function design, construction design, and implementation design—is prone to errors. The functional specifications are different from the user expectations, the ontological construction specifications do not fully match the functional specifications, and the resulting implementation model does not fully match the ontological model of the object system. Therefore, he proposes that model-driven engineering could enable any complete, correct implementation of the ontological model of the using system to make the using system come 'alive' once it is put into operation. He has developed a DEMO processor that executes the model of the using system. Essentially, the ontological model is the executable software. Changes in the software, as required by any agile enterprise, are brought about 'on the fly', through regeneration, based on the modified ontological model of the enterprise's organization. Guerreiro defines and implements an ontological solution for the control of the run-time business transactions guided by the concepts of accountability, authority, competency, delegation and responsibility. Business transactions models prescribe the design freedom restrictions of the business transactions dynamics but *explicitly* do not guarantee that organizational actors perform them accordingly. Enterprise dynamic systems control guarantees that the prescriptions are followed in the operation by performing a continuous cycle of observation, decision and action.

Having outlined these four different approaches to the implementation of the information systems based on enterprise ontology, it becomes clear that all these approaches focus on certain aspects of the implementation model of the organization, but that it is not embedded in an integral ontological model of the B-organization, I-organization, and D-organization. Within this integral model, the actor roles of different kinds have been assigned a central position. These actor roles are fulfilled by subjects or agents (automatic actors). This will be further elaborated in the next subsection. It is important to note that the scope of the organization in the context of this chapter falls within the boundaries of an enterprise.

4.3 The Implementation Model

The implementation model of the organization of an enterprise must be understood as an implementation variant of the integral model of its B-organization, I-organization, and D-organization. We speak of an implementation variant because the actor roles in the three different aspect systems can be fulfilled in many different ways. Coordination acts can be performed explicitly or implicitly. Actor roles can be supported partially or fully by IT

systems. The IT infrastructure, i.e. the physical organization, is usually organization specific. In other words, an unlimited amount of implementation scenarios can be derived from one particular ontological model. In this study we refrain from making any normative judgments on the choice of a particular implementation variant. There are several features that make one particular variant more suitable for a particular organization than another variant. These features can be investigated in a follow-up study. Our subject of study is to find guidelines for a 'way of working' in order to develop an implementation model that is based on the integral model of the B-organization, I-organization, and D-organization. But prior to this we will first need to learn more about the characteristics of an implementation model.

According to the φ-theory there is a difference between the enterprise as a concrete system as it manifests itself in the objective world, the enterprise as a conceptual system in the mind of the observer, and the enterprise as a symbolic system that should be considered as the representation of the conceptual system (cf. Fig. 4.1). The symbolic system manifests itself in a series of symbols framed by a grammar. The symbolic system consists not only of descriptive texts and diagrams in formal or natural languages that describe the process structure and work procedures for actors in the enterprise; it also consists of the source code of implemented business applications. More generally, the symbolic system of an implementation model of an enterprise encompasses formulations of the behavior of all actors in the organization.

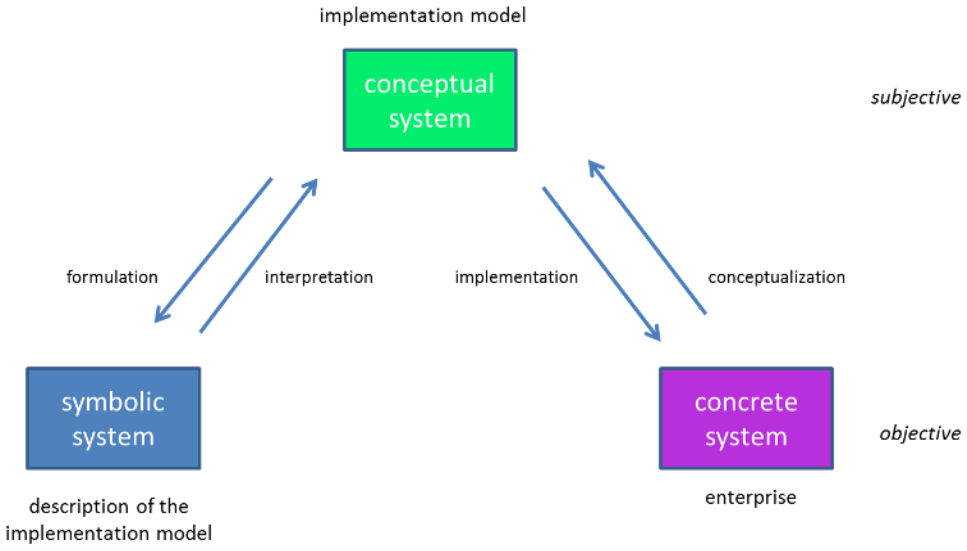


Fig. 4.1 The implementation model triangle

In order to perform a production act, the actor must use one of the following abilities: the performa, the informa or the forma ability. In performing a coordination act, all three abilities must be used in turn. Actors use the performa ability when they create an original P-fact or a C-fact. These actors are always subjects, because only subjects are able to perform original acts. Two examples: entering into a commitment between two actors is an original act; approving the quality of a product to be delivered is an original act too. Because these acts are autonomously made decisions, their results are unique, unlike e.g.

computations [Tribolet 2009; Tribolet 2012][4]. The results are created as soon as they are performed.

4.4 Scoping the implementation scenario for an EIS

A part of an implementation model, also called implementation scenario, of an enterprise can be covered by an EIS. We focus in particular on that part of the implementation model for which we would like to develop an EIS. The first step in designing an implementation scenario for an EIS is the determination of its scope. Normally, an EIS does not cover the entire production world of an organization. It covers a sub system of the organizational system. The sub system is determined on the basis of the OCD and associated BCT on the one hand and the FM on the other hand. The object types which have to be covered by the EIS are specified in the FM. From the set of selected object classes a set of result kinds, that refer to these object classes, have to be determined. The set of result kinds determines the set of O-actors, subsequently, whose P-facts should be created within the scope of EIS. The selection of a set of object classes needs further explanation, because objects are usually no isolated 'things' in the production world. Objects are related to each other by dependent fact kinds. So, it can be that object classes within the scope of the EIS and object types outside of it are related by dependent fact kinds. Designers of EIS solve this by taking over the external object class inside the EIS. That is, however, not without consequences, because the result kinds which are related to the object class are kept outside the EIS. We shall return to this topic in the next chapter.

Once one has definitively established which O-actors are supported by the EIS, the I-actors and D-actors involved in remembering and providing facts fall within the scope of EIS. This also applies to the actors that provide facts that are derived from the part of the production or coordination world that is covered by the EIS.

After defining the set of O-actor roles which have to be supported by the EIS, the O-actor roles can be clustered. A cluster is defined as a set of actor roles for which it holds that they are allocated to the same organizational function. The identification of clusters is important, because of the possibility for (1) optimizing the implementation of the mutual coordination acts within a cluster, and (2) optimizing fact sharing. It does not make sense to share specific facts again if they are already invoked by the subject that fulfills actor roles in the cluster.

[4] We prefer this formulation to that of Tribolet, who speaks of the exchange of carbon-based actors and silicon-based actors and thus seems to pay less attention to the fact that acts that are performed using a performa ability are always original and therefore cannot be contained in the implementation model.

4.5 Implementing Actor Roles

4.5.1 Fulfillment

Some topics about the fulfillment of O-actor roles which have to be supported by the EIS are discussed in this section. We pay attention to four issues.

Firstly, an actor role can be fulfilled by several subjects. This can be done in two different ways [Dietz 2006]:

- *Concurrently*—individual subjects create the same kind of P-facts separate from each other. Two different situations may arise: (1) subjects who are completely interchangeable, (2) subjects who are specialized in a specific domain. For example, salespeople who are specialized in business with a dedicated group of countries or, more generally, in a specific customers category.
- *Collectively*—P-facts are created by subjects collectively; they are held accountable for these facts collectively (e.g. a management team).

It should be taken into account that a split may occur between bearing responsibility and performing coordination acts or production acts. Tasks which are allocated to a specific actor role are not always performed by the actor itself. They may be delegated to other subjects.

Secondly, each actor role should have its own 'look and feel'. This concerns the way in which used facts are shared and in which new C-facts and P-facts are remembered. One must be aware that the physical organization cannot be ignored. Actors deal with documents which are imprinted on a medium. Imprinted documents have to be understood as physical objects or as 'things' that are manipulated by production acts of physical actors.

Thirdly, for each actor role it should be determined to what extent it needs to be automated. A fully automated actor role is called an agent. Actor roles are fulfilled either by subjects or by agents, whereby subjects can be supported by software tools. The question that immediately arises is whether agents are able to fulfill an actor role entirely. From the ψ-theory it is known that O-actor roles have to be performed exclusively by subjects and that I- and D-actors could be performed by agents in many cases. That is, however, only valid for production acts, subjects are always indispensable for the execution of communication acts. Let us consider the performance of a coordination act. Every actor must briefly use the performa ability when performing a coordination act. Dietz [Dietz 2006]: "Well obviously, it means that actor roles can never be taken over completely by ICT systems, because of the performative exchanges in which they inevitably participate, ….". We come back to this issue in the next section.

Table 4.1: Responsibility Areas (RAs) in Organizations

Responsibility areas	Comments
RA-1. Function model	The designer of the function specifications of the organization is responsible for the *raison d'être* of the organization for the stakeholders.
RA-2. Ontological model	The designer of the ontological model is responsible for the ontological model that is used for designing the implementation model It is based on the function model of the organization.

RA-3. Implementation model	The designer of the implementation model is responsible for the implementation model that has been implemented in the enterprise. It contains the action rules, work instructions and algorithms for human actors and agents based on the ontological model of the enterprise.
RA-4. Allocation of subjects and agents	The allocator is responsible for assigning subjects or agents or supporting tools that are capable of performing the work instructions or algorithms that have been defined for the actor roles in RA-3, to actor roles.
RA-5. Operational implementation of actors	All actors are responsible for the created P-fact (O-actors) or for the derived P-fact (I-actors), unless the actor is implemented as an agent.

Fourthly, According to the ψ-theory the responsibilities of an actor are reflected in the coordination acts of the actor. The same theory says that only subjects can bear responsibilities. O-actors create new, original, P-facts and are therefore responsible for the quality of these facts. I-actors and D-actors only execute predefined instructions. If it appears that a subject is not competent to fulfill an allocated actor role, then the allocator of this role must be called to account for it. Agents cannot be held responsible for their performance. A technical construct cannot bear responsibility. It might of course also be possible that a subject or the supporting software operates properly but that the set of instructions to be performed is not correct. Then the designer of the implementation model must be held responsible. The designer, in turn, is dependent on the quality of the ontological model and the functional specifications. The distinctions between the responsibility areas are based on the GSDP and summarized in Table 4.1.

4.5.2 Sharing Facts

In performing coordination acts the performer raises three validity claims towards the addressees: the claim to justice, the claim to sincerity, and the claim to truth. All three of them have to be accepted by the addressee in order to make the coordination act successful [Perinforma 2012]. The acceptance occurs on the basis of the current state of the internal and external production and coordination world. This is based on what we discussed in section 3.4.2. We distinguished two service types for sharing facts: sharing agenda and sharing facts (C-facts and P-facts).
Before we elaborate implementation specific issues of every service type separately, we would like to discuss some common issues.

Facts are shared by initiating an infological transaction with an I-actor. This actor is either a subject or an agent. We would like to discuss differences between the exchange of facts between human subjects and the exchange of facts between human subjects and agents.

The ψ-theory distinguishes four different levels of communication: physical interaction, notational correspondence, cognitive correspondence and social correspondence (cf. Fig. 2.6). See also [Dietz 2006]. We speak of physical interaction or of physical exchange if imprinted documents are exchanged. Actually, only physical objects can be exchanged between subjects or between subjects and agents. In case of notational correspondence, there must be significational understanding between both actors. This means that the ad-

dressee of a message must be able to recognize the syntactic form and be able to interpret it. For example, a document in a digital format is not easily interpretable by a subject. The actor that would like to share this document with a subject should transform it to a natural language text. In case of cognitive correspondence, there must be intellectual understanding between both actors. This basically means that an agent that fulfills an I-actor role must be able to 'understand' a request regarding some proposition. This 'understanding' by an agent takes place in an artificial manner by using a formal language. Finally, in case of social correspondence, an agent as addressee is not able to make a commitment with the performer, and vice versa. However, human behavior can be mimicked.

Let us discuss next the two mentioned service types:

Firstly, an actor asks for the selection of an agendum (cf. Fig. 3.12). The agendum is a coordination act with an addressee that matches with the actor. Generally, a C-fact is passed to an actor Tom (addressee) by an actor John (performer) as the result of a coordination act within a transaction that John and Tom are carrying through. The following aspects should be addressed in the implementation model:

- Tom regularly loops through the actor cycle. In every loop he invokes the 'sharer agenda' in order to get the next task to perform.
- The 'sharer agendum' has to be implemented for general usage. All O-actors within the scope of the EIS have to be authorized to invoke the 'sharer agendum'. It knows all the recallers of C-facts within the scope of EIS and it collects all created C-facts continuously in order to share them with the addressees.
- Assume that John fulfills an actor role outside the scope of the EIS and Tom fulfills an actor role inside the scope. Tom invokes the 'sharer agenda' for getting an agendum. The 'sharer agenda', however, does not know John and is therefore unable to get his C-facts. This means that John has to ask a subject who fulfills an actor role within EIS to be his recaller. This can be solved, for example, by giving John access to the EIS through a website.
- The agendum always relates to a particular object. Then Tom knows the P-fact to which the agendum is connected. He also knows the dependent P-facts which have been linked to that object.
- An agendum has a specific form (e.g. English text, sound block) and a medium. Its form should be such that Tom is able to interpret it. Agents are more limited in interpreting documents, due to a lack of human senses.
- In case that Tom has some colleagues who fulfill the same actor role and that it does not matter to John which subject should perform the agendum then a scheduling mechanism has to be implemented. Agenda can be divided on the basis of available time of the actors, or on specific competences of the actors, or on other internal priority rules.
- Agenda can be bundled. For example, assume that Tom has submitted one request for payment for a bundle of delivered goods to John. Tom accepts or rejects only the payment of the total amount. However, in the ontological model separate payment transactions are defined for each delivered product. This means that Tom receives as the 'stated' C-fact a bundle of 'stated' C-facts which must be responded to with an acceptance of each paid product separately.

Secondly, an actor asks for a specific fact (cf. Fig. 3.14), a C-fact or a P-fact. The following aspects of sharing a C-fact or a P-fact should be addressed in the implementation model:

In case of C-facts (let us consider Peter and Bill who fulfill different actor roles):

- The 'sharer C-facts' has to be implemented for general usage. All O-actors within the scope of the EIS have to be authorized to invoke the 'sharer C-facts'. It knows all the recallers of C-facts within the scope of the EIS and it invokes them in order to share a C-fact which is requested by an O-actor.
- Suppose that Peter fulfills an O-actor role outside the scope of the EIS and Bill fulfills an O-actor role inside the scope of the EIS. Peter initiates a transaction with Bill. Bill wants to invoke the 'sharer C-facts' for getting the C-fact of Peter. The C-fact that Peter creates was entered in the EIS by its delegate who fulfills the 'recaller C-facts' role within the EIS.
- A P-fact is at any moment during the transaction in a certain state (e.g. requested, promised, declined, stated, accepted, or rejected). It must be possible that this state can be inspected.
- To each C-fact is connected the identification of its performer. That makes it possible to gain more insight into 'who' is responsible for a particular transition of the C-world or P-world of the organization. This is important from the viewpoint of transparency and accountability.
- A C-fact has a specific form (e.g. English text, sound block) and a medium. Its form should be such that the addressee is able to interpret it. Agents are more limited in interpreting documents, due to a lack of human senses.

In case of P-facts:

1. The 'sharer P-facts' has to be implemented for general usage. All O-actors within the scope of the EIS have to be authorized to invoke the 'sharer P-facts'. It knows all the recallers of P-facts and the providers of derived facts within the scope of the EIS and it invokes them in order to share a (derived) fact that is asked by an O-actor.
2. Suppose that Bill fulfills an O-actor role outside the scope of EIS and Peter fulfills an O-actor role inside the scope of EIS. The P-facts of Bill are remembered in an external fact bank. Peter wants a fact that is created by Bill and invokes the 'sharer P-facts' for getting the fact. The 'sharer P-facts', however, does not know Bill and is therefore unable to get his facts. It means that within the scope of EIS a 'recaller P-facts' must be defined as a delegate of Bill for making his P-facts available within the EIS. Bill and his delegate have to make an agreement for the fact exchange.
3. Suppose that Bill fulfills an O-actor role outside the scope of the EIS and Peter fulfills an O-actor role inside the scope. Peter initiates a transaction with Bill. Peter does not want to invoke the 'sharer P-facts' for getting the P-fact of Bill. The P-fact is received by Peter along with the 'state' which is performed by Bill. Peter receives this 'state' by invoking the 'sharer C-facts'. The 'state' that Bill creates was entered in the EIS by its delegate who fulfills the 'recaller C-facts' role within the EIS.
4. A P-fact that is received has a specific form (e.g. English text, sound block) and a medium. Its form should be such that Peter is able to interpret it. Agents are more limited in interpreting documents, due to a lack of human senses.

5. One takes into account that if a P-fact has not been formally accepted, it does not exist. So, it cannot be received by a request to 'sharer P-facts'.
6. A fact to share must have the semantic meaning that is expected by the requester. This need not be the case, especially for facts that are shared from outside the organization. For example, assume that the place of residence of a specific person is requested. The requester expects the place in which the person spends at least five days a week. However, the authority issuing the place of residence information defines 'place of residence' as the place where the person is registered. This is called a lack of semantic interoperability.

4.5.3 Remembering Facts

C-facts and P-facts that are created in the O-organization have to be remembered. The BCT includes the facts kinds that have been defined for every actor role. Let us start by looking at the C-facts. The following aspects should be addressed in the implementation model:

1. C-facts are not always created explicitly. For example, John sends a 'state' to Tom. Then, Tom does not always perform an 'accept' explicitly. It assumes, in many cases, that if Tom does not say 'no', he says 'yes' implicitly. Then, it is necessary to make a clear agreement between John and Tom which will allow John to reliably determine which C-fact he should have received at a given time. For example, if John does not receive a 'reject' within 24 hours after sending a 'state', he may assume with certainty that there is an 'accept'.
2. It is necessary to remember the relationship between a C-fact and a P-fact, because, a C-fact determines the state of a P-fact.
3. The performer of a C-fact is considered to be the owner of the C-fact.
4. A C-fact is given a specific form (e.g. English text, sound block) and a medium. The form and the medium should be such that the fact can be remembered and recalled—after a transformation of form and/or medium—by actors that are authorized for it.
5. C-facts can be bundled. For example, a payment request to a specific customer concerning the delivery of a bundle of delivered goods or services.

Let us now turn to the P-facts. The following aspects of remembering a P-fact should be addressed in the implementation model:

1. The production act has to be executed within the intended settlement time of the C-fact. If the requested production time lies in the past, it means that the initiator wants something to start to exist in retroaction. An example is the start of a subscription with a starting day in the past. However, it could also be a future point of time; it will take until then before the fact becomes existent. An example is the start of a membership from the first day of some future month [Perinforma 2012]. The time that the P-fact actually becomes real is called the P-event time. The P-event time differs from the actual creation time of the P-fact. The actual creation time of the P-fact is equal to the creation time of the C-fact 'accepted'.
2. A P-fact is owned by its executor. This subject is responsible for its quality. In case that an agent fulfills the actor role, the allocator of the agent is responsible for the quality of the derived fact.

3. A P-fact is 'remembered' in a specific format and on a specific medium so that it can be recalled and interpreted by other actors - possibly after a transformation of form and/or medium- at a later time. There is a distinction between original P-facts and derived P-facts. Original facts always have to be remembered otherwise they will be lost. This does not apply to derived facts. Derived facts can be derived from other facts every time it is needed. However, it could be necessary – for performance reasons- to remember some derived facts too. For example, the inventory level of products, or the material requirements plan in an industrial company.

So far, the aspects that have to be taken into account during the development of an implementation scenario are discussed from the perspective of coordination and production. It is based on the different aspect models of the essential model. The point of view was the fulfillment of the actor role. We only use the conceptual model of the production world for determining the scope of the EIS. However, this conceptual model also ought to be engineered into a part of the stated implementation scenario. This is already done by Halpin [Halpin 2001]. The C-world as well as the P-world can be recorded in a language that is derived from *Object-Role Modeling*, ORM in short. ORM simplifies the design process by using natural language, expressing the model in terms of natural concepts like *objects* and *roles.* Halpin [Halpin, Evans et al. 2003] developed an algorithm for mapping a design modeled in ORM onto a normalized relational database schema. Using a tool, the conceptual design can be entered in either graphical or textual form, and automatically mapped onto a relational schema for use in a variety of relational DBMSs. Thus the production world that is modeled from the ontological perspective in the FM can be transformed into a production world that is described from the implementation perspective. We do not make this transformation a subject for further research in our study.

4.6 Examples of Implementation Scenarios

We look again at the example of the Pizzeria as presented in section 3.2. Assume that the actor O-A02 requests a price list occasionally. We discuss various different implementation scenarios for sharing the price list. In doing so we will understand how the concepts of 'content', 'form' and 'medium' are given a concrete shape in an implementation model.

In dealing with the 'requested' agendum of transaction T02, O-A02 inquires the current price of the requested pizza kind (cf. Fig. 3.2). Figure 3.1 exhibits the CM of the Pizzeria. This model shows that the facts with regard to pizza kinds are contained in the external production bank PB60 and that the current prices are contained in the internal production bank PB05. Ontologically, the price of a pizza kind is requested for each transaction O-T02. Concerning the I-organization, Figure 4.2 exhibits a diagram that is a limited version of the diagram that is exhibited by Figure 3.14. The information in the banks PB60 and PB05 is recalled by I-A602 and I-A0531, respectively. Actor I-A14 then needs to derive the current price per pizza kind and to share this with I-A0031. This actor shares all P-facts which are invoked by O-A02. In response to the requests from I-A14 the actor I-A0531 searches for facts in O-PB05. D-A0531 subsequently asks actor P-A0000 to transmit the requested fact to a location that is accessible by I-A0531. After interpreting the document,

I-A0531 creates the C-fact ‘state’ in I-T0531. Along with this C-fact the corresponding P-fact of I-A0531 is sent to I-A14. Next, along with the C-fact ‘state’ in I-T14 the corresponding P-fact of I-A14 is sent to I-A0031, the ‘sharer P-facts. Finally, the corresponding P-fact of I-A0031 is sent to O-A02 by the C-fact ‘state’ in I-T0031.

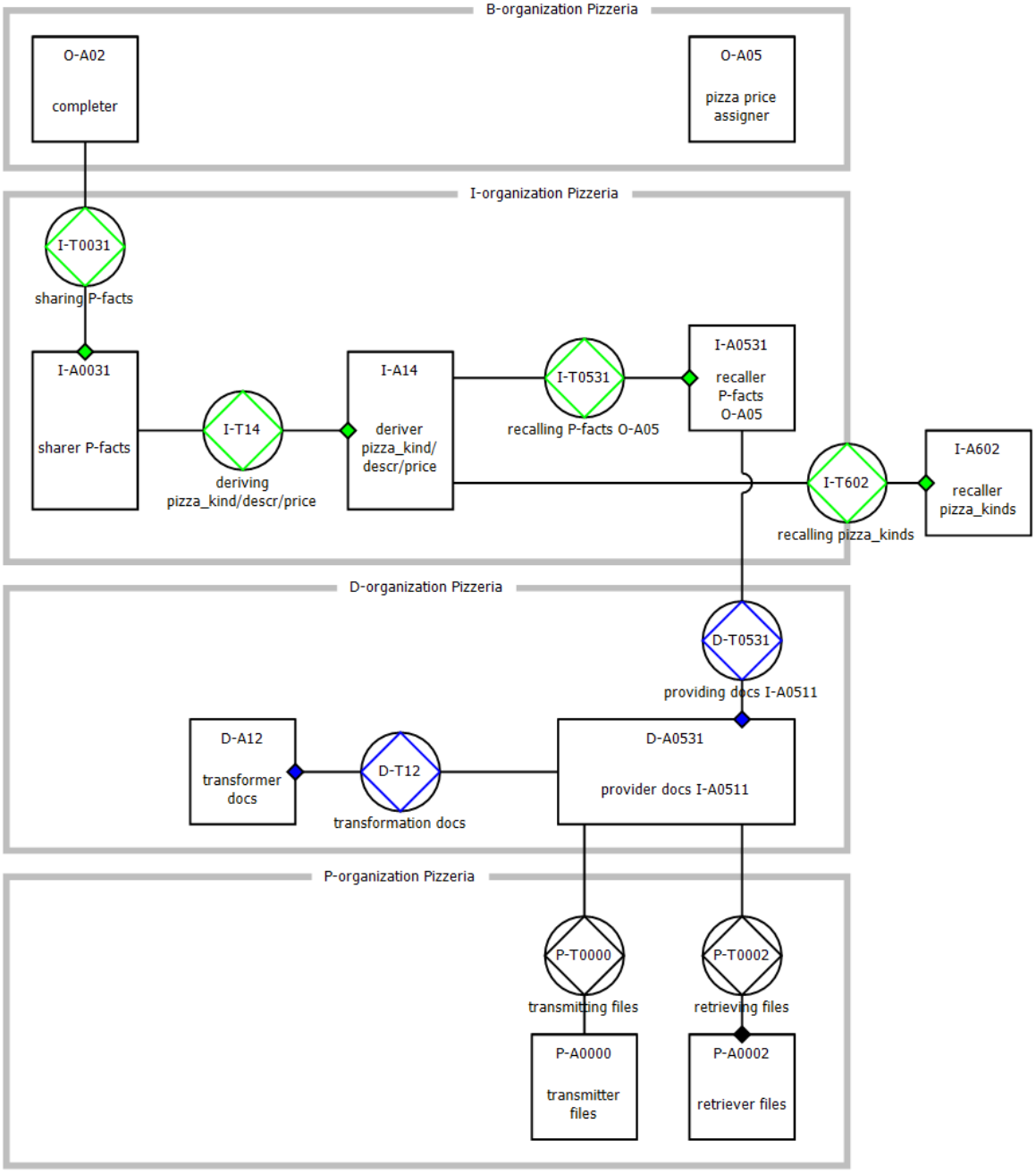


Fig. 4.2 Part of the IAM for engineering the price list

We shall discuss the next different implementation scenarios.

Scenario 1:
The actor roles I-A0031 and I-A14 are fulfilled by John and the actor roles I-A0531, D-A0531, and D-A12 are fulfilled by Tom. John asks Tom for recalling the current price of pizza kind PK. Note that John actually asks two things.

First of all, he asks for the recall of the current price of PK (infological) and secondly he asks for sharing the price with him verbally (physical). We assume that the facts which were created by Tom as O-A05 are remembered in the mind of Tom. Tom (as I-A0531) searches in his mind for the current price of PK. That fact must be transformed into a sound block by D-A12. John, as I-A14, receives the sound fragment with the 'state' of I-T0531 and interprets it subsequently in order to give an 'accept' or a 'reject'.

The pizza kinds are remembered in an external bank and are recalled by the external actor I-A602. This bank is not accessible from an internal actor. Therefore, within the scope of the EIS an internal 'recaller' must be implemented. This 'recaller' has to be fulfilled by a delegate of the external actor I-A602, for example Tom. The external 'recaller' and Tom must make a mutual agreement about the exchange of pizza kind facts.

Scenario 2:
Suppose Tom transmits his P-fact by an e-mail message to John. Although the P-fact of Tom remains unchanged, the form in which his production result is expressed differs from the previous scenario. Tom transforms his production result to natural text, instead of to a sound block. John, as I-A14, receives the text fragment with the 'state' of I-T0531 and interprets it subsequently in order to give an 'accept' or a 'reject'.

Scenario 3:
We change the way of performing the P-act of Tom as I-A0531. Up to now, the production algorithm has been stored in Tom's mind. That makes the P-act vulnerable; it is too dependent of a particular subject. This can be resolved by putting the algorithm on an external medium. The facts that Tom creates as O-A05 are no longer be remembered by him. The algorithm has to obtain facts from an external medium too. Tom needs a software program to find the current price of PK. The production result of I-A531 remains unchanged. The software imprints the result in an email message. John, as I-A14, receives the text fragment with the 'state' of I-T0531 and interprets it subsequently in order to give an 'accept' or a 'reject'.

Scenario 4:
In this scenario Tom fulfills no longer the actor role I-A0531. He is replaced by an agent. The same holds also for D-A0531. D-A12 is not implemented at all. John initiates I-T0531 on his computer. The agent receives the 'request' from the 'sharer agenda' and gives an implicit 'promise'. The agent imprints its production result in an email message which is sent with a 'state' to John.

Scenario 5:
According to the AM in Figure 3.2 the price of a pizza kind must be recalled for each order. It also shows that the price of a pizza kind remains unchanged for a specific period.

The conclusion could be that it is a waste of time and energy to share the current price of the same pizza kind several times per day. It is more efficient to print the pricing list once and to put it next to the counter. To this end, the ontological model of the Pizzeria must be changed: O-A02 invokes a new derived fact, namely a pricing list which is produced by I-A14. Suppose that Bill, who fulfills actor role O-A02, asks John to deliver a list with the current prices for all available pizza kinds. John shares this list with O-A02 by the 'state' of I-T0031. In case John is replaced by an agent, Bill initiates I-A0031 on his computer and receives the pricing list through his computer.

Let us discuss the question of responsibility for the quality of the pricing list. Besides John and Tom, Steve is involved in producing the list. Steve fulfills the external actor role O-A60. He creates, remembers and recalls pizza kinds. The only actors which are directly involved in the creation of the price list are I-A14, O-A05, I-A0531, O-A60, I-A602. These roles are fulfilled by John (I-A14), Tom (O-A05, I-A0531) and Steve (O-A60, I-A602). Tom and Steve create the original facts and are, for that reason, responsible for their quality. The pricing list, which is a derived fact, is created by John. John is responsible for the quality of the list. If these subjects are replaced by an agent, the situation differs. They could not be responsible for their performance. It is the allocator who can be held responsible if these subjects or agents do not operate according to the expectations. But, if appears that action rules or working algorithms are not designed correctly; the subject who is responsible for the implementation model is held responsible (cf. Table 4.1).

Lastly, we would like to come back to the discussion we had in the first scenario. It concerns the external bank with the pizza kinds, PB60. The question has to be answered why a fact bank is defined as external, this does not seem to be obvious. The answer is easy. If a CM has not been produced for the purpose of using it as a basis for the development of an EIS, it may happen that certain banks are positioned outside the organization for the reason that they are not relevant for the purpose for which the CM was designed. Therefore, it is advisable to see whether PB60 could be brought within the scope of the organization. This reduces the complexity of the EIS to develop.

4.7 Conclusion

In conclusion, the implementation model of the organization is based on the integral ontological model of the organization. It is implemented by assigning human actors or agents to the actor roles in the model. An EIS is defined as a software application that supports a set of O-actors. The support takes places by I-actors and D-actors that are mainly agents. In this chapter a large number of aspects have been identified which are important for designing an EIS on the basis of an ontological model of the B-organization, I-organization, and D-organization of the enterprise. It shows that a methodical approach for designing an implementation scenario that is made operational by an EIS is necessary. The next chapter will provide such an approach. It presents a framework for implementation design.

5 A Specification Framework for designing an EIS

5.1 Introduction

In the previous chapter the notion implementation scenario is discussed based on the integral model of the B-organization, I-organization and D-organization of an enterprise. We focused in particular on an implementation scenario for which we would like to develop an EIS. We defined an EIS as a software application that supports a set of O-actors. See section 4.4. These O-actors are supported by I-actors and the I-actors are supported by D-actors. See section 3.4.3 and section 3.5.3. These I-actors and D-actors are mainly agents in the EIS. In this chapter, a framework is presented in order to design an implementation scenario of an enterprise that is implemented by an EIS. The framework is discussed in Section 5.2. It is called the 'PID-Framework' ('PID' stands for 'Procedure for Implementation Design'). Based on this framework a number of practical rules for designing an EIS are presented in section 5.3. The chapter ends with concluding remarks in Section 5.4.

5.2 A Framework for Specifying the EIS

The PID-Framework that is exhibited in Figure 5.1 is intended for practitioners who want to specify an EIS or who wish to assess the suitability of an implemented EIS for an organization on the basis of the ontological model of the organization.

The proposed approach starts with selecting the O-actor roles that have to be supported by the EIS. After selecting these actor roles, the implementation of each actor role is elaborated. The elaboration is done on the basis of a number of aspects, which are grouped in the PID-Framework. Three aspect groups are discerned: general characteristics, sharing facts and remembering facts.

In the remainder of this section, all aspects of the aspect groups are discussed. In section 5.3, we provide one or more practical rules regarding each aspect.

The area within the organization that is covered by the EIS is determined within the aspect group *Context.*

- *Scoping* - refers to the definition of the scope of the EIS. As a basis for defining the scope of the EIS the production world of the organization is taken.
  1. First of all, all object classes that fall within the scope of the EIS are determined. These object classes are defined in the FM of the B-organization. Being covered by the EIS implies that all result kinds that are based on these object classes have to be defined in the EIS too. Next, all O-actor roles that are linked to these result kinds have to be supported by the EIS.

2. There may be binary facts within the scope of the EIS that have one domain (reference class) inside this scope and one outside. The question then is how to implement the external object class. Let us provide an example for clarification. Suppose that in the FM the internal object class SALES_ORDER refers to the external object class CUSTOMER through the binary fact kind 'the customer of [sales_order] is [customer]'. This binary fact kind can only be maintained within the EIS by creating a subset of the external object class CUSTOMER in the EIS. Only those instances of CUSTOMER are remembered and made available within the EIS that are considered relevant. So, the solution to the problem is that a subclass of the external object class is made internal to the EIS.
3. An O-actor in its informa shape can be an initiator of infological transactions with I-actors. If an O-actor is within the scope of the EIS, this inevitably implies that the executors of these infological transactions also fall within the scope of the EIS (see Rules 1, 2 and 3).

- *Clustering* – the sets of actor roles are determined for which it holds that they will be allocated to the same organizational function (see Rule 4).

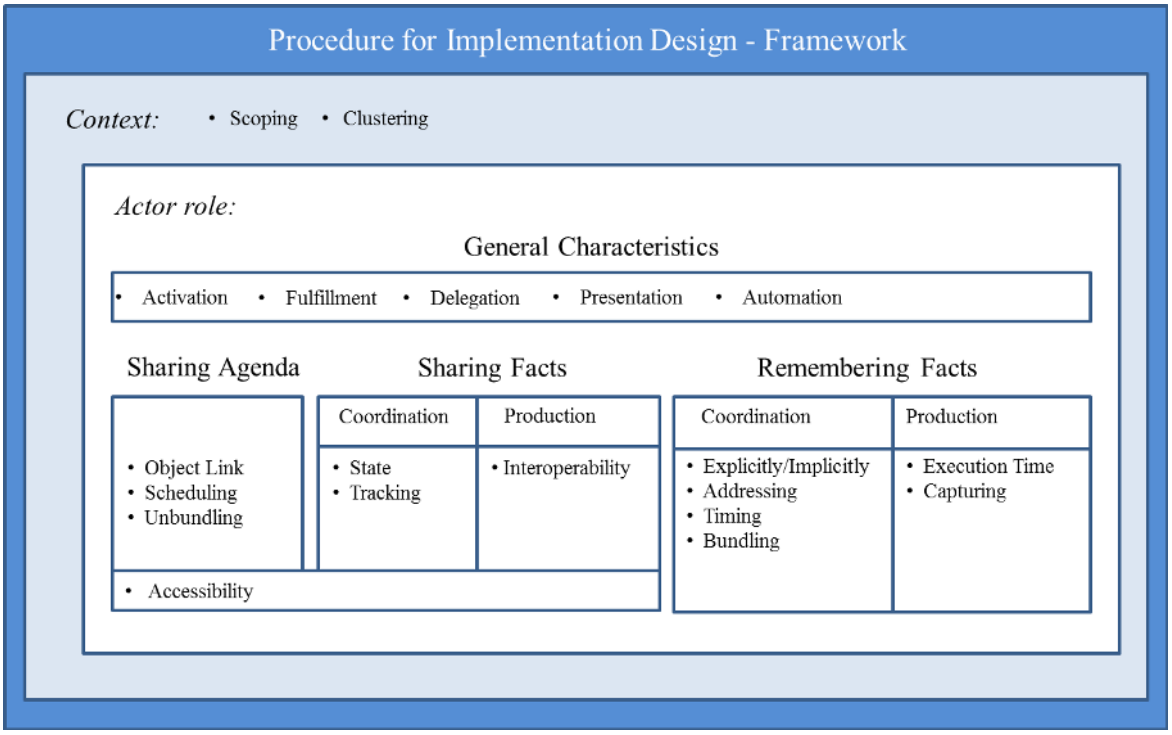


Fig. 5.1 The PID-Framework

Specific aspects of each actor role, based on a number of criteria, will be investigated in the aspect group *General Characteristics*. The following aspects are briefly covered:

- *Activation* –The actor cycle needs to be implemented for every O-actor role within the scope of the EIS (see Rule 5).

