Computational Intelligence, Volume 18, Number 2, 2002

TRENDS IN AGENT COMMUNICATION LANGUAGE

B. Chaib-draa
Laval University, Canada

F. Dignum
Utrecht University, The Netherlands

Agent technology is an exciting and important new way to create complex software systems. Agents blend many of the traditional properties of AI programs—knowledge-level reasoning, flexibility, proactiveness, goal-directedness, and so forth—with insights gained from distributed software engineering, machine learning, negotiation and teamwork theory, and the social sciences. An important part of the agent approach is the principle that agents (like humans) can function more effectively in groups that are characterized by cooperation and division of labor. Agent programs are designed to autonomously collaborate with each other in order to satisfy both their internal goals and the shared external demands generated by virtue of their participation in agent societies. This type of collaboration depends on a sophisticated system of inter-agent communication. The assumption that inter-agent communication is best handled through the explicit use of an agent communication language (ACL) underlies each of the articles in this special issue. In this introductory article, we will supply a brief background and introduction to the main topics in agent communication.

Key words: agent communication language, KQML, FIPA-ACL, semantics, speech acts, conversations.

1. BASIC COMPONENTS AND ORIGINS OF AN ACL

Multi-agent systems (MAS) are the subject of research for researchers studying systems made up of multiple heterogeneous intelligent software entities (called agents). The agents in a MAS can compete, cooperate, or simply coexist. MAS differs from *distributed problem solving* in the sense that there is no common global goal to be solved which is known at design time; on the contrary, a multi-agent system is generally populated by different agents having different purposes.

In recent years the interest in MAS has grown tremendously, and today multi-agent technology is being used in a large range of important industrial application areas. These applications range from information management through industrial process control to electronic commerce. All these applications have one thing in common. Agents must be able to "talk" to each other to decide what action to take and how this action can be coordinated with others' actions. The language used by the agents for this exchange is the *agent communication language* (ACL). An ACL stems from the need to coordinate the actions of an agent with that of the other agents. It can be used to share information and knowledge among agents in distributed computing environments, but also to request the performance of a task. The main objective of an ACL is to model a suitable framework that allows heterogeneous agents to interact, to communicate with meaningful statements that convey information about their environment or knowledge (Kone 2000). An important part of the agent approach is the principle that agents (like humans) can function more effectively in groups that are characterized by cooperation and division of labor. Agent programs are designed to autonomously collaborate with each other in order to satisfy both their internal goals and the shared external demands generated by virtue of their participation in agent societies. The balance between collaboration and fulfilling its own goals is made by each agent individually and depending on the situation. Due to the autonomy of the agents the collaboration needs a sophisticated system of agent communication. In this introduction, we will supply a brief background and introduction to the main topics relevant for agent communication.

1.1. Basic Components of an ACL

ACLs are high-level languages whose primitives and structures are expressly tailored to support the kinds of collaboration, negotiation, and information transfer required in multiagent interaction. ACLs exist in a logical layer above transport protocols such as TCP/IP, HTTP, or IIOP. Such protocols deal with communication issues at the level of data and message transport, while ACLs address communication on the intentional and social level. ACLs themselves are complex structures composed of different sublanguages that specify the message content, interpretation parameters such as the sender and the ontology, the propositional attitude under which the receiver should interpret the message content, and several other components. Typical ACLs also have a characteristic mentalistic semantics that is far more complex than standard distributed object protocols. This means that ACL design is a delicate balance between the communicative needs of the agent with the ability of receivers to compute (in tractable time) the intended meaning of the message. Further, it is important that the syntax, semantics, and pragmatics of the various components of an ACL be as precise and explicit as possible, so that the agent systems using that ACL can be as open and accessible to developers beyond the original group.

This last point bears some emphasis. Historically, many MASs have been built using somewhat *ad-hoc* and developer-private communication mechanisms.[1] Although these systems often contain many independent agents and can exhibit impressive accomplishments, the agents involved often rely on a large number of communicative assumptions that are not true of arbitrary agent collections. These assumptions range from the presumption of a shared *ontology* and problem domain to specific nonstandard meanings for messages (or the absence of a message) that are tailored to particular contexts. These, often undocumented, assumptions are made by agent developers for reasons of communication efficiency or developer convenience, and knowledge of them is critical to properly interpret the agent message traffic in these systems. So, while such purpose-built agent collections are important to test and validate different hypotheses and approaches to agent problems, they can be extremely difficult to generalize and extend without extensive interaction with the original developers. The locus of this problem can be traced to these implicit domain-specific assumptions in the agent communication design.

The articles in this special issue all address a set of issues in general ACL and agent interaction design. The ACLs that they discuss are intended to have explicit principles surrounding their proper interpretation in a context of use.[2] Further, these ACLs are also designed to be generally applicable to a wide variety of agent interaction types. The combination of explicitness and generality leads to extremely expressive languages with well-defined semantics that are grounded in powerful logics.

