Agent Communication and Cooperative Information Agents

Frank Dignum

Faculty of Mathematics & Computer Science,
Eindhoven University of Technology
The Netherlands
e-mail: dignum@win.tue.nl

Abstract. In any multi-agent system the coordination between the agents is an important element that determines to a large extent the effectiveness of the system. One way to facilitate the coordination between the agents are agent communication languages. In this paper we will present a short introduction to agent communication languages in general and will indicate the main topics of interest in agent communication research. We will relate these developments to the characteristics of cooperative information agents in particular. This leads to a number of short term goals for the development of agent communication between cooperative information agents, which are for some part very practical and also deviate in emphasis from agent communication in general.

1 Introduction

Agent technology is an exciting and important new way to create complex software systems. Agents blend many of the classical properties of AI programs - autonomy, adaptivity, pro-activeness, goal-directedness, and so forth - with insights gained from distributed computing, negotiation and teamwork theory, and the social sciences. This blend promises programs that continuously adjust their behavior to balance changes in their environment, collaborating with other agents to satisfy both their internal goals and the external demands generated by virtue of their participation in agent society. And, although it is possible to design agent programs that are completely unaware of each other, much of the power of the agent approach derives from the principle that agents (like humans) can function more effectively in societies characterized by division of labor and cooperation in service of shared goals. In this paper we will study the communication that has to take place in order for agents to cooperate. First we will look at agent communication in general. Then we will indicate which parts are particularly relevant for cooperative information agents and what extra requirements might hold for this situation.

Agents typically employ specialized *Agent Communication Languages (ACLs)* in order to support the types of multiagent interaction alluded to above.[1] Formally, ACLs

[1] Agents can also communicate with actions other than classically linguistic productions, simply by making observable changes in the environment that have semantic force. For example, an agent that locks a resource for itself might be assumed to communicating to an observer its

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.
Although the above gives some indication about the requirements for ACLs, a term like "high-level" can be interpreted in many ways. The way it is meant here can be made most clear by refering to the ISO layers that can be distinguished in communication. On the lowest levels one talks about the hardware and the routing software. In top of that there are message transport protocols such as TCP/IP, HTTP, or IIOP. They deal with communication protocols at the level of message transport. In top of these protocols one can use standards like CORBA that regulates the exchange of data between programs by providing standard formats. ACLs are situated on the level above the data exchange level. ACL's are dealing with communication on the mental and social level. This means that a message in an ACL indicates the purpose for which it is send. E.g. whether it is a request for information or a request to perform a task. Typical ACLs also have a characteristic semantics (see below) that is far more complex than standard distributed object protocols. The formal elements of an ACL must be carefully structured to balance the communicative needs of the agent with the ability of receivers to derive (ideally in tractable time) the intended meaning of the message. Further, it is important that the syntax, semantics, and pragmatics of the ACL be as precise and explicit as possible, so that the resultant agent systems can be as open and accessible to developers beyond the original group.

This last point bears some emphasis. Historically, many multiagent systems have been built using somewhat *ad-hoc* and developer-private communication mechanisms. 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 generally true of arbitrary agent collections. These range from the general assumption of a shared ontology and problem domain to specific nonstandard meanings for messages (or the absence of a message) in particular contexts. Often domain specific solutions are implemented for reasons of efficiency. However, knowledge of these often-undocumented assumptions 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 are extremely difficult to generalize and extend without extensive interaction with the original developers. The locus of this problem can be traced to these implicit assumptions in the agent interaction design. We will come back to this point later on when discussing ACLs in the context of cooperative information agents.

Although most ACL's described in the literature are very generic, they all make some assumptions about the agents and/or the environment. It is important to note that it is hardly possible to device an ACL without any assumption at all about the agents, but it is equally important to make those assumptions explicit! The combination of explicitness and generality leads to extremely expressive languages with well-defined

---

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.

semantics that are grounded in powerful logics. These ACLs demand a great deal from the agents that must interpret them: computing the meaning of an arbitrary message may require extensive deductive machinery, and can even be formally intractable. However, using a strong ACL with a semantics that can be precisely known in advance is a great advantage when creating heterogeneous agent systems that are designed to be easily extendible and broadly useful. One of the challenges for ACL research at this time is to find the right balance between richness and generality of the language and computational efficiency of the communication.

In the next section we will introduce some of the current available ACLs and some of the current research topics. In section 3 we discuss the characteristics of CIA systems in the light of their communication requirements. In section 4 we will indicate some consequences for the development of ACLs for this domain. In section 5 we discuss some future developments and expectations. We conclude the paper in section 6.