- *Fulfillment* – an actor role can be fulfilled by several subjects. The fulfillment can take place in two different ways [Dietz 2006]: concurrently and collectively (see Rule 6).
- *Delegation* – some transaction steps can be performed by somebody else instead of the authorized subject. This is called delegation. The authorized actor remains accountable (see Rule 7).
- *Presentation* - the work environment of an actor role has to be established. An O-actor is involved in remembering and sharing facts. For remembering, it must be able to capture facts in documents and to imprint the corresponding files on a medium. For sharing, it must be able to retrieve documents from mediums and to interpret these documents. A document is received through the 'state' of a 'sharing agenda' (cf. Fig. 3.12) or 'sharing facts' transaction (cf. Fig. 3.14). Physically, the document is transmitted from the executor to the initiator of the sharing transaction by an actor from the physical organization (see Rule 8).
- *Automation* - it should be determined to what extent the actor role is fulfilled by a subject, possibly supported by technology, or completely fulfilled by an agent (see Rule 9).

Next, we will discuss the aspect groups that are related to the O-actor roles. The specific characteristics of both kinds of actor activities are addressed in the *Sharing Facts* and *Remembering Facts* aspect groups.

First, the *Sharing Facts* aspect group will be discussed.

Agenda:
- *Object Link* - if it is not at the top of a process tree, an agendum always links to a specific object. This object is sent along with the C-fact to which the agendum refers. So, the object is available for the actor without an explicit request for a P-fact (see Rule 10).
- *Scheduling* - if there are several actors able and allowed to respond to an agendum, it could be necessary to implement a scheduling mechanism in order to distribute the agenda on the basis of available time of the actors, or on specific competences of the actors, or on the basis of other internal priority rules. The scheduling mechanism takes the facts that are related to the object that is received along with the agendum into account (see rule 11).
- *Unbundling* - an agendum can consist of a bundle of C-facts. For example, suppose that an initiator has submitted a combined request for payment for delivered products to an executor. The executor accepts or rejects the payment of the bundle. However, in the ontological model separate payment transactions are defined for the delivered products. That means that the initiator receives a bundle of 'stated' C-facts with the 'stated' agendum which must be responded to by an acceptance of each paid product separately. Bundling can be done on the basis of specific attributes of the object to which it relates (see Rule 12).

Coordination facts:
- *State*—during a transaction process, the P-fact is at any moment in a certain state (e.g. requested, promised, declined, stated, accepted, or rejected). This state can be inspected (see Rule 13).

- *Tracking* - To each C-fact is linked the identification of its performer. That makes it possible to gain more insight into 'who' is responsible for a specific transition of the C-world or P-world of the organization. This is important from the viewpoint of transparency and accountability (see Rule 14).

Production facts:

- *Interoperability*—it is important to ensure that an actor assigns the semantic meaning to a received fact that is intended by the executor of the fact sharing transaction. One must be particularly alert to this with facts that are shared from outside the organization. This issue is also referred to as 'semantic interoperability' (see Rule 15).

General:

- *Accessibility*—an O-actor initiates an infological transaction with the 'sharer agenda' for a request of sharing the agenda, with the 'sharer P-facts' for a request of sharing P-facts and with the 'sharer C-facts' for a request of sharing C-facts. These actors are invoked by each O-actor and know all C-fact and P-fact 'recallers', respectively, and P-fact 'derivers' within the EIS. The 'sharer agenda' shares agenda, dependent on the addressees of the C-facts, with the O-actors in EIS. The 'sharer C-facts' and the 'sharer P-facts' shares facts with the O-actor, when asked for.
  The C-facts and P-facts that are needed by actors within the EIS may be created outside the scope of EIS. Such C-facts and P-facts are unknown to the 'sharer agenda', 'sharer C-facts' and 'sharer P-facts'. This problem has to be solved by defining a 'recaller C-facts' actor role within de scope van EIS for each external 'recaller C-facts' of the concerning C-facts and a 'recaller P-facts' actor role for each external 'recaller P-facts' of the concerning P-facts. These new actor roles within the EIS are fulfilled by delegates of the external actors that fulfill 'recaller C-facts' and 'recaller P-facts' actor roles (see Rules 16, 17 and 18).

Next, the *Remembering Facts* aspect group will be discussed:

Coordination facts:

- *Explicitly/Implicitly*—C-facts are created explicitly or implicitly. A C-fact is created implicitly when there is an underlying agreement between the performer and addressee, such that the performer does not need to perform an explicit act. Actually, the C-fact is remembered and recalled by a delegate within the EIS (see Rule 19).
- *Addressing*—although ontologically each C-fact has an addressee, this addressee is usually not known in practice when an actor creates a 'request'. If this is the case, a fictitious addressee is assigned to the C-fact (see Rule 20).
- *Timing* - the performer of the C-fact is understood as the owner of the C-fact. The ownership is designated by a timestamp showing when the ownership came into being (see Rule 21).
- *Bundling*—C-facts can be bundled. An example is a sales order that consists of several order lines. Every line is a separate C-fact. The bundled facts belong to the same sales order because of common attributes, like customer and date. A request for payment is always executed for the purpose of the bundle. After receiving the 'state', the 'accept' or 'reject' of the payment is also executed for the bundle (see Rule 22).

Production fact:

- *Execution Time*—the production act has to be executed within the intended settlement time. The actual creation time of the P-fact is equal to the creation time of the C-fact 'accepted' by the initiator of the transaction. The actual creation time can differ from the P-event time which can also lie in the past or in the future. There must be implemented a monitoring mechanism for monitoring the completion of the transaction (see Rule 23, 24).
- *Capturing*— an O-actor can create instances of a binary fact kind of which one of the domains is an external object class. Then, a subset of the external object class is created in the EIS. Only those instances are remembered and made available within the EIS that are considered relevant (see Rule 25).

5.3 Rules for Practitioners

This section provides rules to practitioners for each aspect in the PID-Framework.

Context

1. *Determine the set of O-actor roles and the related transaction kinds that must be supported by the EIS.*

   First, determine the object classes that have to fall within the EIS, from the FM of the B-organization. Second, determine the result kinds are linked to these object classes. Third, determine the O-actor roles on the basis of the result kinds.

2. *Determine the part of the I-organization and D-organization that has to be covered by the EIS.*

   With the capture of the O-actors within the scope of the EIS, the I-actors that remember created facts, the I-actors that make facts available, the I-actors that derive facts, the D-actors that archive the corresponding documents and the D-actors that make the documents available, fall within the scope of the EIS.

3. *Determine the binary fact kinds within the scope of the EIS that have one domain (reference class) inside this scope and one outside.*

   Such binary fact kinds can only be maintained within the EIS by creating a subset of the external object class in the EIS.

4. *Determine the actor roles within the scope of the EIS that will be allocated to the same organizational function.*

   Actor roles are grouped in clusters. Each cluster is allocated to an organizational function. The identification of clusters is important, because of the possibility for (1) opti-

mizing the implementation of the mutual coordination acts within a cluster, and (2) optimizing fact sharing.

Actor – General Characteristics

5. *The actor cycle needs to be implemented for every O-actor role within the scope of the EIS.*

   An actor actively waits for an agendum. An agendum corresponds with a C-fact that has the actor as its addressee. On the request by the O-actor the 'sharer agenda' shares the next agendum to deal with.

6. *Actor roles may be fulfilled by several subjects, either concurrently or collectively. Two different methods of implementation must be distinguished.*

   There are the following two ways of fulfillment:
   i) In concurrent fulfillment, two different situations may arise: (1) subjects who are fully interchangeable and who perform the same activities, (2) subjects who each have their own specialization within the actor role. E.g., salespeople who are specialized in specific countries or, a teacher who is specialized in a particular field. A request to a math teacher to provide a lesson should not be directed to a teacher English language.
   ii) Subjects who collectively create a P-fact can be held collectively accountable for that fact (e.g. a management team).

7. *Specify the acts that may be performed by other subjects than the actor itself.*

   Authorized actors may delegate specific coordination acts and/or production acts to other subjects. The authorized actors keep the responsibility for the results of their delegates. The coordination between an actor and its delegate is considered to be out of the scope of this study.

8. *Because O-actors request for remembering and sharing facts, a good design of the working environment takes into account specific requirements for the presentation of these transaction results.*

   For remembering, actors must be able to capture facts in documents and to imprint the corresponding files on a medium. For sharing, actors must be able to retrieve documents from mediums and to interpret these documents. Take into account that the actor role can be identified as a part of a cluster. The document is received through the 'state' of a 'sharing agenda' or 'sharing facts' transaction. Physically, the document is transmitted from the executor to the initiator of the sharing transaction by an actor from the physical organization. The used technology determines how the fact is received. For example, a sound fragment can be received by telephone or in a face-to-face conversation and a text fragment can be mailed or displayed on the screen of a computer or put on paper through a printer.

9. *In principle, O-actor roles are fulfilled by subjects. They can be supported by information technology. Determine to what extent each actor role should be automated.*

   Actor roles can be fulfilled partly or fully by information technology. O-actors are not excluded, because of mimicking by information technology. The extent to which information technology is used differs. If an actor role is entirely fulfilled by software, one speaks of an agent. A subject who fulfills an actor role is fully responsible for the facts that it creates. Information technology cannot hold any accountability. Then, its allocator is responsible for the quality of its results. An actor must always be uniquely identified within the operational system so that it can be linked to the facts it creates.

Actor - Sharing facts

Agenda:

10. *Every agendum refers to a P-fact concerning some object. This object is accessible for actors that fulfill the actor role.*

    If it is not at the top of a process tree, an agendum always links to a specific object. This object is sent along with the C-fact to which the agendum refers. So, the object with dependent and independent facts is available for the actor without an explicit request for a P-fact. If is at the top of the process tree, an object is created during the 'request' C-act. Along with this C-act dependent facts are sent from the performer of the C-act to the addressee.

11. *An agendum can be assigned to any actor that fulfills an actor role concurrently with others. In that case a scheduling mechanism may have to be implemented.*

    If there are several actors able and allowed to respond to an agendum, it could be necessary to implement a scheduling mechanism in order to distribute the agenda on the basis of available time of the actors, or on specific competences of the actors, or on the basis of other internal priority rules. The scheduling mechanism takes the facts that are linked to the object that is received along with the agendum into account.

12. *An agendum may be a bundle of agenda. Such an agendum has to be unbundled.*

    An agendum can consist of a bundle of C-facts. For example, suppose that an initiator has submitted a combined request for payment for delivered products to an executor. The executor accepts or rejects the payment of the bundle. However, in the ontological model separate payment transactions are defined for the delivered products. That means that the initiator receives a bundle of 'stated' C-facts with the 'stated' agendum which must be responded to by an acceptance of each paid product separately.

Coordination facts:

13. *At any moment, a P-fact has a specific state, which corresponds with the progress of the transaction. Actors must be able to inspect the state of the P-fact.*

    During the transaction process, a P-fact is at any moment in a certain state (e.g. requested, promised, declined, stated, accepted, or rejected).

14. *To each C-fact must be linked the identification of its performer. To each P-fact must be linked the identification of its creator.*

    The identification of the creator gives each fact a 'face'. It makes the EIS more transparent. Consequently, this should increase the quality of information and the accountability of the actors.

Production facts:

15. *The semantic meaning of a fact that is requested by an initiator must be equal to the semantic meaning of the fact that is shared by the executor of the fact sharing transaction.*

    The exchange of facts between different actors can only take place whether there is interoperability of the semantic meaning of the facts. The semantic interoperability is present within the boundary of the organization, due to the common production world of the actors in an organization. One must be alert to it with facts that are requested from outside the organization.

General:

16. *An agendum or a fact must be accessible by the actor that uses the agendum or the fact.*

    An O-actor initiates an infological transaction with the 'sharer agenda' for a request of sharing the agenda, with the 'sharer P-facts' for a request of sharing P-facts and with the 'sharer C-facts' for a request of sharing C-facts. These actors are invoked by each actor in the EIS and know all C-fact and P-fact 'recallers', respectively, and P-fact 'derivers' within the EIS. The 'sharer agenda' shares agenda, dependent on the addressees of the C-facts, with the O-actors in EIS. The 'sharer C-facts' and the 'sharer P-facts' shares facts with the O-actor, when asked for.
    The C-facts and P-facts that are needed by actors within the EIS may be created outside the scope of EIS. Such C-facts and P-facts are made available by newly created 'recaller' actor roles within the EIS that are fulfilled by delegates of the external 'recallers'. P-facts that are exchanged by the concerning O-actor with other O-actors along with a C-fact can be obtained from the 'sharer C-facts'. This does not hold for P-facts that did not come into existence. Figure 3.6 exhibits that 'remembering' of the P-fact takes

place after its acceptance by the initiator of the transaction (cf. Fig. 3.6).

17. *C-facts from external banks must be captured in the EIS in order to be able to share them with actors within the scope of the EIS.*

    An efficient obtain procedure is needed in order to transmit the external generated C-facts in EIS so that these C-facts can be recalled by a delegate of the external 'recaller' within the scope of the EIS. The obtain procedure can be implemented by using information technology or by using a manual procedure by which, for example, the external 'recaller' calls or mails a secretary for making a 'request' available for other actors within the scope of the EIS.

18. *P-facts from external banks must be captured in the EIS in order to be able to share them with actors within the scope of the EIS.*

    An efficient obtain procedure is needed in order to transmit the external generated P-facts in EIS so that these P-facts can be recalled by a delegate of the external 'recaller' within the scope of the EIS. Similar to the procedure for obtaining C-facts from external banks, the obtain procedure for P-facts from external banks can be implemented by using information technology or by using a manual procedure by which, for example, the external 'recaller' calls or mails a secretary for making a fact available for other actors within the scope of the EIS. In case of using information technology, there are three ways of getting P-facts from outside the EIS. Suppose that the external 'recaller' provides result R in response to question A. The following situations may arise: (1) question A always has the same R (e.g. provide John's date of birth), (2) question A always has a different R (e.g. provide the current stock of item B), and (3) R changes periodically (e.g. provide the current training schedule). In the first situation the possibility must be considered that question A is treated within the EIS, i.e. the infological production act should be implemented within the EIS. This can also be considered for the third situation. It depends on the degree of periodicity. Redundant capturing of facts is not a solution for the second situation. Then, an interface between de external 'recaller' and the internal 'recaller' must be developed.

Actor - Remembering facts

Coordination facts:

19. *Determine whether a C-fact must be created implicitly or explicitly.*

    C-facts are created explicitly or implicitly. A C-fact is created implicitly when there is an underlying agreement between the performer and addressee, such that the performer does not need to perform an explicit act. Then, the C-fact is remembered and recalled by a delegate within the EIS. Take into account that the delegate can use a software tool that actually supports him in fulfilling the mentioned actor roles.

20. *If the intention of the C-fact is 'requested', it can contain a fictitious addressee as the executor.*

    In many cases, the performer of the C-fact 'requested' does not have a specific addressee in mind. It does not know specific addressee identification or it does not prefer someone. Therefore, a fictitious identification for the addressee is assigned to the C-fact.

21. *The performer of the C-fact is understood as the owner of the C-fact. The ownership is designated by a timestamp showing when the ownership came into being.*

    Remembering the creation time of the C-fact is important for tracking purposes. Besides that, the P-fact to which the C-fact refers has to be created within the intended settlement time. Therefore, the creation time of the C-fact must be remembered.

22. *It must be possible for an actor to bundle C-facts that meet a specific criteria and create one single C-fact for the entire bundle.*

    A C-fact to create can be comprised of a set of C-facts that complies with one or more specific attributes which are specified in the FM of the organization.

Production facts:

23. *The P-act has to be executed within the intended settlement time.*

    The actual creation time of the P-fact is determined when the initiator of the concerning transaction creates the C-fact 'accepted'. There must be implemented a monitoring mechanism for monitoring the completion of the transaction.

24. *The actual creation time of a P-fact can differ from the P-event time.*

    Sometimes the initiator wants something to start to exist in retroaction. An example is the start of a subscription with a starting day in the past. However, it would also be a future point of time; it will take until then before the fact becomes existent. The time that the P-fact actually becomes real is called the P-event time. The P-event time can differ from the actual creation time of the P-fact.

25. *An O-actor can create instances of a binary fact kind of which one of the domains is an external object class. Then, a subset of the external object class is created in the EIS.*

    Only those instances are remembered and made available within the EIS that are considered relevant.

5.4 Conclusion

The PID-Framework has been developed in this chapter. The framework provides a number of practical rules for designing EIS based on the coherent ontological model of the B-organization, I-organization and D-organization. The actor role is the central element in this approach. The EIS is aimed to support subjects who fulfill these actor roles. Actor roles within the I-organization and D-organization are fulfilled by agents.

PART III

Applications

6 Case 1: Conciliation Board for Consumers

6.1 Introduction

The case study concerns a project that was performed at the Conciliation Board for Consumers (also known as The Foundation for Consumer Complaints Boards and in Dutch abbreviated to 'SGC': Stichting Geschillencommissies Consumentenzaken). The case study was not executed by the researcher personally. The necessary information about the project is obtained through an interview with the person responsible for executing the original case study and the project manager of the SGC project. Furthermore, the researcher has been guided by various publications about this project [Mulder and Van Reijswoud 1997; Goedvolk 2006; Mulder 2006]. The SGC case description dates from 1996. The SGC in its current state cannot be compared with the organization at the time of the case-study.

The following introduction of the SGC is largely derived from [Mulder and Van Reijswoud 1997]. The Conciliation Board for Consumers is a non-profit organization with the aim to negotiate quickly, less costly and easy solutions for disagreement between consumers and suppliers. In 1996, SGC operates in 22 different areas including for example banking, travelling, and postal and telephone services. For each of these areas there is a separate committee. The mediation fees of both consumers and suppliers, supported by the branch organizations, cover the costs of the mediation. Individual suppliers only pay a mediation fee if the board proves the supplier to be in the wrong. The remaining costs of the mediation are covered by the branch organizations. Consumers always contribute when they submit a request for mediation. If a case is submitted for mediation, the board passes a binding advice.

The Conciliation Board for Consumers exhibits the following organizational structure. One director manages SGC; he is also secretary of the 22 different committees. A general secretariat and a group of legal advisors support the Director. For each of the 22 committees there is a secretariat which acts as a coordinator. Associated with the SGC there are the committees, each composed of an independent member, a member of the Dutch consumers' organization and a member for the suppliers. Affiliated to the SGC are experts that can be consulted for an expert opinion. A request for mediation of the Conciliation Board for Consumers needs to be started with a letter of the consumer in which the nature and the magnitude of the complaint is explained. On the basis of this letter a first rough selection is made as to whether the complaint is taken into consideration. The complainer is informed as to whether the complaint will be mediated by SGC or not. If the complaint is considered to be admissible, this is the case for about 70% of the complaints, SGC sends the consumer a complaint form that has to be used to officially file the complaint for mediation. The complaint form consists of a questionnaire in which the necessary details of the complaint are asked. In the same letter, the consumer is also requested to pay a complaint fee, and

possibly to deposit the remaining amount of the invoice. When the complaint is filed, the supplier is requested for a bank guarantee if there is no collective guarantee by the branch organization. When the request for mediation by SGC is granted, the supplier is informed by mail. At the same time the supplier is requested for his defense or may propose an agreement. In addition to the complaint of the customer and the defense of the supplier, the board can initiate an expert examination. All documents of the consumer, supplier and possibly the experts together serve as the input for a meeting of the board for which all the parties involved are invited. In this meeting the committee reaches a decision. This winding up comprises the complaint fee, the deposit of the remaining amount of the invoice, and the expenses of the members of the committee and the experts. If the supplier fails to comply with his terms of payment, the consumer can appeal to a regulation that assures payment. At the same time the branch organization of the supplier is informed. Then, the file is closed.

6.2 Context

Figure 6.1 exhibits the CM of SGC and the corresponding Transaction Result Table. The actor O-A01 receives the request for mediation. O-A01 defines a COMPLAINT and determines whether the request can be processed. If that is the case the applicant is informed about this in writing.

A standard questionnaire is then also sent to the applicant. The applicant is expected to fill out the form correctly and to return it to O-A02. Agreements with the consumer and the supplier regarding the financial settlement are made on the basis of this form by O-A07 and O-A06, respectively. The mediation process only starts once these arrangements have been made. The actor roles O-A04, O-A05, and O-A03 are specified within the scope of the organization. This is different from the model presented by Mulder [Mulder 2006]. Mulder positioned these actor roles as external roles of the organization. Therefore, they are considered out of scope. We consider these roles within scope in order to make it possible to implement their transactions within the organization.

This study concerns the implementation design of an EIS that supports the actors O-A01 through O-A07. O-A04, O-A06 and O-A07 are left out in order to avoid more of the same models. Figure 6.2 exhibits the FM. The relevant part of the AM will be shown when discussing the individual actor roles.

Transaction Kind	**Result Kind**
O-T01 - admission	O-R01 - [complaint] has been admitted
O-T02 - mediation	O-R02 - [complaint] has been mediated
O-T03 - defense	O-R03 - [complaint] has been defended
O-T04 - expert advice	O-R04 - [complaint] has been adviced
O-T05 - judgment	O-R05 - [complaint] has been judged
O-T06 - s_payment control	O-R06 - s_payment [complaint] has been controlled
O-T07 - c_payment control	O-R07- c_payment [complaint] has been controlled

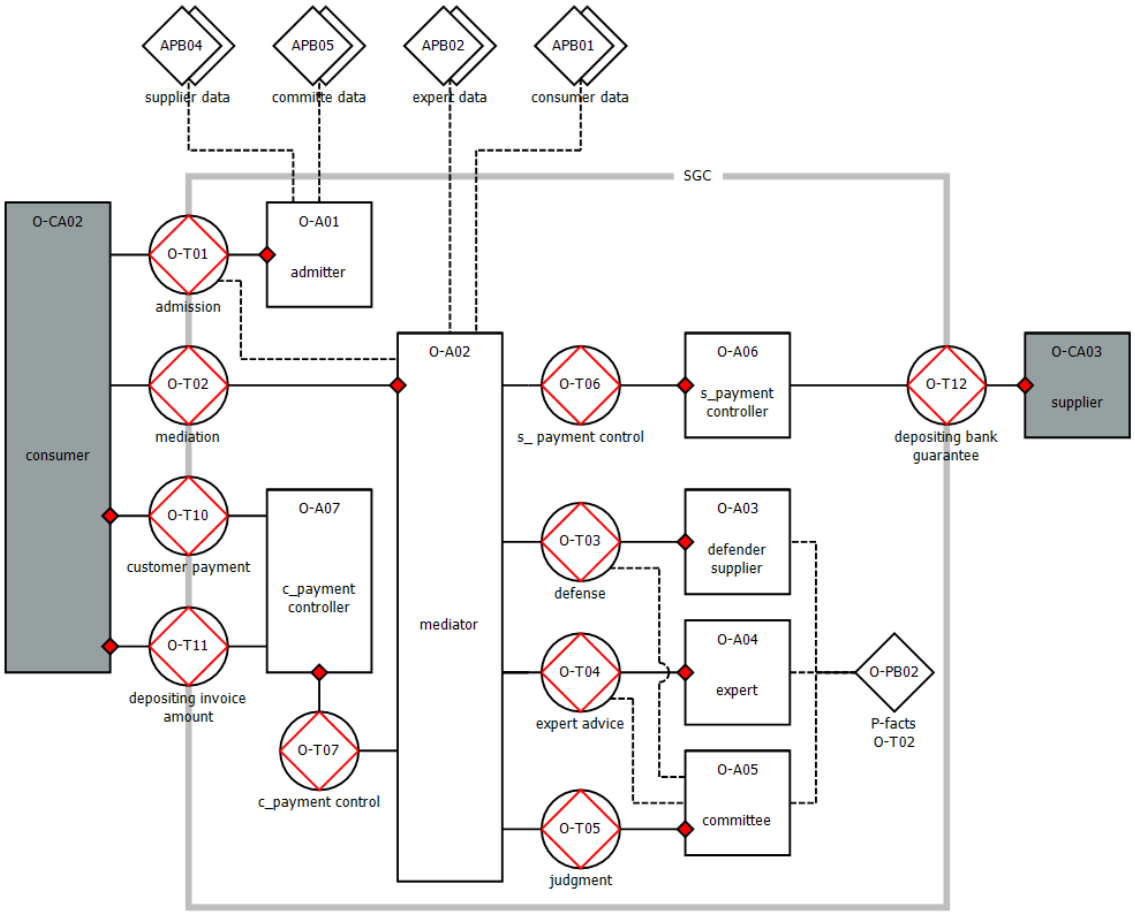


Fig. 6.1 Construction model of SGC

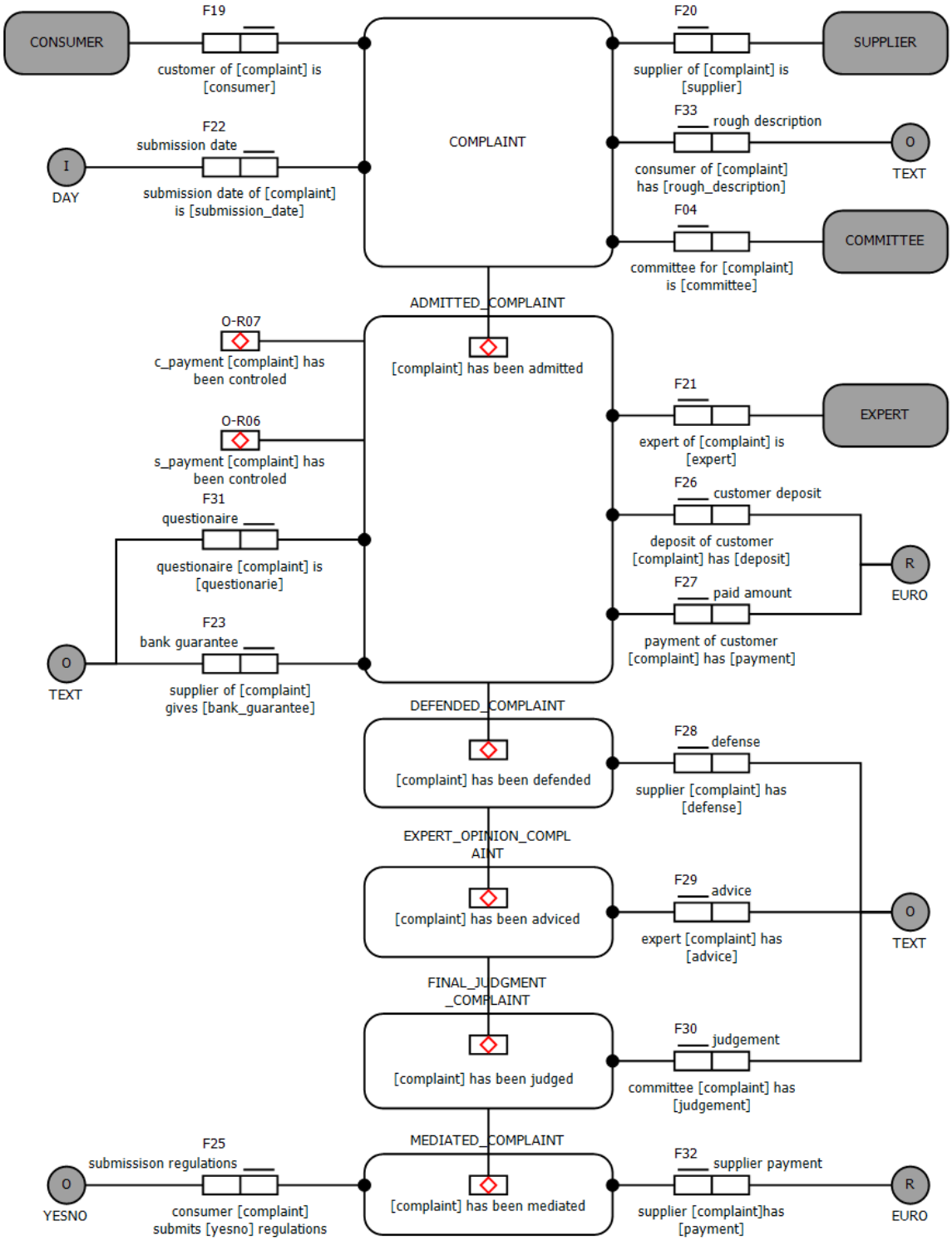


Fig. 6.2 Fact Model of SGC

The first step to take is the establishment of the scope of the EIS system that supports the mediation within SGC. Therefore, we look at the PID-Framework and discuss the rules 1 till 4. The numbering corresponds to the numbering of the rules.

1. The result kinds which are linked to the COMPLAINT object class (cf. Fig. 6.2) are used to determine the actors that have to be supported by the EIS. These result kinds are linked to the actors O-A01 through O-A07.
2. We will refrain from discussing here the question of which I-actors and D-actors fall inside or outside the scope of the EIS. We will deal with this topic in more detail when discussing the individual actor roles.
3. There are four binary fact kinds within the scope of the EIS that have one domain inside its scope en one outside. These are 'customer of [complaint] is customer', 'supplier of [complaint] is supplier', 'expert of [complaint] is expert', and 'committee for [complaint] is committee'. The external domains are CONSUMER, SUPPLIER, EXPERT and COMMITTEE, respectively.
4. There is no clustering at the level of the ontological actors

6.3 Actor Roles

Next, the implementations of the actor roles O-A01, O-A02, O-A03 and O-A05 are discussed.

6.3.1 Implementation of O-A01

A request for mediation of the Conciliation Board for Consumers needs to be started with a letter of the consumer in which the nature (concerning housing, computers, travels etc.) and the magnitude of the complaint is explained as well as the name and address of the supplier is included. This may be a simple letter. On the arrival of a letter, a file is opened. The file is identified by the committee the complaint relates to, a unique identification number of the complaint, the name of the complainer and the date that the complaint was submitted. In the course of the procedure the file is used to archive additional information.

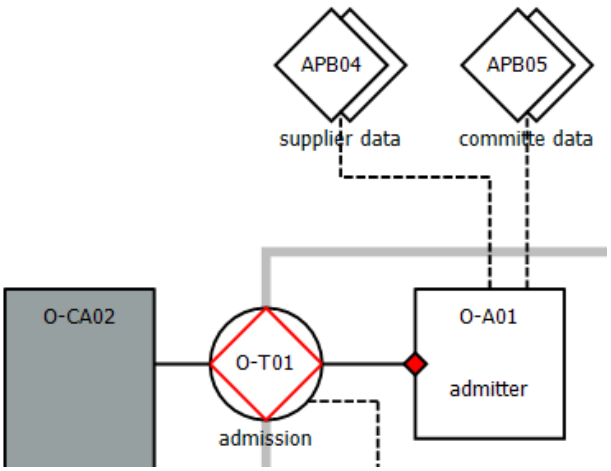


Fig. 6.3 CM: Complaint admission

On the basis of the first letter a first rough selection is made as to whether the complaint is taken into consideration. The complainer is informed in a letter as to whether the com-

plaint will be mediated by SGC or not. This letter is sent by mail to the address of the consumer as was indicated in the first letter. If the complaint is considered to be admissible, SGC sends the consumer a complaint form that has to be used to officially file the complaint for mediation. The relevant part of the CM of SGC is exhibit in Figure 6.3. Figure 6.4 shows the part of the AM that contains the action rules for O-A01.

Actor O-A01:

	when admission **for** new Complaint is requested **with** **the** consumer **of** Complaint **is** a Consumer **the** supplier **of** Complaint **is** a Supplier **the** admission date **of** Complaint **is** current date **the** rough description **of** Complaint **is** Rough_description	O-T01/rq
	assess *justice:* **the** Performer **of the** request **is the** consumer **of** Complaint *sincerity:* <no specific condition> *truth:* <supplier data is available> <committee data is available>	
	if *complying with request is considered justifiable* **then** promise admission **for** Complaint **else** decline admission **for** Complaint	

	when admission **for** Complaint is promised	O-T01/pm
	assess *justice:* **the** Performer **of the** promise **is the** admitter **of** Complaint *sincerity:* <no specific condition> *truth:* <no specific condition>	
	if *complying with promise is considered justifiable* **then** execute admission **for** Complaint state admission **for** Complaint	

Fig. 6.4 Action rules for O-A01

O-A01 requests for the following services:

Remembering facts:
a. C-facts and P-facts of O-A01 are remembered by I-A0110 and I-A0111, respectively.

Sharing agenda:
b. Agenda for O-A01 are shared by I-A0020 on request by O-A01.

Sharing facts:
c. Supplier data for O-A01 are shared by I-A0031 (I-T1131)
d. Committee data for O-A01 are shared by I-A0031 (I-T1231)

Figure 6.5 exhibits the I-organization with the I-actors that are invoked by O-A01. The dark colored part is covered by the EIS to be developed. It turns out that this only applies to the 'rememberer C-facts', 'rememberer P-facts' and the 'sharer agenda'. However, I-A0020 has a problem with the accessibility of I-CA0230. See rule 17 for the proposed so-

lution. All requested P-facts are delivered from outside the organization. SGC decided not to implement these facts in the EIS.

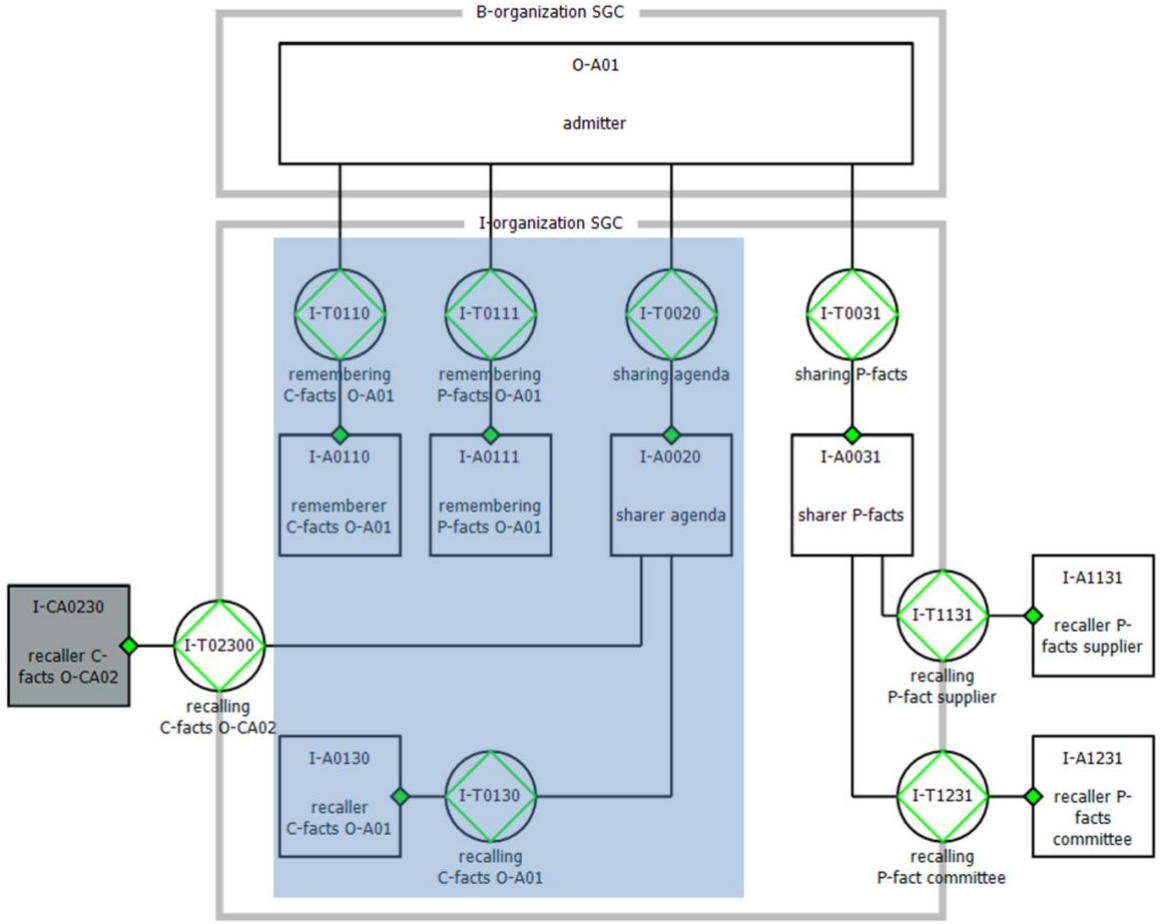


Fig. 6.5 Remembering and Sharing Facts for O-A01

The discussion about the implementation of O-A01 in the EIS continues on the basis of the rules from the PID-Framework related to the *Actor General Characteristics*.

5. O-A01 is activated when a request for mediation is received in the form of a letter. O-A01 then performs an initial check to see whether it makes sense to start a mediation process. The committee to which the complaint relates is also identified.
6. O-A01 can be fulfilled by several people who are completely interchangeable.
7. It is unclear whether there are appointed delegates for specific acts of O-A01.
8. The infological transactions ‘rememberer C-facts’, ‘rememberer P-facts’ and the ‘sharer agenda’ have to be implemented in the EIS. The C-fact ‘request’ from O-CA02, along with the letter, has to be scanned and entered into the EIS by an employee of SGC. This subject acts as a delegate of O-CA02 in this respect. This employee of somebody else has to be understood as the ‘recaller’ of the C-facts of O-CA02 in the EIS. It is requested by the ‘sharer agenda’ for sharing its C-facts. O-A01 receives the scanned letter from the ‘sharer agenda’ through the EIS. All requested P-facts are de-

livered from outside the EIS (I-T1131, I-T1231). The coordination with respect to these P-facts takes place in the Dutch language through letters, emails and phone calls.
9. O-A01 is a business actor and cannot be replaced by an agent. Facts created by O-A01 are remembered within the scope of the EIS. They can be recalled in the EIS and are made available for the purpose of O-A02 (cf. Fig. 6.1).