1.2. Origins of ACLs

A first attempt to come to a standardized agent communication language (ACL) came forth from the ARPA knowledge sharing project and produced knowledge query and

[1] Agents can also communicate with actions other than classically linguistic productions, simply by making observable changes in the environment that have semantic force (this sort of communication is generally based on signs or signals). For example, an agent that locks a resource for itself might be assumed to communicate to an observer its need for the resource at that time. However, without a general semantic and pragmatic theory of action, it is impossible for other agents to precisely characterize the meaning of such actions, or to understand them as communicative. And, the provision of such a theory simply turns the applicable actions into a *de facto* communication system, albeit one with an unorthodox syntax.

[2] Although the ACLs described in this special issue are very generic, they all make certain assumptions about the agents that use them and/or the environment in which they are used. It is not possible to create an ACL without any assumptions at all; hence the emphasis on making these assumptions explicit.

manipulation language (KQML). In the context of this project, researchers developed two main components: (1) a representation language for the contents of messages (called Knowledge Interchange Format (KIF), which is an extension of first-order logic; and (2) a communication language KQML which consists of a set of communication primitives aiming to support interaction among agents in MAS. KQML includes many performatives of speech acts, all assertives (i.e., when it states a fact) or directives (i.e., when it reflects command or request), which agents use to assert facts, request information, or subscribe to services. A sample KQML message has the following syntax (tell :sender A :receiver B :content "snowing"), that is, the agent A tells to agent B that the proposition "it is snowing" is true. The semantics of KQML presupposes that each has its own virtual knowledge base (KB). In these conditions, telling P corresponds to reporting that P is in its KB; asking for P is attempting to extract P from the addressee's KB, etc. The main advantage of KQML is its ability to support a wide range of agent architectures with its extensible set of performatives. The early version, however, presented some confusions and ambiguities in the usage of its performatives, mainly due to a lack of a precise semantic definition of the performatives. Later on its authors gave it a semantics issued from the theoretical foundation of Searle and Vanderveken (1985) and limited the use of some performatives in order to avoid some problems.

More recently another effort to come to a standard ACL has started through the Foundation for Intelligent Physical Agents (FIPA) initiative. This foundation is a nonprofit association whose objective consists of promoting the success of emerging agent-based technology. It operates through the open international collaboration of companies and universities who are active members in the field. FIPA assigns tasks (ontologies, semantics, architectures, gateways, compliance) to technical committees, each of which has primary responsibility for producing, maintaining, and updating the specifications applicable to its tasks.

FIPA's agent communication language (or FIPA-ACL) is also based on speech act theory and messages are also considered as communicative acts whose objective is to perform some action by virtue of being sent. Precisely, FIPA's effort brings ARCOL to bear on ACL, a language developed by France Télécom (Sadek 1991). In FIPA-ACL the set of primitives is smaller than in KQML (but new performatives can be defined by formally combining primitives) and this set also includes assertives or directives as in KQML. ARCOL has a formal semantics based on Cohen and Levesque's approach on speech acts (1990). Contrary to KQML, in ARCOL, agent A can tell agent B that P only if A believes that P and believes that B does not believe P. Thus, ARCOL gives preconditions on communicative acts as specified by its semantics. Although theoretically nice, it is also seen by some as its main weakness. It will be very difficult to determine whether the listening agent believes a fact or not and therefore whether a fact can be told to that agent.

Finally, we will point out an important area of agent communication and ACL use that the articles in this special issue do not emphasize. Successfully using an ACL to communicate between two agent programs is dependent on the proper functioning of a great deal of non-ACL communications infrastructure. This infrastructure involves message ordering and delivery, formatting and addressing, directory services, gateway-style translation, quality-of-service, and other standard networking and communications issues. In practice, implemented agent systems have used both centralized strategies that handle all aspects of messaging between agents (often implemented in KQML through the introduction of a router that receives a message from the registered agents and routes the message to the correct receiver), as well as more decentralized systems that devolve this functionality to the communication handlers of the agents themselves (as is typically done in FIPA-ACL). Agent systems exhibit little standardization in this area. Further, implementations of ACLs often expose more details of this infrastructure than one might strictly like. The safest thing to say is that there are many deep and difficult problems when designing a general agent communications infrastructure that will be efficient and reliable for different communication topologies and networking systems.

2. ISSUES IN AGENT COMMUNICATION

In this section we will introduce a set of issues that have been important in the development of ACLs and agent communication theory generally. Although we will use KQML and FIPA-ACL as examples and point out a number of potential problems with them, we do not intend to criticize the effort put into the design of these languages. Much of our knowledge of these problems came from the practical experience of researchers applying KQML or FIPA-ACL in their systems, and we are indebted to the many researchers who have devoted enormous efforts to these ACLs.

2.1. Theories of Agency

One of the core research issues in the agent communication community involves the linkage between the semantic underpinnings of the ACL and the theory of agency that regulates and defines the agent's behavior. In order for the messages of an ACL to be formally coherent, these two theories must be aligned.