2 Agent communication

A first attempt to produce an ACL that was both standard and general came out of the DARPA knowledge sharing initiative. The Knowledge Query and Manipulation Language (KQML)[8] was originally devised as a means for exchanging information between heterogeneous knowledge systems. However, because of the generality of its high level primitives and its message orientated structure, KQML also functions well as language for agent communication. KQML is the most widely implemented and used ACL in the agents community. The messages are based on the theory of Speech Acts [1, 11]. The main innovation of this theory was to take communication not just as an exchange of information, but to see the utterance of a sentence as an action in itself. These actions (the communication) thus also have a purpose and an (intended) effect, etc. just like other actions. This viewpoint relates very closely to the way communication between agents is viewed. At this level it is not just the exchange of messages that counts, but the purpose for which they are exchanged. For instance, whether a message is a request or an inform or an order. These elements are needed if an agent has to initiate a conversation. In that case it has to determine how the communication is going to fulfil some goal of its own. This is only possible if the agent can reason about the expected effects of the communication. E.g. if the agent wants to have certain information and it expects that another agent has the information, than it can request the other agent to send the information. Note that the main novelty here over more traditional information exchanges is that the agent reasons about the information that it needs and the way that it can get it. There is no standard in-built procedure that forces the agents to exchange all their information periodically or on the cue of some external user.

Basically speech acts consist of three elements: the illocution of the message, the context of the message and the content of the message. The illocution determines the purpose of the message. E.g. a request, an inform, an order, a commitment, etc. The context of the message determines who are the sender and receiver of the message, whether the message is a reply to another message, the conversation that the message is part of, the language that is used to describe the content, the time of the sending of the message, etc. Finally, the content of the message describes what the message is about.

These elements are all incorporated in most ACLs as well. The following is a typical message in KQML format:

```
(tell :sender       CDagent
      :receiver     Dignumagent
      :in-reply-to  id9100.145
      :ontology     EC.CD
      :language     Prolog
      :content      "price(cd263956,20)")
```

It states that the CDagent replies to a message from Dignumagent telling the price of a particular CD. It uses Prolog to describe the content and a particular ontology (EC.CD) that clarifies what ”cd263956” stands for and also what the currency of the ”20” is taken to be.

The separation of the content of the message from the other elements of the message makes it easy to exchange messages without being tied to a specific formalism to describe the contents. Agents only have to agree upon the illocutions that can be used and know which fields can be present in the message (i.e. which elements of the context are specified) to process the messages. What an agent does with the content of a message that it receives does not have to be known by the sending agent. It can be different for every agent. However, the message handling mechanism can be uniform for all agents. See [9] for a more thorough overview of KQML.

More recently, the Foundation for Intelligent Physical Agents (FIPA) has proposed its own ACL, FIPA-ACL, to remedy some perceived weaknesses of KQML. The FIPA-ACL is also based on speech act theory and the syntax of the messages does not differ widely from that of KQML. Although this is not the place to make an extensive comparison between KQML and FIPA-ACL, we would like to point out a few of the main differences that indicate why FIPA decided to use a new ACL of its own.

The first difference is that FIPA-ACL is heavily based on a multi-modal logical semantics for communication. The advantage is that it faciltates a precise formal description of the communication and its effects. The disadvantage is that agents are usually not equiped with the logical abilities that would be needed to actually act according to this multi-modal logics. In [10] some more concerns on the semantics of the FIPA-ACL are given. One of their main points is that the semantics is based for a large part on believes of agents, which in one hand are unknown for other agents and in the other hand can change quickly and easily. This makes many of the preconditions on communication either difficult to check or easy to fulfil, which was not the intended purpose of these pre-conditions. One famous example is the sincerity condition, which states that agents only tell what they believe in themselves. Of course, it is usually impossible for another agent to check what an agent believes in. Also when an agent really wants to tell something it does not believe in it can add the belief temporarily and retract it again after the message is sent.

KQML does have a formal specification, but this one is much simpler and puts less burden on the agents. The disadvantage being that some messages still have an ambiguous meaning.

Another big difference is that KQML has a number of special administrative and networking messages. E.g. to ”register” an agent in a kind of yellow-guide. These messages facilitate finding other agnts that an agent wants to communicate with. In FIPA-ACL they have to be performed using the ”normal” messages. Although this makes the ACL appear cleaner, it also moves a number of problems to a lower level, where they have to be solved and of which little can be seen (and changed!) on the top level.