The aspects in the *Sharing Agenda* aspect group are discussed hereafter:

10. O-A01 is on the top of a process tree. The business process is started by O-CA02. Therefore, the agendum 'requested' that O-A01 receives from O-CA02 is not linked to an existent COMPLAINT object. An object is created during the C-act 'request'. O-CA02 sends some dependent facts along with the C-fact, such as the consumer and the supplier identification and the rough description of the complaint (cf. Fig. 6.4).
11. There is no scheduling mechanism needed.
12. Not applicable.

Next, the aspects in the *Sharing Facts* aspect group are discussed:

Coordination facts:

13. Not applicable.
14. The identification of the consumer is stated in the attached letter with the 'request'. With the 'promise', the consumer is informed about the person who fulfills O-A01.

Production facts:

15. Not applicable

General:

16. O-A01 does not know anything about the location of an agendum or any other requested fact. Therefore, two generic I-actors 'share agenda' and 'sharer P-facts' have to be implemented. They are responsible for sharing requested facts with O-A01. De external actor I-CA0230 needs a delegate for recalling its result within the scope of the EIS. The 'sharer P-facts' is implemented outside the EIS.
17. This rule concerns the way of exchanging C-facts between I-CA0230 and its delegate who recalls these facts within the EIS. It is a manual procedure, executed by a secretary who scans every incoming letter and enters it in the EIS.
18. Not applicable.

Lastly, the aspects in the *Remembering Facts* aspect group are discussed:

Coordination facts:

19. O-A01 creates the status 'promised' and the status 'stated' of transaction O-T01. The 'promise' is performed implicitly. The 'state' is performed explicitly by sending a mail with a confirmation that the complaint will be taken in treatment to the consumer.

20. The addressee of the ‘promised’ and the ‘stated’ messages is the sender of the letter.
21. O-A01 remembers the dispatching time of the ‘state’.
22. Not applicable.

Production facts:

23. Not applicable.
24. Not applicable.
25. Instances of the binary fact kinds of which one of the domains is one of the following object classes CONSUMER, SUPPLIER, EXPERT and COMMITTEE are created in the EIS at the same time that the P-fact is remembered.

6.3.2 Implementation of O-A02

If the complaint is considered to be admissible, SGC sends the consumer a complaint form that has to be used to officially file the complaint for mediation. The complaint form consists of a questionnaire in which the necessary details of the complaint are asked and a part in which the consumer has to state that he/she will submit him/herself to the regulations of the Conciliation Board for Consumers (the regulations are enclosed). In the same letter, the consumer is also requested to pay a complaint fee (based on the amount of the invoice of the complaint), and possibly to deposit the remaining amount of the invoice (only if the invoice amount is not fully paid). When the complaint is filed, the supplier is requested for a bank guarantee if there is no collective guarantee by the branch organization. The bank guarantee is also based on the amount of the invoice. When the questionnaire is returned properly filled out, it is checked whether the complaint is still within the margins of competence and responsibility of SGC on the basis of detailed questions of the form. Then the money is transferred by the consumer, and, if necessary, the bank guarantee is provided by the supplier. The actual mediation handling by SGC can start.

When the request for mediation by SGC is granted, the supplier is informed by mail and the execution of the procedure is documented in the so-called complaint book. This complaint book was set up for the purpose of progress monitoring. At the same time the supplier is supplied with the documents of the file (first letter and complaint form) and is requested for his defense. In addition to the complaint of the customer and the defense of the supplier, the board can initiate an expert examination. These experts will investigate the nature and the magnitude of the complaint on-site. Their findings are reported in an expert-report. All documents of the consumer, supplier and possibly the experts together serve as the input for a meeting of the board for which all the parties involved are invited. In this meeting the committee reaches a decision. About one month after the meeting, the parties involved are informed by mail about the judgment of the committee. After the judgment of the committee, the financial matters between the customer and the supplier are settled.

We refer to Figure 6.1 for the CM. Figure 6.6 shows the part of the AM that contains the action rules for A02.

Actor A02:

when mediation **for** Complaint is requested	O-T02/rq
assess *justice:* **the** Performer **of the** request **is the** consumer **of** Complaint *sincerity:* <no specific condition> *truth:* <all needed information is available> <consumer is committed to the regulations> <consumer pays complaint fee and deposits the remaining amount of the voice> <supplier guarantees payment>	
if *complying with request is considered justifiable* **then** promise mediation **for** Complaint **else** decline mediation **for** Complaint	

when mediation **for** Complaint is promised	O-T02/pm
assess *justice:* **the** Performer **of the** promise **is the** mediator **of** Complaint *sincerity:* <no specific condition> *truth:* <no specific condition>	
if *complying with promise is considered justifiable* **then** request c_payment control **for** Complaint request s_payment control **for** Complaint	

when c_payment control **for** Complaint is stated	O-T07/st
assess *justice:* **the** Performer **of the** state **is the** c_payment controller **of** Complaint *sincerity:* <no specific condition> *truth:* <financial agreement with consumer is OK >	
if *complying with statement is considered justifiable* **then** accept c_payment control **for** Complaint	

when s_payment control **for** Complaint is stated	O-T06/st
assess *justice:* **the** Performer **of the** state **is the** s_payment controller **of** Complaint *sincerity:* <no specific condition> *truth:* <financial agreement with supplier is OK >	
if *complying with statement is considered justifiable* **then** accept s_payment control **for** Complaint	

when s_payment control **for** Complaint is accepted	O-T06/ac
assess *justice:* **the** Performer **of the** accept **is the** s_payment controller **of** Complaint *sincerity:* <no specific condition> *truth:* c_payment control **of** [complaint] is stated <check whether an expert advice is needed>	

<check availability of expert if needed>
if *complying with accept is considered justifiable* **then** request defense **for** Complaint **if** <expert advice is desirable> **and** **<** expert is available> **then** request expert advice **for** Complaint

when defense **for** Complaint is stated	O-T03/st
assess *justice:* **the** Performer **of the** state **is the** defender supplier **of** Complaint *sincerity:* <no specific condition> *truth:* <defense is available >	
if *complying with statement is considered justifiable* **then** accept defense **for** Complaint	

when expert advice **for** Complaint is stated	O-T04/st
assess *justice:* **the** Performer **of the** state **is the** expert **of** Complaint *sincerity:* <no specific condition> *truth:* <expert advice is available >	
if *complying with statement is considered justifiable* **then** accept expert advice **for** Complaint	

when defense **for** complaint is accepted	O-T03/ac
assess *justice:* **the** Performer **of the** accept **is the** mediator **of** Complaint *sincerity:* <no specific condition> *truth:* expert advice **of** [complaint] is accepted	
if *complying with accept is considered justifiable* **then** request judgment **for** Complaint	

when judgment **for** complaint is stated	O-T05/st
assess *justice:* **the** Performer **of the** state **is the** committee **of** Complaint *sincerity:* <no specific condition> *truth:* <final report is available >	
if *complying with statement is considered justifiable* **then** accept judgment **for** Complaint	

	when judgment **for** complaint is accepted	O-T05/ac
	assess *justice:* **the** Performer **of the** accept **is the** mediator **of** Complaint *sincerity:* <no specific condition> *truth:* <no specific condition>	
	if *complying with accept is considered justifiable* **then** execute mediation **for** Complaint state mediation **for** Complaint	

Fig. 6.6 Action rules for O-A02

O-A02 requests for the following services:

Remembering facts:
a. C-facts and P-facts of O-A02 are remembered by I-A0210 and I-A0211, respectively.

Sharing agenda:
b. Agenda for O-A02 are shared by I-A0020 on request by O-A02.

The financial agreements with the consumer and the supplier, the supplier defense, the expert advice and the final report of the committee are exchanged through the agenda ('state'). Therefore, a separate request for the concerning P-facts is not possible, because they still do not exist that time.

Sharing facts:
c. Checking C-fact: c_payment control **of** [complaint] is stated (I-T0730)
d. Checking C-fact: expert advice **of** [complaint] is accepted (I-T0430)
e. The filled out questionnaire (I-T1031)
f. Whether or not the consumer has agreed to the regulations (manual check)
g. Checking whether an expert advice is needed (manual check)
h. Checking whether an expert is available (manual check)

Figure 6.7 exhibits the structure of the I-organization that is used by O-A02. The dark colored part is covered by the new EIS to be developed. Figure 6.7 shows that the EIS not only supports coordination with actors O-A06 and O-A07, but also supports coordination with actors O-A03, O-A04 and O-A05. The subjects who fulfill these actor roles are not employees of SGC. They are rented to fulfill these actor roles with regard to a specific mediation project. These actors usually do not have a workplace at SGC. Therefore, they need a remote connection with the EIS.

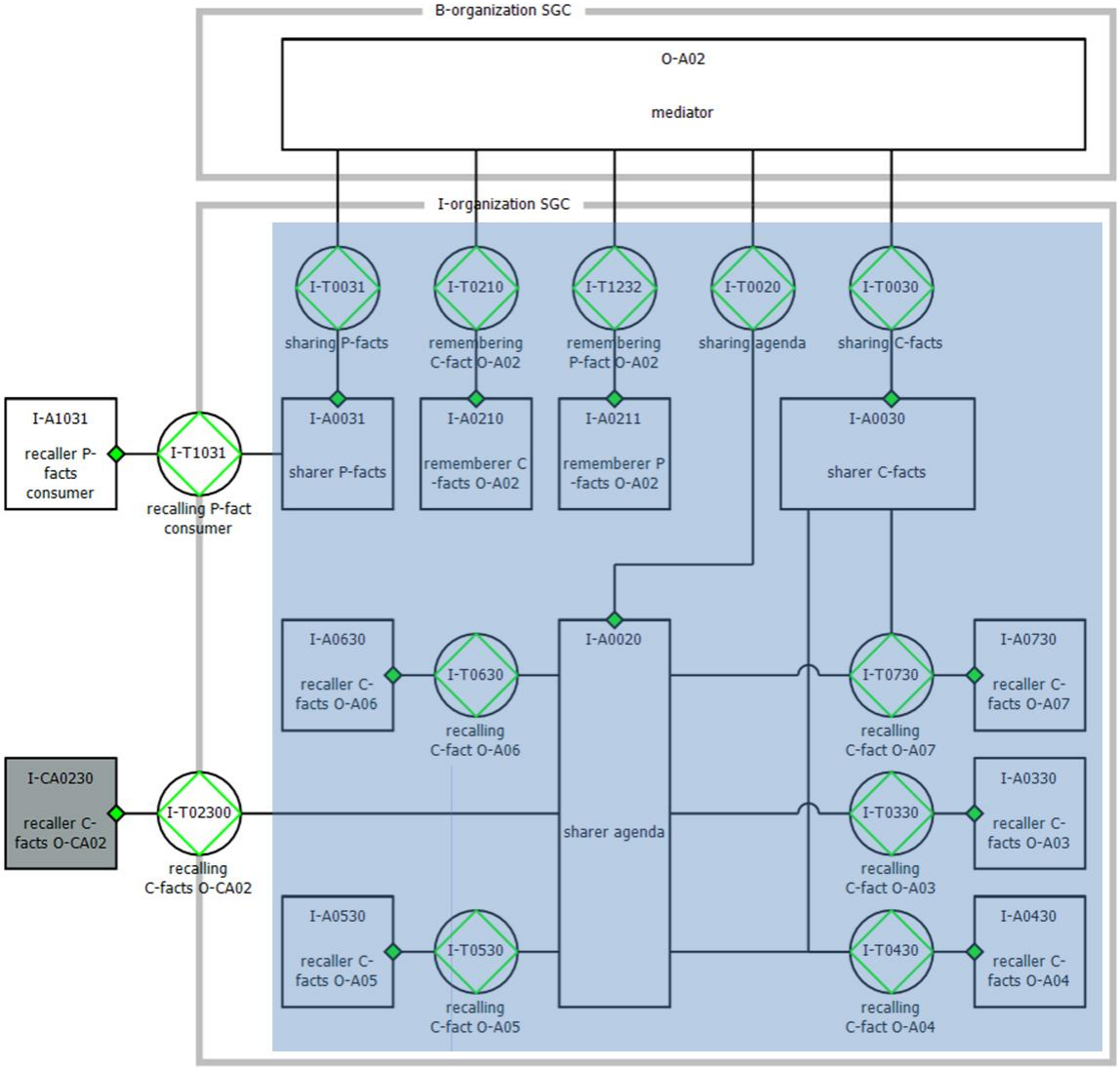


Fig. 6.7 Remembering and Sharing Facts for O-A02

The discussion of the implementation of O-A02 in an EIS continues on the basis of the rules from the PID-Framework which are related to the *Actor General Characteristics*:

5. O-A02 handles admitted complaints. Its process starts with the request of O-CA02 for starting up the mediation of a specific complaint. O-CA02 has received the complaint identification from O-A01.
6. O-A02 can be fulfilled by several people who are completely interchangeable.
7. According to the text, the SGC director is responsible for dealing with complaints. However, there is a general secretariat that supports the director in his work. The director has appointed one or more people of the secretariat for being his delegate.
8. The infological transactions ‘remember C-facts’, ‘remember P-facts’, ‘share agenda’, ‘share C-facts’ and ‘share P-facts’ have to be implemented in the working environment of O-A02 in the EIS. The C-facts from O-CA02 have to be entered into the EIS by an

employee of SGC. This employee acts as a delegate of I-CA0230 and has to be understood as the 'recaller' of the C-facts of O-CA02 in the EIS. The P-facts from I-A1031 have to be entered into the EIS by a delegate as well. This delegate has also to be understood as the 'recaller' of the P-facts in the EIS.

9. O-A02 is a business actor and can therefore not be replaced by an agent. Figure 6.7 exhibits that all information services are supported by the EIS.

The aspects in the *Sharing Agenda* aspect group are discussed hereafter:

10. An object COMPLAINT is linked to the agendum 'requested' that corresponds with the C-fact that has been created by O-CA02. It means that the P-fact of O-A01 is available for O-A02 without an explicit request for fact sharing. O-A02 is also involved in five other ontological transactions: O-T03, O-T04, O-T05, O-T06 and O-T07. It initiates these transactions and receives a 'state' when they finish their production act. Along with the 'state' the COMPLAINT object is received with the dependent and independent facts that have to be accepted.
11. Not applicable.
12. Not applicable.

Next, the aspects in the *Sharing Facts* aspect group are discussed.

Coordination facts:

13. This is done by I-A0030. A C-fact is requested for determining whether a specific state of a P-fact is achieved (I-T0430, I-T0730).
14. The identification of the consumer is already stated in the COMPLAINT object. O-A02 is able to detect the performer of the 'request' of O-CA02 on the basis of the complaint identification. With the 'promise', the consumer is informed about the person who fulfills O-A02.

Production facts:

15. Not applicable.

General:

16. O-A02 does not know anything about the location of an agendum or any other requested fact. Therefore, three generic I-actors 'share agenda', 'sharer C-facts' and 'sharer P-facts' have to be implemented. They are responsible for sharing requested facts with O-A02. De external actors I-CA0230 and I-A1031 need a delegate for recalling their results within the scope of the EIS.
17. This rule concerns the way of exchanging C-facts between I-CA0230 and its delegate who recalls these facts within the EIS. It is a manual procedure.
18. I-A1031 recalls the questionnaire and the confirmation with the SGC regulations of the consumer. These facts are sent by mail to SGC. A secretary, as the delegate of I-A1031, enters these facts into the EIS for being available for O-A02.

Lastly, the aspects in the *Remembering Facts* aspect group are discussed

Coordination facts:

19. O-A02 creates the transaction statuses ‘promised’ and a ‘stated’ in transaction O-T02. The ‘promise’ is performed implicitly. The ‘state’ is performed explicitly by sending a letter with the final decision of the committee. O-A02 initiates five transactions kinds. The ‘request’ in each transaction takes place explicitly. The ‘accept’ takes place implicitly for all these transaction kinds.
20. O-A02 creates five different kinds of ‘requests’. Every request must have an addressee. The addressee of O-A06 and O-A07 are well known. It is the administrator of SGC who maintains contacts about financial issues with consumers and suppliers. The addressees of the actor roles O-A03, O-A04 and O-A05 are difficult to determine. These actor roles are fulfilled by external subjects. They delegate the coordination with O-A02 to a secretary of SGC.
21. OK
22. Not applicable.

Production facts:

23. Not applicable.
24. Not applicable.
25. Not applicable.

6.3.3 Implementation of O-A03

When the request for mediation by SGC is granted, the supplier is informed by mail, At the same time the supplier is supplied with the documents of the file (first letter and complaint form) and is requested for his defense (cf. Fig. 6.8). Figure 6.9 shows the part of the AM that contains the action rules for A03.

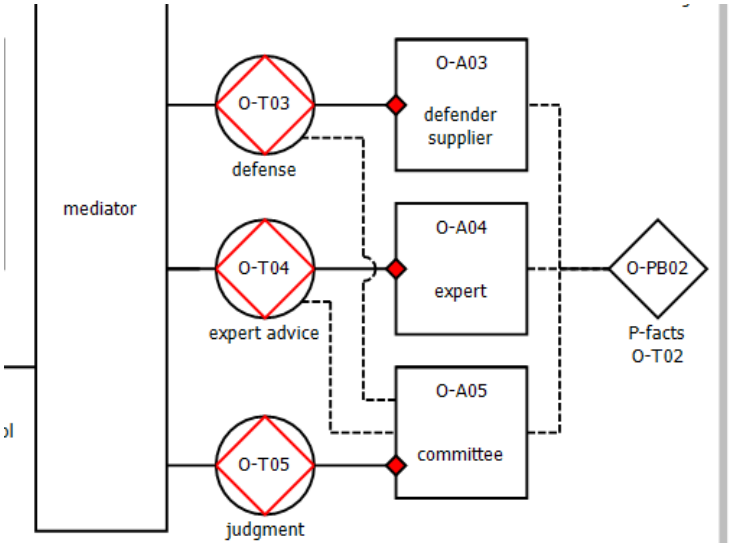


Fig. 6.8 CM: Complaint handling

	when defense **for** Complaint is requested	O-T03/rq
	assess *justice:* **the** Performer **of the** request **is the** mediator **of** Complaint *sincerity:* <no specific condition> *truth:* <no specific condition>	
	if *complying with request is considered justifiable* **then** promise defense **for** Complaint	

	when defense **for** Complaint is promised	O-T03/pm
	assess *justice:* **the** Performer **of the** promise **is the** defender supplier **of** Complaint *sincerity:* <no specific condition> *truth:* <no specific condition>	
	if *complying with promise is considered justifiable* **then** execute defense **for** Complaint state defense **for** Complaint	

Fig. 6.9 Action rules for the actor A03

O-A03 requests for the following services:

Remembering facts:
a. C-facts and P-facts of O-A03 are remembered by I-A0310 and I-A0311, respectively.

Sharing agenda:
b. Agenda for O-A03 are shared by I-A0020 on request by A03.

Sharing facts:
None.

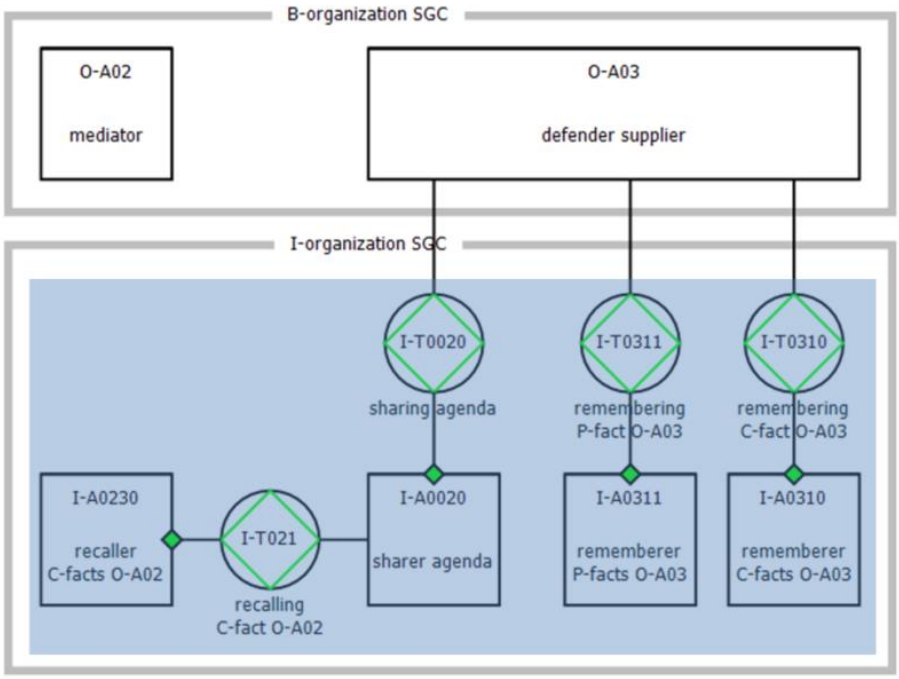


Fig. 6.10 Remembering and Sharing Facts for O-A03

Figure 6.10 exhibits the structure of the I-organization that is used by O-A03. The dark colored parts are covered by the new EIS to be developed. O-A03 is fully supported by the EIS. The discussion of the implementation of O-A02 in an EIS continues on the basis of the rules from the PID-Framework related to the *Actor General Characteristics*:

5. The actor role O-A03 is fulfilled by the defender of the supplier who is linked to the complaint. He acts on behalf of the supplier. He is activated by the supplier who has got a message with the request for defense from SGC.
6. Each supplier has his own defender. Therefore, many suppliers fulfill the actor role independent from each other.
7. Not applicable.
8. The actors get a web enabled interface in order to fill in their defense. The complaint information is sent along with the 'request' by O-A02.
9. O-A03 is a business actor and can therefore not be replaced by an agent. See rule 8.

The aspects in the *Sharing Agenda* aspect group are discussed hereafter:

10. The COMPLAINT object is linked to the agendum 'requested'. Al dependent and independent facts are available for A03.
11. There is no scheduling mechanism needed. The 'request' is sent directly to the supplier or its defender. The supplier is a dependent fact that already is available.
12. Not applicable.

Next, the aspects in the *Sharing Facts* aspect group are discussed.

Coordination facts:

13. Not applicable.
14. OK.

Production facts:

15. Not applicable.

General:

16. O-A03 does not know anything about the location of an agendum Therefore, the I-actor ‘share agenda’ has to be implemented.
17. Not applicable.
18. Not applicable.

Lastly, the aspects in the *Remembering Facts* aspect group are discussed:

Coordination facts:

19. O-A03 creates the transaction statuses ‘promised’ and ‘stated’ in transaction O-T03. The ‘promise’ takes place implicitly, the ‘stated’ takes place explicitly by sending the defense to O-A02.
20. Not applicable.
21. Not applicable.
22. Not applicable.

Production facts:

23. According to the regulations of SGC, the P-fact has to be created (providing the supplier defense) within a fixed number of days.
24. Not applicable.
25. Not applicable.

6.3.4 Implementation of O-A05

All documents of the consumer, supplier and possibly the experts together serve as the input for a meeting of the committee for which all the parties involved are invited. In this meeting the committee comes to a final decision. The director of the SGC is also secretary of all the committees. He has access to the EIS. See Figure 6.8 for a relevant part of the CM and Figure 6.13 for the action rules of O-A05.

Actor A05:

	when judgment **for** Complaint is requested	O-T05/rq
	assess *justice:* **the** Performer **of the** request **is the** mediator **of** Complaint *sincerity:* <no specific condition> *truth:* <no specific condition>	
	if *complying with request is considered justifiable* **then** promise judgment **for** Complaint	

	when judgement **for** Complaint is promised	O-T05/pm
	assess *justice:* **the** Performer **of the** promise **is the** committee **of** Complaint *sincerity:* <no specific condition> *truth:* <no specific condition>	
	if *complying with promise is considered justifiable* **then** execute judgment **for** Complaint state judgment **for** Complaint	

Fig. 6.13 Action rules for the actor A05

The CM exhibits that several information links are drawn from O-A05 to production banks. We will see in the AM that these information links do not correspond with infological transactions that have to be modeled in the I-organization. It corresponds with a need for information that is gained with the C-fact 'request' from O-A02. Along with this 'request' the COMPLAINT object comes available for O-A05. This object contains already the dependent facts that are needed.

O-A05 requests for the following services:

Remembering facts:
a. C-facts and P-facts of O-A05 are remembered by I-A0510 and I-A0511, respectively

Sharing agenda:
b. Agenda for O-A05 are shared by I-A0020 on request by O-A05.

Sharing facts:
None.

Figure 6.14 exhibits the structure of the I-organization that is used by O-A05. The dark colored parts are covered by the new EIS to be developed. O-A05 is fully supported by the EIS.

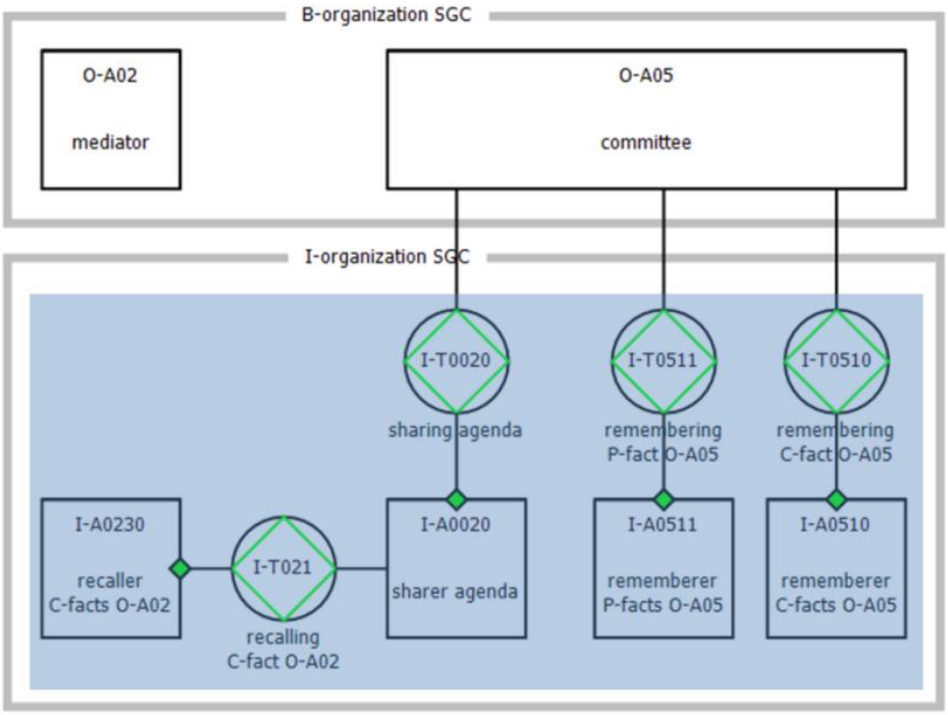


Fig. 6.14 Remembering and Sharing Facts for O-A05

The discussion of the implementation of O-A05 in an EIS continues on the basis of the rules from the PID-Framework related to the *Actor General Characteristics:*

5. The actor role O-A05 is fulfilled by committee which is linked to the complaint. The director of the SGC is also secretary of all the committees.
6. There is a committee that is constituted of the SGC director and three other subjects. Each committee handles complaints that belong to a specific category. A committee has a collective responsibility.
7. The actor delegates the coordination to a secretary of SGC.
8. The actors get a web enabled interface in order to fill in their defense. The complaint information, included the defense and the expert advice, is sent along with the 'request' by O-A02 to O-A05 and can be viewed by it.
9. O-A05 is a business actor and can therefore not be replaced by an agent. See rule 8.

The aspects in the *Sharing Agenda* aspect group are discussed hereafter:

10. The COMPLAINT object is linked to the agendum 'requested'. Al dependent and independent facts are available for A05.
11. There is no scheduling mechanism needed. The 'request' is sent directly to the secretary who the delegate of the committee.
12. Not applicable.

Next, the aspects in the *Sharing Facts* aspect group are discussed.

Coordination facts:

13. Not applicable.
14. OK.

Production facts:

15. Not applicable.

General:

16. O-A05 does not know anything about the location of an agendum Therefore, the I-actor 'share agenda' has to be implemented.
17. Not applicable.
18. Not applicable.

Lastly, the aspects in the *Remembering Facts* aspect group are discussed.

Coordination facts:

19. O-A05 creates the transaction statuses 'promised' and 'stated' in transaction O-T05. The 'promise' takes place implicitly, the 'stated' takes place explicitly by sending the final decision to O-A02.
20. Not applicable.
21. Not applicable.
22. Not applicable.

Production facts:

23. According to the regulations of SGC, the P-fact has to be created (providing the file decision) within a fixed number of days.
24. Not applicable.
25. Not applicable.

6.4 Validation and Conclusions

The design of an EIS for the SGC was elaborated on the basis to the PID-Framework. We noticed in the introduction that, in the past, an attempt was already made to derive an EIS from the ontological model of the SGC. A report of this case study can be found in the PhD thesis of Mulder [Mulder 2006]. Although he has not documented every step extensively, A clear picture of the approach he has followed, emerges from the available publications [Mulder and Van Reijswoud 1997; Mulder 2006]. Some of the main objections to this approach are dealt with in the case study as described in this chapter. One major draw-

back is that Mulder does not provide a functional model of the I-organization that has to be implemented by the EIS. Consequently, an ontological model of the I-organization is also lacking. We have designed an EIS from the integral ontological model of the B- and I-organization. Mulder followed a different approach. He defined a direct link between the ontological model of the B-organization and an implementation model of the infological and datalogical organization. In hindsight, Mulder has skipped several steps in the τ-theory [Dietz 2008], without proper argumentation.

A validation of Mulder's application with employees of the current SGC organization was not possible. The findings were only discussed with two designers of the model that was developed in 1996. This application has to be considered as a scientific exercise in order to make a more thorough demonstration of the applications in the next chapters possible.

7 Case 2: The Education Center EC

7.1 Introduction

Mprise is part of Mprise Group B.V. (established in 1991). Mprise is an IT service provider for training and implementation of Microsoft ERP software. The Training Unit organizes standard training courses that are offered in Mprise's Academy as well as customized training offered in-house at the customer's premises. The training courses are short, ranging from one day to one week. On an annual basis, there are about 1800 students.

This application focuses on the organization of the standard training courses offered in the Education Center, abbreviated to EC. Besides the usual office IT products, an EIS, which is called COURSE, has been implemented in the EC about vijf years ago. The need has arisen within the EC to assess the value added by COURSE to the organization. The PID-Framework and the associated rules (see 5.3) have been used for this assessment.

7.2 Context

In EC seven different processes can be distinguished with regard to the provision of standard training courses, namely:

1. *The scheduling process*. A new training program is made every six months. It is made in an Excel spreadsheet. Afterwards the program is entered in COURSE, printed in booklets and published on the website.
2. *The enrollment process*. Usually, companies enroll participants in trainings through the website. It is also possible to enroll participants by telephone or email. There is a discount on the enrollment if the applicant already enrolled previously and then withdrew within two weeks before the start of the training in question in order to take it at a later date.
3. *The cancellation process*. According to the general conditions of EC, registration for a specific training can be canceled free of charge if the scheduled date is more than two weeks before the cancellation date. A cancellation that occurs within two weeks of the scheduled training date cannot be made free of charge. Then, an adjusted invoice will be sent.
4. *The delay process*. According to the general conditions of EC, postponing of the training within two weeks before the start of the training is possible. According to the general training conditions an invoice will still be sent to the participant.
5. *The control process*. This process involves examining whether a specific training has enough applicants to enable it to take place.
6. *The completion process*. In this process, trainings are prepared, performed, and evaluated, after the decision has been taken that it will definitely take place.

7. *The invoicing process.* Each week, the trainings that are completed in the previous week are billed.

In this chapter the enrollment and the completion processes are partly elaborated. Figure 7.1 exhibits the relevant part of the CM and Figure 7.2 the FM.

Transaction Kind	**Result Kind**
O-T01 - enrollment	O-R01 - [enrollment] has been enrolled
O-T03 - time table control	O-R03 - time table for [6_months] has been controlled
O-T04 - training schedule	O-R04 - [training] has been scheduled
O-T05 - training completion	O-R05 - training [week] has been completed
O-T06 - preparation	O-R06 - [training] has been prepared
O-T07 - execution	O-R07 - [training] has been executed
O-T08 - evaluation	O-R08 - [training] has been evaluated
O-T12 - turnover control	O-R12 - turnover [week] has been controlled
O-T15 - financial collection	O-R15 - financials [enrollment] has been collected

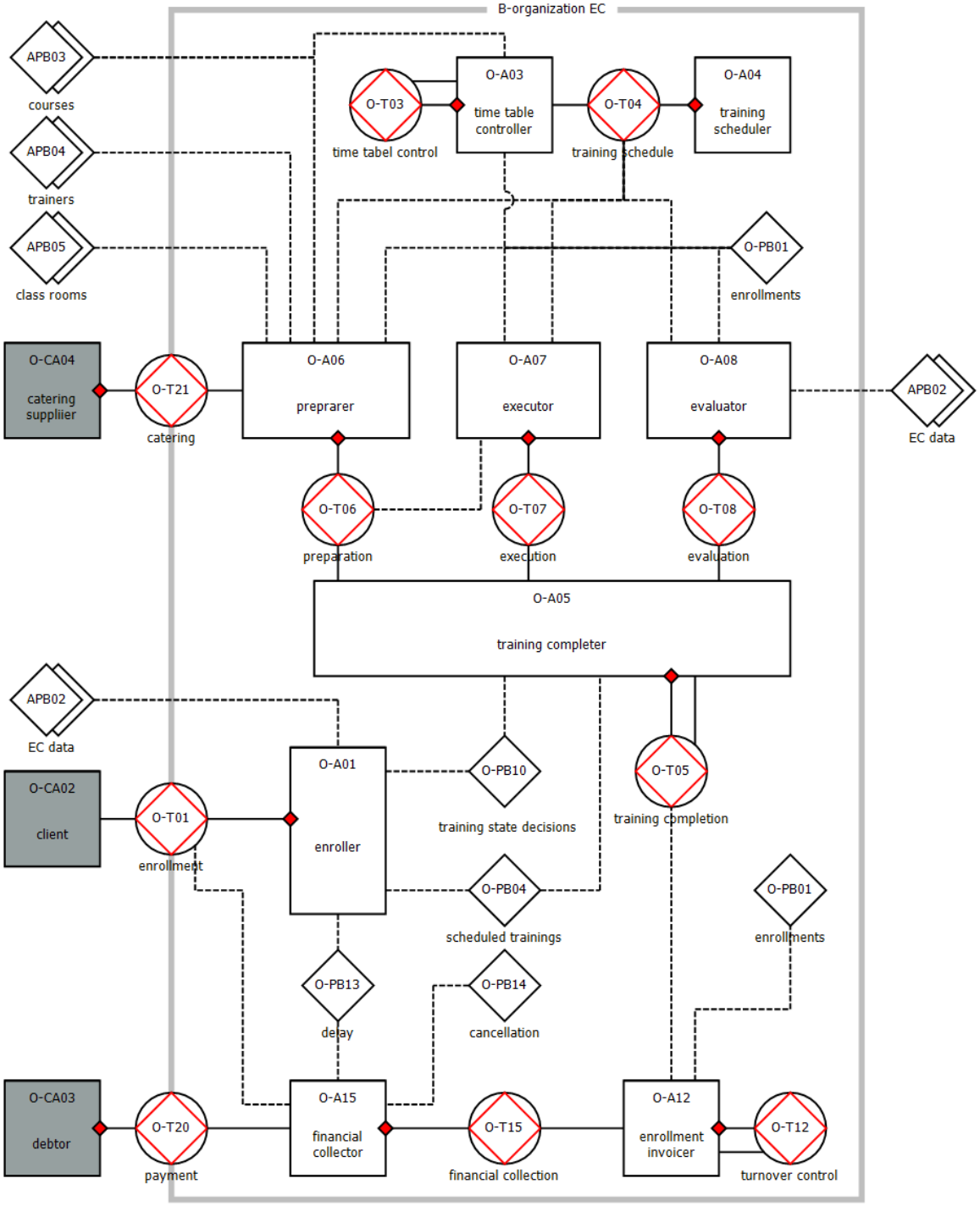


Fig. 7.1 Construction Model of EC

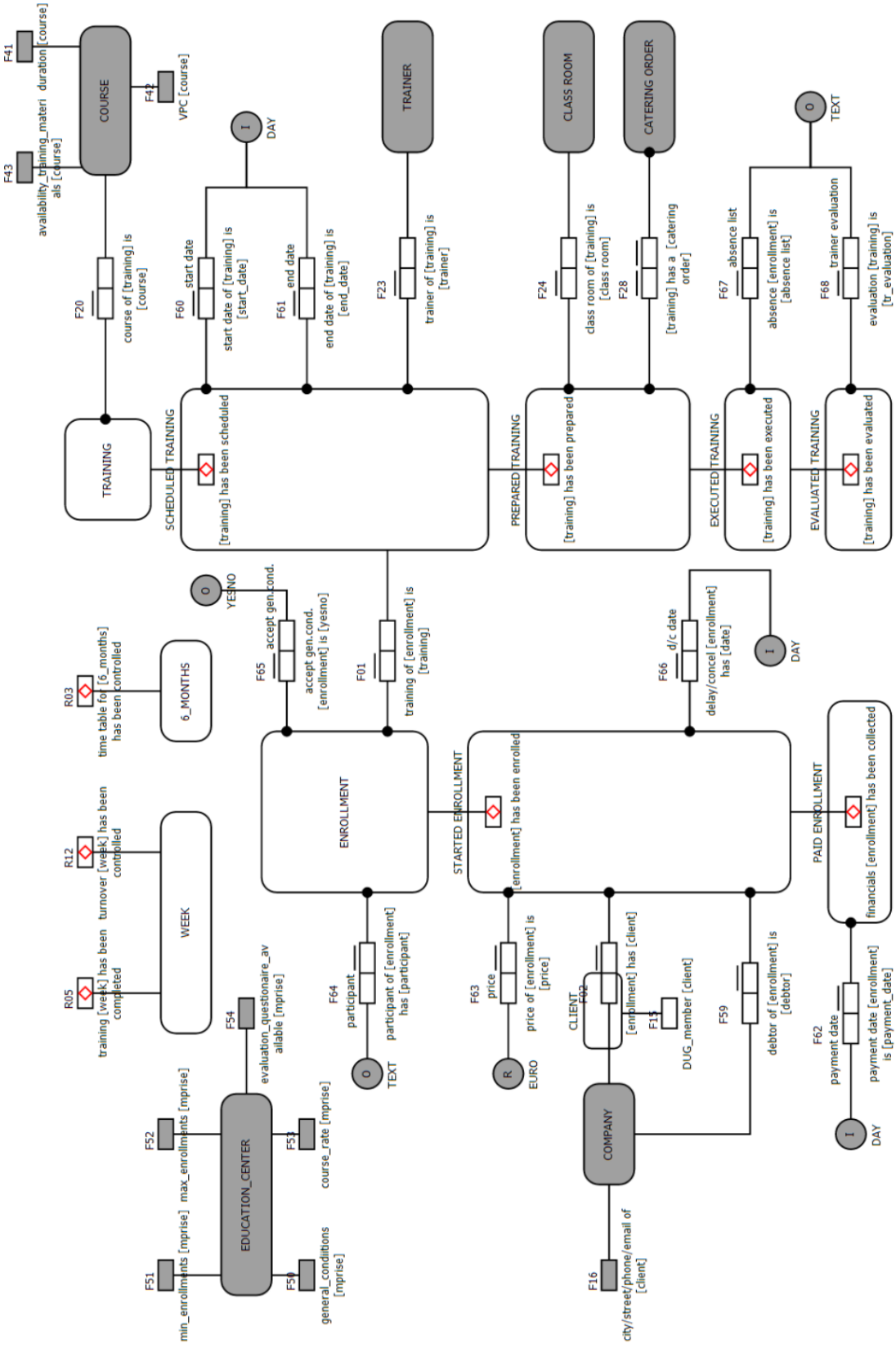


Fig. 7.2 Fact Model of EC

The individual aspects that are specified within the *context* aspect group are discussed point by point based on the rules of the PID-Framework (see 5.3). The numbering corresponds to the numbering of the rules.