A theory of agency is a general formal model that specifies what actions an agent can or should perform in various situations. Like the Turing model of computability, it abstracts away from any particular implementation, and functions as a normative theory that is useful for analysis. Theories of agency for software agents are usually based on a small set of primitives derived from the propositional attitudes of philosophy (e.g., belief, desire, and intention—that led to BDI architecture) and a set of axioms or axiom schema that defines their entailment relations. A complete theory of agency also includes accounts of the agent's general reasoning strategy and deductive model, its theory of action and causality, its account of planning and goal satisfaction, its system of belief dynamics and revision, and so forth. An agent need not directly implement its theory of agency, but it must behave as if it did. Examples of the elements that compose a theory of agency include Moore's accounts of knowledge and action (1985), Georgeff and Rao's BDI architecture (1999), Singh's know-how and branching time systems (1998), Cohen and Levesque's intention theories (1990), and so forth. Different agent systems will combine different elements to comprise their own theory of agency.

An agent's communicative behavior is among the behaviors regulated by a theory of agency. Because of this, the semantic theories that define the meaning of an ACL message must ultimately be linked to the entities provided by the agent's baseline theory of agency. Current versions of both KQML and FIPA-ACL handle the linkage between the semantic theory and the theory of agency by appealing to a simplified version of natural language speech act theory (originally developed by Searle (1969)). In this approach, agent communication is treated as a type of action that affects the world in the same way that physical acts affect the world. Precisely, message types of ACLs are considered as *speech acts*, which in turn are described and defined in terms of beliefs, desires, and intentions.

The current semantic theory of FIPA-ACL depends on a theory of agency that supplies a set of BDI-style primitives. The semantics of FIPA-ACL is based on mentalistic notions such as belief and intention, and (because of its speech-act theory component) treats agent messaging as a type of action. Formally, this means that FIPA-ACL's semantic theory is expressed in an extremely powerful quantified multimodal logic involving both belief and intention as primitive operators, as well as a simple theory of action. As a result, agents that aspire to use FIPA-ACL in a semantically coherent way are required to adhere to a BDI-style theory of agency. They also face the somewhat daunting task of acting as if they implemented a reasoning engine for the semantic account.

In contrast to the FIPA-ACL, KQML did not originally assume a full BDI architecture of the agents. Rather, the original KQML semantics were defined in terms of a very simple theory of agency centered on adding and deleting assertions from a virtual knowledge base. The assumptions made about the required behavior of KQML agents were very weak, and the resultant semantics of KQML messages were much more permissive than that of FIPA-ACL. As is now well known, this permissiveness allowed wide latitude in KQML implementations, and contributed to the proliferation of different and incompatible KQML dialects. Labrou's (1997) second-generation semantics for KQML was much more precise, and based on a sophisticated BDI-style theory of agency similar to that of FIPA-ACL. However, Labrou's use of modal logic to specify the preconditions, postconditions, and completion conditions for each KQML communicative act type made the complexity of semantic reasoning for KQML messages comparable to that required by FIPA-ACL.

Mismatches between the theory of agency and the semantic theory can occur when the theory of agency licenses communicative actions that are not expressible in the ACL semantics. The *sincerity* condition on agent ACL usage is one such example. Sophisticated theories of agency often allow agents to act with the intent to deceive if it furthers the agent's goals. This is often cited as a requirement for optimal behavior in electronic commerce applications and adversarial negotiations generally; for example, the revenue-maximizing strategy of an agent might involve deceiving another agent about the first agent's true valuation of a good. However, in order to make the message semantics as useful as possible, most ACL semantic theories (such as the KQML and FIPA-ACL theories) require that agents never use that ACL to assert something that they do not themselves believe. This is a strengthening of the analogous principle for humans: we do not typically assume that our interlocutors are lying to us. But it also makes possible the situation that an agent might desire to communicate something while the semantic preconditions of the ACL message forbid him to do so. Can we actually presuppose that the other agents also will be sincere in their communication? Or will they just deviate from the required semantics whenever it is convenient? One point in question is that it will be very difficult to verify the sincerity of another agent. Theoretically, an agent could even temporarily change its believes to be sincere (while changing them back right after performing the speech act!). See Pitt (2000) for more details on this problem.

The sincerity condition serves as a simplifying assumption for agent communication. Another such assumption involves the ability of an agent to reliably observe the effects of another agent's actions. Applied to agent communication, this is often taken to mean that the inter-agent communication channels are error-free. Agent systems routinely assume that all messages eventually arrive at their intended recipients and are not distorted by the environment (or malicious actors) in transit. Often, it is further assumed that the order of the messages that are sent to the same destination does not change during the transportation. Depending on the agent's execution context, these assumptions may not be appropriate. Again, the possibility exists that simplifying assumptions in the ACL could foreclose certain types of desirable or rational behavior relative to the agent's theory of agency.