A last difference worth mentioning is the way new message types can be defined. In FIPA-ACL this can only be done by combining existing messages using a few well-defined operators. This has the advantage that all messages have a well-defined semantics. In KQML it is possible to define new message types at will. This makes it easy to define some message types that are useful for a specific domain and can be implemented efficiently. The disadvantage is that the new messages do not necessarily have a well-defined semantics and also many ”dialects” might arise.

It is worthwhile to point out an important area of agent communication and ACL use that is usually not mentioned. 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, and other standard distributed communications issues. In practice, implemented agent systems have used both centralized strategies that handle all aspects of messaging between agents (implemented in KQML through the introduction of a ”facilitator” agent), as well as more decentralized systems that devolve this functionality to the communication handlers of the agents themselves (as is done in the FIPA-ACL). Agent systems exhibit little standardization in this area. Further, implementations of ACLs often expose more of the details of this infrastructure than one might strictly like. However, the problems of designing a general agent communications infrastructure that is efficient and reliable for different communication topologies and networking systems are deep and difficult.

In the following subsection we will introduce a set of issues that have been important in the development of ACLs and agent communication theory generally.

2.1 Issues in Agent Communication

Although in this section we often 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. The topics that are presented in this section are taken from [3] which discusses current issues in agent communication in more detail.

semantics of ACL messages

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. These theories are usually based on logics of mental attitudes such as beliefs, goals and intentions. A complete theory of agency also includes things like the agent's general reasoning strategy and (perhaps resource-aware) 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 which compose a theory of agency include Moore's accounts of knowledge and action, Rao and Georgeff's BDI architectures, Linder's KARO system including abilities and opportunities, Cohen and Levesque's intention theories, 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 speech act theory as developed in Searle [1, 11] and Grice [4]. Communicative acts are performed in service of intentions in the same way as any other act. Unlike physical acts, however, the primary effect of a communicative act is to change the mental attitudes of the parties involved in the communication.[2]

The current semantic theory of FIPA-ACL requires that the theory of agency supply a set of BDI-style primitives. The semantics of FIPA-ACL is based on mentalistic notions such as belief and intention, and (because of the speech-act theory component) takes messaging to be a type of action. This approach has, in turn, led to the requirement that FIPA-ACL's semantics be expressed in an extremely powerful quantified multi-modal logic involving both belief and intention 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 (but not necessarily implement) 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 fairly permissive. As is now well know, this permissiveness allowed wide latitude in KQML implementations, and the proliferation of different and incompatible KQML dialects. Labrou's second-generation semantics for KQML ([8]) struck a clever compromise between KQML's original virtual knowledge base theory of agency and a BDI-style theory, but at the cost of introducing modal operators back in the semantic theory, with the associated computability issues.

[2] Certain special classes of acts, parallel to the *explicit performatives* of natural language, can also in some circumstances directly change the properties of objects in the world. For example, a priest can marry a couple by simply pronouncing them husband and wife. In the agent world, there are examples of this when agents register themselves with a yellow-guide or when they give other agents some access rights on a database they manage.

Mismatches between the theory of agency and the semantic theory can occur when the theory of agency licenses communicative actions that are not accounted for by the semantics. The *sincerity* condition on agent communicative behavior 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 electronic commerce applications and adversarial negotiations generally; for example, the locally optimal behavior of an agent might involve deceiving another agent about its 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 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. However, because the semantic and pragmatic foundations of human communication allow for the possibility of deception, our natural language semantics allows for this as well.

The sincerity condition thus 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 interagent communication channels are error-free. Agent systems routinely assume that all messages eventually arrive to 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.

Ontologies

Both FIPA-ACL and KQML include a field in the message structure in which an ontology identifier can be given. The ontology identifier is typically a string which can be used to identify the source of the vocabulary used in the message. This is designed to make these languages independent of particular application vocabularies. However, 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 which include agents based in different organizations. The problems associated with learning meanings and reasoning with a new set of terminology are the same as have been addressed in the area of database integration and cooperative information systems for several years as well.

Somehow the ontologies that the different organizations use have to be "integrated." Of course, it does not mean that ontologies have to be actually unified, but at least translation rules should be defined to 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 design these rules fully automatically. Consequently, agents can only really communicate if they already have a common ontology, or if a set of preexisting translation rules are provided. Although this seems very restrictive, it is not so disastrous in reality. E.g., in (international) trade it is very common to have standards for product descriptions. In the same way, in many open systems the agents communicate initially through some third party that initiates the contact between the agents. This third party may impose an ontology on all agents. This is for instance the case in electronic auctions, where selling agents have to specify their product using predefined forms.