1. All O-actors which have P-acts on one of the object classes TRAINING and ENROLLMENT have to be supported by the COURSE application. While it is true that the actor O-A04 also fall within the scope of the EIS, O-A04 has already been satisfactorily supported by an Excel application. This has the consequence that P-facts that were created by O-A04 are stored on two different places physically: within the scope of the Excel application as well as within the scope of COURSE.
2. Given the size of this case, we will refrain from discussing here the question of which I-actors and D-actors fall inside or outside the scope of the EIS. We will deal with this in more detail when discussing the individual actor roles.
3. There is one binary fact kind within the scope of COURSE which have one domain inside COURSE and one outside. It concerns the object class COMPANY. There are more binary fact kinds with the domain TRAINING which have also a domain outside: TRAINER, COURSE, CLASS_ROOM and CATERING_ORDER. The current implemented version of COURSE does not contain subsets of these external object classes. These objects with their dependent and independent P-facts are currently remembered in separated systems and by heart.
4. No clusters are distinguished for the actor roles to discuss.

Facts that were created by O-A04 are frequently consulted by actors within the scope of the EIS, while these facts are captured outside the scope of COURSE in an Excel file. Because this Excel file is only updated twice a year, the decision was taken to periodically transform these facts into a different form and medium, and capture them in COURSE as well.

7.3 Actor roles

The following actor roles are elaborated: O-A01, O-A05, O-A06 and O-A07.

7.3.1 Implementation of O-A01

A customer enrolls on a specific training course. He/she has several options at his disposal to enroll on training at the EC. He/she may call by phone or it may send an email or a letter. EC prefers to start the enrollment process by filling out a standard form on the website. Then, a number of steps need to be carried out whereby all the necessary information is obtained in a structured manner. In one of these steps the applicant indicates that he accepts the general training conditions of EC. When the enrollment process completes, a message appears on its screen thanking him for the enrollment and stating that an email with a confirmation of the enrollment will be sent. The customer considers the email as a confirmation that the enrollment has been received well by the EC.

The email contains the following text:

'We would like to thank you cordially for your request for enrollment. We process it and send you further information as soon as possible.'

The request has arrived in an auxiliary table of COURSE at the EC. The secretary reads the request and looks at whether there are vacant places in the training. If that is the case, she sends a confirmation that the enrollment on the training is definitive. It may happen that the customer has enrolled previously on the same training course and that he decides to postpone his participation in the training within two weeks before the start date of the training. Then the full cost of the training will be charged. However, a new request of that customer for the same training later provides a discount of 50 or 25 percent, depending on the time that remained previously till the start date of the training.

The relevant part of the CM and the AM are exhibited in Figure 7.3 and 7.4. The relevant part of the PM is disregarded; it does not add anything to what is described in Figure 7.4. The price of the training is determined by means of the second action rule prior to the actual production act taking place. It depends on whether the customer has registered for this training previously and has withdrawn it within two weeks before the start date.

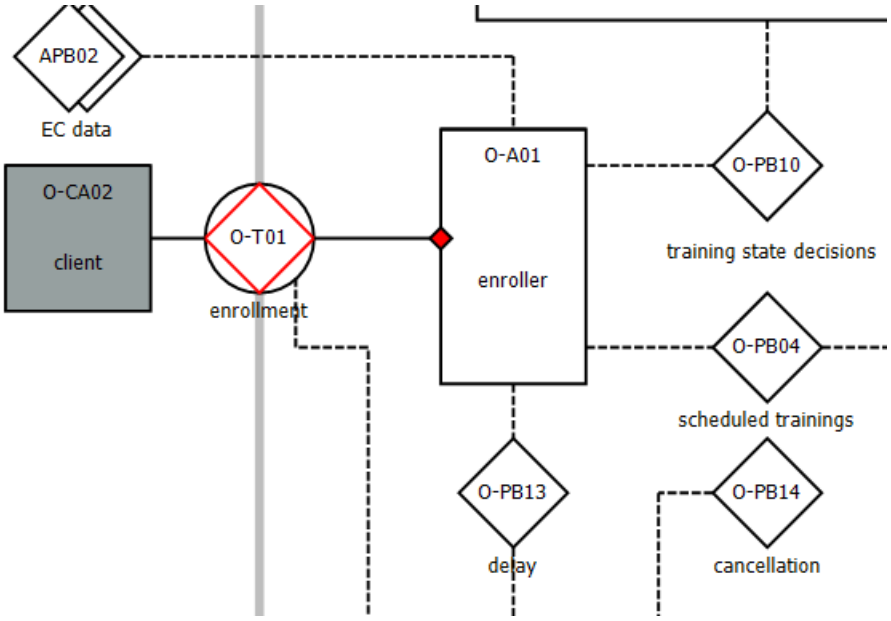


Fig. 7.3 The enrollment in a training program

Actor O-A01:

when enrollment **for** new Enrollment is requested		O-T01/rq
with	**the** participant **of** Enrollment **is** a Participant **the** training **of** Enrollment **is** a Training **the** client **of** Enrollment **is** a Client **the** debtor **of** Enrollment **is** a Debtor **the** accept.gen.cond. **of** Enrollment **is** a Acc.gen.cond.	
assess	*justice:* **the** Performer **of the** request **is the** client **of** Enrollment *sincerity:* <no specific condition> *truth:* <training is known> <training is not expired> <check #participants < available seats> <general conditions are accepted>	
if	*complying with request is considered justifiable*	
then	promise enrollment **for** Enrollment	
else	decline enrollment **for** Enrollment	

when enrollment **for** Enrollment is promised		O-T01/pm
assess	*justice:* **the** Performer **of the** promise **is the** admitter **of** Enrollment *sincerity:* <no specific condition> *truth:* <determine training price>	
if	*complying with promise is considered justifiable*	
then	execute enrollment **for** Enrollment state enrollment **for** Enrollment	

Fig. 7.4 Action rules for O-A01

Figure 7.5 exhibits the I-organization with the I-actors that are invoked by O-A01. The dark colored parts are covered by COURSE. O-A01 requests for the following services:

Remembering facts:

a. C-facts and P-facts of O-A01 are remembered by I-A0110 and I-A0111, respectively

Sharing agenda:

b. Agenda for O-A01 are shared by I-A0020 on request by O-A01.

Sharing facts:

c. Check whether training is planned (I-T0431)
d. Check whether training is expired (I-T1031)
e. The maximum number of participants and standard training fee from the general EC data (I-T6031)
f. Check the number of vacant seats (I-T27)
g. Determine the price of the training (I-T26)

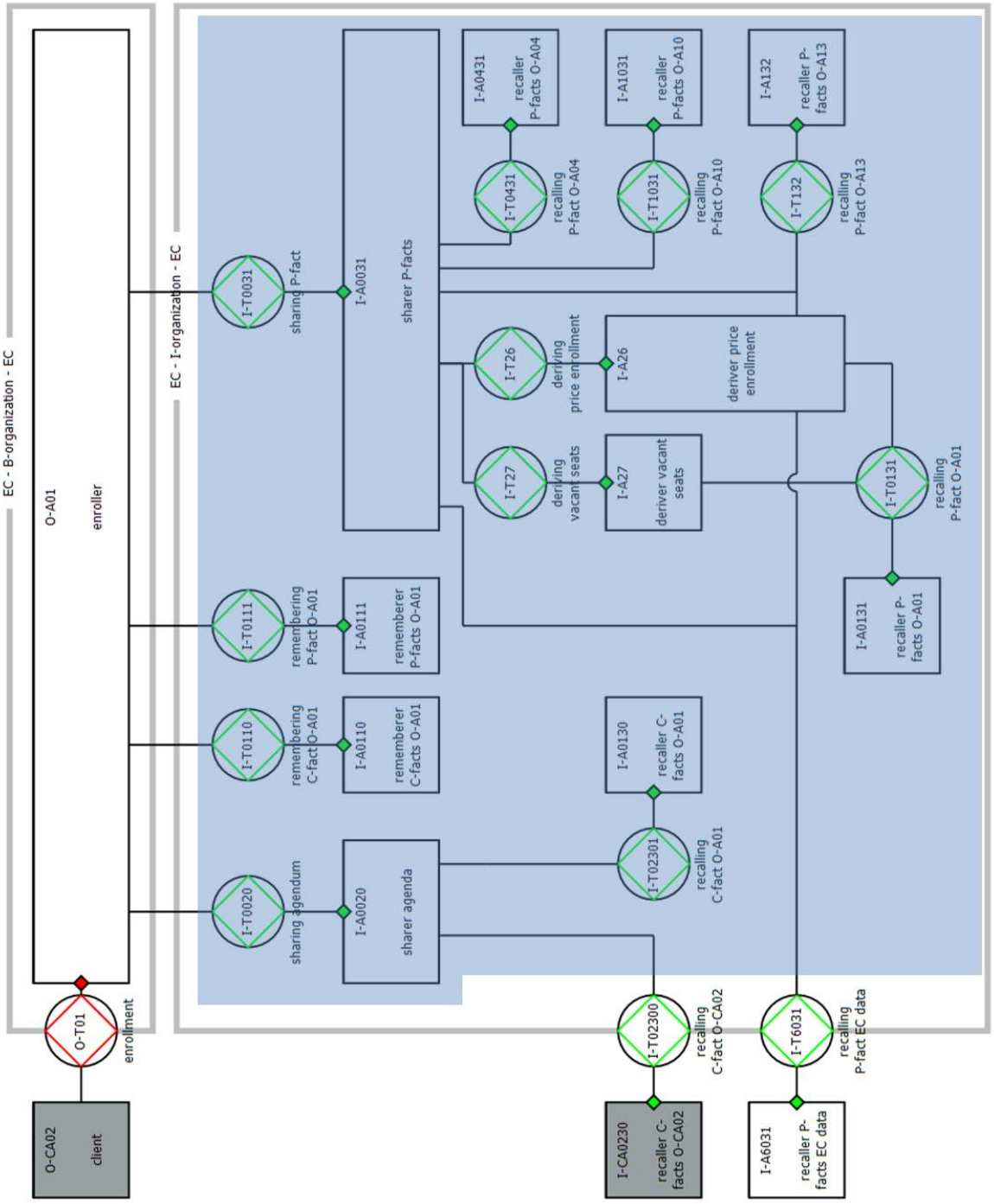


Fig. 7.5 Remembering and Sharing Facts for O-A01

We will continue the discussion of the implementation of O-A01 in the COURSE application on the basis of the rules from the PID-Framework related to the *Actor General Characteristics*.

5. O-A01 is activated when a request for enrollment is received. This occurs in several ways. EC prefers to receive the requests for enrollment through its website.
6. O-A01 is fulfilled by several people independent of each other. All actors make use of a common interface of COURSE and are able to take over each other's work.
7. The subjects who perform this role are considered as delegates of the department manager. Many questions that arise during enrollment are fed back to the manager who takes the final decision.
8. A 'request' from O-CA02 is shared by the 'sharer agenda' with O-A01. Figure 7.4 exhibits that some dependent facts are sent along with the 'request'. Entering these facts can be forced before submitting the web form on the website. Requests for enrollments are also allowed by email, phone and other communication media. Then, the needed depended facts have to be requested explicitly afterwards.
9. O-A01 is a business actor and cannot be replaced by an agent. Facts are remembered in the EIS. They can be recalled in the EIS and are made available for the purpose of other actors. (cf. Fig. 7.1).

The aspects in the *Sharing Facts* aspect group are discussed hereafter:

Agenda:

10. The 'when' clause of the action rule of O-A01 contains the dependent fact which must be sent along with the 'request'. During the 'request' act an object ENROLLMENT is created.
11. A scheduling mechanism is not applicable. The subjects who fulfill O-A01 are fully replaceable.
12. Not applicable.

Coordination facts:

13. Not applicable.
14. The identification of the client is stated in one of the dependent facts that are along with the 'request'. With the 'promise', the client is informed about the person who fulfills O-A01.

Production facts:

15. O-A01 retrieves client facts from I-CA02. Among these is the debtor. In the EC case, the debtor is the organization to which the invoice has to be sent for the training course. This assumption can differ from the definition which is used by the client. For example, it could be that the mentioned debtor only deals with invoices for the purchase of raw materials and intermediate goods and that invoices for expenses have to be sent to another address. Problems with semantic interoperability can largely be obviated by using structured forms with detailed explanations for the enrollment process.

General:

16. O-A01 does not know anything about the location of an agendum or any other requested fact. Therefore, two generic I-actors 'share agenda' and 'sharer P-facts' have to be implemented. They are responsible for sharing requested facts with O-A01. The external actor I-CA0230 needs a delegate for recalling its result within the scope of COURSE.
17. The requests for enrollments are collected from the website in an intermediate database table in COURSE. The secretary of EC looks at all of them thoroughly before processing them in COURSE. She operates actually as the delegate of the external recaller of the C-facts of the client.
18. Not applicable.

Next, the aspects in the *Remembering Facts* aspect group are discussed:

Coordination facts:

19. O-A01 remembers its C-facts which are addressed to O-CA02. Along with the 'promise' an email is sent that the enrollment will be processed. A confirmation is sent along with the 'state'. It is also sent by email.
20. Not applicable.
21. The date of the 'promise' indicates the sequence of enrollments to treat. The date of the 'state' indicates the final date of the registration.
22. Each registration for a single training is treated as a separate registration.

Production facts:

23. O-CA02 does not request an intended settlement time. It is important to process the enrollment as soon as possible in order to avoid given away an available seat to somebody else.
24. Not applicable.
25. The ENROLLMENT object class has two dependent binary fact kinds with COMPANY, namely the client of the enrollment and the debtor of the enrollment. A subclass of COMPANY is created in COURSE in order to remember the instances of the dependent binary facts.

Note:

- An external actor that initiates an ontological transaction sends along with the 'request' some dependent facts which are needed for creating the object on which P-act need to be executed. The completeness of the number of facts that are sent along can be enforced by having the request made through a web form on the website. Additional information has to be required through an infological transaction with an external I-actor. The external I-actor could be fulfilled by the same subject. Then, take into account that there is a matter of two parallel transactions between likely the same subjects. Prevent confusion through clearly specified coordination acts.

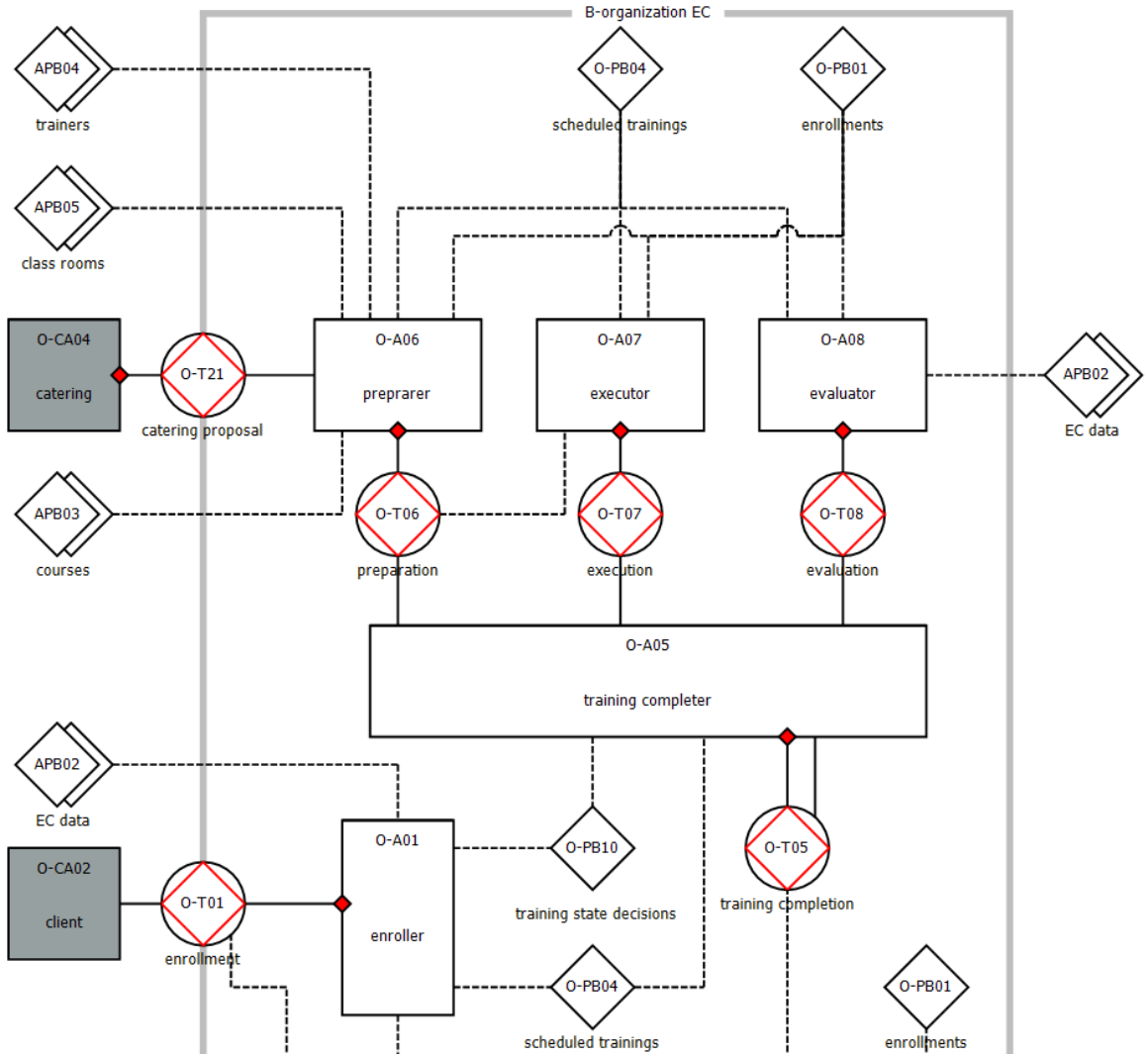


Fig. 7.6 CM of the training completion

7.3.2 Implementation of O-A05

Figure 7.6 shows actor O-A05 reading facts from two different production banks, namely scheduled trainings from O-PB04 and whether the training has been expired from O-PB10. The manager of the EC fulfills O-A05. In practice, she delegates the C-acts as well as the P-acts to her secretary.

The current implementation of COURSE hardly supports O-A05. This is a gap in the design of COURSE. Therefore, the proposed implementation design of actor O-A05 in this section must be primarily seen as a description of the desired implementation.

As mentioned, the secretary that fulfills the role of O-A05 as a delegate prints the trainings schedule that will definitely be provided in the next week. On the basis of this schedule she initiates transactions with O-A06 for preparing the training. O-PB10 is consulted for the current state of the training. It could be possible that the training has been expired. O-A05 also initiates a transaction with O-A07, the trainer, for providing the training. We

skip the evaluation part of this application. The relevant part of the FM is given in Figure 7.7.

Actor O-A05:

when training completion **for** Week is requested	O-T05/rq
assess *justice:* **the** Performer **of the** request **is the** training completer **of** Week *sincerity:* <no specific condition> *truth:* <no specific condition>	
if *complying with request is considered justifiable* **then** request training_completion **for** next Week **with** requested_creation_time = now + 1 week promise training completion **for** Week	

when training completion **for** Week is promised	O-T05/pm
assess *justice:* **the** Performer **of the** promise **is the** training completer **of** Week *sincerity:* <no specific condition> *truth:* <no specific condition>	
if *complying with promise is considered justifiable* **then** **do for** <all Trainings in Week that have not been expired> request preparation **for** Training **od**	

when preparation **for** Training is stated	O-T06/st
assess *justice:* **the** Performer **of the** state **is the** preparer **of** Training *sincerity:* <no specific condition> *truth:* <preparation of Training is OK>	
if *complying with request is considered justifiable* **then** accept preparation **for** Training request execution **for** Training	

when execution **for** Training is stated	O-T07/st
assess *justice:* **the** Performer **of the** state **is the** executor **of** Training *sincerity:* <no specific condition> *truth:* <no specific condition>	
if *complying with promise is considered justifiable* **then** accept execution **for** Training request evaluation **for** Training	

Fig. 7.7 Action rules for O-A05

Figure 7.8 exhibits the I-organization with the I-actors that are invoked by O-A05. The dark colored parts are covered by COURSE. O-A05 requests for the following services:

Remembering facts:
a. C-facts and P-facts of O-A05 are remembered by I-A0510 and I-A0511, respectively

Sharing agenda:
b. Agenda for O-A05 are shared by I-A0020 on request by O-A05.

Sharing facts:
c. A scheduled training from O-PB05 for Week that is not expired (I-T35).

We will continue the discussion of the implementation of O-A05 in the COURSE application on the basis of the rules from the PID-Framework related to the *Actor General Characteristics*.

5. O-A05 activates itself every week on Wednesday and prints a list with the scheduled training for the next week.
6. O-A05 is fulfilled by the manager of the EC, but in practice all acts are delegated to a secretary
7. See 6.
8. The secretary activates a print process in COURSE for getting the list with scheduled trainings for the next week. This list also contains the current number of enrolled participants. All coordination acts are performed outside COURSE. This means that O-A05 cannot guarantee that the list is sent to O-A06. This could lead to misunderstandings and to a badly well prepared training. No further training information is exchanged between O-A05 and O-A06 and O-A07.
9. COURSE is also not used for capturing the C-facts and P-facts of O-A05. See also 8.

The aspects in the *Sharing Facts* aspect group are discussed hereafter:

Agenda:

10. O-A05 executes its P-act on a WEEK object. This object is assumed to be present. No dependent facts are sent along with the 'request'. A TRAINING object is sent along with the 'request' to O-A06 and O-A07. A 'state' which is shared with O-A05 contains also the TRAINING object. Along with the 'state', dependent facts which are created by O-A06 and O-A07 become available to O-A05. So, based on the received facts of O-A06 the secretary has to decide whether the preparation may be accepted. Unfortunately, in practice it does not work in that way, due to the missing implementation of the C-act 'accept'.
11. There is no scheduling mechanism implemented.
12. Not applicable.

Coordination facts:

13. Not applicable.
14. Not applicable.

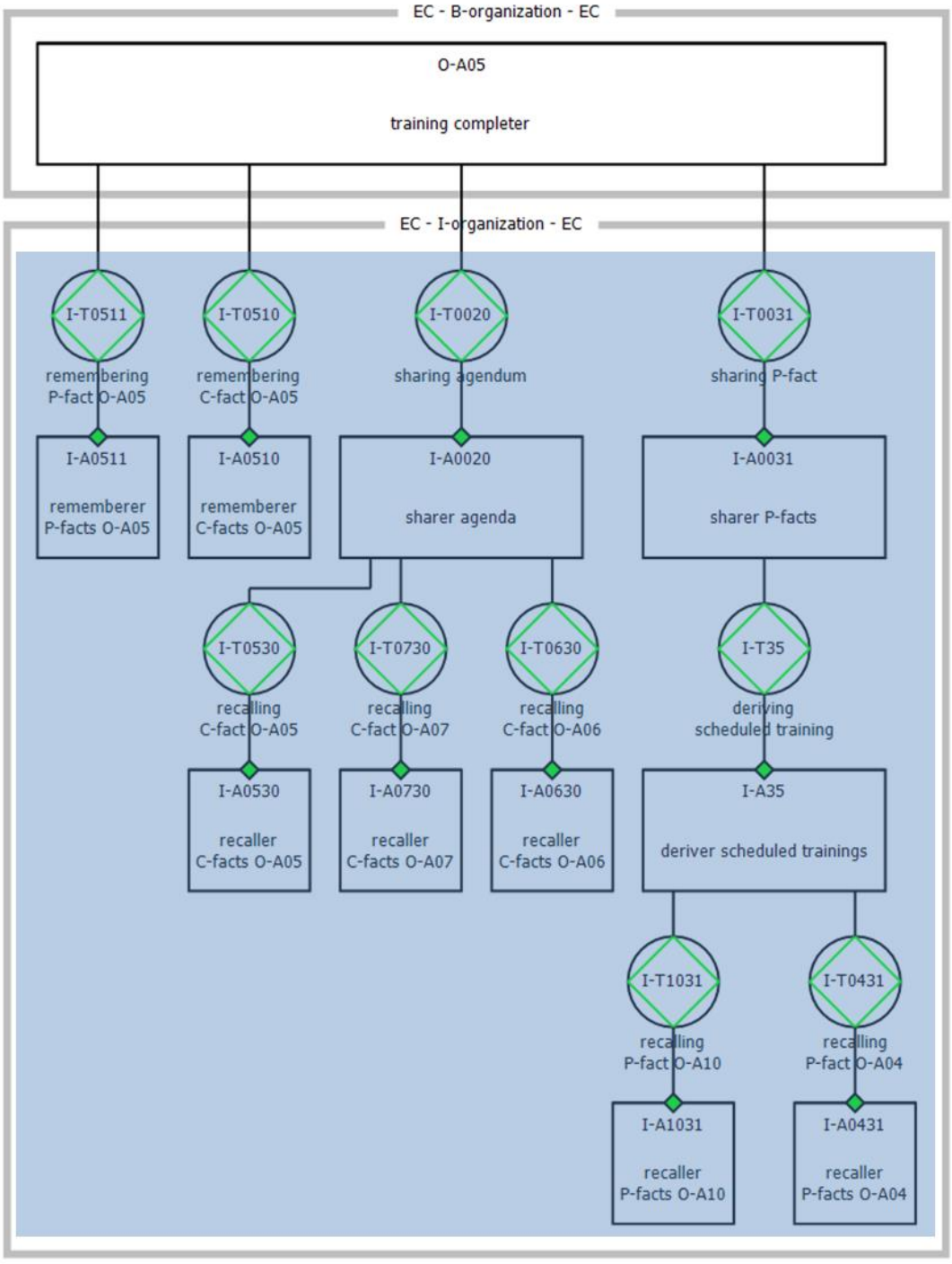


Fig. 7.8 Remembering and Sharing Facts for O-A05

Production facts:

15. Not applicable

General:

16. O-A05 does not know anything about the location of an agendum or any other requested fact. Therefore, two generic I-actors 'share agenda' and 'sharer P-facts' have to be implemented. They are responsible for sharing requested facts with O-A05. All facts are recalled or derived within the scope of COURSE.
17. Not applicable
18. Not applicable

Next, the aspects in the *Remembering Facts* aspect group are discussed:

Coordination facts:

19. The C-fact 'request' which is addressed to O-A06 and O-A07 is performed outside COURSE. An 'accept' is implicitly performed. Actually, there is no real acceptance procedure. O-A05 hopes (!) that the preparation is well done. This is a clear issue for improvement.
20. The trainer has been linked to TRAINING object by O-A04. On the basis of this dependent fact the secretary is able to know which subject to assign to the addressee of the 'request' for preparing and executing a specific training.
21. It should be better when the 'accept' of the preparation is registered with a time stamp. Then, it is clear for everybody when the preparation phase of a specific training has been completed formally.
22. Not applicable.

Production facts:

23. The P-fact is created every Wednesday.
24. Not applicable.
25. Not applicable.

Note:

- Concerning the transaction O-A06 between O-A05 and O-A06:
  Every week O-A05 prints out a list with trainings for the coming week. These trainings need to be prepared. Training materials must be constituted, class rooms prepared with the right number of laptops, and so on. A05 gives the list to A06 with the tacit request to prepare the trainings. A06 does not give a formal promise. It does also not give a formal 'state'. The current cooperation between O-A05 and O-A06 relies entirely on mutual trust. This rarely leads to problems in practice. One of the consequences is, however, that the preparation of trainings remains in the phase of improvisation. Consequently optimizing this process is hardly possible.

7.3.3 Implementation of O-A06

O-A06 prepares trainings which are provided in the class rooms of EC. A training is only prepared if a number of conditions are met (cf. Fig. 7.10). Figure 7.9 shows that actor O-

A06 uses facts which are remembered in the external production banks APB03, APB04, and APB05. The number of enrollments on the training to prepare can be derived from O-PB01 which corresponds to transaction O-T01. The final dates of the trainings are remembered in O-PB04. This production bank corresponds with O-T04. O-A06 prepares trainings based on a weekly schedule that O-A06 receives from O-A05. Several people can be involved in preparing trainings. A secretary copies trainings material, her colleague prints attendance lists and someone of the IT department installs the right number of PC's with the required software configuration in the class room. Several people are involved in the implementation of A06, whereby the trainer has the final responsibility.

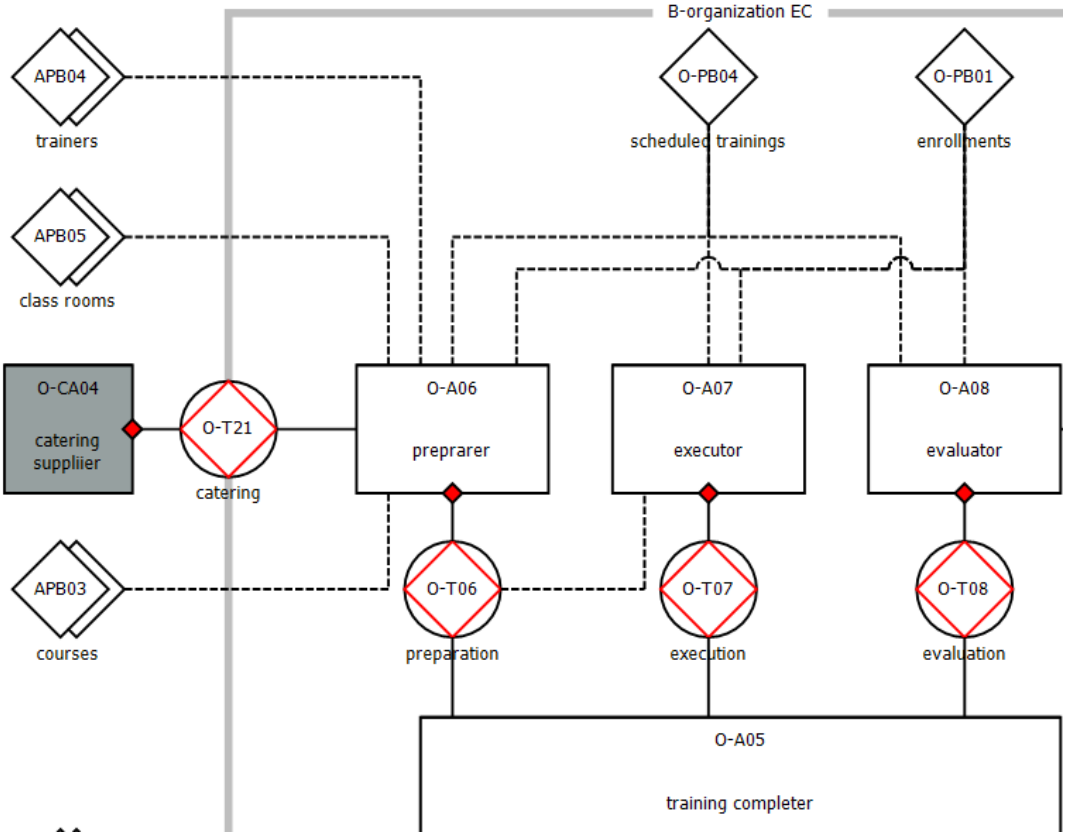


Fig. 7.9 CM of the training preparation

Actor O-A06:

	when preparation **for** Training <u>is requested</u>	O-T06/rq
assess	*justice:* **the** Performer **of the** <u>request</u> **is the** training completer **of** Week *sincerity:* <no specific condition> *truth:* <Training materials are available> <Class room is available and well equiped> <number of enrollments on Training is known> <specific Trainer wishes are known>	
if **then**	*complying with request is considered justifiable* <u>promise</u> preparation **for** Training	

when preparation **for** Training is promised	O-T06/pm
assess *justice:* **the** Performer **of the** promise **is the** preparer **of** Training *sincerity:* <no specific condition> *truth:* <no specific condition>	
if *complying with promise is considered justifiable* **then** request catering **for** Training execute preparation **for** Training state preparation **for** Training	

when catering **for** Training is stated	O-T21/st
assess *justice:* **the** Performer **of the** state **is the** catering supplier **of** Training *sincerity:* <no specific condition> *truth:* <catering proposal for Training is OK>	
if *complying with request is considered justifiable* **then** accept catering **for** Training	

Fig. 7.10 Action rules for O-A06

Figure 7.11 exhibits the I-organization with the I-actors that are invoked by O-A06. The dark colored parts are covered by COURSE.O-A06 requests for the following services:

Remembering facts:

a. C-facts and P-facts of O-A06 are remembered by I-A0610 and I-A0611, respectively.

Sharing agenda:

b. Agenda for O-A06 are shared by I-A0020 on request O-A06 .

Sharing facts:

c. The availability of training materials, It is remembered in APB03 (I-T7031).
d. The availability of the class room. It is remembered in APB05 (I-T9031).
e. The current configuration of the room. The current configuration of the room can be found in PB06 (I-T0631).
f. Specific needs of the trainer related to the training. It is remembered in APB04 (I-T8031).
g. The number of training participants enrolled on the training (I-T24).