2.2. ACLs Semantics

Recall that the pre- and postconditions of actions in programming languages can be expressed in terms of variables and their values before and after the action, because the relevant types of actions are limited to manipulating the values of variables. However, communicative acts in an ACL do not directly manipulate variables and their values. They are conceived to operate at the higher level of abstraction given by the theory of agency, and refer to the primitives supplied by this theory. Therefore, the preconditions and postconditions for communicative acts are typically expressed in terms of the mental attitudes of the involved agents. For example, the precondition of KQML's tell message states that the sender believes what it tells and that it knows that the receiver wants to know that the sender believes it. The postcondition of sending the tell message is that the receiver can conclude that the sender believes the content of the message.[3] The semantics for FIPA-ACL is based on a similar semantic approach that involves specifying a message's feasible preconditions and rationally expected effects.

Although the precondition/postcondition approach can supply a minimal meaning for messages in an ACL, situations frequently occur where it is desirable to overload this minimal meaning with a more precise and context-specific gloss. This leads to a tension in ACL semantic theory. On one hand, we want the semantics to be flexible enough to be applicable in all situations where agents use the ACL. Therefore we formulate very general pre- and postconditions in the formal statement of the semantics. On the other hand, the resulting pre- and postconditions are often so abstract that they are not fully adequate in all situations. Furthermore, it is often very difficult to verify whether the agent's state in fact satisfies the pre- and postconditions. This is partly due to the fact that, although we routinely ascribe mental attitudes to agents, agents are almost never actually programmed using these concepts directly. For example, how would one verify, for agents i and j and a proposition p, that "i knows that j wants to know that i believes p"?

To illustrate these considerations on pre- and postconditions, let's consider first the KQML semantics. This semantics has been expressed in terms of "preconditions," "postconditions," and "completion conditions." As expressed in Labrou (1997), preconditions indicate the necessary states for an agent to send a performative, and for the receiver to accept it. If these preconditions do not hold, then error or sorry is generated. Postconditions, on the other hand, describe the states of the sender after the successful utterance of a performative, and of the receiver after the receipt and processing of a message. Postconditions concerning the sender and receiver hold unless error or sorry is sent as response to report the unsuccessful sending or processing of the message. Finally, a completion condition indicates the final state, which generally corresponds to the fulfillment of the intention that starts the conversation. As we can see, in KQML, precondition, postconditions, and completion conditions describe mental states of agents as suggested by Searle and Vanderveken (1985) and there is no semantics associated to those mental states. In this case, the "semantics problem" is just transposed from performatives to agents' mental states.

The FIPA-ACL semantics is sustained by a formal language called SL. This language is a quantified multimodal logic with modal operators for beliefs, desires, uncertain beliefs, and persistent goals (a form of intention). Such language goes back to the work of Cohen and Levesque (1990) and has been extended by Sadek (1991) in the context of ARCOL. It is used for FIPA-ACL semantics by ascribing to each communicative act (inform, request, etc.) sets of SL formulae describing the act's *feasibility preconditions* and *rational effects*. According to this semantics, an inform act, for example, in which agent i informs agent j of content p, leads to the following: (1) i believes that p holds; (2) i does not already believe that the receiver has any knowledge of the truth of p; (3) i intends that the receiving agent should also come to believe that the proposition is true.

Comparatively to KQML, FIPA-ACL offers a small set of primitives that can be combined. It also has a formal semantics that can support the interoperability. Its limitations revolve

[3] KQML also has the notion of a completion condition, which roughly corresponds to the conditions that are obtained after the successful performance of the act in a normal and cooperative communicative context.

around the fact that its minimal semantics, rests only on the belief states of communicating agents, offers no clue on how to infer the mental states of the receiver. In addition, FIPA-ACL has a fixed context with the sender agent which can be an impediment to the heterogeneity (Kone 2000).

To summarize we can state that some work has been done on semantics of ACLs but that the semantics of KQML and FIPA-ACL is based on the mental agency, that is, communicative acts are described in terms of beliefs, intentions, desires, and similar mental states. However, agents are almost never actually programmed using such mental states directly. Therefore it is almost impossible to verify whether the messages are used correctly by the agents and the link between theory and practice in ACL use is still very big.

Lately an alternative to the mental agency has been advocated by, for example, Singh (2000): the social agency. This promising approach considers communicative acts as part of ongoing social interaction. In this case, even if we cannot determine whether agents have a specific mental state, we are sure that communicating agents follow some social laws that sustain conversations. Agent designers have usually assumed that the networks of obligation and power relationships that characterize human social behavior are not relevant to multi-agent systems. In practice, however, idiosyncratic social conventions have ended up being implicitly embedded in agent architectures and interaction protocols; for instance, the simple norm that an agent will usually try to answer a query (and even tries to find the best answer). Whether an agent is cooperative or not is, however, for the large part determined by the commitments, obligations, etc. Because these concepts are only implicitly incorporated into the behavior (and protocols) of the agents the result is that different agent systems exhibit significant incompatibilities in this area. More research is needed into characterizing these fundamental communicative concepts in a multi-agent systems context. This includes concepts such as "commitment," "obligation," "convention," "power" (in the sense of hierarchical relations), and so forth. Once these concepts are clarified, it then becomes possible to build a unified ACL semantics and pragmatics that take account of these concepts.