Finally, although the use of ontology tags makes an ACL much more flexible, specifying the ontology and content completely separate from other message components, like the performative used, also makes it impossible to specify any constraints that involve both performatives and contents. E.g., an agent might request for an action to be performed before a time that has already passed when the message arrives. One could say that the message is thus not complying to the general rules for requests for actions. However, this can only be detected after the contents of the message are parsed and the ontology used to determine its meaning.
Another point is related to a problem mentioned by Cohen and Levesque in [2]. It is casued by the nesting of KQML messages in the content of other KQML messages. In general this nesting is allowed in KQML except for the "deny" message. A nesting within the deny message of for instance a tell message would lead to the question whether the content of the tell message is now denied or whether the action of telling it is denied. However, the problem does not disappear by just prohibiting KQML as language for the deny message. The content of the message can be expressed in Prolog or first order logic or something else, but still express a tell action. Then the problem of the exact meaning of the deny remains.

These are only two examples that show that it is sometimes difficult to give a precise meaning of message types without knowing anything about the possible content of the message. Also it is often difficult or impossible to check the preconditions that should hold for a message to be sent if the content cannot be checked against the precondition. So, the above leads to the fact that some rules range over both content of the message as well as over the performative (or other message components). It shows that it is not possible to determine the meaning (and correctness) of some message types completely independent of the message content. Therefore, using explicit ontologies and ontology tags in a message can help to abstract away a bit from the actual contents, but not completely (as some people would like to believe).

Standard set of message types
An issue that also comes up in many other communication standards is whether it is possible to come up with a basic set of message types (usually equated with performatives) that can be used to express all types of communication. With the intuitive idea that this set is "minimal" and "disjunct". I.e. the set should not contain redundant messages that could be expressed by other messages already and the meaning of messages should not overlap (too much). An interesting example in this respect is the X.12 EDI standard. It started with a limited set of messages (which worked in most cases), but has been repeatedly extended over the years every time some situation occurred that could not be easily captured with the existing message types. The main point was that there was a lot of attention for a precise and complete syntax description, but there was only a very informal semantics for the messages. Also they started with the message types that seemed to occur most in practice, without checking whether these message types had overlapping meanings, etc. This practical but unprincipled approach makes it

very difficult at the moment, even for humans, to choose which message type to use in a particular case.

A similar case can be made for KQML. Although there is a basic set of performatives that seem to cover most aspects of communication, there is no justification of this basic set. Also, a more close inspection of the performatives shows that all performatives are either directives or assertions (in the classification used by Searle [11]). There are no performatives to express an agent's commitment (promise or something like that). This type of performatives also is absent in the FIPA ACL specification!

KQML is explicitly designed in a way that users can easily extend the basic set of performatives with their own performatives. Although this is very practical, it also leads to many situations where the designers of a multi-agent system define their own set of performatives. Although agents from this system might be able to communicate with agents of a different system using KQML, they cannot "understand" each other anymore, because they use completely different performatives.

There is one more point to be made on the syntax of the messages in ACLs. We already remarked that existing ACLs seem to have an ad-hoc set of performatives. Also the possible values of other arguments in the messages do not seem to be determined in a very principled way. One example that came up recently is the use of message identifiers. There is a simple way to refer to the message and conversation for which the present message is a reaction. In both KQML and FIPA ACL this can be done using an identifier (characterstring or number). However, it is not very well possible to parameterize this field or to refer to a message from another conversation. These kinds of references are quite common in multi-party conversations. So, also on the level of the syntax of the individual arguments of messages some improvements should be made and some more research is needed to find out which are the real needs for each argument for each type of application.

Semantics and Conversation Policies

The final issue we will address involves the theoretical framework necessary to chain individual messages together into semantically coherent agent conversations. The issues here are complex, involving binding together the semantics and pragmatics of messages.

A common way of specifying the semantics of actions involves defining the action's preconditions and postconditions. For example, the semantics of $x = 2.1$ might be expressed by the precondition that x should be the name of a location able to store a value of type real. The postcondition of this action is that the value stored in the location denoted by x is 2.1. For communicative acts, the preconditions and postconditions are typically expressed in terms of mental attitudes of the involved agents. For example, the precondition of KQML's TELL message states that the sender should believe 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. The FIPA-ACL semantics specifies similar preconditions and postconditions for its INFORM message type.