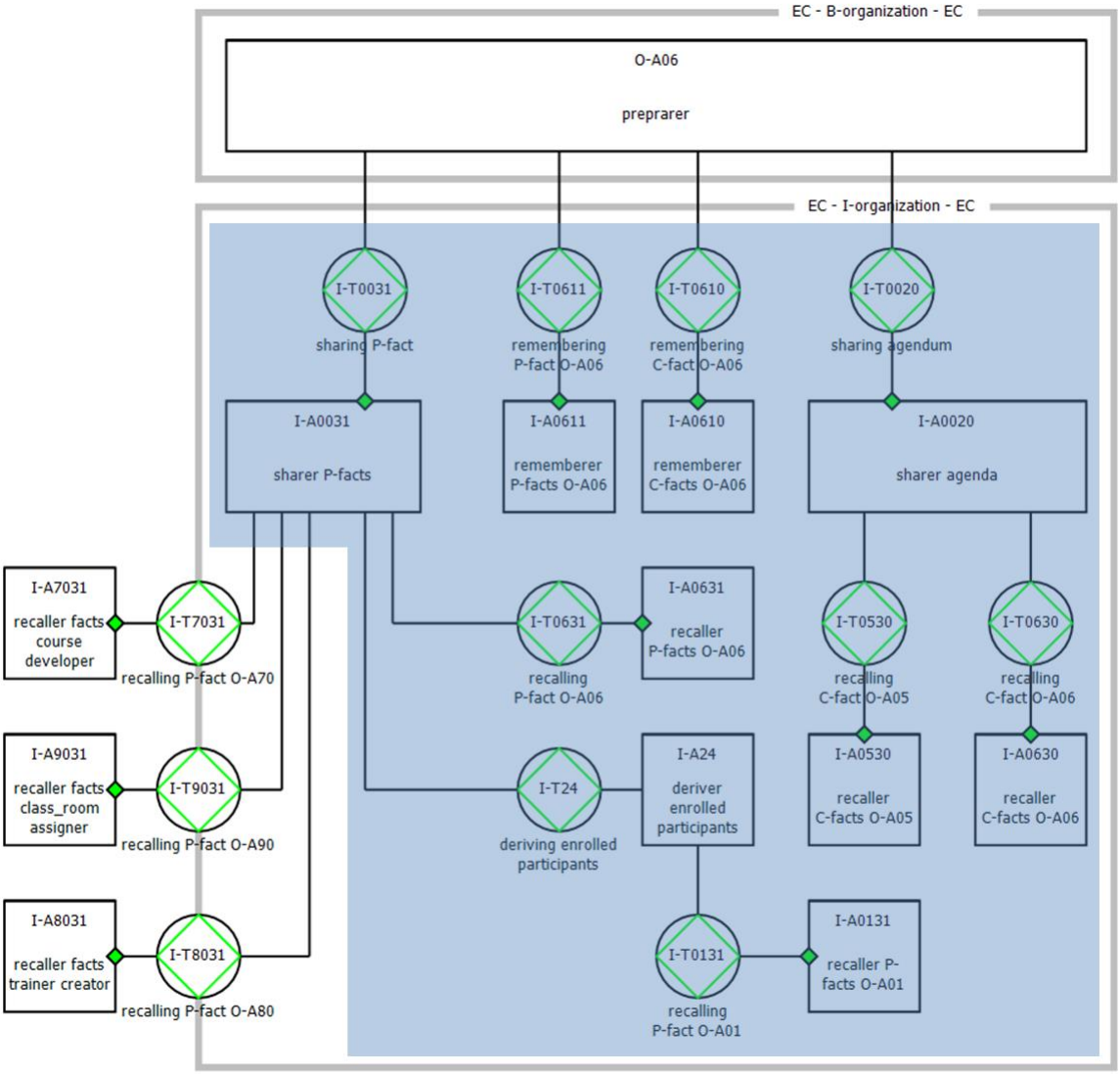


Fig. 7.11 Remembering and Sharing Facts for O-A06

We will continue the discussion of the implementation of O-A06 in the COURSE application on the basis of the rules from the PID-Framework related to the *Actor General Characteristics*.

5. O-A06 receives a list of trainings that should be provided by the EC the following week. The list could be considered as a bundle of 'requests' to O-A06.
6. O-A06 is performed by the trainer. Officially, it coordinates the preparation. In practice, however, everyone goes about their own individual business. Things usually work out fine due to the long-term experience of the individuals who execute acts that are linked to this actor role. Sometimes there are problems (e.g., someone forgot to copy part of the training material, or a laptop has been incorrectly installed in the classroom). This is discovered when the training is underway. It is therefore recommended not to assign teachers to this actor role. They are often unable (because they are too busy) to check the P-fact of O-A06 (O-R06).

7. As said at rule 6, several people are involved in the preparation of trainings: secretaries for copying training materials, IT people for preparing the IT configuration in the class room.
8. There is no good working environment for O-A06. It is all improvisation.
9. O-A06 is not supported by information technology. There is at most some coordination verbally and by email between the actors and their delegates. COURSE is not used. Neither the C-facts with O-A05 nor the P-fact of O-A06 are captured in COURSE. Therefore, no formal 'state' is returned to O-A05. This is a major shortcoming of COURSE.

The aspects in the *Sharing Facts* aspect group are discussed hereafter:

Agenda:

10. Along with the request, the TRAINING object has to be sent from O-A05 to O-A06. Currently, the actor only receives the name of the training and the date. Based on these facts, the current number of enrolled participants can be requested.
11. The 'request' is always sent by O-A05 to a specific subject. There is no need for a scheduling mechanism.
12. Not applicable.

Coordination facts:

13. Not applicable.
14. Not applicable.

Production facts:

15. Not applicable.

General:

16. O-A06 does not know anything about the location of an agendum or any other requested fact. Therefore, two generic I-actors 'share agenda' and 'sharer P-facts' have to be implemented. They are responsible for sharing requested facts with O-A06. The P-facts which are remembered in external production banks are not accessible through COURSE. They are requested outside COURSE. Actually, these facts are located in the minds of employees of EC.
17. Not applicable
18. Facts which are located in the minds of EC employees are sometimes hard to access by other actors. They have to be entered in COURSE manually and to be recalled in COURSE by agents.

Next, the aspects in the *Remembering Facts* aspect group are discussed:

Coordination facts:

19. Currently, O-A06 creates only explicit C-facts with O-CA04 (catering). All C-facts with O-CA05 are implicit and not registered.
20. Not applicable.
21. If the 'state' is created with a time stamp, it is clear for everybody when the preparation completes.
22. Not applicable.

Production facts:

23. The preparation has to be executed before the training actually starts.
24. Not applicable.
25. Not applicable.

7.3.4 Implementation of O-A07

A trainer provides the training for which he is scheduled. He may assume that the training is well prepared: the training documentation has been copied, the computer systems have been properly configured, and the attendance lists and evaluation forms have been printed. See Figure 7.9 for a relevant part of the CM and Figure 7.12 for action rules of O-A07.

Actor O-A07:

	when execution **for** Training is requested	O-T07/rq
assess	*justice:* **the** Performer **of the** request **is the** training completer **of** Week *sincerity:* <no specific condition> *truth:* <Training is well prepared> <list of participants available>	
if **then**	*complying with request is considered justifiable* promise execution **for** Training	

	when execution **for** Training is promised	O-T07/pm
assess	*justice:* **the** Performer **of the** promise **is the** executor **of** Training *sincerity:* <no specific condition> *truth:* <no specific condition>	
if **then**	*complying with promise is considered justifiable* request catering proposal **for** Training execute execution **for** Training state execution **for** Training	

Fig. 7.12 Action rules for O-A07

Figure 7.13 exhibits the I-organization with the I-actors that are invoked by O-A07. The dark colored parts are covered by COURSE.

O-A07 requests for the following services:

Remembering facts:
a. C-facts and P-facts of O-A07 are remembered by I-A0710 and I-A0711, respectively.

Sharing agenda:
b. Agenda for O-A07 are shared by I-A0020 on request by O-A07.

Sharing facts:
c. The participants of the training. The participants of the training are remembered in O-PB01 (I-T40).

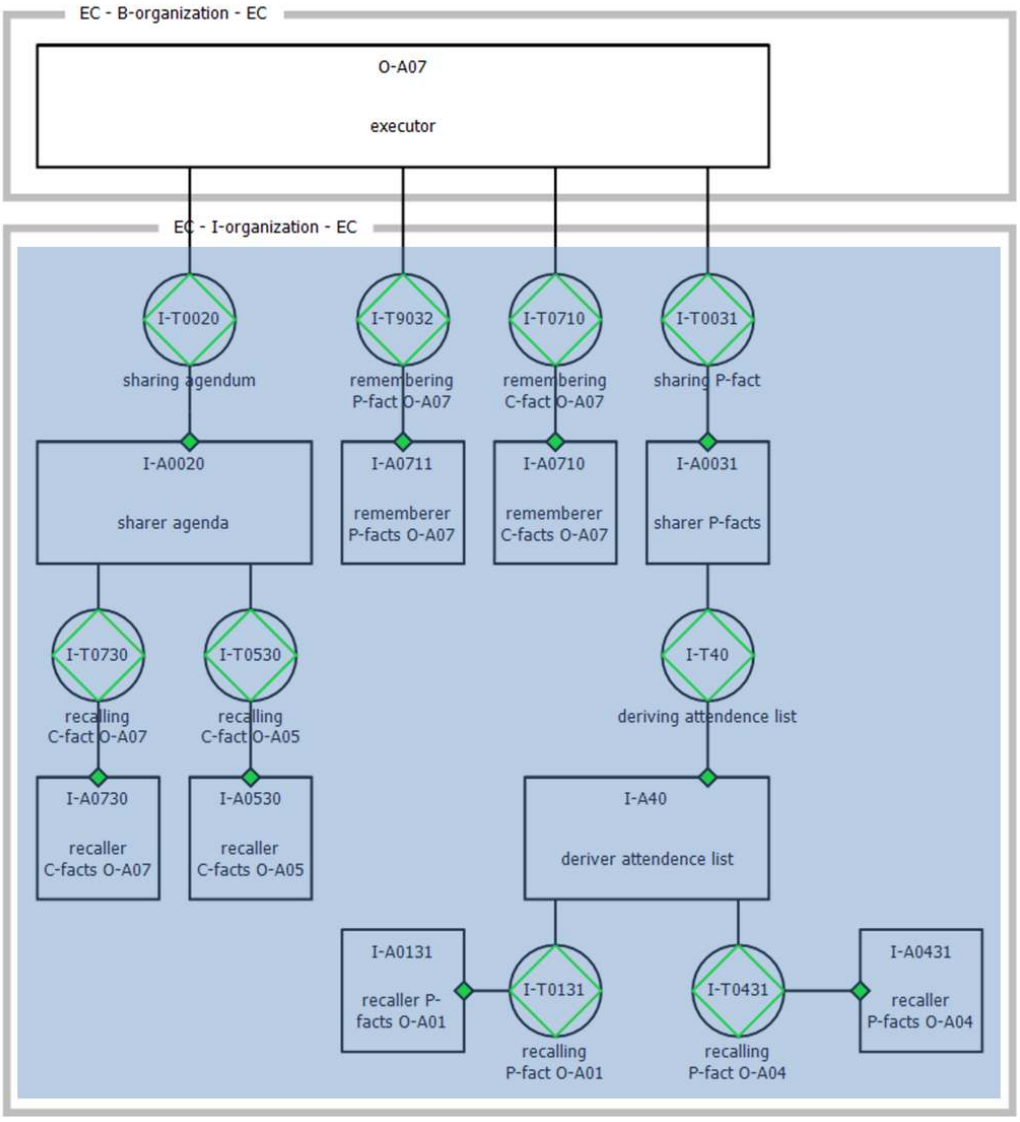


Fig. 7.13 Remembering and Sharing Facts for O-A07

We will continue the discussion of the implementation of O-A07 in the COURSE application on the basis of the rules from the PID-Framework related to the *Actor General Characteristics*.

5. Trainers use their own agenda, which are periodically aligned with O-PB05.
6. Each teacher is distinguished from other teachers by his specific skills and is therefore specialized in particular trainings. A request from O-A05 for providing training is always addressed to a specific trainer.
7. The attendance list is printed by the secretary of EC as a delegate of the actor.
8. It is important to take into account the specific nature of the trainer's role when specifying a good working environment for trainers. Trainers spend most of their time in front of a group of training participants. It is important for them to have an overview of all participants on paper, possibly with some background information.
9. Trainers have no access to COURSE. They use only their agenda and email facilities for coordination with O-A05.

The aspects in the *Sharing Facts* aspect group are discussed hereafter:

Agenda:

10. Along with the request, the TRAINING object has to be sent from O-A05 to O-A07. Currently, the actor does not receive anything. It knows already the identification of the training so that it is able to print the attendance list (by a delegate).
11. There is no scheduling mechanism. The teachers are distinguished from each other by specific competences and skills; they are specialized in providing a limited number of trainings. A trainer must receive a 'request' that is specifically addressed to him.
12. Not applicable.

Coordination facts:

13. Not applicable
14. Sometimes a teacher would like to see the enrollment facts from a participant. In particular, he is asking for these facts in order to check the prior knowledge or the expectations of the participant. Unfortunately, these facts are not registered during the enrollment.

Production facts:

15. Not applicable.

General:

16. O-A07 does not know anything about the location of an agendum or any other requested fact. Therefore, two generic I-actors 'share agenda' and 'sharer P-facts' have to be implemented. They are responsible for sharing requested facts with O-A06.
17. Not applicable.
18. Not applicable.

Next, the aspects in the *Remembering Facts* aspect group are discussed:

Coordination facts:

19. The ‘promise’ as well as the ‘state’ is implicitly implemented. Regarding the ‘promise’, the actor was involved in the preparation of the training. Therefore, ‘declining’ the execution of the training has not been taken place in the past. Regarding the ‘state’, the training has been scheduled for a fix number of days. The end of the last training day must be considered as an implicit ‘state’.
20. Not applicable.
21. Not applicable.
22. Not applicable.

Production facts:

23. The P-facts must be executed on a fixed date. This date is specified in the training schedule.
24. Not applicable.
25. Not applicable.

Note:
One aspect that is addressed in the discussion of this actor role is the importance of the ‘state’ coordination act. The registration of the result of O-A07 is essential for the EC. If there is a complaint regarding to a specific training, it is always related to the P-act of the trainer. Therefore, it is of importance to specify the dependent facts explicitly to avoid a sufficiently reliable memory of the EC production world.

7.4 Validation and Conclusions

The current implementation of COURSE in the EC has been systematically assessed based on the rules described in section 5.3. Not all actor roles have been discussed. A selection has been made from the collection of actor roles that may be considered representative in the EC. This case study shows that the current implementation of COURSE contains many shortcomings. It also indicates how these shortcomings could be solved.

Regarding the validation of the PID-Framework, we give a brief description of the operational organization of EC first. One of the key persons of EC is a secretary who has been appointed specifically to fulfill the actor roles O-A01 and O-A05. She is assisted by a junior secretary who spends her time mainly on a number of tasks outside EC. The actor roles O-A06 and O-A07 are fulfilled by trainers. The results of applying the PID-Framework for the implementation of the four mentioned actor roles have been discussed with the persons who fulfill those actor roles. It is remarkable that these actors seem satisfied about the current implementation of COURSE. That is not what the researcher did expect; he saw many issues that could be improved. How is it possible that the actors are quite satisfied? It became clear that these people, being a relatively small team with few personnel changes for

several years, were able to anticipate on many unexpected events and failures though improvisation. However, it became also clear, after discussing the results of this application, that 'working by improvisation' has many disadvantages, such as inefficient behavior, adverse effects on quality, lack of required information and unnecessary stressful situations. Based on the results of applying the PID-Framework they changed their mind. They saw that, through this research, the proper way of cooperation between employees of EC was made explicitly. The shortcomings of COURSE could be discussed. The following points were listed from this discussion:

- The aspect 'actor responsibility' has never been addressed explicitly in the past. This has been changed through the proposed approach in which actors plays a central role. Then the question arises whether the secretary fulfilled an actor role herself or that she was a delegate of her manager. The answer to this question had been given by the management of EC. It turned out that the secretary acts as a delegate, while her manager did not appoint her as a delegate explicitly.
- The secretary O-A01 has a transaction with client O-CA02. To detect a request for a new enrollment, she knows exactly where to look for it in COURSE. She told that the 'request' is implemented in an acceptable way. P-facts, which have to be invoked for performing the production act, are also available through COURSE. She also told that if the customer has his concerns about the registered enrollment, for example, the registration of a wrong course, he always reacts. Therefore, an explicit 'accept' of the enrollment is not necessary. To avoid misunderstandings in coordination, the general course conditions are tightened at this point.
- The degree of coordination between the actor O-A05 on one hand and the actors O-A06 and O-A07, on the other, seems very poor. The coordination between those actors is based on involvement and improvisation. Both the efficiency and the quality can be improved. The C-facts, as well as the P-facts of O-A06 and O-A07, are not captured in COURSE. This limits the availability of these facts to other actors. Additional questions from the researcher reveal that COURSE only provides the facility to print a list with trainings in a particular week. The list with trainings the week after, can only be printed then. It is sent to several subjects who fulfill the actor role O-A06. This list is considered as a list of requests from O-A05 to O-A06. After preparing a training, O-A06 did not send back a 'state' to O-A05. This is not correct; O-A05 now remains uncertain about the status of the 'prepare' transaction.
- O-A06 has no access to COURSE. Required facts have to be requested from other people in the organization. This is no problem if the size of the organization is relatively small, and people are always available, but that is not always the case in practice. Therefore, the current way of working is neither efficient nor practical.
- Preparation of trainings comprises various aspects: it regards the system configuration in the training room, printed manuals, the attendance list, special wishes from attendants, etc. During preparation, some aspects are forgotten easily. Through improvisation, however, issues can be resolved soon. This is not the professional way of working they prefer.

  The secretary and the teachers indicate that they are glad about the proposed model. The model clearly shows what the communication lines should be and what should be paid attention to.

- O-A07 also has no access to COURSE. The list of training participants is sent to the trainer by the secretary (O-A05) by email. It is a pity that trainers are unable to follow the enrollments of the respective training. It could improve the quality of their preparation on the training if they know the company of the participant early.
- The training report that has to be written after providing the course by the trainer, is not captured in COURSE, due to the absence of a COURSE interface for O-A07. The report is written in MS Word and sent to the secretary by e-mail. In practice, this is often neglected due to which the fact comes into oblivion.

Summarizing, both secretary and teachers indicate that the model clearly shows how the coordination between the actors, ontological as well as infological, should take place. They indicate that the model also clearly reveals the bottlenecks that they were not always aware of. Whether it is worthwhile to extend the functionality of COURSE fell outside the scope of the discussion.

Applying the PID-Framework in assessing the implementation of COURSE gives the following benefits:

From the perspective of the using system (organization):

- COURSE is used for about six years. The current users are satisfied with it. Evaluating the current implementation of COURSE with the use of the PID Framework shows both the risks of their current way of working and the lack of information for fulfilling their actor roles professionally.
- The implementation of COURSE within EC shows the importance of implementing an EIS in an organization. An EIS enforces the explicit execution of the successive process steps. Although some process steps can be executed implicitly, the choice to execute an implicit process step must be done consciously. The case study shows that this not always the case. Consequently, failures are solved through improvisation. They do not lead to structural improvement of the implementation of COURSE.

From the perspective of the engineer:

- The actor role is a central element in the proposed approach. While an organizational function may cover one or more actor roles, the facts to ask for and to remember can be specified for an organizational function. Employees are more able to talk from their organization function than from their actor roles. For that reason, discussions with them can be efficient.
- The use of the PID-Framework leads to a standardized approach in evaluating COURSE based on the ontological model of the organization. For example, the assessment of each actor role implementation is executed according to the same rules.
- The information needs per actor role are elaborated in the integral ontological model of the organization. This model appears to be a good starting point for weighing implementation options regarding the choice of technology.

8 Case 3: Wholesale Flower Bulbs

8.1 Introduction

Around 85 percent of the global trade in flower bulbs is performed by a few wholesalers in the Netherlands. The trade in flower bulbs can be subdivided in “flower grower sales” and “dry sales”. The trade in “flower grower sales” consists of the supply of bulbs to enterprises that grow cut flowers. The trade in “dry sales” focuses on retail businesses, wholesale enterprises, garden centers, etc. The bulbs are packed in boxes or bags. The boxes and bags are delivered in all kinds of shapes and sizes, even in racks. The flower bulbs are purchased either directly from the growers or via intermediaries. The bulbs are usually packed on the wholesaler’s premises but this task is sometimes outsourced to specialist enterprises. Bulbs differ from many other products in that they can suffer from rot. Furthermore, different lots of the same type of bulb can differ in quality.

A number of “dry sales” wholesalers have joined forces in an ICT think tank under the leadership of the trade association Anthos. There was a commonly felt lack of good-quality software for supporting the “dry sales” business processes. The researcher was asked to give a presentation on the subject of how to develop a common EIS that would be suited for this line of business. The suggestion was made to design a so-called sector model that could serve as the basis for the specifications for a supporting business application. The sector model was designed using the principles of enterprise engineering. Seven different “dry sales” wholesalers were visited and interviews were held that lasted about two hours. They are all recorded. A draft sector model was drawn up on the basis of these interviews. This sector model was presented at a meeting of the ICT think tank. Three enterprises also stated at the meeting that they would be prepared to validate the model. A single day was reserved for the validation.

This chapter focuses on the design of an implementation model for wholesalers that are specialized in “dry sales”. In view of the model’s size, a selection has been made of the parts to be subjected to detailed elaboration. The selection was done in order to avoid more of the same. It does not make sense to focus on the application of the PID-Framework in situations that have been dealt with previously.

8.2 Context

The sector model for the flower bulb “dry sales” wholesale sector, hereafter referred to as the FB-DS sector, consists of five groups of related composite actor roles, namely CA07, CA08, CA09, CA10, and CA11 (cf. Fig. 8.1). Six external parties have been identified. These are actor role CA02 that concludes the sales contract, actor role CA03 that receives the goods, actor role CA05 with which the dealer closes the purchasing contract, actor role CA06 that delivers the bulbs, actor role CA12 that supplies the packing material,

and the external inspector CA04. The ontological model of the organization is restricted to modeling the essence of the organization. Some fact banks are positioned external to the model in order to avoid making the model excessively complex. This consideration can be seen in Figure 8.1. The fact banks APB05, APB06, APB07, APB10, and APB11 have been defined as external. This may be an acceptable choice from the point of view of the implementation of business processes. However, the decision to make these fact banks external is open to debate (with the exception of APB11) from the point of view of the implementation of information processes. We shall return to this point later.

The case to discuss in this section is restricted to the composite actor role CA08. A more detailed CM of a part of CA08 is exhibited in Figure 8.2. We focus on the area within the grey encircling in the CM. The accompanying FM is exhibited in Figure 8.3.

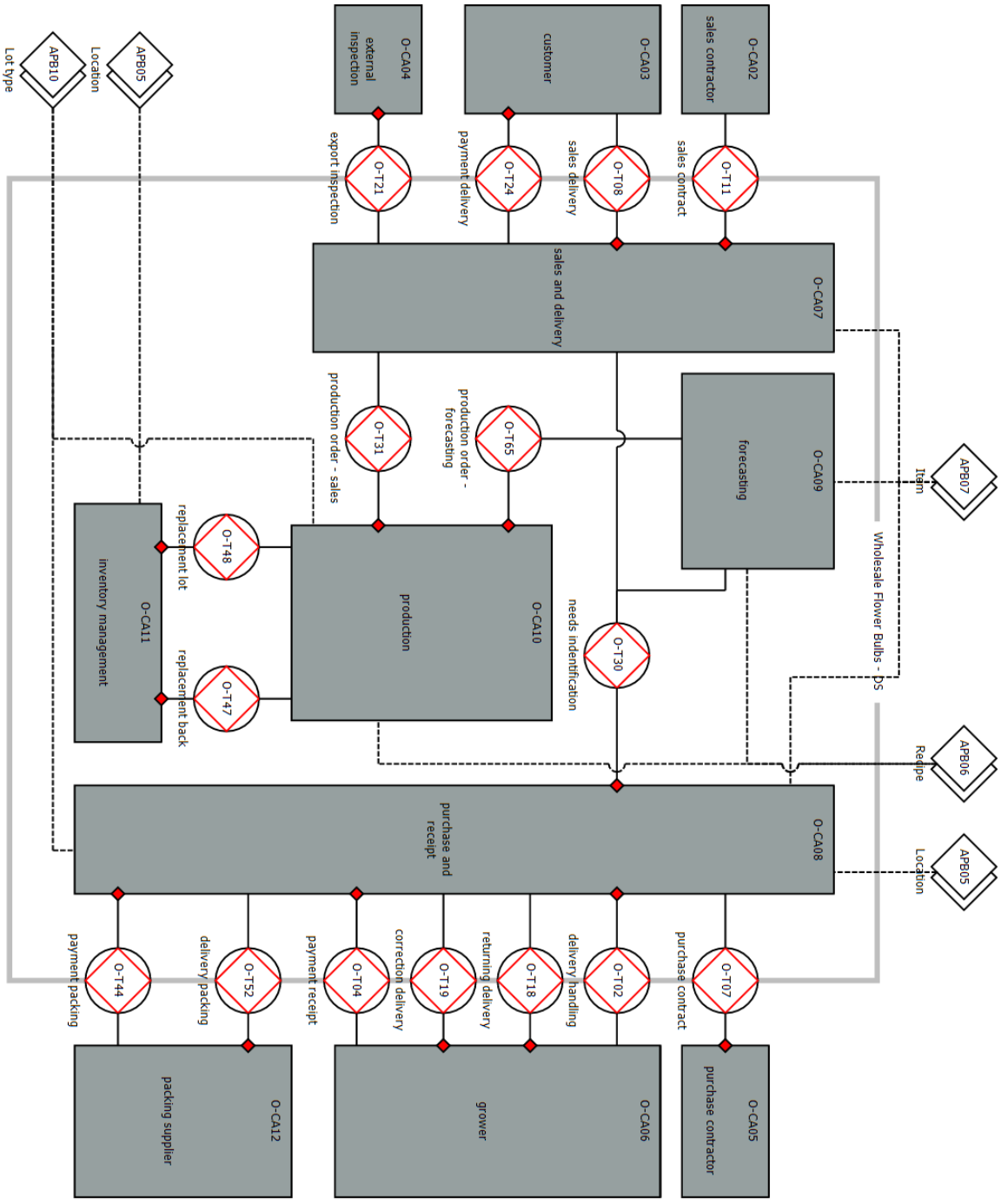
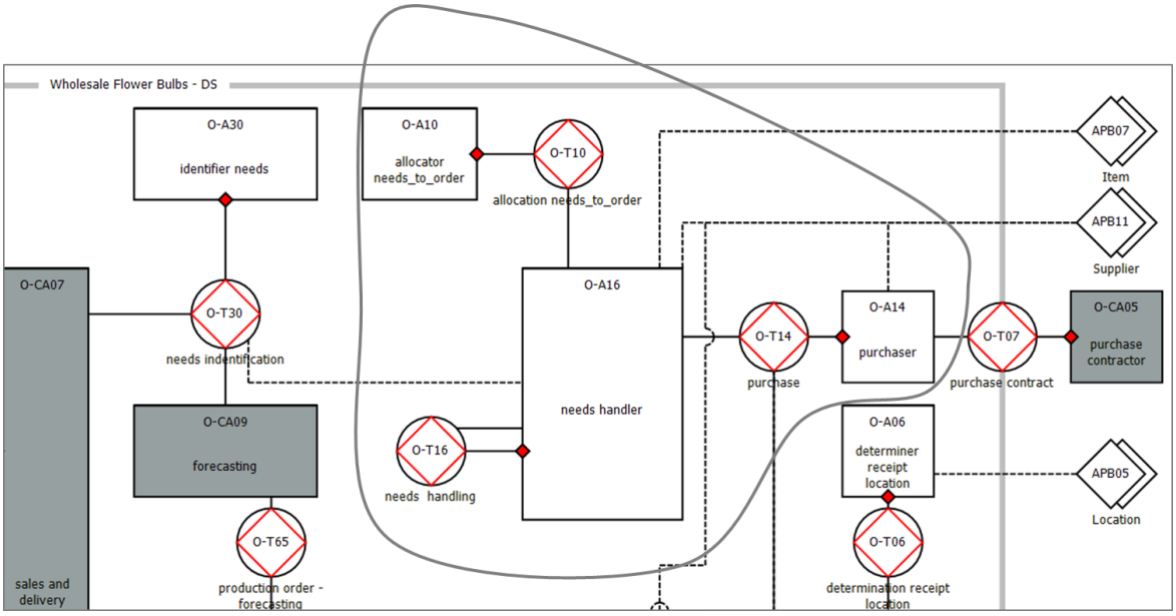


Fig. 8.1 Construction Model of the FB-DS sector



Transaction Kind	Result Kind
O-T10 - allocation needs_to_order	O-R10 - [order _ need] has been allocated
O-T14 - purchase	O-R14 - [purchase order] has been concluded
O-T16 - needs handling	O-R16 - needs on [hour] has been handled
O-T30 - needs identification	O-R30 - [need] has been identified

Fig. 8.2 Construction Model of a part of O-CA08

We will now discuss the *context* of the FB-DS sector point by point, based on the rules of the PID-Framework (see section 5.3). The numbering in this chapter corresponds to the numbering of the rules.

1. The result kinds linked to the categories in the FM (cf. Fig. 8.3) are used to determine the actors in the CM (cf. Fig. 8.2) that fall within the scope of the EIS.
2. We discuss the question of which I-actors fall within or outside the scope of the EIS when elaborating the individual actor roles.
3. We will not discuss this point in any further detail in view of the fact that the EIS to be developed is more extensive than the part we are focusing on in this case study.
4. The clustering aspect is hard to implement in a sector model. The individual enterprises within the sector make distinct choices in this respect. Even so, a few clusters can be established on the basis of Figure 8.2. Let us first consider the actor roles O-A10, O-A14, and O-A16. It turned out that these actor roles are fulfilled by the same person in each enterprise.

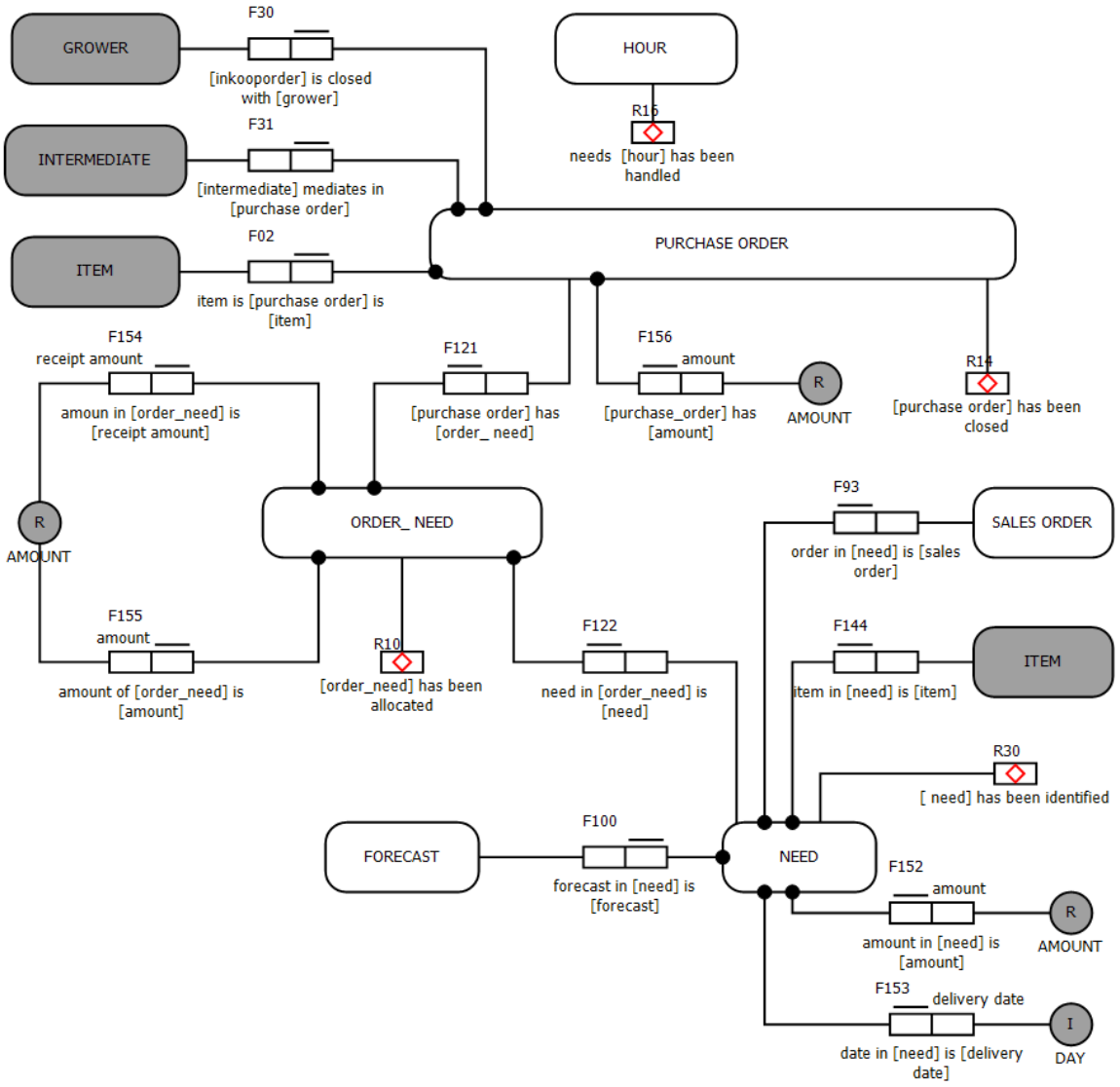


Fig. 8.3 Fact Model of part of O-CA08

8.3 Actor Roles

Next, we elaborate the specifications for actor roles based on the PID-Framework. We will look at the actor roles in the process ‘Generating purchase orders on the basis of purchasing needs’ (sections 8.3.1 through 8.3.3).

8.3.1 Implementation of O-A16

Flower bulbs are purchased from bulb growers or from purchasing agencies that act as an intermediary between growers and traders. The acquisitions are made on the basis of purchasing needs. Purchasing needs are created at two different points within the organization:

from sales (order-driven purchasing needs) and from forecasts ("anonymous" purchasing needs). Each purchasing need comprises a specification of the kind of flower bulb to be purchased plus the quantity and the desired delivery date. The purchasing needs drawn up internally are used in the purchasing department as the basis for contracts concluded with suppliers. The purchaser is free to operate completely independently in this regard.

The model is developed in such a way that the salesperson can always see whether the bulbs for a specific sales transaction have been purchased. The purchasing needs can be found in the transaction bank O-PB30 (cf. Fig. 8.2). At regular intervals, a purchaser reads the purchasing needs and searches in his network for a supplier who is able to deliver these bulbs. If he reaches an agreement with a supplier, he records the agreement in a purchase order so that it is registered how many bulbs of a particular type are expected from this supplier. Then, he specifies for each purchasing need the extent to which it has been converted into purchase orders.

It is possible for the purchaser to buy more bulbs than the quantity specified in the purchasing needs. On delivery, the excess bulbs are added to the free available stock. The free available stock is taken into account when generating purchasing needs, to prevent the wholesaler from being left with surpluses. Figure 8.2 shows that the total performance of the purchaser can be split into the performance of three different actor roles: (1) the handling of the prepared purchasing needs by O-A16, (2) the establishment by O-A14 of a purchase order with a supplier, and (3) the allocation by O-A10 of the prepared purchasing needs to the established purchase order. These roles are all part of the same cluster. The relevant part of the AM is exhibited in Figure 8.4.

Actor O-A16:

	when needs handling **for** Hour is requested	O-T16/rq
assess	*justice:* **the** Performer **of the** request **is the** needs handler **of** Hour *sincerity:* <no specific condition> *truth:* <no specific condition>	
if **then**	*complying with request is considered justifiable* request needs handling **for** next Hour **with** requested_creation_time = now + 1 Hour promise needs handling **for** Hour	

	when needs handling **for** Hour is promised	O-T16/pm
assess	*justice:* **the** Performer **of the** promise **is the** needs handler **of** Hour *sincerity:* <no specific condition> *truth:* <no specific condition>	
if **then**	*complying with promise is considered justifiable* **do for** < each genus/variety/size of flower bulb in O-PB30 > request purchase **for** new Purchase_order **with** **the** genus **of** Purchase_order **is** a Genus **the** variety **of** Purchase_order **is** a Variety **the** size **of** Purchase_order **is** a Size **the** supplier **of** Purchase_order **is** a Supplier **the** quantity **of** Purchase_order **is** a Quantity **od**	

when purchase **for** Purchase_order is stated	O-T14/st
assess *justice:* **the** Performer **of the** state **is the** purchaser **of** Purchase_order *sincerity:* <no specific condition> *truth:* <Purchase_order has been agreed>	
if *complying with statement is considered justifiable* **then** accept purchase **for** Purchase_order	

when purchase **for** Purchase_order is accepted	O-T14/ac
assess *justice:* **the** Performer **of the** accept **is the** needs handler **of** Hour *sincerity:* <no specific condition> *truth:* <needs in O-PB30 available for allocation>	
if *complying with acceptance is considered justifiable* **then** **do for** <all bulbs in purchase order> request allocation needs_to_order **for** new Order_need **with** **the** purchase_order **of** Order_need **is** a Purchase_order **the** need **of** Order_need **is** a Need **od**	

when allocation needs_to_order **for** Order_need is stated	O-T10/st
assess *justice:* **the** Performer **of the** state **is the** allocator needs_to_order **of** Order_need *sincerity:* <no specific condition> *truth:* <Purchase_order has been agreed>	
if *complying with statement is considered justifiable* **then** accept allocation needs_to_order **for** Order_need	

when allocation needs_to_order **for** Order_need is accepted	O-T10/ac
assess *justice:* **the** Performer **of the** accept **is the** needs handler **of** Hour *sincerity:* <no specific condition> *truth:* <Purchase_order is de last accepted purchase and Order_need is the last accepted allocation needs_to_order>	
if *complying with acceptance is considered justifiable* **then** execute needs handling **for** Hour state needs handling **for** Hour	

	when needs handling **for** Hour is stated	O-T16/st
	assess *justice:* **the** Performer **of the** request **is the** needs handler **of** Hour *sincerity:* <no specific condition> *truth:* <no specific condition>	
	if *complying with statement is considered justifiable* **then** accept needs handling **for** Hour	

Fig. 8.4 Action rules for O-A16

Figure 8.5 exhibits the I-organization with the I-actors that are invoked by O-A16. The dark colored part is covered by the EIS to be developed. O-A16 requests for the following services:

Remembering facts:

a. C-facts and P-facts of O-A16 are remembered by I-A0710 and I-A0711, respectively.