Although some work has been done on the semantics of individual speech acts in KQML and FIPA-ACL, little is known about the semantics of conversations and the relations between speech acts and the conversations of which they form a part. A clear semantics of conversations can facilitate "extensibility" and "scalability" of conversations between agents in the sense that it can be easier for a user to extend existing conversations with new performatives (communicative acts). It is interesting to see that several articles in this special issue touch on this important issue.

2.3. Verification

Although the point has been made above already, we think it is worth pointing out explicitly that the verification of the semantics of an ACL as well as the verification of an instantiation of a protocol to a protocol specification is an issue that is, at least, underrated. The only publication to date explicitly addressing the issue of verification of the semantics of an ACL is that of Wooldridge (2000).

If the semantics of an ACL has been given in some kind of logic then the verification whether the agents that use the ACL actually comply with the semantics comes down to a proof of the semantics of the ACL from the semantics of the agent programs. Here we encounter the first problem. Although it will probably be possible to give a formal semantics of any program, this can usually only be done in terms of some kind of temporal logic. It therefore does not include concepts such as "belief," "intention," "goals," etc. If the agent implementation would have been made strictly on the basis of a BDI theory there might be a possible interpretation in these terms, but rarely would it be unique and unambiguous.

So, there are two possible choices to be made. Either verification of the ACL can only be done when the agent systems conform completely and formally to the mentalistic concepts used in the semantics of agent theories (and ACLs) or the semantics of ACLs also has to be given in a simpler logic than the multi-modal logics commonly used for, e.g., KQML and FIPA-ACL. Wooldridge tries the second approach in his paper, but therewith also reduces agent communication to a non-intentional communication language. However, it might be a more practical approach than trying for a complete semantics that defines both the agent programs and the ACL completely in mentalistic concepts.

The second, related, verification problem mentioned above is that of verifying an instance of a communication protocol. This is a very hard problem. Even when the protocol specification is given as a finite state diagram it is difficult to verify whether a certain protocol behaves according to that diagram. The problem becomes harder when Petri nets are used to specify the protocols. Due to the inherent parallelism that can be expressed in this formalism the verification problem can only be solved in certain "neat" sub-cases. This is certainly an important practical issue for open agent systems. When agents agree to use a certain protocol for their interaction one wants to be sure they both implement the protocol in the same way and the resulting interaction is conform to the specification! This is one of the cornerstones of standardization in industry and should also be a main concern for agent interactions.

2.4. Ontologies

An issue that is closely related to ACL semantics is the proper treatment of ontologies in an ACL. Both FIPA-ACL and KQML include an element that is used to identify the source of the vocabulary used in the message content. This is designed to make these ACLs independent of particular application vocabularies, and to give the message recipient a way to interpret the nonlogical terms in the message content. In the original specification of KQML, this element was designed to refer to an ontology specified in Ontolingua. In FIPA-ACL, the semantics of the ontology tag is effectively user-defined.

Obviously, an ontology that has a *broad coverage*, *relevant* to its domain, and *extensible*, is needed in many ACLs. Thus, an ontology should exhibit a "coverage" of its domain in order to allow multiple agents to share knowledge in several contexts. It is however very important to keep in mind that a broad coverage can lead to a "voluminous" ontology and then agents can spend much time to find the meaning of contents of their ACL instead of interacting with others. A good ontology should also be extensible in order to allow designers to add new elements. Finally, an ontology must be domain-dependent and its taxonomy and relationships should show clearly their relevance to that domain.

The way that an agent would make use of the KQML or FIPA-ACL ontology specification to interpret unfamiliar parts of an ACL message has never been precisely defined. Merely supplying an ontology tag does not solve the problem of how agents acquire and use the common ontological knowledge base that is a prerequisite for successful communication. This is a particularly acute problem in open systems that include agents based in different organizations. The problems associated with learning meanings and reasoning with a new set of terminology are very similar to those in the area of database integration and cooperative information systems: somehow the ontologies that the different agents use have to be "integrated." Of course, ontological integration does not mean that the terminological structures actually have to be unified, but at minimum there must exist "translation rules" that convert relevant terms from one ontology into the other. Although a human working with representatives of each terminological community can often hash out a satisfactory set of rules, it is almost impossible to construct these rules fully automatically (Klusch 1999;

Papazoglou 1997). Consequently, agents can only fully communicate if they already share a common ontology, or if a set of preexisting translation rules is provided.