Although the above descriptions give a fair approximation of the intended meaning of these message types, it is impossible to specify a complete semantics. One of the problems is the distinction between the actual effect and the intended effect of a speech act. Whenever an agent tells some information to another agent it does that with the intention that the other agent will at least believe that the sending agent believes that information. However, an agent can never directly change the beliefs of another agent. Therefore the agent does not have complete control over the actual effect of the speech act.

Another point is that an agent will perform a certain communicative act with a certain expectation of getting a reaction of the receiving agent. For instance, a request for information is send with the expectation that the other agent will give that information or tell that it does not have that information (or does not want to give that information). This presupposes some (subtle) social relations between the agents. Just as any other action, the communicative acts also have to be planned by an agent. The only reason for an agent to plan a request is if it has a reasonable expectation that it will get some (valuable) reaction. Otherwise the performance of the request would be useless. However, this precondition is usually not expressed in the speech acts. Also the fact that after the speech act the sending agent is having some expectation is not easily expressed. In the same way there arises some kind of obligation for the receiving agent to react, which is difficult to express.

The above shows the social aspect of communication. It is an aspect that has hardly been addressed until now. One of the arguments is that agent societies do not have to have the same aspects as human societies. Therefore these social aspects are not important. However, it usually means that the designers of multi-agent systems take some social attitudes of agents for granted and built them into the agent architecture, while designers of other systems might not share the same values and capture different social conventions. Again this can lead to big misunderstandings between agents in open systems, because conventions are different and it is not clear that they differ.

In my opinion more research is needed into the fundamental concepts that play a role in communication. This includes concepts such as “commitment”, “obligation”, “convention”, “power” (in the sense of hierarchical relations), etc. If there is a clear idea about which concepts are involved it becomes possible to define the semantics of communication in terms of these concepts.

In the above sections we have discussed research issues related to single messages. A last (big) issue that we want to point out is how to get from individual messages to conversations. As already mentioned in the previous section, messages hardly ever are send in isolation. For the description of communication protocols several techniques have been used. For agent communication often Finite State Diagrams have been used, but lately the use of (colored) Petri-Nets has been advocated. The most important use of these techniques is to describe the possible sequences of messages. The use of Petri-Nets gives the additional possibility to check e.g. for deadlocks and other properties of the protocol.

Although these techniques work fine for simple protocols, the figures they produce for real-life protocols are hardly readable. One of the main problems is that at any point in the protocol many exceptions can arise. Each of these exceptions has to be modeled. This causes one to loose track of the main flow of the protocol very easily.
Another disadvantage of these techniques is that they do not fix any relation between the semantics of the individual messages and the order in which they might appear in

the protocols. E.g. if an agent asks information about a product it will never expect to get as answer the identical question. Although it is possible to describe all these types of restrictions they are not inherent in the techniques with which the protocols are described (as one would like).
Of course we also have to decide what kind of properties we would like to prove for conversations and how these properties can be expressed in the description formalism. It seems that the properties and semantics we use for conversations are quite different from the properties of the protocols for which most of the existing formalisms have been developed. Therefore none of these formalisms is a natural "best candidate".

Another important point (first raised in [12]) is how one can verify compliance to a protocol. Suppose an agent has to use the Contract net protocol in order to delegate its tasks. It uses some mechanism to manage its communication. Is it possible to proof that this mechanism will lead to compliance of the Contract net protocol. Even if the mechanism and the Contract net protocol are both described using e.g. Petri-Nets it is not trivial to proof that the agent's protocol is equal to (part of) the Contract net protocol.

There are also more practical issues. Is it possible to define a basic library of conversation protocols that can be used for all types of communications? I.e. can we define a kind of conversation language? If this cannot be done, agents first have to find out which conversation protocol their partners use, before they can start a conversation. A similar discussion as for the use of ontologies also applies for conversation protocols. I.e., how to integrate different protocols, how to learn protocols, etc.

Of course, the issues mentioned in all the previous sections are not the only issues to be answered. We merely tried to shed some light on the state of the ACL research and highlight the most important categories and give a general impression of which topics deserve our attention in agent communication research. In the following section we will briefly describe our view on Cooperative Information Agents with an emphasis on their communication.