Sharing agenda:

b. Agenda for O-A16 are shared by I-A0020 on request by O-A16.

Sharing facts:

c. Based on the needs for purchase the item data is requested. These data is remembered in the external bank APB07 (I-T9231).
d. The supplier data is requested. It is remembered in the external bank APB11 (I-T9331).
e. Purchase needs are requested from O-PB30. These needs are grouped on item and supplier (I-T3031).

We will continue the discussion of the implementation of O-A16 in the EIS on the basis of the rules from the PID-Framework related to the *Actor General Characteristics*.

5. O-A16 is a self-activating actor role. If one or more purchasing needs are generated, O-A16 picks them up and handles them. A subject who fulfills this role is an expert in acquiring specific bulb kinds. In the action rule is specified a one hour periodicity. In practice, it is more flexible.
6. O-A16 is fulfilled by several people. Mostly, some of them are specialized in purchasing specific bulb genius. This can be specified in the employee object class. It is not needed for this application. An actor knows its own specialism (by heart) and requests only the purchasing needs from I-A3031 which can be treated by it.
7. Not applicable.
8. O-A16, O-A14, and O-A10 belong to the same cluster. All their coordination activities are geared to each other and take place without using the EIS. It is important to design

a common interface for these actors so that the same facts retrieved by actors in different roles only have to be retrieved once within the interface.

9. O-A16 has to be supported by the EIS for all information services (cf. Fig. 8.5).

The aspects in the *Sharing Agenda* aspect group are discussed hereafter:

10. The agendum 'requested' is linked to an HOUR object. There are no dependent facts linked to this object. In case O-A16 initiates O-A10 or O-A14, a PURCHASE_ORDER_NEED object or a PURCHASE_ORDER object was sent along with the 'state', respectively. Through the 'state' O-A16 acquires all current dependent facts of these objects.
11. There is no scheduling mechanism needed.
12. Not applicable.

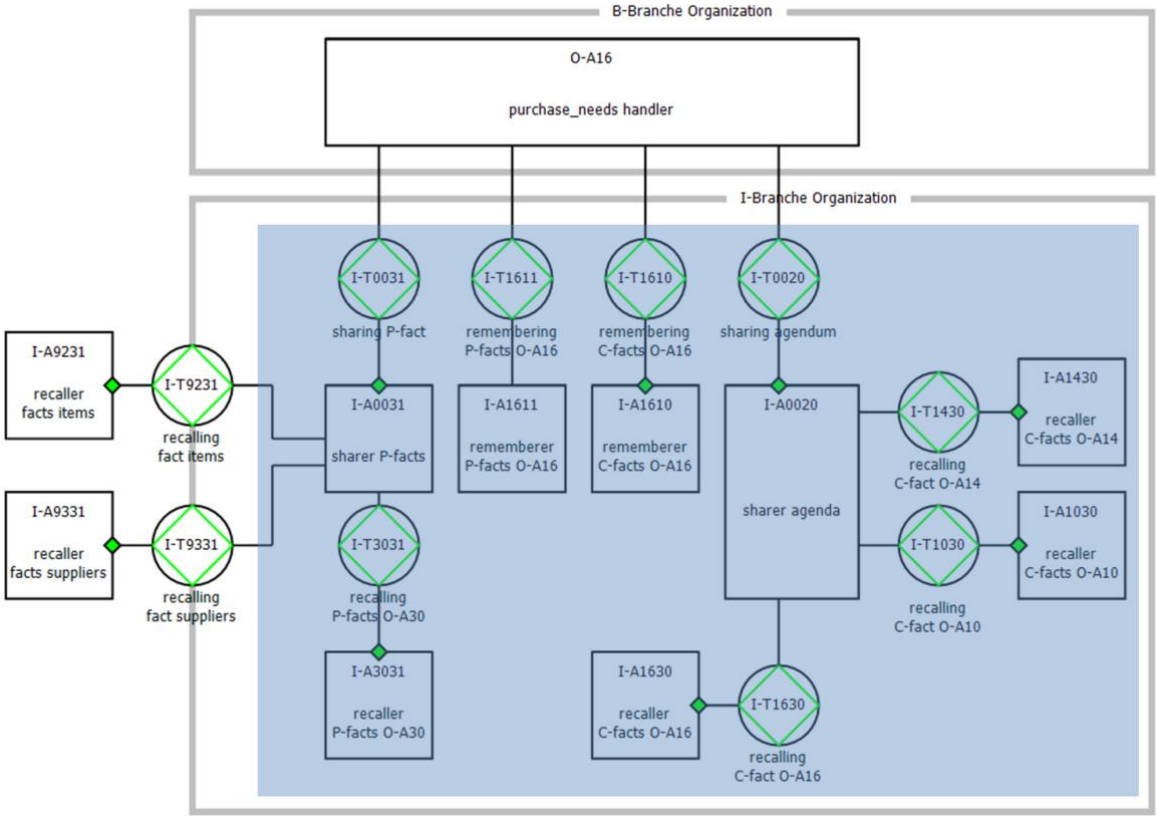


Fig. 8.5 Remembering and Sharing Facts for O-A16

Next, the aspects in the *Sharing Facts* aspect group are discussed.

Coordination facts:

13. Not applicable.
14. Not applicable.

Production facts:

15. Not applicable.

General:

16. O-A16 does not know anything about the location of an agendum or any other fact. Therefore, two generic I-actors 'sharer agenda' and 'sharer P-facts' have to be implemented. They are responsible for sharing requested facts with O-A16.
17. Not applicable.
18. O-A16 requests item information from I-A9231. I-A9231 is consulted at high frequency. Therefore, it is important to bring the rememberer and the recaller of items within the scope of the EIS. The situation with suppliers is different. The supplier data is mainly remembered in the mind of the purchaser. The actor I-A9331 is in most cases the same subject as its requester who fulfills O-A16. The supplier data change from day to day. In season, O-A16 calls its suppliers daily in order to be updated about available quantities, bulb genius, varieties and sizes.

Lastly, the aspects in the *Remembering Facts* aspect group are discussed.

Coordination facts:

19. The C-facts are implicit. O-A16 initiates transactions with actors that are in the same cluster.
20. Not applicable, because the addressees are in the same cluster.
21. Not applicable.
22. Not applicable.

Production facts:

23. Not applicable.
24. Not applicable.
25. Not applicable.

Notes:

- Self-initiation is rigidly defined in the action rules. There is a fixed periodicity in the model. In practice this is usually not the case. Hours may pass before O-A16 starts a new initiation. On other days there may be multiple initiations carried out by different people at any time.
- The item bank APB07 is positioned outside the organization. This is not unusual in designing the ontological model of an organization. It is a means to keep the model simple. However, for developing an EIS it is preferred to bring this fact bank within the scope of the EIS, given the great frequency with which it is consulted.

8.3.2 Implementation of O-A14

O-A16 provides O-A14 with the quantity of flower bulbs of a specific genius/variety/size that a specific supplier is able to deliver. O-A14 actually closes contracts (O-T07) with the purchase contractor O-CA05. Its request for a contract is preceded by a request to I-A9331 for the conditions that the supplier sets for the delivery of bulbs of a particular genius/variety/size. This includes the price and the delivery terms.

Actor O-A14:

	when purchase **for** Purchase_order is requested	O-T14/rq
	assess *justice:* **the** Performer **of the** request **is the** needs handler **of** Hour *sincerity:* <no specific condition> *truth:* <no specific condition>	
	if *complying with request is considered justifiable* **then** promise purchase **for** Purchase_order	

	when purchase **for** Purchase_order is promised	O-T14/pm
	assess *justice:* **the** Performer **of the** promise **is the** purchaser **of** Purchase_order *sincerity:* <no specific condition> *truth:* <supplier conditions are agreed>	
	if *complying with promise is considered justifiable* **then** request purchase contract **for** Purchase_order	

	when purchase contract **for** Purchase_order is stated	O-T07/st
	assess *justice:* **the** Performer **of the** state **is the** purchase contractor **of** Purchase_order *sincerity:* <no specific condition> *truth:* <Purchase_contract has been agreed>	
	if *complying with statement is considered justifiable* **then** accept purchase contract **for** Purchase_order execute purchase **for** Purchase_order state purchase **for** Purchase_order	

Fig. 8.6 Action rules for O-A14

That request is made to a flower-bulb grower or to an intermediary. The intermediary offers some lots of flower bulbs on the behalf of growers. These arrangements are made by telephone. Generally, the 'request' by O-A14 is made through a phone call; the 'promise' of the supplier is mostly made through the same phone call and is confirmed by fax or by mail. The relevant part, of the CM and the AM are exhibited in Figure 8.2 and 8.6, respectively.

Figure 8.7 exhibits the I-organization with the I-actors that are invoked by O-A14. The dark colored part is covered by EIS to be developed. O-A14 requests for the following services:

Remembering facts:

a. C-facts and P-facts of O-A14 are remembered by I-A1410 and I-A1411, respectively.

Sharing agenda:

b. Agenda for O-A14 are shared by I-A0020 on request by O-A14.

Sharing facts:

c. The supplier conditions are requested. It is remembered in the external bank APB11 (I-T9331).

We will continue the discussion of the implementation of O-A14 in the EIS on the basis of the rules from the PID-Framework related to the *Actor General Characteristics*.

5. O-A14 is initiated by O-A16 with the aim of concluding a purchase order with a bulb supplier. The initiation takes place at arbitrary points in time. It may occur several times a day but also once a week or once a month.
6. O-A14 belongs to the same cluster as O-A16 and O-A10.
7. Not applicable.
8. There is common working environment with O-A16 and O-A10.
9. O-A14 is a business actor and therefore cannot be replaced by an agent. The new application is intended to support remembering production orders and invoking agenda.

The aspects in the *Sharing Agenda* aspect group are discussed hereafter:

10. Along with an agendum the object PURCHASE_ORDER is sent to O-A14. O-CA05 sends the contract data with the ‘state’. The ‘request’ to O-CA05 is usually done by telephone. Both parties negotiate by telephone and during the same call O-CA05 creates a ’promised’ or a ‘declined’. O-CA05 confirms its sales order by fax or by email. The confirmation contains both the ‘state’ of O-T22 and the P-fact of O-CA05. O-A14 takes note of it and uses it in its P-act.
11. Not applicable.
12. Along with the ‘promise’ and the ‘state’ O-A14 receives a bundle of purchase orders. O-A14 unbundles these C-facts and allocates them to the separate purchase orders in the bundle. These single purchase orders are known, because the initial ‘request’ to O-CA05 was bundled by O-A14.

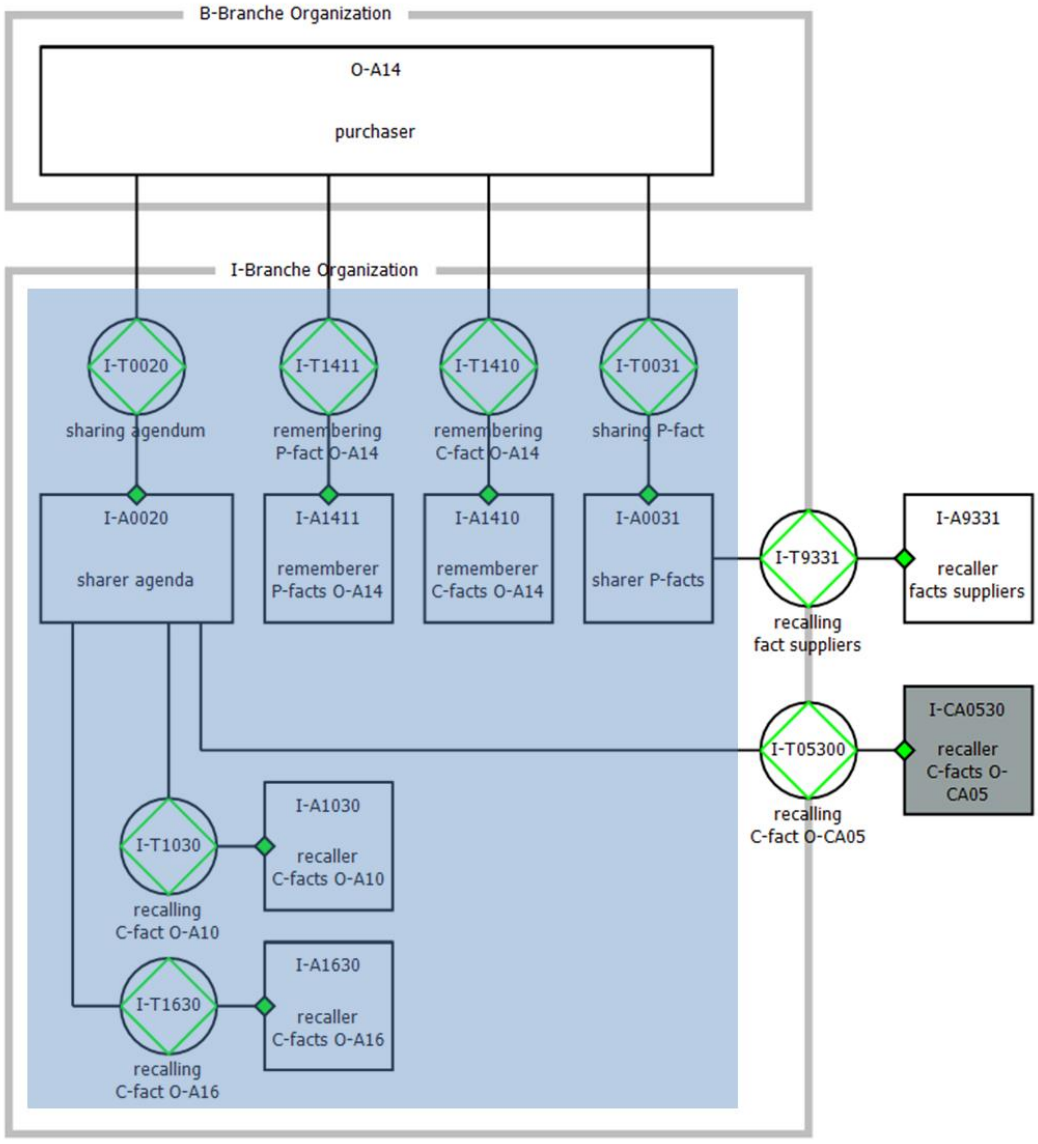


Fig. 8.7 Remembering and Sharing Facts for O-A14

Next, the aspects in the *Sharing Facts* aspect group are discussed.

Coordination Facts:

13. Not applicable.
14. The creator of the 'state' within O-T22 is the contractor from the supplier who concludes the contract. This fact has to be remembered.

Production facts:

15. Not applicable.

General:

16. O-A14 does not know anything about the location of an agendum or any other fact. Therefore, two generic I-actors 'sharer agenda' and 'sharer P-facts' have to be implemented. They are responsible for sharing requested facts with O-A14. For making the C-facts of O-CA05 available within the EIS, a recaller of these facts has to be implemented within the scope of the EIS.
17. The internal recaller of O-CA05 must be fulfilled by a delegate of O-CA05. It could be the subject that fulfills O-A14. This subject receives all C-facts by mail or by phone. Then it's easy to put these facts also in the EIS.
18. The supplier data is mainly remembered in the mind of the purchaser. The actor I-A9331 is in most cases the same subject as the one who fulfills O-A14. The most important communication medium for O-A14 is the telephone and the email system. Is does not make sense to implement a recaller within the scope of the EIS.

Lastly, the aspects in the *Remembering Facts* aspect group are discussed.

Coordination facts:

19. The C-facts with O-A16 are implicit created. The C-facts with O-CA05 are created explicitly.
20. A 'request' to O-CA05 has a specific addressee. The addressee is mostly known to O-A14. O-A14 negotiates through the telephone with someone from the supplier. That person is the addressee of the 'request'.
21. The addressee of the 'request' is also the performa owner of the 'promise' and of the 'state'.
22. O-A14 receives a request from O-A16 for each purchase order. However, purchase orders for the same purchase contract party (grower or intermediary) and within the same transaction O-T16 are offered to O-CA05 as a bundle (purchase order with purchase order lines).

Production facts:

23. Not applicable.
24. Not applicable.
25. A subclass of the external object class SUPPLIER is created in the EIS in order to be able to create instances of the binary fact kinds of which one of the domains is the external object class SUPPLIER.

Notes:

- An internal actor that has a transaction with an external actor remembers the production result of the external actor as part of its own production result in a transaction bank within the scope of the EIS.
- An internal actor that initiates a transaction with an external actor receives the P-fact along with the 'state'. The internal actor is able to execute an 'accept' with respect to the P-fact without requesting to share it explicitly.

- This section contains a good example of bundling purchase orders. We start from the assumption that an actor that constitutes a bundle is also able to split up the bundle in single purchase orders.

8.3.3 Implementation of O-A10

Closed purchase orders need to be allocated to the purchasing needs. Once that has been done, salespeople for instance are able to see which purchase orders are linked to a particular purchasing need established by them. The relevant parts of the CM and the AM are exhibited in Figure 8.2 and 8.8, respectively.

Actor O-A10:

	when allocation needs_to_order **for** Order_need is requested	O-T10/rq
	assess *justice:* **the** Performer **of the** request **is the** needs handler **of** Hour *sincerity:* <no specific condition> *truth:* <no specific condition>	
	if *complying with request is considered justifiable* **then** promise allocation needs_to_order **for** Order_need	

	when allocation needs_to_order **for** order_need is promised	O-T10/pm
	assess *justice:* **the** Performer **of the** promise **is the** allocator needs_to_order **of** Order_need *sincerity:* <no specific condition> *truth:* <no specific condition>	
	if *complying with promise is considered justifiable* **then** execute allocation needs_to_order **for** Order_need state allocation needs_to_order **for** Order_need	

Fig. 8.8 Action rules for O-A10

Figure 8.9 exhibits the I-organization with the I-actors that are invoked by O-A10. The dark colored part is covered by the EIS to be developed. O-A10 requests for the following services:

Remembering facts:

a. C-facts and P-facts of O-A10 are remembered by I-A1010 and I-A1011, respectively.

Sharing agenda:

b. Agenda for O-A10 are shared by I-A0020 on request by O-A10.

Sharing facts:

c. None.

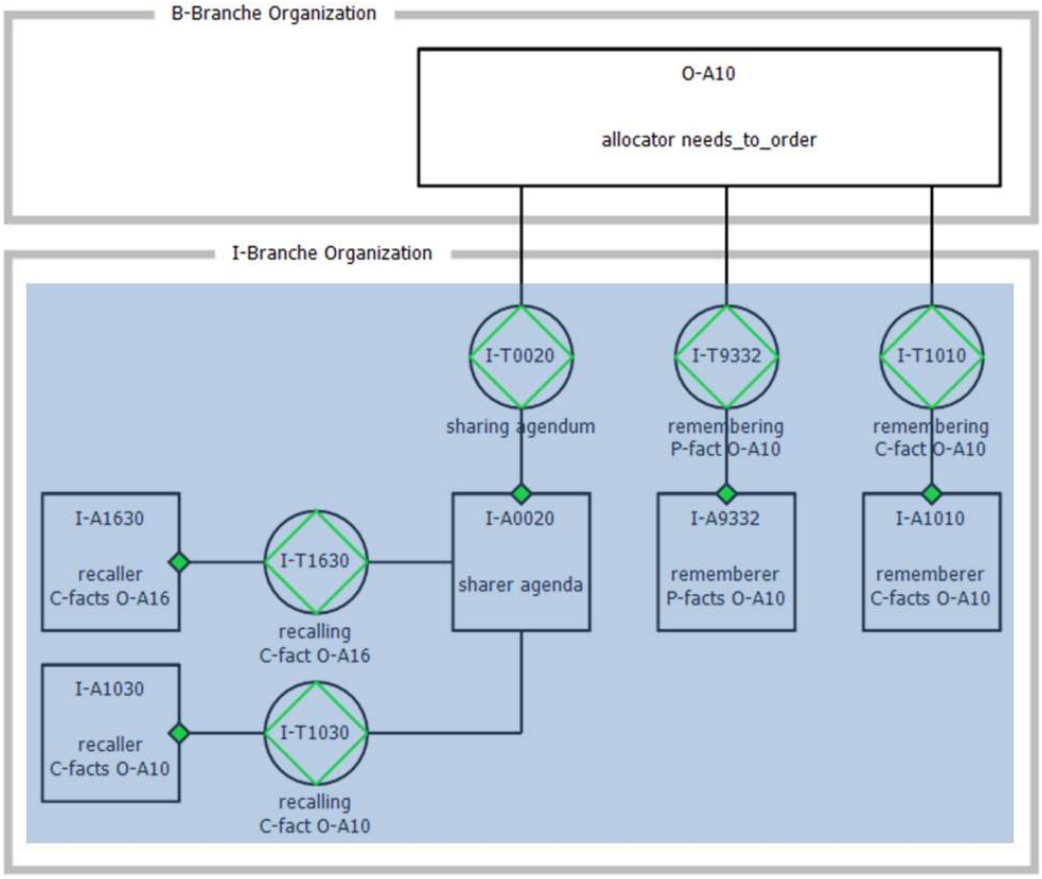


Fig. 8.9 Remembering and Sharing Facts for O-A10

We will continue the discussion of the implementation of O-A14 in the EIS on the basis of the rules from the PID-Framework related to the *Actor General Characteristics*.

5. O-A16 submits a request to O-A10 to allocate a purchasing need from the collection of purchasing needs to a particular purchase order.
6. Actor role belongs to a cluster together with O-A16 and O-A14
7. Not applicable.
8. The actors O-A16, O-A14, and O-A10 are in the same cluster. All their coordination activities are geared to each other and take place implicitly.
9. The new application is intended to support O-A10 in remembering facts and invoking agenda.

The aspects in the *Sharing Agenda* aspect group are discussed hereafter:

10. The agenda are linked to an object PURCHASE_ORDER_NEED.
11. Not applicable.
12. Not applicable.

The aspects in the *Sharing Facts* aspect group are discussed hereafter.

Coordination facts:

13. Not applicable.
14. Not applicable.

Production facts:

15. Not applicable.

General:

16. O-A10 does not know anything about the location of an agendum. Therefore, the generic I-actor 'sharer agenda' has to be implemented.
17. Not applicable.
18. Not applicable.

Lastly, the aspects in the *Remembering Facts* aspect group are discussed:

Coordination facts:

19. They are created implicitly because of the common cluster with O-A16.
20. Not applicable.
21. Not applicable.
22. Not applicable.

Production facts:

23. Not applicable.
24. Not applicable.
25. Not applicable.

8.4 Validation and Conclusions

The researcher has drawn up a draft sector model on the basis of interviews with IT staff and business management of the participating companies. The IT think tank, which was founded by the participating companies, has discussed this sector model in general terms during one of their meetings. The participants did not discuss IT issues but business issues, like actor roles, transactions and facts that actors need to know. The attendants found this discussion awkward. Some of them were lacking a required level of abstract thinking. For example, they had a lot of trouble to discuss actor roles and transaction kinds and preferred to not getting away from current organizational structures and organizational roles. Therefore, we scheduled another meeting. During a full day, three representatives of the IT think tank discussed the model in depth with the researcher. All of them were information ana-

lyst and all of them were heavily involved in maintaining business processes and aligning these processes with the IT configuration in their companies. They were assumed to be analytical and abstract thinkers. An animated discussion emerged in this group. The participants focused strongly on the actor roles and transaction types which are defined in the model. A good discussion leader appeared to be crucial. He has to play chess on two boards at the same time; firstly, on the board of the ontological sector model and secondly on the board of the implementation model. The discussion moved continually to the implementation model level. This is understandable because these are people from practice. However, the goal of the researcher was to validate the results of applying the PID-Framework on the ontological sector model. He continually had to translate the outcomes of the discussion regarding implementation scenarios to its meaning for the ontological sector model. In general, playing chess on two boards is a necessity. People only detect mistakes in the outcome of applying the PID-Framework through discussing implementation scenarios. Once again, the discussion has shown that the participants still were struggling to come off the existing organizational roles in their companies. In case the researcher exactly knew the actor roles in the sector model that covers a particular organizational role, a discussion about organizational roles were not a problem for him. He just had to connect specified information needs which were discussed on organization level, to the right actor role in the sector model. In this way, it appeared to him that during the meeting the results of applying the PID-Framework were basically validated. It turned out that for validating the actor roles in detail the relevant actors from the organization should be consulted. This was not possible for all actors. Just some of the people surveyed could say that the information needs were correctly defined or that they contained failures. They were, however, unable to imagine additional information needs. The presumption arose that during implementing an implementation scenario further information needs become apparent.

Applying the PID-Framework in designing an EIS gives the following benefits over the traditional way of working:

From the perspective of the using system (organization):

- The project has been started during a meeting of a group of seven entrepreneurs in the wholesale market of flower bulbs. This group was formed by Anthos, a Dutch interest group for the trade in flower bulbs. Their common purpose was to develop an EIS that supports their business processes. They were convinced that such a system should be developed for a market and not only for an individual enterprise. During that meeting, the researcher has explained his strategy: developing an EIS is not only an IT project; it is also a business project. One ought to start with the design of a construction model of the organization. The actor roles and transaction kinds are central elements in de organization, and the starting point for defining the information requirements. Requirements are not based on ‘nice-to-have’ but on the actual needs of business actors. For defining a common EIS, one has to abstract from implementation topics, because these topics could be different for the enterprises in the group.
  The attendees, who are business people and not IT experts, responded very positively. They were able to understand the approach, and were very motivated by the idea that IT does not support processes but human beings in business. For the researcher, this

meeting was also a revelation that people, who lack IT knowledge completely, can become very motivated for this approach.

- After producing the ontological models, the discussion with information managers in the organization becomes well structured. They were able to understand the models in a short time. However, discussions about information needs linked to actor roles are difficult for them. They tend to mix actor roles and organizational functions continually. Therefore, highly abstracted discussions should be avoided. An important benefit from this approach is that discussions need not bog down in discussions about implementation questions.
- If the required information services are defined at actor role level, then a discussion about priorities in supporting these services with IT is fruitful.
- Discussions with employees of these organizations about actor responsibilities are difficult. It is much easier for them to talk about tasks and activities than about responsibility and accountability.

From the perspective of the engineer:

- The use of the PID-Framework leads to a standardized approach in designing and engineering the EIS. The implementation of every actor role is elaborated stepwise after designing the integral ontological model of the (common) enterprise. Designing and engineering the EIS is clearly discerned. This improves the intellectual manageability of the development process.
- The ontological model of the organization is used as an implementation independent reference model. Each implementation scenario has to be founded on this reference model. That makes discussions with people from the using organization about implementation scenarios framed and well structured.
- Separation of concerns can be applied for interfacing with the external world. The wholesales companies purchase their bulbs from growers or intermediaries. For the exchange of standard messages, the EDIbulb protocol has been developed. The implementation of this protocol in organizations can be confined to the implementation of a few actor roles.

PART IV

Conclusions

9 Conclusions and further research

9.1 Answers to Research Questions

The first chapter of this dissertation presented the main research question that we have sought to answer. This main question is motivated by the desire for a methodical approach to develop an EIS or to select an existent EIS on the basis of the integral ontological model of the organization. Simply put, there is a need for a limited number of rules based on a specification framework in order to keep the implementation design of the EIS manageable.

A correct implementation model, in the sense used in this thesis, does not mean that the using organization will necessarily consider it to be an acceptable one. The word 'correct' is used in the sense of an implementable model that is correctly derived from a given ontological model of the organization. The acceptance of an implementable model by the organization is also dependent on the choices which are made for linking technology to the implementation components. This aspect has been kept outside the scope of this thesis.

Many aspects are associated with an implementation model. That is why the research is limited to that part of the implementation model that is often covered by an EIS. The development of an EIS should start with producing the integral ontological model of the organization: the ontological model of the B-organization, the I-organization and the D-organization. On the basis of this integral model the realization of the organization is elaborated.

This brings us to the first sub-question which is elaborated in chapter 3, namely:

What is understood by the realization of the essential model of an organization that constitutes the basis for the design of an implementable model of the information provision in the organization?

The ontological models of the B-organization, I-organization, and D-organization collectively provide a coherent implementation-independent model of an organization as a basis for developing an EIS. It clearly shows that essentially actors in the B-organization create new facts and use original or derived facts. These facts are shared, i.e. made available, by specific I-actors, namely 'sharer agenda', 'sharer C-facts' and 'sharer P-facts'. So, O-actors need not know about fact locations. If they need an agendum, a C-fact or a P-fact, they only have to request the 'sharer agenda', 'sharer C-facts' and 'sharer P-facts', respectively, for it. These 'sharers' get the facts they need from 'recallers' from 'derivers'. Facts are archived and provided by D-actors.

The second sub-question is elaborated in chapter 4, namely:

What does an implementation model of information provision based on the realization of the essential model of the organization consist of?

The implementation model of the information provision to be developed, is created from the integrated implementation-independent model of the B-organization, the I-organization, and the D-organization. Firstly, the scope of the model is determined. It could be that binary facts within the scope of the EIS have one domain inside its scope and one outside. Then we have to create a subset of the external object class within the EIS. Secondly, the EIS supports O-actor roles within the B-organization. The I- and D- actor roles are contained in the EIS, by assigning them to agents. In order to design an implementation model of an enterprise that is implemented by an EIS the PID-framework is presented. The specified aspects in this framework are identified on the basis of literature sources that have been consulted by the researcher and on the basis of nearly thirty years of practical experience with the development of complex EISs. Based on this experience and the literature review, in Chapter 5 the final sub-question is answered, namely:

What guidelines are needed for the design of an implementation model of the organization, on which the design of an enterprise information system could be based?

Based on the PID-Framework twenty five rules for practitioners are formulated. The applicability of these rules are validated through three practical applications, presented and discussed in Chapters 6, 7 and 8.

9.2 Benefits of using the PID Framework

Applying the PID-Framework in designing an EIS has the following benefits over the traditional way of working:

From the perspective of the using system (the organization):

- A project for developing an EIS is understood as a business project and not as an IT project. The application for the trade in flower bulb exhibits that
  business people react positively to this approach. They are very motivated by the insight that IT does not support processes but human beings in business. It sounds very logical to them to hear that all information services must be defined from the point of view of the actor roles in the B-organization and that information technology ought to link software components to model components.
- After having produced the ontological models of the organization the discussion with information managers in the organization become well structured. It concentrates on the information needs of the business organization instead of all kinds of implementation issues.
- Identifying actor roles is considered as a part of the design process of the construction model of the organizations. This avoids defined actor roles from the view ‘nice to have’ and an implemented EIS that is not operating effectively.

- The integral ontological model of the organization serves as a reference point for designing an implementation scenario for an EIS.
- Defining the required information services on actor role level is a good basis for having a transparent discussion about priorities in supporting these services with information technology.
- If the enterprise prefers to buy a standard EIS, then the selection process of the EIS can be done more thoroughly if the standard EIS is mapped on the information services that are identified for each actor role that must be supported by the EIS.

From the perspective of the engineer:

- The PID-Framework is based on a well-founded way of thinking. This enables to intellectually managing the complexity of the design of an EIS. O-actors create coordination facts and production facts and they request for coordination facts and production facts. This is the starting point for implementation design.
- The use of the PID-Framework leads to a standardized approach in designing and engineering the EIS. The implementation of every actor role is stepwise elaborated on the basis of the integral ontological model of the enterprise. Designing and engineering the EIS is clearly discerned. This improves the intellectual manageability of the development process.

9.3 Future Research

Several issues have fallen outside the scope of this study because each of them would deserve a separate research project. They are mentioned hereafter in random order.

- The conducted research is concentrated on what is often called the success layer in the organization. It is quite common in practice that either the initiator or an executor revokes a pending transaction [Dietz 2006; Perinforma 2012]. This topic has not been covered. Further research into the consequences for the PID-Framework of the implementation of revocation or cancellation patterns is desirable.

- The organization theorem (section 2.4) serves as one of the grounding theories for this research. Combined with the Generic System Development Process, this theorem enables the development of the I-organization as an object system supporting the B-organization as a using system (section 3.4). However, sometimes facts are requested by an external actor. For example, the government may request an enterprise to share a certain report periodically. Then, the enterprise provides this report at the request of an actor that belongs to the organization of the government. Another example, it occurs that an enterprise needs information from its supplier in order to achieve an optimal alignment in its supply chain. Then, an ontological actor in the enterprise needs to carry out an infological transaction with an actor at the supplier. These situations are currently not addressed in the PID-Framework. Further research would be required for this.

- Another point that is not discussed in this study and that would require further research concerns the sourcing of the I-organization and/or the D-organization wholly or partly. Although Op ‘t Land [Op 't Land 2008] has done research on sourcing at the level of the business organization, research on the sourcing of the I-organization and/or D-organization in the enterprise engineering domain is still an unexplored area.

- Although the implementation model (which is composed on the basis of the PID-Framework) is an effective model, it need not necessarily be the most desirable model for the organization. In this study the arguments for allocating a specific infological actor role to an agent or to a human subject are not discussed. There is still a wide field of study open on how to arrive at a desired model for an operational organization of an enterprise from an implementation-independent model.

- Another important topic for future research is the elaboration of the chosen path of development of the PID-Framework into a full-fledged design methodology for EIS’s. It was difficult to maintain a clear and comprehensive overview of the implementation model during the design of the EIS. The availability of proper tooling could be indispensable in this regard.

- The integral, implementation-independent model of the B-organization, I-organization, and D-organization provides insight into the construction of the organization. It is a white-box model in which concepts such as “responsibility” and “authorization” can be clearly defined. These concepts are often difficult to deal with when implementing enterprise information systems and they often end up being implemented differently from what was intended. In addition, this issue is also closely associated with legislation concerning governance. More research in this area is desirable.

PART V

References

References

Adam, F. and P. O'Doherty (2003). ERP projects: good or bad for SME's? Cambridge, Cambridge University Press.

Aken van, J. E. (2004). "Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-Tested and Grounded Technological Rules." Journal of Management Studies **41**(1): 219-246.

Aken van, J. E. (2005). "Management research as a design science: Articulating the research products of Mode 2 knowledge production in management." British Journal of Management(16): 19-36.

Albani, A. (2006). The Benefit of Enterprise Ontology in Identifying Business Components. 19th IFIP World Computer Congress, Santiago de Chile, Chile, Springer-Verlag.

Albani, A. and J. L. G. Dietz (2011). "Enterprise ontology based development of information systems." Int. J. Internet and Enterprise Management **7**(1): 41-63.

Alexander, C. (1964). Notes on the Synthesis of Form, Harvard University Press.

Apostel, L. (1960). "Towards the formal study of models in the non-formal sciences." Synthese **12**.

Austin, J. L. (1962). How to do things with words. Cambridge, MA, Harvard University Press.

Baan, J. (2005). The way to market leadership. Putten, Vanenburg Group B.V.

Barbier, F. and C. Atkinson (2003). Business Components, , Kluwer Academic Publishers Group.

Barjis, J. A. (2008). "The importance of business process modeling in software systems design." Science of Computer Programming **71**: 73–87.

Bennett, A. (1990). The Death of the Organization Man. New York, William Morrow.

Bernroider, E. W. N. (2008). "IT governance for enterprise resource planning supported by the DeLone–McLean model of information systems success." Information & Management **48**(5): 257-269.

Briggs, R. O., G. J. Vreede de, et al. (2003). "Colaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems." Journal of Management Information Systems **19**(4): 31-64.

Bryson, C. and W. Currie (1995). "IT strategy: formal rational orthodoxy or contingent adhocracy?" Omega **23**(6): 677-689.

Bunge, M. A. (1979). Treatise on Basic Philosophy, vol.4, A World of Systems. Dordrecht, The Netherlands, D. Reidel Publishing Company.

Carr, N. G. (2003). "IT doesn't matter." Harvard Business Review **81**(5).

Chan, Y. E. and B. H. Reich (2007). "IT alignment: an annotated bibliography." Journal of Information Technology **22**(4): 316.

Checkland, P. (1988). "Information systems en systems thinking: Time to unit?" International Journal of Information Management **8**.

Chen, P. P. (1976). "The Entity-Relationship Model - Toward a Unified View of Data." ACM Transactions on Database Systems **1**(1): 9-36.

Choo, C. W. (2002). Information management for the intelligent organization: The art of environmental scanning, 3rd Ed. Medford, NJ.

Choo, C. W. (2008). What is Information Management? University of Toronto, http://choo.fis.utoronto.ca/Imfaq/ **2012**.

Ciborra, C. U. (1997). "De profundis? Deconstructing the concept of strategic alignment." Scandinavian Journal of Information Systems **9**: 67-82.

Codd, E. F. (1969). Derivability, Redundancy and Consistency of relations stored stored in Large Data Banks. San Jose, California RJ599, IBM Research Report.

Codd, E. F. (1970). "A relational model of Data for Large Shared Data Banks." Communications of the ACM **13**(6).

Cronk, M. C. and E. P. Fitzgerald (1999). "Understand IS business value: derivation of dimensions." Logistics Information Management **12**(1-2): 40-49.

Davenport, T. (1993). Process Innovation. Boston, Harvard Business Review Press.

Davenport, T. (1993). Process innovation: Reengineering work through information technology. Boston, MA, Harvard Business School Press.