Although this may seem very restrictive, it has not been so disastrous in reality. For example, standards for product descriptions are very common in trade groups. And, in many open systems the agents communicate initially through some third party that initiates the contact between the agents. This third party will often mandate an ontology that all agents will use, and ontologies thus mandated will typically be built into the agents by their developers. This is the case in most electronic auctions, where selling agents include specialized code to specify their product using predefined terminology. Nevertheless, the general ontological problem is still the subject of active research.

2.5. Completeness of ACL Message Types

When heterogeneous agents interact by means of an ACL, the meaning of such exchange is characterized by *communicative acts*. According to the philosophy of language, *all* these acts fall into one of the following categories:

- *Representatives* or *assertives*, which represent a state of affairs,
  e.g., statements such as: the file is empty;
- *directives*, which order or ask the hearer to do something,
  (1) orders such as: close the file, or
  (2) queries such as: can you check the file?;
- *commissives*, which lead the speaker to commit herself to doing something,
  e.g., promises such as: I will check the file;
- *expressives*, which express a certain psychological state,
  e.g., giving congratulations or expressing emotional states such as: I am afraid that all files would be infected;
- *declaratives*, which bring something about in the world,
  e.g., a declaration such as: I name this file READ-ME;
- *permissives*, which give permission for act,
  e.g., permissions such as: you may delete the file TITI;
- *prohibitives*, which ban some act,
  e.g., prohibitions such as: you may not open the file TITI.

Should all these categories also be covered by communication between agents? Or can they suffice with only a few of these categories? When can we say that the set of agent message types is "complete"? Because ACLs can be used in arbitrary communicative contexts, one important goal is that their basic set of message types be sufficient to express all possible kinds of communicative intent that are allowed by the underlying theory of agency. Without a complete message set, agents and their developers may find themselves in situations where they are forced to invent additional *ad hoc* meanings for certain ACL messages, with the attendant decline in interoperability. Thus for instance, KQML and FIPA-ACL have limited coverage since all primitives are either assertives or directives. So, at least the important category of commisives, which are used to convey commitments to a course of action, are missing. In FIPA-ACL, we can simulate commissives using other performatives. However, it will be difficult to get the exact semantics that we expect for commissives. This means that these popular ACLs are incapable of expressing all agent intentions that are possible in powerful theories of agency, because several classes of performatives are absent from both. The practical effect of these omissions is limited because both KQML and FIPA-ACL are extensible ACLs: users are free to invent new application-specific performatives as long as they do not overlap or clash with the standard set. However, the semantics of these new performatives can be defined differently by different groups and thus has as consequence the development of different incompatible dialects of these ACLs.

2.6. Conversation Policies

In the preceding sections we have discussed ACL research issues that are primarily related to the generation and interpretation of individual ACL messages. A final topic we will address is how to bridge the gap between these individual messages and the extended message sequences, or *conversations*, that arise between agents. As part of its program code, every agent must implement tractable decision procedures that allow an agent to select and produce ACL messages that are appropriate to its intentions. This is not purely a problem of matching ACL semantics to agent intention: except in the most limited of agent systems, these decision procedures must also take into consideration the context of prior ACL messages and other agent events. Paradoxically, taking this context into account can actually *simplify* the computational complexity of ACL message selection for an agent. By engaging in preplanned or stereotypical conversations, much of the search space of possible agent responses can be eliminated, while still being consistent with the ACL semantics. The specification of these conversations is accomplished via *conversation policies* (CPs).

Because of this computational advantage, virtually all multi-agent systems employ some type of explicit or implicit conversational layer.[4] Theory has lagged behind practice in this area. Unlike research in ACL semantics, work on formal accounts of agent conversation remains in its infancy. Terminology and theoretical approaches are still being worked out, formal approaches and metrics are still fairly unsettled, and the role of research in natural language pragmatics and discourse theory is still being evaluated. The wide variety of approaches that are discussed in this special issue testify to the exploratory nature of the research. The theory tries to find a middle ground between completely fixed protocols (like those used in distributed systems) and using some high level rules that can generate protocols on the fly. Completely fixed protocols are usually too rigid to be used in an MAS environment or they get too complex (taking into account every possible exception that might occur). However, generating every next step in a protocol based on the present situation is highly computational intensive and therefore not practical for most agent implementations. In the article by Kumar et al. in this special issue, a way between a fixed protocol and computation of each step is proposed that might lead the way forward. In this introduction we will not try to formulate how a complete theory for conversation policies should look, but simply introduce some of the questions that are central to research in this field, and that agents researchers hope to answer in the coming years.