3 Cooperative Information Agents

Since 1997 a sequence of workshops have been held on the topic of cooperative information agents (see [5–7]). However, just like happened with software agents in general, no single definition of cooperative information agents has been adopted yet. There seem to be two basic approaches towards CIAs. The first one starts from the users and employs agents to get the right information to the user. In this perspective usually one finds a number of different types of agents (or agents with different roles). There are user-interface agents that learn about the needs of the user and present the information in a personalized way. There are information fusion agents that integrate the data from heterogeneous sources. And there are information gathering agents that try to find the most relevant data for the user and sometimes also play a role in maintaining that data.

The second perspective sees CIAs as a further development of cooperative information systems. In this perspective each agent is related with a certain (type of) data source and the agents can communicate to keep their data consistent and to integrate the information from different sources. Usually it is assumed that a user will approach one of the CIAs with an information request. This agent will then try to provide the answer by checking its own sources and requesting the information from the other agents.

Although the two perspectives are by no means exclusive they emphasize different topics. The first perspective emphasizes the different roles of the agents and looks more at information filtering and data mining (in order to find regularities in user groups) issues. Because the emphasis is on getting information for the user, there is often little attention for information management (updating and maintaining consistency of information). Sometimes this is forced by the fact that the information sources are not under the control of the system (like the WWW) and are by definition not manageable. The agents that play different roles usually also have different architectures geared to the role they play. (Often one can even question whether all the components are really agents or just complex objects.)
The second perspective emphasizes the data integration issues and discusses common ontologies for the heterogeneous data sources and agents. In this perspective one assumes that the data sources might be autonomous but can still be analyzed and e.g. a meta-data model can be defined for (at least some of) them.

As we said earlier it is hardly possible to give a general definition of CIAs let alone a general architecture. However, the fact that the agents are CIAs does have some consequences for their components. A first point is that the planning module that manages the tasks of the agent is usually not very complicated. The domain is relatively simple with respect to the number of possible types of tasks that can be performed and the way a goal is achieved is often quite straightforward. For instance, if the goal is to find information about a topic then the agent has to perform a query on a private information source or request the information from one or more other agents. Once the answers come back it might still have to perform an integration step, but the number of steps leading to the result is usually not more than 3 or 4. This is quite different from e.g. robots that have to move and react to a physical environment.
A second point that becomes obvious very quick is that most actions that are performed are communications with other agents or users. So, the communication component is important for this type of agents (which is not the same as saying that it has to be very complex!). We will come back to this component in the next section.
In most multi-agent systems the agents must have knowledge about the other agents in the system. This is also true for CIAs. In some systems it is important to have a thorough knowledge of all the capabilities of the agents, because they might differ widely. However, in CIA systems there are only a few possible roles and related capabilities. Therefore it is relatively easy to keep track of the capabilities of the agents. For CIAs the knowledge that they must have of other agents is concerned with the type of information that they can provide. This often leads to some kind of module that describes per agent which sources it can access and possibly some meta-data about these resources. For instance agent ”167” has access to the personnel data which contains names, addresses, ages and functions of all employees. As said before, a common ontology or rules to translate information are also an important factor in this respect.
A last component that we consider important for CIAs is a component that contains ”social” information on other agents. It can contain for instance information about which agent provides the best information on certain topics. But can also contain information

about contracts with other CIAs representing e.g. subscriptions to information. E.g. a company might consult reports of Forrester, but has to pay a certain amount for each report it consults. Also this component might contain obligations that the agent has towards other agents to supply information (both on request and pro-actively, as in subscriptions).

In the next section we will discuss in more detail what are the communication requirements for CIAs and which of the developments sketched on ACLs are (also) important for CIAs.

4 CIAs and communication

As with all multi-agent systems the complexity of the communication between the agents depends to a large extend on the autonomy of the agents. If the agents are designed to cooperate and thus many social conventions are fixed in all agents then the communication becomes less complex, because many things are already determined by the context. If the agents are operating in an open environment where they interact with agents of other organizations and agents can appear and disappear at will the ACL has to determine more elements of the context and thus is more complex c.q. rich.

For CIAs operating within one organization often the first situation arises. That is, the agents might be autonomous, but the social conventions are actually completely determined by the organizational structure and programmed in the agents. As a consequence there is e.g. no real difference between orders and requests. Also there is usually no need for an agent to commit itself to an action. Requests are always handled and agents are by default committed to requests from other agents. This leads in general to a need for much less performatives than e.g. provided in KQML.

Another aspect that is usually handled centrally is a common ontology for the agents. This can e.g. be arranged by the use of a thesaurus that is often already present in larger companies.