Davenport, T., M. Hammer, et al. (1989). "How executives can shape their company`s information systems." Harvard Business Review **67**(2): 130-134.

Davenport, T. H. (1998). "Putting the enterprise into the enterprise system." Harvard Business Review **76**(4): 121-131.

Deming, W. E. (1986). Out of the Crisis. Cambridge, MA USA, MIT Press.

Deming, W. E. (2000). The New Economics for Industry, Government, Education Cambridge, MA, USA, MIT Press.

Denyer, D., D. Tranfield, et al. (2008). "Developing Design Propositions through Research Synthesis." Organization Studies, SAGE Publications **29**(3): 393-413.

Detlor, B. (2004). Towards knowledge portals: From human issues to intelligent agents. Dordrecht, The Netherlands, Kluwer Academic Publishers.

Detlor, B. (2010). "Information Management." International Journal of Information Management **30**(2): 103-108.

Dietz, J. L. G. (1990). A Communication Oriented Approach to Conceptual Modelling of Information Systems. Lecture Notes in Computer Science. Berlin, Springer Verlag**:** 441-460.

Dietz, J. L. G. (2001). "DEMO: Towards a discipline of organisation engineering." European Journal of Operational Research **128**: 351-363.

Dietz, J. L. G. (2003). Deriving Use Cases from Business Process Models. Conceptual Modeling - ER 2003, Springer Verlag.

Dietz, J. L. G. (2004). Towards a LAP-based Information Paradigm. 9th International Working Conference on the Language Action Perspective on Communication Modelling, New Brunswick, New Jersey, USA.

Dietz, J. L. G. (2006). Enterprise Ontology – theory and methodology, Springer Verlag.

Dietz, J. L. G. (2008). Architecture, building Strategy into Design. The Hague, Academic Service.

Dietz, J. L. G. (2009). Is it φτψ or bullshit? - Farewell lecture. Faculty of Electrical Engineering, Mathematics and Computer Science. Delft, The Netherlands, Technical University of Delft.

Dietz, J. L. G. (2013) DEMO-3, Models and Representations, version 3.6c March 2013. http://www.ee-institute.com

Dietz, J. L. G., A. Albani, et al. (2010). "Enterprise Engineering: The concise Manifesto - version 9."

Dietz, J. L. G. and J. A. P. Hoogervorst (1213). "The discipline of enterprise engineering." Int. J. Organisational Design and Engineering **3**(1): 86-114.

Dietz, J. L. G. and J. A. P. Hoogervorst (2007). "Enterprise Ontology and Enterprise Architecture – how to let them evolve into effective complementary notions." GEAO Journal of Enterprise Architecture **1**.

Dietz, J. L. G. and J. A. P. Hoogervorst (2012). The Principles of Enterprise Engineering. Second Enterprise Engineering Working Conference, EEWC 2012, Delft, The Netherlands, Springer Berlin Heidelberg.

Dietz, J. L. G. and J. B. F. Mulder (1998). Transformation of organisations requires constructional knowledge of business systems. 31st Hawaii International Conference on System Sciences Los Alamitos CA, IEEE Computer Society Press.

Dietz, J. L. G. and G. A. M. Widdershoven (1991). Speech Acts or Communicative Action? . The Second European Conference on Computer Support Cooperative Work, Amsterdam.

Donovan, J. J. (1994). Business Reengineering with Information Technology. Englewood Cliffs, Prentice Hall.

Duplaga, E. and M. Astani (2003). "Implementing ERP in manufacturing." Information Systems Management **20**(3): 68-76.

Fayol, H. (1920). Administration industrielle et generale. Paris, Dunod.

Finlayson, J. G. (2005). Habermas, a Very Short Introduction. New York, Oxford University Press.

Frick, N. and P. Schubert (2009). Packaged ERP software: A Study of Future Requirements. University of Koblenz-Landau, Germany, Institute for IS Research.

Gluchowski, P., R. Gabriel, et al. (2008). Management Support Systeme und Business Intelligence, Springer Berlin Heidelberg.

Godwin, J. U. (1992). "Rethinking effectiveness measures of decision support systems." Information and Management **22**(2): 123-135.

Goedvolk, H. (2006). Case SGC. Master of Informatics, Module EEP 2.3, HU Utrecht.

Goldkuhl, G. and K. Lyytinen (1982). A language action view of information systems. 3rd International Conference on Information Systems, TIMS/SMIS/ACM.

Grabski, S. and S. Leech (2007). "Complementary controls and ERP implementation success." International Journal of Accounting Information Systems **8**(1): 17-39.

Guerreiro, S. L. P. D. (2012). Enterprise Dynamic Systems Control enforcement of run-time business transactions using DEMO: principles of design and implementation. Instituto Superior Tecnico, Universidade Tecnica de Lisboa. **PhD**.

Habermas, J. (1981). Theorie des Kommunikatives Handelns, Erster Band. Frankfurt am Main, Suhrkamp Verlag.

Halpin, T. A. (2001). Information Modeling and Relational Databases. San Francisco, Morgan Kaufmann.

Halpin, T. A., K. Evans, et al. (2003). Database Modeling with Microsoft Visio for Enterprise Architects. San Francisco, Morgan Kaufmann.

Hammer, M. (1990). "Reengineering Work: Don 't Automate, Obiliterate,." Harvard Business Review.

Harvey, E. (1968). "Technology and the Structure of Organizations." American Sociological review **33**: 241-259.

Henderson, J. C. and N. Venkatraman (1993). "Strategic Alignment: Leveraging Information Technology for Transforming Organizations." IBM Systems Journal **32**(1): 4-16.

Hevner, A. R. (2007). "A Three Cycle View of Design Science Research." Scandinavian Journal of Information Systems **19**(2): 87-92.

Hevner, A. R., S. T. March, et al. (2004). "Design Science in Information Systems Research." MIS Quarterly **28**(1): 75-105.

Hice, G. F., W. Turner, et al. (1970). System Development Methodology. Amsterdam, North-Holland Publ. Co.

Hirschheim, R., H. K. Klein, et al. (1995). Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations. Cambridge, Cambridge University Press.

Holland, C. and B. Light (1999). "A Critical Success Factors Model For ERP Implementation." IEEE Software **May/June**: 30-36.

Hong, K. and Y. Kim (2002). "The critical success factors for ERP implementation: an organizational fit perspective." Information & Management **40**(1): 25-40.

Hoogervorst, J. A. P. (2004). "Enterprise Engineering & -Architecture: een antwoord op falende strategie implementaties." Holland Management Review(12): 20-31.

Hoogervorst, J. A. P. and J. L. G. Dietz (2005). "Kernbegrippen omtrent Enterprise Architectuur en Architectureren." Tiem(10).

Hoogervorst, J. A. P. and J. L. G. Dietz (2008). "Enterprise Architecture in Enterprise Engineering." Enterprise Modelling and Information Systems Architecture **3**(1).

Huysmans, P. (2011). On the Feasibility of Normalized Enterprises: Applying Normalized Systems Theory to the High-Level Design of Enterprises. Department of Management Information Systems, University of Antwerp. **PhD**.

Huysmans, P., D. Bellens, et al. (2010). Aligning the Constructs of Enterprise Ontology and Normalized Systems. 6th International Workshop, CIAO!, St. Gallen, Springer-Verlag.

Iivari, J. (2007). "A Paradigmatic Analysis of Information Systems As a Design Science." Scandinavian Journal of Information Systems **19**(2): 39-64.

Jong de, J. (2009). Integration Aspects between the B/I/D-Organizations of the Enterprise. CIAO!/EOMAS 2009, Amsterdam, The Netherlands, Springer-Verlag Berlin Heidelberg.

Jong de, J. (2011). Designing the information organization from ontological perspective. 1ste Enterprise Engineering Working Conference, Antwerp, Belgium, Springer - LNBIP.

Jong de, J. and J. L. G. Dietz (2010). Understanding the Realization of Organizations. CIAO! 2010, St. Gallen, Switzerland, Springer-Verlag Berlin Heidelberg.

K., K. and J. Hillegersberg (2000). "Enterprise resource planning experiences and evolution." Communications of the ACM **43**(3): 22-26.

Kallunki, J.-P., E. K. Laitinen, et al. (2011). "Impact of enterprise resource planning systems on management control systems and firm performance." International Journal of Accounting Information Systems **12**(1): 20-39.

Karim, N. S. A. and R. Hussein (2008). "Manager's perception of information management and the role of information an knowledge managers: The Malaysian perspectives." International Journal of Information Management **28**(2): 114-127.

Keen, P. G. W. (1991). Shaping the Future: Business Design through Information Technology. Harvard, Harvard Business School Press.

Keller, E. L. (1999). "Lessons learned." Manufacturing Systems **17**(11): 44-50.

Kelly, D. (2006). "A study of design characteristics in evolving software using stability as a criterion." Software Engineering, IEEE Transactions on **32**(5): 315-329.

Kervel van, S. J. H. (2012). Ontology driven Enterprise Information Systems Engineering. Tilburg, Delft University of Technology. **PhD**.

Kervel van, S. J. H., J. L. G. Dietz, et al. (2012). Enterprise Ontology Driven Software Engineering. The 7the International Conference on Software Paradigm Trends.

Koedijk, A. and A. Verstelle (1999). ERP in Bedrijf. Woerden, KPMG Consulting.

Koning de, F. (2004). "ERP implementaties: managementprobleem of softwareprobleem?" Maandblad voor Accountancy en Bedrijfseconomie (10): 435-443.

Koningsveld, H. and J. Mertens (1986). Communicatief en strategisch handelen: inleiding tot de handelingstheorie van Habermas. Muiderberg, Coutinho.

Krouwel, M. R. and M. Op 't Land (2011). Combining DEMO and Normalized Systems for Developing Agile Enterprise Information Systems. First Enterprise Engineering Working Conference EEWC, Antwerp, Belgium, Springer-Verlag.

Langefors, B. (1977). "Information System Theory." Information Systems **2**(4): 207-219.

Lehman, M. (1980). "On understanding laws, evolution, and conservation in the large program life cycle." Journal of Systems and Software **1**: 213-221.

Lehman, M. M. and J. F. Ramil (2001). "Rules and tools for software evolution planning and management." Ann.l Software Engineering **11**(1): 15-44.

Liu, K. (2000). Semiotics in Information Systems Engineering. Cambridge, Cambridge University Press.

Lundeberg, M., G. Goldkuhl, et al. (1981). Information Systems Development: A Systematic Approach. Englewood Cliffs, Prentice Hall.

Maij, E., P. J. Toussant, et al. (2002). "Use cases and DEMO: aligning functional features of ICT-infrastructure to business processes." International Journal of Medical Informatics **65**(2002): 179-191.

Mallens, P. J. M., J. L. G. Dietz, et al. (2001). The Value of Business Process Modeling with DEMO prior to Information Systems Modeling with UML. EMMSAD'01: proceedings , 6th CAISE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design, Interlaken.

Mannaert, H. and J. Verelst (2009). Normalized Systems - Re-creating Information Technology Based on Laws for Software Evolvability. Kermt, Belgium, Koppa.

March, S. T. and G. F. Smith (1995). "Design and natural science research on information technology." Decision Support Systems **15**: 251-266.

Martin, J. and C. Finkelstein (1981). Information Engineering. Lancaster, Savant Institute.

McGee, J. V. and L. Prusak (1993). Managing information strategically. Toronto, Canada, John Wiley & Sons.

Melville, N., K. Kraemer, et al. (2004). "Information technology and organizational performance: an integrative model of IT business value." MIS Quarterly **28**(2): 283-322.

Merriam-Webster (2012). Merriam-Webster Dictionary, http://www.merriam-webster.com/dictionary/rule.

Mohr, L. B. (1971). "Organizational Technology and Organization Structure." Administrative Science Quarterly **16**: 444-459.

Morris, C. W. (1938). Foundation of the Theory of Signs.

Mulder, J. B. F. (2006). Rapid Enterprise Design. Delft, Technical University Delft. **PhD**.

Mulder, J. B. F. and J. L. G. Dietz (2002). Business Architecture based on the integration of Communication, Actors and Production. 6th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, US.

Mulder, J. B. F. and V. E. Van Reijswoud (1997). "The Language Action Alternative for Information Systems Development, A Case Study."

Mumford, E. (1985). "Defining systems requirements to meet business needs: A casestudy example." The Computer Journal **28**(2).

Nijssen, G. M. and T. A. Halpin (1989). Conceptual Schema and Relational Database Design: A Fact Oriented Approach. Sydney, Prentice Hall.

Op 't Land, M. (2008). Applying Architecture and Ontology to the Splitting and Allying of Enterprises. Delft, Delft University of Technology. **PhD**.

Op 't Land, M. and H. A. Proper (2007). Impact of principles on Enterprise Engineering. ECIS 2007 - The 15th European Conference on Information Systems, St. Gallen.

Orlicky, J. A. (1975). Material Requirements Planning: The New Way of Life in Production and Inventory Management. New York, McGraw-Hill

Orlicky, J. A., G. W. Plossl, et al. (1972). "Structuring the bill of material for MRP." Production and Inventory Management **4th Qtr**: 19-42.

Perinforma, A. P. C. (2012). The Essence of Organization, An Introduction to Enterprise Engineering. Delft, Sapio B.V.

Porter, M. E. (1985). Competitive Advantage, creating and sustaining superior performance The Free Press.

Porter, M. E. and V. E. Millar (1985). "How information gives you competitive advantages." Harvard Business Review.

Purgathofer, P. (2006). "Is informatics a design discipline." Poiesis & Praxis **4**(4): 303-314.

Reijswoud van, V. E. (1996). The Structure of Business Communication: Theory, Model and Application. Technical Mathematics and Informatics. Delft, Delft University of Technology. **PhD**.

Reijswoud van, V. E., J. B. F. Mulder, et al. (1999). "Communicative Action Based Business Process and Infomation Systems Modelling with DEMO." Information Systems Journal **9**: 117-138.

Reijswoud van, V. E., J. B. F. Mulder, et al. (1999). "Speech Act Based Business Process and Information Modeling with DEMO." Information Systems Journal.

Robertson, J. (2005). Ten principles of effective information management. http://www.steptwo.com.au/papers/kmc effectiveim/index.html.

Sauer, C. and J. Burn (1997). The Pathology of Strategic Alignment. San Francisco, Jossey-Bass Publisher.

Scheer, A.-W. and F. Habermann (2000). "Making ERP a Success." Communications of the ACM **43**(4): 57-61.

Scott Morton, M. S. (1991). The Corporation of the 1990's: Information Technology and Organizational Transformation. New York, Oxford University Press.

Searle, J. R. (1969). Speech Acts, an Essay in the Philosophy of Language. Cambridge MA, Cambridge University Press.

Searle, J. R. (1995). The Construction of Social Reality. New York, The Free Press.

Seligmann, P. S., G. M. Wijers, et al. (1989). Analyzing the structure of IS methodologies, an alternative approach. The First Dutch Conference on Information Systems, Amersfoort, the Netherlands.

Shang, S. and S. P. (2002). "Assessing and managing the benefits of enterprise systems: the business manager's perspective." Information Systems Journal **12**(4): 271-299.

Shishkov, B. and J. L. G. Dietz (2003). Deriving use cases from business processes: the advantages of DEMO. The Fifth International Conference on Enterprise Information Systems, Angers, France.

Simon, H. A. (1996). The Sciences of the Artificial. Cambridge MA, the MIT Press.

Smith, H. and P. Fingar (2003). Business Process Management (BPM): The Third Wave. Tampa, Florida, USA, Meghan-Kiffer Press.

Smith, H. and P. Fingar (2003). IT doesn't Matter, Business Processes Do. Tampa, Florida, USA, Meghan-Kiffer Press.

Sneller, A. C. W. (2010). Does ERP Add Company Value? A Study for the Netherlands and the United Kingdom. Alblasserdam, Nyenrode Business University. **PhD**.

Stamper, R. K. (1973). Information in Business and Administrative Systems. New York, Wiley.

Stamper, R. K. (1996). Signs, Information, Norms and Systems. Signs of Work: Semiotics and Information Processing in Organisations. P. Holmqvist, P. B. Anderson, H. Klein and R. Posner, Walter de Gruyter.

Stamper, R. K., K. Liu, et al. (2000). "Understanding the Roles of Signs and Norms in Organisations." Journal of Behaviour and Information Technology **19**(1): 15-27.

Standish (2004). Third Quarter Research Report. West Yarmouth, MA, The Standish Group International, Inc.

Strassmann, P. (1997). The Squandered Computer. Evaluating the Business Alignment of Information Technologies. New Canaan, Connecticut, The Information Economics Press.

Sumner, M. (2000). "Risk factors in enterprise-wide ERP projects." Journal of Information Technology **15**: 317-327.

Tallon, P., K. Kraemer, et al. (2000). "Executives' Perceptions of the Business Value of Information Technology: A Process-Oriented Approach." Journal of Management Information Systems **16**(4): 145-173.

Tapscott, D. and A. Caston (1993). Paradigm Shift - The New Promise of Information Technology. New York, New York, USA, McGraw-Hill.

Taylor, F. W. (1903). Shop Management. New York, Harper & Brothers Publishers.

Taylor, F. W. (1911). The Principles of Scientific Management. New York, Harper & Row.

Te'eni, D. (2001). "A cognitive-affective model of organizational communication for designing IT." MIS Quarterly **25**(2): 251–312.

Terlouw, L. (2011). Modularization and Specification of Service-Oriented Systems. Delft, Technical University Delft. **PhD**.

Terlouw, L. and K. E. Maarse (2009). A Service Specification Framework for Developing Component-Based Software: A Case Study at the Port of Rotterdam. 5th International Workshop, CIAO! 2009, Amsterdam, Springer-Verlag.

Tribolet, J. (2009). Dietz Engineered a Marvelous Ontology! Technical University Delft, The Netherlands.

Tribolet, J. (2012). The emergence of Enterprise Engineering. EFCA Conference. Lisbon, Portugal.

Wikipedia Droste effect, http://en.wikipedia.org/wiki/Droste-effect.

Wilson, T. D. (2003). Information management. International encyclopedia of information and library science. London, UK, Routledge**:** 263–278.

Winograd, T. and F. Flores (1986). Understanding Computers and Cognition: A New Foundation for Design, Ablex Corporation, Norwood, NJ.

Winter, R. (2008). "Design science research in Europe." European Journal of Information Systems(17): 470-475.

Woodward, J. (1958). Management and Technology. London, Her Majesty's Stationary Office.

Woodward, J. (1965). Industrial Organization: Theory and Practice. London, Oxfort University Press.

xAF (2004). Extensible Architecture Framework, version 1.1 (formal edition). J. L. G. Dietz, NAF.

Yin, R. K. (2003). Casestudy research: design and methods, Sage Publications.

Yourdon, E. and L. L. Constantine (1979). Structured Design. Englewood Cliffs, New Yersey, Prentice Hall.

Samenvatting (Dutch)

Probleemstelling

Een bedrijf heeft met de aanschaf van een Enterprise Information System, afgekort als EIS, een keuze gemaakt voor een operationele werkwijze, zonder dat men veelal van de werking van het pakket goed op de hoogte is en de leverancier van het pakket in detail veelal weet hoe binnen het bedrijf wordt gewerkt. Beide partijen realiseren zich dat wel, maar er wordt een algemeen aanvaard standpunt ingenomen dat het niet anders kan en dat mogelijke afstemmingsproblemen tussen het bedrijf en het pakket tijdens het implementatietraject moeten worden opgelost. Fricties die ontstaan bij de implementatie van het EIS worden opgelost door het op maat maken van de standaard software of door aanpassingen aan te brengen in de werkwijze van de organisatie. Uiteraard moeten de wijzigingen begrensd zijn want de leverancier moet voor wat betreft kosten en tijd binnen een specifieke bandbreedte opereren.

De beschreven aanpak is vanaf begin van de jaren ’80 van de vorige eeuw gemeengoed geworden. Het is een aanpak die getypeerd kan worden als een benadering vanuit ‘best practices’. Mede door de toegenomen configuratie mogelijkheden van de standaard software applicaties is deze aanpak breed geaccepteerd. De onderzoeker is vanaf begin jaren ’80 van de voorgaande eeuw in management posities bij toonaangevende leveranciers van standaard ERP systemen hierin intensief betrokken. Allereerst in de ontwikkeling van standaard software producten en later in de verzorging van IT diensten aan bedrijven die omvangrijke EISs hebben geïmplementeerd. Daarbij kwam de vraag regelmatig boven, die ligt ook aan de basis van dit onderzoek, of bedrijven wel de software aanschaffen die ze in werkelijkheid nodig hebben. En dan gaat het niet uitsluitend over ‘het nodig hebben in het heden’ maar ook over ‘het nodig hebben in de nabije toekomst’. Die vraag werd ook gevoed door contacten bij een groot aantal bedrijven uit veel verschillende branches. Vanuit die contacten bleek dat (1) bedrijven de mogelijkheden van een aangeschaft EIS maar voor een deel benutten, (2) bedrijven zich soms de vraag stellen of hun organisatie niet weg groeit van het gebruikte EIS, en (3) bedrijven het EIS soms zien als een drempel voor het doorvoeren van veranderingen.

Hoe moet de relatie tussen de organisatie en een EIS worden gezien? We gaan daar in deze samenvatting niet uitgebreid op in, maar het is duidelijk dat een EIS de medewerkers van de onderneming ondersteunt in de uitvoering van hun werkzaamheden. Een EIS is in de organisatie vaak zo manifest aanwezig dat de wijze waarop de processen geoperationaliseerd worden voor een belangrijk deel door de constructie van het EIS wordt bepaald. Een vorkheftruck, bijvoorbeeld, moet worden beschouwd als een stuk gereedschap in handen van een medewerker van de organisatie voor het uitvoeren van een specifieke productiehandeling. Een EIS echter is van een heel andere orde. Dat moet beschouwd worden als

een stuk gereedschap voor het operationaliseren van een deel van de bedrijfsorganisatie. Het bepaalt hoe medewerkers met elkaar werken. Het vervult de informatiebehoefte van medewerkers en het 'onthoudt' geproduceerde feiten om ze later weer als informatie beschikbaar te kunnen stellen. Het belang van een passend EIS voor de organisatie moet niet worden onderschat. Immers de prestatie van de onderneming wordt bepaald door de constructie van de onderneming!

Dit realiserende is er alle reden om ons vragen te stellen die aan de basis van dit onderzoek liggen, namelijk: hoe ontwikkelen we een EIS dat bij de organisatie past en hoe bepalen we of een EIS datgene bevat wat de organisatie nodig heeft. Deze vragen veronderstellen enerzijds dat we kunnen bepalen wat een organisatie nodig heeft en anderzijds dat we de uitkomst daarvan op zodanige wijze tegen een bestaand EIS kunnen aanhouden zodat daar zinvolle conclusies uit getrokken kunnen worden. Dat blijkt inderdaad te kunnen.

Onderzoeksaanpak

Dit onderzoek heeft plaatsgevonden door gebruik te maken van de Design Science Research methodologie. In de terminologie van deze methodologie wordt voor een klasse van problematische contexten is een interventie type ontwikkeld waarbij een aantal generatieve mechanismen te hulp worden geroepen om een gewenste uitkomst te realiseren. De generatieve mechanismen verklaren waarom de interventie tot de gewenste uitkomst moet leiden. Onder de klasse van problematische contexten moet de inbedding van een standaard EIS's binnen de constructie van organisaties worden verstaan. Die context is problematisch omdat vanuit de praktijk blijkt dat de inbedding van het EIS door de organisatie vaak als niet optimaal wordt ervaren met als gevolg dat de organisatie niet optimaal functioneert. Het interventietype dat hiervoor is ontwikkeld bestaat uit een verzameling van richtlijnen die gebaseerd zijn op het Procedure for Implementation Design – Framework, afgekort als het PID-Framework. Met de inzet van een interventie van dit type wordt een EIS op de juiste wijze ingebed in een organisatie. D.w.z. de organisatie krijgt daarmee een EIS dat ze nodig heeft. Onder generatieve mechanismen worden theorieën verstaan waarop het PID- Framework is gefundeerd. In het tweede hoofdstuk van deze dissertatie zijn deze theorieën beschreven. We hebben ze aangeduid als de ψτφ – theorieën. Met deze theorieën wordt verklaard waarom het PID-Framework tot het gewenste resultaat leidt. De ψ-theorie beschouwt de organisatie als een sociaal systeem waarbinnen actoren vanuit drie verschillende gedaanten coördinatiehandelingen en productiehandelingen verrichten. De τ-theorie beschrijft het generieke proces om vanuit een gebruikend system een object system af te leiden. Dat object systeem wordt in verschillende stappen afgeleid. Allereerst wordt het function model van het object systeem gedefinieerd. Dit function model dient als basis voor de ontwikkeling van een implementatie onafhankelijk constructie model, ook wel een ontologie model genoemd. Vanuit het ontologiemodel wordt vervolgens een implementatiemodel geworpen. De τ-theorie laat zien dat een software applicatie niet als een zelfstandig object systeem, dat een gebruikend systeem ondersteunt, moet worden beschouwd, maar dat het de technologie omvat die toegekend moet worden aan componenten van het implementatiemodel. Daardoor ontstaat er een operationele organisatie. Het is van belang in dit verband te noemen dat vanuit een ontologiemodel meerdere implementatiemodellen kunnen worden afgeleid. Daar gaan we in het kader van deze dissertatie niet op in. Het PID-Framework beperkt zich tot het aangeven van richtlijnen waaraan een implementatie-

model moet voldoen. De φ-theorie, waarop de ψ-theorie zich ook op beroept, werpt een helder licht op het begrip ‘feit’ in de infologische en datalogische organisatie van een onderneming,

Er zijn drie applicaties uitgewerkt om enerzijds de werking van het Framework te illustreren, en anderzijds ook zijn gebruikt om het Framework aan te scherpen. De eerste applicatie betreft een situatie die al uitgebreid in de bestaande literatuur is beschreven. Het betreft de casus van de Stichting Geschillen Commissie. De tweede applicatie richt zich op het beoordelen van een EIS dat al in een organisatie voor commerciële trainingen is geïmplementeerd en dat al enige jaren in gebruik is. De derde applicatie richt zich op de ontwikkeling van een EIS voor een typische Nederlandse branche, namelijk die van het exporteren van bloembollen.

Onderzoeksresultaten

In het huidige Enterprise engineering onderzoek wordt een sterke nadruk gelegd op het beschrijven van de essentie van de onderneming in de vorm van een ontologiemodel van de bedrijfsorganisatie. Hoewel binnen dit ontologiemodel zowel de originele feiten als de afgeleide feiten worden gespecificeerd wordt daarin geen aandacht geschonken aan de wijze waarop feiten worden afgeleid en worden gedeeld tussen de actoren. Daarvoor moet het essentiële model worden uitgebreid met een ontologisch model voor de infologische en datalogische organisatie Volgens de gehanteerde terminologie binnen het Enterprise engineering domein spreekt men dan van realisering van het essentiële model. Er zijn verschillende redenen te noemen waarom het van belang is voor het ontwerp van een EES om naast het ontologische model van de business organisatie ook de ontologische modellen van de Infologische en Datalogische organisatie te ontwerpen.

In de eerste plaats moet het ontologiemodel van de infologische organisatie beschouwd worden als een implementatie onafhankelijk model voor het onthouden, het afleiden en het beschikbaar stellen van coördinatiefeiten en productiefeiten. Daarbij worden de inhoud en de vorm van een feit van elkaar onderscheiden. Dit onderscheid is van belang omdat de inhoud van een feit op verschillende wijzen vorm kan worden gegeven. Het ontologiemodel van de datalogische organisatie moet beschouwd worden als een implementatie onafhankelijk model voor het archiveren, transformeren en ophalen van documenten.

In de tweede plaats wordt naast de business organisatie, de infologische organisatie en de datalogische organisatie ook de fysieke organisatie onderscheiden. Hoewel deze organisatie in dit onderzoek niet wordt uitgewerkt is het wel van belang inzicht te hebben in de productiehandelingen die in de fysieke organisatie t.b.v. de datalogische organisatie plaatsvinden. Het ontologische model van de fysieke organisatie levert een implementatie onafhankelijke model voor het opslaan, kopiëren, vernietigen, verplaatsen en opvragen van op media vastgelegde documenten. In de fysieke organisatie worden documenten uit de D-organisatie beschouwd als ‘dingen’, ook wel aangeduid met files.

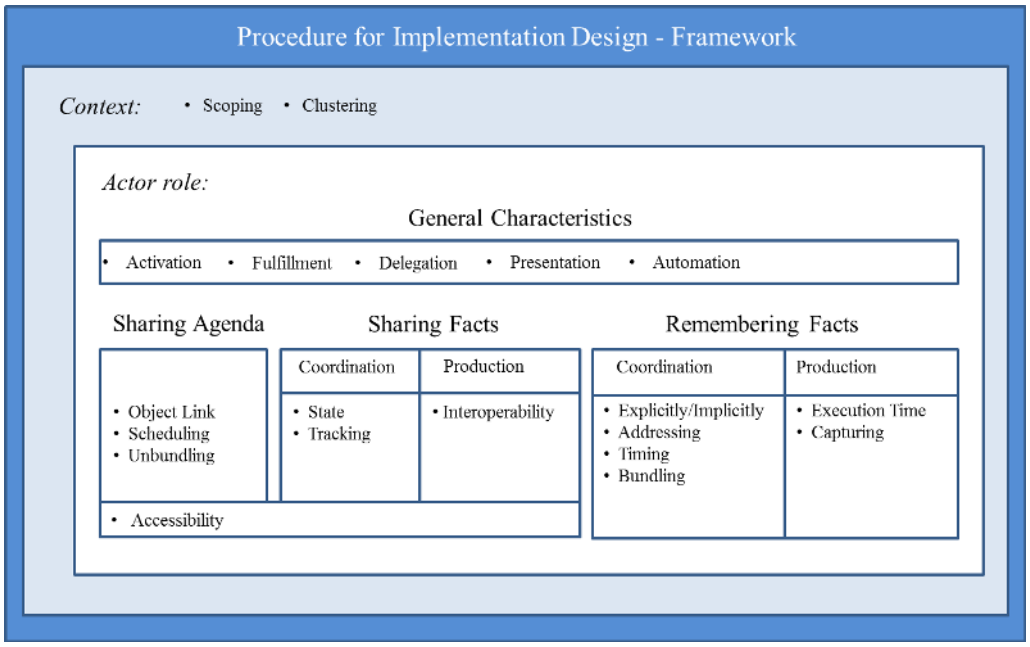
In de derde plaats wordt de onderneming beschouwd als een sociaal systeem. In het precomputer tijdperk was de afhandeling van feiten een gezamenlijke menselijke activiteit. Door de ontwikkeling van de informatie techniek zijn we het zicht daarop kwijtgeraakt. Zo wordt een EIS beschouwd als een gegevensverwerkend systeem waarvan in veel gevallen volslagen onduidelijk is welke actor de verantwoordelijkheid draagt voor de kwaliteit van

de brongegevens. Door het EIS te ontwikkelen vanuit een samenhangend ontologisch model van de informatievoorziening krijgen we dat inzicht weer terug.

In de vierde plaats zouden we nog willen noemen dat het ontologische model van de infologische en datalogische organisatie de actorrollen bevat die door actoren moeten worden vervuld. Voor de specificatie van het EIS is het nodig om voor elk van de actorrollen de afweging te maken in hoeverre ze ingevuld moeten of kunnen worden met informatie technologie.

Afsluitend, de beschikbaarheid van een integraal samenhangend ontologisch model is een belangrijke voorwaarde voor het ontwikkelen van een EIS. Dit ontologische model moet worden beschouwd als het startpunt voor het ontwerp van het te ontwikkelen of te selecteren EIS. In dit onderzoek wordt niet ingegaan op de criteria waarin de verschillende scenario's zich van elkaar kunnen onderscheiden. De vraag die in dit onderzoek centraal staat betreft de vraag naar de set van richtlijnen die tijdens de ontwikkeling van het EIS moet worden gevolgd wil het EIS gefundeerd zijn op het onderliggende ontologische model. De mate waarin de actorrollen zijn geautomatiseerd en de technologie die daarbij wordt gebruikt is daaraan secundair.

Het PID-Framework bevat een aantal aspectgroepen. Elke aspectgroep omvat meerdere aspecten waaraan vervolgens één of meerdere richtlijnen zijn verbonden. Deze richtlijnen moeten als concrete werkinstructies worden beschouwd in het ontwerpproces van het EIS. Deze instructies zijn zodanig opgesteld dat ze niet alleen kunnen worden gebruikt voor het specificeren van een nieuw EIS vanuit het ontologiemodel van de organisatie, maar ook kunnen worden gebruikt voor het beoordelen van de toepasbaarheid van een EIS voor een onderneming.



In het derde deel van dit onderzoek worden drie cases uitgewerkt. In deze cases wordt aan de hand van het PID-Framework voor een aantal actor rollen geïllustreerd met welke aspecten bij het ontwerp van een implementatie scenario rekening moet worden gehouden. Deze aspecten uit het PID-Framework zijn voor een belangrijk deel afgeleid uit de weten-

schappelijke literatuur en voor een ander deel vastgesteld uit enkele tientallen jaren van praktische ervaring van de onderzoeker met het ontwikkelen en implementeren van EIS's. De validatie van de aspecten uit het PID-Framework heeft in elke case plaatsgevonden door een intensieve discussie met direct betrokkenen. Bij de eerste en derde case waren dat zowel de engineers als de gebruikers van het te ontwerpen EIS. Bij de tweede case was er sprake van een geïmplementeerd EIS. De validatie vond hier plaats met de directe gebruikers van dit EIS. Deze gebruikers werden in het kader van de case studie geconfronteerd met een aantal aspecten van het implementatieontwerp die niet binnen het EIS waren geïmplementeerd. Men ontdekte dat, naast het feit dat actorrollen niet expliciet waren benoemd, actorrollen niet of maar voor een klein deel door het EIS werden ondersteund en dat een groot deel van de coördinatie tussen actorrollen was gebaseerd op improvisatie. Wat dit laatste betreft, in de praktijk leverde dat relatief weinig problemen op. Dat was vooral te danken aan het feit dat een beperkt aantal mensen al lange tijd met elkaar samenwerkten en van elkaar wist wat men van elkaar kon verwachten.

Curriculum Vitae

Joop de Jong is geboren op 20 juli 1956 in Ridderkerk. Nadat hij een docentenopleiding voor leraar wiskunde en economie had afgerond studeerde hij vervolgens theoretische informatica aan Technische Universiteit Delft. In 1984 studeerde hij af als wiskundig ingenieur. In dat jaar verwisselde hij ook zijn baan als leraar op een middelbare school voor een baan als ontwerper van ERP software bij het toenmalige IT bedrijf Baan Info Systems. Later bekent onder de naam Baan Company. In dit bedrijf dat tien later als ontwikkelaar en leverancier van standaard ERP softwarepakketten tot de top 3 van de wereld zou gaan behoren maakte Joop snel carrière. Al na enkele jaren kreeg hij de leiding van de afdeling R&D die verantwoordelijk was voor de ontwikkeling van vierde generatie software ontwikkeltools en nieuwe generaties ERP software. Als vice president Product Development verliet Joop begin 1991 het bedrijf. Hij richtte vervolgens het bedrijf Profuse op met het doel om gebruikers van BAAN ERP software te ondersteunen in het gebruik daarvan. Na een succesvolle groei met vestigingen in Duitsland, België en Roemenië met in totaal meer dan 250 medewerkers verkocht Joop in 2007 dit bedrijf. In 2003 had hij een par-time lectoraat ‘extended enterprise studies’ aanvaard aan de Hogeschool Utrecht. In samenwerking met een collega heeft Joop daar de geaccrediteerde masteropleiding ‘Master of Informatics’ opgezet. In 2010 heeft Joop zijn dienstverband aan de HU beëindigd en is sindsdien weer fulltime actief in het bedrijfsleven met Mprise. Mprise is een afsplitsing van Profuse en ondersteunt gebruikers en leveranciers van Microsoft Dynamics ERP in het gebruik van deze software door trainingen, adviezen, consultancy, business solutions en softwareontwikkeling.

Joop de Jong was born on July 20, 1956 in Ridderkerk, the Netherlands. After completing teacher training to become a mathematics and economics teacher, he studied theoretical computer science at Delft University of Technology. In 1984, he graduated with a MSc. in mathematics. In that year, he traded in his job as a teacher at a high school for a job as a designer of ERP software in the IT company Baan Info Systems, later known as Baan Company. Joop’s career progressed rapidly in this company, which ten years later would become one of the top three vendors of standard ERP software in the world. He was soon put in charge of the R&D department. This department was responsible for the development of the fourth generation software development tools and new generations of ERP software. Joop was vice president of Product Development when he left Baan in early 1991 to start the company Profuse. Its aim was to provide support to BAAN ERP software users. After successful growth with offices in Germany, Belgium and Romania and with over 250 employees, this company was sold by Joop in 2007. In 2003, he had accepted the part-time job as professor in “Extended Enterprise Studies” at the University of Applied Sciences Utrecht. In collaboration with a colleague, Joop set up the accredited Master’s degree programme: “Master of Informatics”. Joop left the university in 2010 and had since fully rejoined the business world with Mprise. Mprise is a company that split off from Profuse in

2005, and that supports users and vendors of Microsoft Dynamics ERP software in their use of that software by training, consultancy, business solutions and software development.