Possibly the overriding theoretical question in the field concerns the linkage between the ACL's semantic theory and its account of conversation. On the one hand, it seems obvious that large-scale properties of agent conversations, such as overall information flow and the establishment of commitments, are a consequence of the individual meanings of the messages that make up the conversation. In this view, the ACL semantics is primary, and every conversational property logically derives from the composition of some collection of semantic properties of the individual messages and their sequence. On the other hand, there is a significant thread of research that takes conversational sequences themselves to be semantically primitive, and the precise meaning of the individual messages is derived through their role in the overall conversation. In this view, the conversational semantics are primary, and because

[4] The original KQML specification had only token support for agent conversations; this shortcoming was noticed as early as 1995 by Finin and has been largely corrected using a variety of methods.

of the dependence of ACL semantics on context, the same message might have slightly different meanings when used in the context of different agent conversations. In the context of this view precisely, an ACL is viewed as a sort of conversation between software agents and not as a set of speech acts. Its semantics is the semantics of a conversation and it cannot be reduced to the conjunction or composition of semantics of its speech acts (Vongkasem and Chaib-draa 2000). Whether one takes conversational semantics or ACL semantics as being the starting point will affect the answers to several definitional questions, such as:

- What exactly is (and is not) a conversation policy? What important properties of agent interaction should CPs capture?
- How are CPs individualized? When are two agent conversations instances of the same policy? Are there interesting equivalence classes of CPs? Is there a useful type hierarchy of CPs?
- How should CPs be extended? How should they be composed?, etc.

Once one has settled on a basic theoretical perspective on the linkage between CPs and ACL semantics, then there are still a number of technical questions that remain. For example, there are several non-equivalent candidates for a CP specification language, ranging from transition nets like finite state machines and Petri nets, to various types of logic-based specifications, subgoal trees, and network protocol specification systems. These formalisms vary widely in the degree of conversational rigidity they entail, the models of concurrency they support, the computational complexity of their execution, and the availability of tools and techniques to help verify the analytical properties of the represented conversation policy.

Finally, the use of conversation policies to guide agent communication behavior engenders a host of practical questions. How should conversation policies be implemented in agent systems? Should CPs be downloaded from a common library, prebuilt into the agent's program, or derived from conversational axioms at runtime? How can conversation policies be negotiated, and unfamiliar policies learned? And finally, how can conversation policies be integrated with the other policies, plans, and rules that define an agent's behavior?

3. ABOUT THIS ISSUE

As the foregoing has indicated, research in agent communication can be placed along a spectrum of maturity, with research in content languages and network protocols being the most developed, research in ACL semantics being somewhere in the middle, and research on conversation policies and social agency being the most exploratory. The articles in this issue clearly reflect this diversity.

- *M. Gaspari* **extends existing ACLs** by some interesting work which pays particular attention to issues of naming and connectivity for a dynamic system of agents (i.e., a system where agents can be created, cloned, and terminated). For example it enables the agents control over name transmission in an MAS so that only certain agents know the names of other agents, and agents have control over the names they reveal to other agents. The proposed system uses a distributed service where each agent contains a facilitator which will resolve agent addresses for anonymous multicasting. Finally, the author presents his **semantics** which maps his ACL into a message passing architecture based on the actor model. The architecture is expressed by means of an algebra of actors which provides rigorous mechanisms to describe how the connections of the underlying agent architecture evolve. Notice that this work provides the basis for discussing dynamic primitives in ACL and for studying properties of dynamic multi-agent systems, for example concerning the behavior of agents and the security of their conversation policies.

- *R. A. Flores and R. C. Kremer* propose an ACL in which semantics are given using **social agency** (precisely, social commitments) which are more easily verified. More precisely, the authors distinguish between conversation policies and conversation protocols. Then, they view policies as a set of inference rules and protocols as sequences of communicative acts. The authors argue for using **conversation policies** rather than conversation protocols because of the flexibility offered by policies (the agents are not constrained to use specific communicative acts in the specified sequence and so policies support emergent conversations). To this end, they propose a basic set of speech acts that enable agents to negotiate social commitments and specify a number of (conversational commitment and social commitment) policies that should govern agent interactions during the negotiation process. They use the Z notation to specify conversation policy formally. They claim that this notation allows designers to check the type of specification and to prove certain properties from the specification.
- *S. Kumar, M. J. Huber, P. R. Cohen and D. R. McGee* propose a formalism for **conversation policies** using concepts from **social agency** and particularly the social concept of joint intention. A central theme of this work has been to formally treat conversation policies (called in their approach: conversation protocols) as joint actions by representing them as joint action expressions and applying joint intention theory to those expressions. The key idea defended in this article is that protocols are meant to achieve certain tasks, that is, they have a goal. Starting from this observation, the authors identify the landmarks or the state of affairs that must be brought about during the execution of a protocol in order to achieve its goal. Therefore the important aspect of protocols are these landmarks rather than the communicative acts needed to achieve the landmarks. Kumar and his colleagues show that families of conversation protocols can be expressed formally as partially ordered landmarks where each landmark is characterized by propositions that are true in the state represented by that landmark. In this context, they treat conversation protocols as joint action expressions and apply the joint intention theory to protocols and their compositions. Finally, they give a formal semantics to group communicative acts and use it to handle group communication in a formal treatment of protocols.
- *C. Reed, T. Norman, and N. R. Jennings* make a number of thought-provoking and valid points concerning the **semantics** of ACLs. The specific issue addressed is that there is no consensus on the core primitives across the leading ACL proposals, despite continuing efforts at standardization. The authors find drawbacks with other approaches, namely those in which the language is generated on the fly by agents or in which agents infer more complex composed primitives from a core set that is shared. The article offers a hybrid approach, namely an approach that allows agents to negotiate the meaning of the primitives they will use, in the context of a particular task. The work puts forward as its contributions (a) the notion of a semantic space and a "systematic framework of defining primitives within it" and (b) a voting mechanism by which agents can engage in the proposed negotiation about the meaning of communication primitives. This is an interesting effort toward the establishment of systems of agent communication which would not be based on fixed and previously shared protocols. It is indeed hard to imagine how there could ever be a worldwide consensus about the ontologies and associated languages for every possible domain of multiagent systems. As multiagent systems are typically open systems, new needs can arise after a convention has been agreed upon.