Taking the above into consideration makes it doubtful whether in many situations where the CIAs operate within one organization they have need for a general ACL which brings a lot of overhead and from which they only use a small part.

This situation changes radically when the CIAs operate across organizational boundaries. Even though the agents might still be cooperative in principle they will operate in a much more autonomous way. One can easily imagine situations where a request for information is send to multiple other agents from different organizations (e.g. to give information about the annual growth of e-commerce in Asia). Each of the agents might respond with a bid on how much it would cost to provide that information and how long it might take. Then one of the agents is selected and a commitment between the agents is established to get the information before a certain deadline and to pay for it.

These types of scenarios require a much richer communication language than the ones that play within one organization. It also brings up the question of a common ontology right away. In the above example the question is what is understood as being "e-commerce"? The definition of this term may have a large influence on the outcome of the request. Also one might question what is meant by Asia. Is it the geographical Asia or are actually the countries in the Pacific rim meant by the term?

It is clear that a general ACL is more appropriate in these situations than in the one where all agents operate within the same organization.

Which are the things that most CIAs do need in their communication? First of all they usually have a number of ways to ask and send information. They can ask for information once, or they can subscribe to changes in that information. They can ask for one instance of a certain type of information or all possible information at once. The agents can always send information on request or send it periodically when they perceive a need, etc.
In all cases the agents also have to keep track of the task to which a message is related. E.g. if an agent is subscribed to information about the stock exchange and also to general news bulletins. It should know where a certain message belongs when it informs about the crash on Wall Street, because this might determine what action it will take. So, although conversations in general have a very simple structure, they can continue over long periods of time and it is not always clear when they are finished!
Another point that is important for CIAs are the messages related to registration of agents and administration of the agents. Whether these messages should be incorporated as explicit message in the ACL or just in the content of standard messages is under debate. However, it is clear that it should be possible to easily add new agents that maintain new resources, have agents going off-line and coming back again, agents changing addresses, etc. Because some task related conversations (like subscriptions) are very long lived, these messages might happen in between the "normal" conversational messages. They should not only be processed properly at the right time, but also their effect should be calculated on existing conversations. E.g. if an agent that has a subscription goes off-line it is no use to send it more messages or there should be some mailbox for that agent to which the messages are subsequently send.

Of the topics that were mentioned in the previous section as being important for ACLs the one that seems most important for communication between CIAs is that of ontologies. In every setting CIAs are used to connect heterogeneous information sources. When the sources are all under control of one organization it is often possible to define a common ontology that can be used to exchange information. This will get harder when external sources (like the WWW or newsfeeds) are also connected to the system.

The second topic that was mentioned concerned the relationship between the semantics of the messages and the models of agency of the agents. In situations where the social conventions are already fixed and sometimes a common ontology can be used, this topic is of little interest. In fact the semantics of the messages is determined by the way the system is set up and the conventions are programmed into the agents. E.g. it is predetermined what the reaction will be to a request for information (nl. a telling of the information if it is known).
However, the more sophisticated the systems become and the more open they are the more important the precise semantics of the messages becomes. E.g. precondition of giving another agent some information is that an agent believes that that other agent does not know that information yet. Now, in case an agent repeats the same query after a short period it might be that it only wants to know the latest updates (i.e. the differences with the last answer to the query) and is not interested to get all information again. Also when the system is open it becomes more important to know exactly

what an agent expresses with certain messages. E.g. when it denies certain information (doesn't it have the information or does it believe the information to be false?).

Even within the limited domain of information systems we doubt whether there will appear a standard set of performatives for the messages before a more thorough theory has been developed to describe the semantics of the communication. Only such a theory will give a basis on which a standard set of performatives can be developed and against which it can be tested on completeness and disjunctiveness. At the moment performatives will arise from the needs in practice. The performatives that are needed depend for a large part on the type of application. In case the application is geared towards information retrieval it will develop many standard ways to query data. When the system is also used to maintain data it probably needs some performatives to control the data, to give authorities to agents to change data, etc. Just like in natural languages the consequence will be that there will be many overlapping performatives each suited for particular situations.

Finally, we have discussed agent conversations. As said before, for CIAs the conversations usually have a very simple structure. Therefore there is little need to prove that a conversation does not have deadlock or that it is guaranteed to end. These properties are checked easily by hand.
What is more interesting is how to interleave several conversations and how administrative messages can be interleaved with the conversations on information. Little or no research about these points has been conducted.