SIKS Dissertatiereeks

1998

1998-1 Johan van den Akker (CWI), DEGAS - An Active, Temporal Database of Autonomous Objects
1998-2 Floris Wiesman (UM), Information Retrieval by Graphically Browsing Meta-Information
1998-3 Ans Steuten (TUD), A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective
1998-4 Dennis Breuker (UM), Memory versus Search in Games
1998-5 E.W.Oskamp (RUL), Computerondersteuning bij Straftoemeting

1999

1999-1 Mark Sloof (VU), Physiology of Quality Change Modelling; Automated modeling of Quality Change of Agricultural Products
1999-2 Rob Potharst (EUR), Classification using decision trees and neural nets
1999-3 Don Beal (UM),The Nature of Minimax Search
1999-4 Jacques Penders (UM),The practical Art of Moving Physical Objects
1999-5 Aldo de Moor (KUB), Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems
1999-6 Niek J.E. Wijngaards (VU), Re-design of compositional systems
1999-7 David Spelt (UT), Verification support for object database design
1999-8 Jacques H.J. Lenting (UM), Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation.

2000

2000-1 Frank Niessink (VU), Perspectives on Improving Software Maintenance
2000-2 Koen Holtman (TUE), Prototyping of CMS Storage Management
2000-3 Carolien M.T. Metselaar (UVA), Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief.
2000-4 Geert de Haan (VU), ETAG, A Formal Model of Competence Knowledge for User Interface Design
2000-5 Ruud van der Pol (UM), Knowledge-based Query Formulation in Information Retrieval.
2000-6 Rogier van Eijk (UU), Programming Languages for Agent Communication
2000-7 Niels Peek (UU), Decision-theoretic Planning of Clinical Patient Management
2000-8 Veerle Coup, (EUR), Sensitivity Analyis of Decision-Theoretic Networks
2000-9 Florian Waas (CWI), Principles of Probabilistic Query Optimization
2000-10 Niels Nes (CWI), Image Database Management System Design Considerations, Algorithms and Architecture
2000-11 Jonas Karlsson (CWI), Scalable Distributed Data Structures for Database Management

2001

2001-1 Silja Renooij (UU), Qualitative Approaches to Quantifying Probabilistic Networks
2001-2 Koen Hindriks (UU), Agent Programming Languages: Programming with Mental Models
2001-3 Maarten van Someren (UvA), Learning as problem solving
2001-4 Evgueni Smirnov (UM), Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets
2001-5 Jacco van Ossenbruggen (VU), Processing Structured Hypermedia: A Matter of Style
2001-6 Martijn van Welie (VU), Task-based User Interface Design
2001-7 Bastiaan Schonhage (VU), Diva: Architectural Perspectives on Information Visualization
2001-8 Pascal van Eck (VU), A Compositional Semantic Structure for Multi-Agent Systems Dynamics.
2001-9 Pieter Jan 't Hoen (RUL), Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes
2001-10 Maarten Sierhuis (UvA), Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design
2001-11 Tom M. van Engers (VUA), Knowledge Management: The Role of Mental Models in Business Systems Design

2002

2002-01 Nico Lassing (VU), Architecture-Level Modifiability Analysis
2002-02 Roelof van Zwol (UT), Modelling and searching web-based document collections
2002-03 Henk Ernst Blok (UT), Database Optimization Aspects for Information Retrieval
2002-04 Juan Roberto Castelo Valdueza (UU), The Discrete Acyclic Digraph Markov Model in Data Mining
2002-05 Radu Serban (VU), The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents

2002-06 Laurens Mommers (UL), Applied legal epistemology; Building a knowledge-based ontology of the legal domain
2002-07 Peter Boncz (CWI), Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications
2002-08 Jaap Gordijn (VU), Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas
2002-09 Willem-Jan van den Heuvel(KUB), Integrating Modern Business Applications with Objectified Legacy Systems
2002-10 Brian Sheppard (UM), Towards Perfect Play of Scrabble
2002-11 Wouter C.A. Wijngaards (VU), Agent Based Modelling of Dynamics: Biological and Organisational Applications
2002-12 Albrecht Schmidt (UVA), Processing XML in Database Systems
2002-13 Hongjing Wu (TUE), A Reference Architecture for Adaptive Hypermedia Applications
2002-14 Wieke de Vries (UU), Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems
2002-15 Rik Eshuis (UT), Semantics and Verification of UML Activity Diagrams for Workflow Modelling
2002-16 Pieter van Langen (VU), the Anatomy of Design: Foundations, Models and Applications
2002-17 Stefan Manegold (UVA), Understanding, Modeling, and Improving Main-Memory Database Performance

2003
2003-01 Heiner Stuckenschmidt (VU), Ontology-Based Information Sharing in Weakly Structured Environments
2003-02 Jan Broersen (VU), Modal Action Logics for Reasoning About Reactive Systems
2003-03 Martijn Schuemie (TUD), Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy
2003-04 Milan Petkovic (UT), Content-Based Video Retrieval Supported by Database Technology
2003-05 Jos Lehmann (UVA), Causation in Artificial Intelligence and Law - A modelling approach
2003-06 Boris van Schooten (UT), Development and specification of virtual environments
2003-07 Machiel Jansen (UvA), Formal Explorations of Knowledge Intensive Tasks
2003-08 Yongping Ran (UM), Repair Based Scheduling
2003-09 Rens Kortmann (UM), The resolution of visually guided behaviour
2003-10 Andreas Lincke (UvT), Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture
2003-11 Simon Keizer (UT), Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks
2003-12 Roeland Ordelman (UT), Dutch speech recognition in multimedia information retrieval
2003-13 Jeroen Donkers (UM), Nosce Hostem - Searching with Opponent Models
2003-14 Stijn Hoppenbrouwers (KUN), Freezing Language: Conceptualisation Processes across ICT-Supported Organisations
2003-15 Mathijs de Weerdt (TUD), Plan Merging in Multi-Agent Systems
2003-16 Menzo Windhouwer (CWI), Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses
2003-17 David Jansen (UT), Extensions of Statecharts with Probability, Time, and Stochastic Timing
2003-18 Levente Kocsis (UM), Learning Search Decisions

2004
2004-01 Virginia Dignum (UU), A Model for Organizational Interaction: Based on Agents, Founded in Logic
2004-02 Lai Xu (UvT), Monitoring Multi-party Contracts for E-business
2004-03 Perry Groot (VU), A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving
2004-04 Chris van Aart (UVA), Organizational Principles for Multi-Agent Architectures
2004-05 Viara Popova (EUR), Knowledge discovery and monotonicity
2004-06 Bart-Jan Hommes (TUD), The Evaluation of Business Process Modeling Techniques
2004-07 Elise Boltjes (UM), Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes
2004-08 Joop Verbeek(UM), Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale Politionele gegevensuitwisseling en digitale expertise
2004-09 Martin Caminada (VU), For the Sake of the Argument; explorations into argument-based reasoning
2004-10 Suzanne Kabel (UVA), Knowledge-rich indexing of learning-objects
2004-11 Michel Klein (VU), Change Management for Distributed Ontologies
2004-12 The Duy Bui (UT), Creating emotions and facial expressions for embodied agents
2004-13 Wojciech Jamroga (UT), Using Multiple Models of Reality: On Agents who Know how to Play
2004-14 Paul Harrenstein (UU), Logic in Conflict. Logical Explorations in Strategic Equilibrium
2004-15 Arno Knobbe (UU), Multi-Relational Data Mining
2004-16 Federico Divina (VU), Hybrid Genetic Relational Search for Inductive Learning
2004-17 Mark Winands (UM), Informed Search in Complex Games
2004-18 Vania Bessa Machado (UvA), Supporting the Construction of Qualitative Knowledge Models
2004-19 Thijs Westerveld (UT), Using generative probabilistic models for multimedia retrieval
2004-20 Madelon Evers (Nyenrode), Learning from Design: facilitating multidisciplinary design teams

2005

2005-01 Floor Verdenius (UVA), Methodological Aspects of Designing Induction-Based Applications
2005-02 Erik van der Werf (UM), AI techniques for the game of Go
2005-03 Franc Grootjen (RUN), A Pragmatic Approach to the Conceptualisation of Language
2005-04 Nirvana Meratnia (UT), Towards Database Support for Moving Object data
2005-05 Gabriel Infante-Lopez (UVA), Two-Level Probabilistic Grammars for Natural Language Parsing
2005-06 Pieter Spronck (UM), Adaptive Game AI
2005-07 Flavius Frasincar (TUE), Hypermedia Presentation Generation for Semantic Web Information Systems
2005-08 Richard Vdovjak (TUE), A Model-driven Approach for Building Distributed Ontology-based Web Applications
2005-09 Jeen Broekstra (VU), Storage, Querying and Inferencing for Semantic Web Languages
2005-10 Anders Bouwer (UVA), Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments
2005-11 Elth Ogston (VU), Agent Based Matchmaking and Clustering - A Decentralized Approach to Search
2005-12 Csaba Boer (EUR), Distributed Simulation in Industry
2005-13 Fred Hamburg (UL),Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen
2005-14 Borys Omelayenko (VU),Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics
2005-15 Tibor Bosse (VU), Analysis of the Dynamics of Cognitive Processes
2005-16 Joris Graaumans (UU), Usability of XML Query Languages
2005-17 Boris Shishkov (TUD), Software Specification Based on Re-usable Business Components
2005-18 Danielle Sent (UU), Test-selection strategies for probabilistic networks
2005-19 Michel van Dartel (UM), Situated Representation
2005-20 Cristina Coteanu (UL), Cyber Consumer Law, State of the Art and Perspectives
2005-21 Wijnand Derks (UT), Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics

2006

2006-01 Samuil Angelov (TUE), Foundations of B2B Electronic Contracting
2006-02 Cristina Chisalita (VU), Contextual issues in the design and use of information technology in organizations
2006-03 Noor Christoph (UVA), The role of metacognitive skills in learning to solve problems
2006-04 Marta Sabou (VU), Building Web Service Ontologies
2006-05 Cees Pierik (UU), Validation Techniques for Object-Oriented Proof Outlines
2006-06 Ziv Baida (VU), Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling
2006-07 Marko Smiljanic (UT), XML schema matching -- balancing efficiency and effectiveness by means of clustering
2006-08 Eelco Herder (UT), Forward, Back and Home Again - Analyzing User Behavior on the Web
2006-09 Mohamed Wahdan (UM), Automatic Formulation of the Auditor's Opinion
2006-10 Ronny Siebes (VU), Semantic Routing in Peer-to-Peer Systems
2006-11 Joeri van Ruth (UT), Flattening Queries over Nested Data Types
2006-12 Bert Bongers (VU), Interactivation - Towards an e-cology of people, our technological environment, and the arts
2006-13 Henk-Jan Lebbink (UU), Dialogue and Decision Games for Information Exchanging Agents
2006-14 Johan Hoorn (VU), Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change
2006-15 Rainer Malik (UU), CONAN: Text Mining in the Biomedical Domain
2006-16 Carsten Riggelsen (UU), Approximation Methods for Efficient Learning of Bayesian Networks
2006-17 Stacey Nagata (UU), User Assistance for Multitasking with Interruptions on a Mobile Device
2006-18 Valentin Zhizhkun (UVA), Graph transformation for Natural Language Processing
2006-19 Birna van Riemsdijk (UU), Cognitive Agent Programming: A Semantic Approach
2006-20 Marina Velikova (UvT), Monotone models for prediction in data mining
2006-21 Bas van Gils (RUN), Aptness on the Web
2006-22 Paul de Vrieze (RUN), Fundaments of Adaptive Personalisation
2006-23 Ion Juvina (UU), Development of Cognitive Model for Navigating on the Web
2006-24 Laura Hollink (VU), Semantic Annotation for Retrieval of Visual Resources
2006-25 Madalina Drugan (UU), Conditional log-likelihood MDL and Evolutionary MCMC
2006-26 Vojkan Mihajlovic (UT), Score Region Algebra: A Flexible Framework for Structured Information Retrieval
2006-27 Stefano Bocconi (CWI), Vox Populi: generating video documentaries from semantically annotated media repositories
2006-28 Borkur Sigurbjornsson (UVA), Focused Information Access using XML Element Retrieval

2007

2007-01 Kees Leune (UvT), Access Control and Service-Oriented Architectures
2007-02 Wouter Teepe (RUG), Reconciling Information Exchange and Confidentiality: A Formal Approach
2007-03 Peter Mika (VU), Social Networks and the Semantic Web
2007-04 Jurriaan van Diggelen (UU), Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach
2007-05 Bart Schermer (UL), Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance
2007-06 Gilad Mishne (UVA), Applied Text Analytics for Blogs
2007-07 Natasa Jovanovic' (UT), To Whom It May Concern - Addressee Identification in Face-to-Face Meetings
2007-08 Mark Hoogendoorn (VU), Modeling of Change in Multi-Agent Organizations
2007-09 David Mobach (VU), Agent-Based Mediated Service Negotiation
2007-10 Huib Aldewereld (UU), Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols
2007-11 Natalia Stash (TUE), Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System
2007-12 Marcel van Gerven (RUN), Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty
2007-13 Rutger Rienks (UT), Meetings in Smart Environments; Implications of Progressing Technology
2007-14 Niek Bergboer (UM), Context-Based Image Analysis
2007-15 Joyca Lacroix (UM), NIM: a Situated Computational Memory Model
2007-16 Davide Grossi (UU), Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems
2007-17 Theodore Charitos (UU), Reasoning with Dynamic Networks in Practice
2007-18 Bart Orriens (UvT), On the development and management of adaptive business collaborations
2007-19 David Levy (UM), Intimate relationships with artificial partners
2007-20 Slinger Jansen (UU), Customer Configuration Updating in a Software Supply Network
2007-21 Karianne Vermaas (UU), Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005
2007-22 Zlatko Zlatev (UT), Goal-oriented design of value and process models from patterns
2007-23 Peter Barna (TUE), Specification of Application Logic in Web Information Systems
2007-24 Georgina Ramírez Camps (CWI), Structural Features in XML Retrieval
2007-25 Joost Schalken (VU), Empirical Investigations in Software Process Improvement

2008

2008-01 Katalin Boer-Sorbán (EUR), Agent-Based Simulation of Financial Markets: A modular, continuous-time approach
2008-02 Alexei Sharpanskykh (VU), On Computer-Aided Methods for Modeling and Analysis of Organizations
2008-03 Vera Hollink (UVA), Optimizing hierarchical menus: a usage-based approach
2008-04 Ander de Keijzer (UT), Management of Uncertain Data - towards unattended integration
2008-05 Bela Mutschler (UT), Modeling and simulating causal dependencies on process-aware information systems from a cost perspective
2008-06 Arjen Hommersom (RUN), On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective
2008-07 Peter van Rosmalen (OU), Supporting the tutor in the design and support of adaptive e-learning
2008-08 Janneke Bolt (UU), Bayesian Networks: Aspects of Approximate Inference
2008-09 Christof van Nimwegen (UU), The paradox of the guided user: assistance can be counter-effective
2008-10 Wauter Bosma (UT), Discourse oriented summarization
2008-11 Vera Kartseva (VU), Designing Controls for Network Organizations: A Value-Based Approach
2008-12 Jozsef Farkas (RUN), A Semiotically Oriented Cognitive Model of Knowledge Representation
2008-13 Caterina Carraciolo (UVA), Topic Driven Access to Scientific Handbooks
2008-14 Arthur van Bunningen (UT), Context-Aware Querying; Better Answers with Less Effort
2008-15 Martijn van Otterlo (UT), The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains.
2008-16 Henriette van Vugt (VU), Embodied agents from a user's perspective
2008-17 Martin Op 't Land (TUD), Applying Architecture and Ontology to the Splitting and Allying of Enterprises
2008-18 Guido de Croon (UM), Adaptive Active Vision
2008-19 Henning Rode (UT), From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search
2008-20 Rex Arendsen (UVA), Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven.
2008-21 Krisztian Balog (UVA), People Search in the Enterprise
2008-22 Henk Koning (UU), Communication of IT-Architecture

2008-23 Stefan Visscher (UU), Bayesian network models for the management of ventilator-associated pneumonia
2008-24 Zharko Aleksovski (VU), Using background knowledge in ontology matching
2008-25 Geert Jonker (UU), Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency
2008-26 Marijn Huijbregts (UT), Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled
2008-27 Hubert Vogten (OU), Design and Implementation Strategies for IMS Learning Design
2008-28 Ildiko Flesch (RUN), On the Use of Independence Relations in Bayesian Networks
2008-29 Dennis Reidsma (UT), Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users, and Other Humans
2008-30 Wouter van Atteveldt (VU), Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content
2008-31 Loes Braun (UM), Pro-Active Medical Information Retrieval
2008-32 Trung H. Bui (UT), Toward Affective Dialogue Management using Partially Observable Markov Decision Processes
2008-33 Frank Terpstra (UVA), Scientific Workflow Design; theoretical and practical issues
2008-34 Jeroen de Knijf (UU), Studies in Frequent Tree Mining
2008-35 Ben Torben Nielsen (UvT), Dendritic morphologies: function shapes structure

2009

2009-01 Rasa Jurgelenaite (RUN), Symmetric Causal Independence Models
2009-02 Willem Robert van Hage (VU), Evaluating Ontology-Alignment Techniques
2009-03 Hans Stol (UvT), A Framework for Evidence-based Policy Making Using IT
2009-04 Josephine Nabukenya (RUN), Improving the Quality of Organisational Policy Making using Collaboration Engineering
2009-05 Sietse Overbeek (RUN), Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality
2009-06 Muhammad Subianto (UU), Understanding Classification
2009-07 Ronald Poppe (UT), Discriminative Vision-Based Recovery and Recognition of Human Motion
2009-08 Volker Nannen (VU), Evolutionary Agent-Based Policy Analysis in Dynamic Environments
2009-09 Benjamin Kanagwa (RUN), Design, Discovery and Construction of Service-oriented Systems
2009-10 Jan Wielemaker (UVA), Logic programming for knowledge-intensive interactive applications
2009-11 Alexander Boer (UVA), Legal Theory, Sources of Law & the Semantic Web
2009-12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin), Operating Guidelines for Services
2009-13 Steven de Jong (UM), Fairness in Multi-Agent Systems
2009-14 Maksym Korotkiy (VU), From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)
2009-15 Rinke Hoekstra (UVA), Ontology Representation - Design Patterns and Ontologies that Make Sense
2009-16 Fritz Reul (UvT), New Architectures in Computer Chess
2009-17 Laurens van der Maaten (UvT), Feature Extraction from Visual Data
2009-18 Fabian Groffen (CWI), Armada, An Evolving Database System
2009-19 Valentin Robu (CWI), Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets
2009-20 Bob van der Vecht (UU), Adjustable Autonomy: Controling Influences on Decision Making
2009-21 Stijn Vanderlooy (UM), Ranking and Reliable Classification
2009-22 Pavel Serdyukov (UT), Search For Expertise: Going beyond direct evidence
2009-23 Peter Hofgesang (VU), Modelling Web Usage in a Changing Environment
2009-24 Annerieke Heuvelink (VUA), Cognitive Models for Training Simulations
2009-25 Alex van Ballegooij (CWI), "RAM: Array Database Management through Relational Mapping"
2009-26 Fernando Koch (UU), An Agent-Based Model for the Development of Intelligent Mobile Services
2009-27 Christian Glahn (OU), Contextual Support of social Engagement and Reflection on the Web
2009-28 Sander Evers (UT), Sensor Data Management with Probabilistic Models
2009-29 Stanislav Pokraev (UT), Model-Driven Semantic Integration of Service-Oriented Applications
2009-30 Marcin Zukowski (CWI), Balancing vectorized query execution with bandwidth-optimized storage
2009-31 Sofiya Katrenko (UVA), A Closer Look at Learning Relations from Text
2009-32 Rik Farenhorst (VU) and Remco de Boer (VU), Architectural Knowledge Management: Supporting Architects and Auditors
2009-33 Khiet Truong (UT), How Does Real Affect Affect Affect Recognition In Speech?
2009-34 Inge van de Weerd (UU), Advancing in Software Product Management: An Incremental Method Engineering Approach
2009-35 Wouter Koelewijn (UL), Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling
2009-36 Marco Kalz (OUN), Placement Support for Learners in Learning Networks

2009-37 Hendrik Drachsler (OUN), Navigation Support for Learners in Informal Learning Networks
2009-38 Riina Vuorikari (OU), Tags and self-organisation: a metadata ecology for learning resources in a multilingual context
2009-39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin), Service Substitution -- A Behavioral Approach Based on Petri Nets
2009-40 Stephan Raaijmakers (UvT), Multinomial Language Learning: Investigations into the Geometry of Language
2009-41 Igor Berezhnyy (UvT), Digital Analysis of Paintings
2009-42 Toine Bogers (UvT), Recommender Systems for Social Bookmarking
2009-43 Virginia Nunes Leal Franqueira (UT), Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients
2009-44 Roberto Santana Tapia (UT), Assessing Business-IT Alignment in Networked Organizations
2009-45 Jilles Vreeken (UU), Making Pattern Mining Useful
2009-46 Loredana Afanasiev (UvA), Querying XML: Benchmarks and Recursion

2010

2010-01 Matthijs van Leeuwen (UU), Patterns that Matter
2010-02 Ingo Wassink (UT), Work flows in Life Science
2010-03 Joost Geurts (CWI), A Document Engineering Model and Processing Framework for Multimedia documents
2010-04 Olga Kulyk (UT), Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments
2010-05 Claudia Hauff (UT), Predicting the Effectiveness of Queries and Retrieval Systems
2010-06 Sander Bakkes (UvT), Rapid Adaptation of Video Game AI
2010-07 Wim Fikkert (UT), Gesture interaction at a Distance
2010-08 Krzysztof Siewicz (UL), Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments
2010-09 Hugo Kielman (UL), A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging
2010-10 Rebecca Ong (UL), Mobile Communication and Protection of Children
2010-11 Adriaan Ter Mors (TUD), The world according to MARP: Multi-Agent Route Planning
2010-12 Susan van den Braak (UU), Sensemaking software for crime analysis
2010-13 Gianluigi Folino (RUN), High Performance Data Mining using Bio-inspired techniques
2010-14 Sander van Splunter (VU), Automated Web Service Reconfiguration
2010-15 Lianne Bodenstaff (UT), Managing Dependency Relations in Inter-Organizational Models
2010-16 Sicco Verwer (TUD), Efficient Identification of Timed Automata, theory and practice
2010-17 Spyros Kotoulas (VU), Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications
2010-18 Charlotte Gerritsen (VU), Caught in the Act: Investigating Crime by Agent-Based Simulation
2010-19 Henriette Cramer (UvA), People's Responses to Autonomous and Adaptive Systems
2010-20 Ivo Swartjes (UT), Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative
2010-21 Harold van Heerde (UT), Privacy-aware data management by means of data degradation
2010-22 Michiel Hildebrand (CWI), End-user Support for Access to\\ Heterogeneous Linked Data
2010-23 Bas Steunebrink (UU), The Logical Structure of Emotions
2010-24 Dmytro Tykhonov, Designing Generic and Efficient Negotiation Strategies
2010-25 Zulfiqar Ali Memon (VU), Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective
2010-26 Ying Zhang (CWI), XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines
2010-27 Marten Voulon (UL), Automatisch contracteren
2010-28 Arne Koopman (UU), Characteristic Relational Patterns
2010-29 Stratos Idreos(CWI), Database Cracking: Towards Auto-tuning Database Kernels
2010-30 Marieke van Erp (UvT), Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval
2010-31 Victor de Boer (UVA), Ontology Enrichment from Heterogeneous Sources on the Web
2010-32 Marcel Hiel (UvT), An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems
2010-33 Robin Aly (UT), Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval
2010-34 Teduh Dirgahayu (UT), Interaction Design in Service Compositions
2010-35 Dolf Trieschnigg (UT), Proof of Concept: Concept-based Biomedical Information Retrieval
2010-36 Jose Janssen (OU), Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification
2010-37 Niels Lohmann (TUE), Correctness of services and their composition
2010-38 Dirk Fahland (TUE), From Scenarios to components
2010-39 Ghazanfar Farooq Siddiqui (VU), Integrative modeling of emotions in virtual agents
2010-40 Mark van Assem (VU), Converting and Integrating Vocabularies for the Semantic Web
2010-41 Guillaume Chaslot (UM), Monte-Carlo Tree Search

2010-42 Sybren de Kinderen (VU), Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach
2010-43 Peter van Kranenburg (UU), A Computational Approach to Content-Based Retrieval of Folk Song Melodies
2010-44 Pieter Bellekens (TUE), An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain
2010-45 Vasilios Andrikopoulos (UvT), A theory and model for the evolution of software services
2010-46 Vincent Pijpers (VU), e3alignment: Exploring Inter-Organizational Business-ICT Alignment
2010-47 Chen Li (UT), Mining Process Model Variants: Challenges, Techniques, Examples
2010-48 Withdrawn
2010-49 Jahn-Takeshi Saito (UM), Solving difficult game positions
2010-50 Bouke Huurnink (UVA), Search in Audiovisual Broadcast Archives
2010-51 Alia Khairia Amin (CWI), Understanding and supporting information seeking tasks in multiple sources
2010-52 Peter-Paul van Maanen (VU), Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention
2010-53 Edgar Meij (UVA), Combining Concepts and Language Models for Information Access

2011

2011-01 Botond Cseke (RUN), Variational Algorithms for Bayesian Inference in Latent Gaussian Models
2011-02 Nick Tinnemeier(UU), Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language
2011-03 Jan Martijn van der Werf (TUE), Compositional Design and Verification of Component-Based Information Systems
2011-04 Hado van Hasselt (UU), Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms
2011-05 Base van der Raadt (VU), Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.
2011-06 Yiwen Wang (TUE), Semantically-Enhanced Recommendations in Cultural Heritage
2011-07 Yujia Cao (UT), Multimodal Information Presentation for High Load Human Computer Interaction
2011-08 Nieske Vergunst (UU), BDI-based Generation of Robust Task-Oriented Dialogues
2011-09 Tim de Jong (OU), Contextualised Mobile Media for Learning
2011-10 Bart Bogaert (UvT), Cloud Content Contention
2011-11 Dhaval Vyas (UT), Designing for Awareness: An Experience-focused HCI Perspective
2011-12 Carmen Bratosin (TUE), Grid Architecture for Distributed Process Mining
2011-13 Xiaoyu Mao (UvT), Airport under Control. Multiagent Scheduling for Airport Ground Handling
2011-14 Milan Lovric (EUR), Behavioral Finance and Agent-Based Artificial Markets
2011-15 Marijn Koolen (UvA), The Meaning of Structure: the Value of Link Evidence for Information Retrieval
2011-16 Maarten Schadd (UM), Selective Search in Games of Different Complexity
2011-17 Jiyin He (UVA), Exploring Topic Structure: Coherence, Diversity and Relatedness
2011-18 Mark Ponsen (UM), Strategic Decision-Making in complex games
2011-19 Ellen Rusman (OU), The Mind ' s Eye on Personal Profiles
2011-20 Qing Gu (VU), Guiding service-oriented software engineering - A view-based approach
2011-21 Linda Terlouw (TUD), Modularization and Specification of Service-Oriented Systems
2011-22 Junte Zhang (UVA), System Evaluation of Archival Description and Access
2011-23 Wouter Weerkamp (UVA), Finding People and their Utterances in Social Media
2011-24 Herwin van Welbergen (UT), Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior
2011-25 Syed Waqar ul Qounain Jaffry (VU)), Analysis and Validation of Models for Trust Dynamics
2011-26 Matthijs Aart Pontier (VU), Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots
2011-27 Aniel Bhulai (VU), Dynamic website optimization through autonomous management of design patterns
2011-28 Rianne Kaptein(UVA), Effective Focused Retrieval by Exploiting Query Context and Document Structure
2011-29 Faisal Kamiran (TUE), Discrimination-aware Classification
2011-30 Egon van den Broek (UT), Affective Signal Processing (ASP): Unraveling the mystery of emotions
2011-31 Ludo Waltman (EUR), Computational and Game-Theoretic Approaches for Modeling Bounded Rationality
2011-32 Nees-Jan van Eck (EUR), Methodological Advances in Bibliometric Mapping of Science
2011-33 Tom van der Weide (UU), Arguing to Motivate Decisions
2011-34 Paolo Turrini (UU), Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations
2011-35 Maaike Harbers (UU), Explaining Agent Behavior in Virtual Training
2011-36 Erik van der Spek (UU), Experiments in serious game design: a cognitive approach
2011-37 Adriana Burlutiu (RUN), Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference
2011-38 Nyree Lemmens (UM), Bee-inspired Distributed Optimization

2011-39 Joost Westra (UU), Organizing Adaptation using Agents in Serious Games
2011-40 Viktor Clerc (VU), Architectural Knowledge Management in Global Software Development
2011-41 Luan Ibraimi (UT), Cryptographically Enforced Distributed Data Access Control
2011-42 Michal Sindlar (UU), Explaining Behavior through Mental State Attribution
2011-43 Henk van der Schuur (UU), Process Improvement through Software Operation Knowledge
2011-44 Boris Reuderink (UT), Robust Brain-Computer Interfaces
2011-45 Herman Stehouwer (UvT), Statistical Language Models for Alternative Sequence Selection
2011-46 Beibei Hu (TUD), Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work
2011-47 Azizi Bin Ab Aziz(VU), Exploring Computational Models for Intelligent Support of Persons with Depression
2011-48 Mark Ter Maat (UT), Response Selection and Turn-taking for a Sensitive Artificial Listening Agent
2011-49 Andreea Niculescu (UT), Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality

2012
2012-01 Terry Kakeeto (UvT), Relationship Marketing for SMEs in Uganda
2012-02 Muhammad Umair(VU), Adaptivity, emotion, and Rationality in Human and Ambient Agent Models
2012-03 Adam Vanya (VU), Supporting Architecture Evolution by Mining Software Repositories
2012-04 Jurriaan Souer (UU), Development of Content Management System-based Web Applications
2012-05 Marijn Plomp (UU), Maturing Interorganisational Information Systems
2012-06 Wolfgang Reinhardt (OU), Awareness Support for Knowledge Workers in Research Networks
2012-07 Rianne van Lambalgen (VU), When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions
2012-08 Gerben de Vries (UVA), Kernel Methods for Vessel Trajectories
2012-09 Ricardo Neisse (UT), Trust and Privacy Management Support for Context-Aware Service Platforms
2012-10 David Smits (TUE), Towards a Generic Distributed Adaptive Hypermedia Environment
2012-11 J.C.B. Rantham Prabhakara (TUE), Process Mining in the Large: Preprocessing, Discovery, and Diagnostics
2012-12 Kees van der Sluijs (TUE), Model Driven Design and Data Integration in Semantic Web Information Systems
2012-13 Suleman Shahid (UvT), Fun and Face: Exploring non-verbal expressions of emotion during playful interactions
2012-14 Evgeny Knutov(TUE), Generic Adaptation Framework for Unifying Adaptive Web-based Systems
2012-15 Natalie van der Wal (VU), Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.
2012-16 Fiemke Both (VU), Helping people by understanding them - Ambient Agents supporting task execution and depression treatment
2012-17 Amal Elgammal (UvT), Towards a Comprehensive Framework for Business Process Compliance
2012-18 Eltjo Poort (VU), Improving Solution Architecting Practices
2012-19 Helen Schonenberg (TUE), What's Next? Operational Support for Business Process Execution
2012-20 Ali Bahramisharif (RUN), Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing
2012-21 Roberto Cornacchia (TUD), Querying Sparse Matrices for Information Retrieval
2012-22 Thijs Vis (UvT), Intelligence, politie en veiligheidsdienst: verenigbare grootheden?
2012-23 Christian Muehl (UT), Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction
2012-24 Laurens van der Werff (UT), Evaluation of Noisy Transcripts for Spoken Document Retrieval
2012-25 Silja Eckartz (UT), Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application
2012-26 Emile de Maat (UVA), Making Sense of Legal Text
2012-27 Hayrettin Gurkok (UT), Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games
2012-28 Nancy Pascall (UvT), Engendering Technology Empowering Women
2012-29 Almer Tigelaar (UT), Peer-to-Peer Information Retrieval
2012-30 Alina Pommeranz (TUD), Designing Human-Centered Systems for Reflective Decision Making
2012-31 Emily Bagarukayo (RUN), A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure
2012-32 Wietske Visser (TUD), Qualitative multi-criteria preference representation and reasoning
2012-33 Rory Sie (OUN), Coalitions in Cooperation Networks (COCOON)
2012-34 Pavol Jancura (RUN), Evolutionary analysis in PPI networks and applications
2012-35 Evert Haasdijk (VU), Never Too Old To Learn -- On-line Evolution of Controllers in Swarm- and Modular Robotics
2012-36 Denis Ssebugwawo (RUN), Analysis and Evaluation of Collaborative Modeling Processes
2012-37 Agnes Nakakawa (RUN), A Collaboration Process for Enterprise Architecture Creation

2012-38 Selmar Smit (VU), Parameter Tuning and Scientific Testing in Evolutionary Algorithms
2012-39 Hassan Fatemi (UT), Risk-aware design of value and coordination networks
2012-40 Agus Gunawan (UvT), Information Access for SMEs in Indonesia
2012-41 Sebastian Kelle (OU), Game Design Patterns for Learning
2012-42 Dominique Verpoorten (OU), Reflection Amplifiers in self-regulated Learning
2012-43 Withdrawn
2012-44 Anna Tordai (VU), On Combining Alignment Techniques
2012-45 Benedikt Kratz (UvT), A Model and Language for Business-aware Transactions
2012-46 Simon Carter (UVA), Exploration and Exploitation of Multilingual Data for Statistical Machine Translation
2012-47 Manos Tsagkias (UVA), Mining Social Media: Tracking Content and Predicting Behavior
2012-48 Jorn Bakker (TUE), Handling Abrupt Changes in Evolving Time-series Data
2012-49 Michael Kaisers (UM), Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions
2012-50 Steven van Kervel (TUD), Ontologogy driven Enterprise Information Systems Engineering
2012-51 Jeroen de Jong (TUD), Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching

2013
2013-01 Viorel Milea (EUR), News Analytics for Financial Decision Support
2013-02 Erietta Liarou (CWI), MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing
2013-03 Szymon Klarman (VU), Reasoning with Contexts in Description Logics
2013-04 Chetan Yadati (TUD), Coordinating autonomous planning and scheduling
2013-05 Dulce Pumareja (UT), Groupware Requirements Evolutions Patterns
2013-06 Romulo Goncalves(CWI), The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience
2013-07 Giel van Lankveld (UvT), Quantifying Individual Player Differences
2013-08 Robbert-Jan Merk(VU), Making enemies: cognitive modeling for opponent agents in fighter pilot simulators
2013-09 Fabio Gori (RUN), Metagenomic Data Analysis: Computational Methods and Applications
2013-10 Jeewanie Jayasinghe Arachchige(UvT), A Unified Modeling Framework for Service Design.
2013-11 Evangelos Pournaras(TUD), Multi-level Reconfigurable Self-organization in Overlay Services
2013-12 Marian Razavian(VU), Knowledge-driven Migration to Services
2013-13 Mohammad Safiri(UT), Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly
2013-14 Jafar Tanha (UVA), Ensemble Approaches to Semi-Supervised Learning Learning
2013-15 Daniel Hennes (UM), Multiagent Learning - Dynamic Games and Applications
2013-16 Eric Kok (UU), Exploring the practical benefits of argumentation in multi-agent deliberation
2013-17 Koen Kok (VU), The PowerMatcher: Smart Coordination for the Smart Electricity Grid
2013-18 Jeroen Janssens (UvT), Outlier Selection and One-Class Classification
2013-19 Renze Steenhuizen (TUD), Coordinated Multi-Agent Planning and Scheduling
2013-20 Katja Hofmann (UvA), Fast and Reliable Online Learning to Rank for Information Retrieval
2013-21 Sander Wubben (UvT), Text-to-text generation by monolingual machine translation
2013-22 Tom Claassen (RUN), Causal Discovery and Logic
2013-23 Patricio de Alencar Silva(UvT), Value Activity Monitoring
2013-24 Haitham Bou Ammar (UM), Automated Transfer in Reinforcement Learning
2013-25 Agnieszka Anna Latoszek-Berendsen (UM), Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System
2013-26 Alireza Zarghami (UT), Architectural Support for Dynamic Homecare Service Provisioning
2013-27 Mohammad Huq (UT), Inference-based Framework Managing Data Provenance
2013-28 Frans van der Sluis (UT), When Complexity becomes Interesting: An Inquiry into the Information eXperience
2013-29 Iwan de Kok (UT), Listening Heads
2013-30 Joyce Nakatumba (TUE), Resource-Aware Business Process Management: Analysis and Support
2013-31 Dinh Khoa Nguyen (UvT), Blueprint Model and Language for Engineering Cloud Applications
2013-32 Kamakshi Rajagopal (OUN), Networking For Learning; The role of Networking in a Lifelong Learner's Professional
2013-33 Qi Gao (TUD), User Modeling and Personalization in the Microblogging Sphere
2013-34 Kien Tjin-Kam-Jet (UT), Distributed Deep Web Search
2013-35 Abdallah El Ali (UvA), Minimal Mobile Human Computer Interaction
2013-36 Than Lam Hoang (TUe), Pattern Mining in Data Streams
2013-37 Dirk Börner (OUN), Ambient Learning Displays
2013-38 Eelco den Heijer (VU), Autonomous Evolutionary Art