Readers interested in other facets of ACLs can refer to Dignum and Greaves' book (2000). The contributions in this book were selected from the workshop on *Specifying and*

Implementing Conversation Policies and the workshop on *Agent Communication Languages: From Speech Acts to Conversations*. This joint book covered most of the research in the field of agent communication. The book contains four themes: semantical aspects of agent communication, conversation policy descriptions, fundamental issues surrounding agent conversations, and the relation between agent communication and general agent task planning issues.

REFERENCES

COHEN, P. R., and H. J. LEVESQUE. 1995. Communicative actions for artificial agents. *In* Proceedings of the International Conference on Multi-Agent Systems (ICMAS-95), pp. 65–72.

DIGNUM, F., and M. GREAVES. 2000. Issues in Agent Communication, LNAI 1916. Springer-Verlag, Berlin.

FININ, T. Y. LABROU, and J. MAYFIELD. 1995. KQML as an agent communication language. *In* Software Agents. *Edited by* J. Bradshaw. The MIT Press, Cambridge, MA.

GEORGEFF, M., and A. RAO. 1998. Rational software agents: from theory to practice. *In* Agent Technology. *Edited by* N. J. Jennings and M. J. Wooldridge. Springer Unicom, pp. 139–160.

GRICE, H. P. 1974. Logic and Conversation. *In* Syntax and Semantics, vol. 3, Speech Acts. *Edited by* P. Cole and J. Morgan. Academic Press, New York, pp. 41–58.

KONE, M. T., A. SHIMAZU, and T. NAKAJIMA. 2000. The state of the art in agent communication languages. Knowledge and Information Systems, **2**:258–284.

KLUSCH, M. (Editor). 1999. Intelligent Information Agents. Springer-Verlag, Berlin.

LABROU, Y. 1997. Semantics for Agent Communication Languages, PhD thesis, University of Maryland, MD.

LABROU, Y., T. FININ, and Y. PENG. 1999. Agent communication languages: the current landscape. IEEE Intelligent Systems, **March/April**:45–52.

MOORE, R. 1985. A formal theory of knowledge and action. *In* Formal Theories of the Commonsense World. *Edited by* J. Hobbs and R. Moore. ABLEX Publishing, Norwood, NJ.

PAPAZOLGOU, M., and G. SCHLAGETER (Editors). 1997. Cooperative Information Systems: Current Trends and Directions. Academic Press, New York.

PITT, J., and A. MAMDANI. 2000. Some legal aspects of inter-agent communication: from the sincerity condition to 'ethical' agents. *In* Issues in Agent Communication, LNAI 1916. *Edited by* F. Dignum and M. Greaves. Springer-Verlag, Berlin, pp. 46–62.

SADEK, M. D. 1991. Attitudes Mentales et Interaction Rationelle: Vers une Theorie Formelle de la Communication. PhD thesis, Université de Rennes I, France.

SEARLE, J. R. 1969. Speech Acts. Cambridge University Press.

SEARLE, J. R., and D. VANDERVEKEN. 1985. Foundation of Illocutionary Logic. Cambridge University Press.

SINGH, M. P. 1998. Toward a model theory of actions: how agent do it in branching time. Computational Intelligence, **14**(3):287–305.

SINGH, M. P. 2000. A social sematics for agent communication languages. *In* Issues in Agent Communication, LNAI 1916. *Edited by* F. Dignum and M. Greaves. Springer-Verlag, Berlin, pp. 31–45.

VONGKASEM, L., and B. CHAIB-DRAA. 2000. ACL as a joint project between participants: a preliminary report. *In* Issues in Agent Communication, LNAI 1916. *Edited by* F. Dignum and M. Greaves. Springer-Verlag, Berlin.

WOOLDRIDGE, M. 2000. Semantic issues in the verification of agent communication languages. Journal of Autonomous Agents and Multi-Agent Systems, **3**(1):9–31.