5 Future developments in communication between CIAs

In the previous section we have already listed a number of topics that will be of interest for communication between CIAs. In this section we will look at the prospects for the near and longer future.
As has been remarked by many people already, there is a discrepancy between agent theory and practice. The theory of agents is often concerned with logics that are developed to formally describe the (internal and external) behavior of agents and that try to give a semantics to it. In practice one is interested in systems that work (efficiently). This also holds for the communication part of agents.

In order for a standard ACL to be useful for communicating CIAs the following points should be addressed. First of all there should be a standard set of performatives that support the communication. Preferably this set should be as small as possible in order to facilitate efficient implementation. However, it should be possible to express all types of information requests easily with these performatives. Preferably the set of performatives would be constructed by surveying the performatives used in practical projects and finding some common denominators. (Note that we are not claiming this to be an ideal set theoretically, but deem it a necessary step towards this ideal set).

Secondly, the communication tools should support the administration of the agents. I.e. their addresses, their capabilities, their availability, etc. In this respect it is particularly important to pay attention to the fact that the agents often have interactions over long periods of time. This point becomes more important when the systems are large (contain many agents that can register and deregister at any time) and open (contain agents with different (and unknown) architecture and behavior).

Thirdly, a good support should be developed to describe the content of the messages. Although it is (probably) not possible to develop a standard content language, it should be easy to describe a content language. This facilitates the translation of the same message to different formats suited for different types of information sources. E.g. a query can be translated into SQL for a relation database and translated to a WWW query for the search engine on the WWW. In this respect XML might play an important role as a standard language to describe the content languages.

All of the above points are important for the short term development of the communication between CIAs. For the long term development the research on data integration and ontologies is of prime importance. This research should give some fundamental answers to the question how agents that do not share the same data models and ontologies can communicate.
Of course the general topics mentioned in section 3 are also of interest for CIA communication, but do not play a major role there.

6 Conclusions

In this paper we have tried to sketch an overview of Agent Communication Languages in general. We have given some arguments to the need for general ACLs and some current developments.
In section 3 we have tried to give an overview of the most important topics of research on ACLs at this moment. Although this overview can never be complete it gives a reasonable idea of the area. Whether all these topics are also of interest for communication between Cooperative Information Agents is discussed in section 5. Because of the type of setting it appears that e.g. conversation structures are usually very simple and thus research in this area is not of prime interest for CIA communication. In the other hand, interactions between CIAs are often spanning long periods of time and many different types of conversations can be interleaved (and even interact) with each other.
Finally we have tried to given some indications as to what type of developments we believe to be important for CIA communication in the future.

References

1. Austin, J.L. *How to do things with words* Oxford: Clarendon Press, 1962.
2. Cohen, P.R. and H.J. Levesque. Communicative actions for artificial agents In proceedings of the international conference on multi-agent systems (ICMAS-95), 1995, pp. 65-72.
3. Dignum, F. and Greaves, M. (eds) *Issues in Agent Communication*, Springer Verlag, (forthcoming).
4. Grice, P.H. Logic and conversation. In P. Cole and J. Morgan (eds.) *Syntax and semantics, vol.3, Speech acts*, Academic Press, New York, pp. 41-58.
5. Kandzia, P. and Klusch, M. (eds.)*Cooperative Information Agents* (LNAI 1202), Springer-Verlag, Berlin, 1997.
6. Klusch, M. and Weiss, G. (eds.)*Cooperative Information Agents II* (LNAI 1435), Springer-Verlag, Berlin, 1998.

7. Klusch, M., Shehory, O. and Weiss, G. (eds.)*Cooperative Information Agents III* (LNAI 1652), Springer-Verlag, Berlin, 1999.
8. Labrou, Y. *Semantics for an agent communication language*, Ph.D. thesis, University of Maryland, USA, 1997.
9. Labrou, Y. and Finin, T. Semantics and conversations for an agent communication language. In M.Huhns and M. Singh (eds.) *Readings in Agents*, Morgan Kaufmann, Los Altos, 1998, pp. 235-242.
10. Pitt, J. and Mamdani, A. Remarks on the semantics of FIPA's agent communication language. In *Autonomous Agents and Multi-Agent Systems*, vol. 2, nr. 4, 1999, pp. 333-356.
11. Searle, J.R. *Speech Acts*, Cambridge University Press, Cambridge, 1969.
12. Wooldridge, M. Verifiable semantics for agent communication languages. In Y. Demazeau (ed.) *Proceedings ICMAS'98*, IEEE Press, New York, 1998, pp. 349-356.