F. Dignum, E. Verharen and H. Weigand. A language/action perspective on cooperative information agents. In N. van der Rijst E. Verharen and J. Dietz, editors, *International Workshop on Communication Modelling (LAP-96)*, pages 40--53, Oisterwijk, 1996.

A Language/Action perspective on Cooperative Information Agents

E. Verharen*
Infolab, Tilburg University
Tilburg, The Netherlands

F. Dignum
Fac. of Maths. & Comp. Sc., Eindhoven University of Technology
Eindhoven, The Netherlands

H. Weigand
Infolab, Tilburg University
Tilburg, The Netherlands

Abstract

Research in Information Systems has switched its focus from data to communication. The communication between different autonomous ICS's (Information and Communication System) requires a certain amount of intelligence of each system. The system should be able to know which queries it can/may handle and also be able to negotiate about the information that it will give. In short these systems evolve into what is called Cooperative Information Agents. We show that basing the information contents of these agents on linguistic concepts and furthermore modelling the communications between the agents using speech act theory provides for a natural and sound setting for these CIA's as well as for Business Modelling. In addition, we describe a lexicon in which the conceptual meaning of the terms of communication can be defined. Together, these models provide an integrated semantics for Cooperative Information Agents. These can be interorganizational, as in EDI applications, or intraorganizational, as in Workflow Management.

Keywords: language/action perspective, communication modelling, linguistics, agents, functional grammar, lexicon

1 Introduction

Traditionally an information system was considered as one central database and a set of users accessing the database through application programs or directly via an SQL interface (e.g. [Date, 1990]). Today, databases are connected to each other and have to be accessible using electronic networks and EDI, while still maintaining their autonomy. Complete integration of the various resources might not be possible for technical or organizational reasons, hence the growing reliance on interaction between systems. To account for this evolution, we use the term *Information and Communication System* (ICS) instead of Information System. Furthermore, for systems to be able to cooperate with others they must have an intelligent interface that can cope with all types of requests and eventualities. In this light ICS's become active in several ways. An ICS actively maintains its information; it can communicate with other systems and reason about the information that it contains. It might decide to search for information that it needs by enquiring for it from other ICS's if it knows that it does not contain the information itself, preferably in ways it negotiates with (and lays down in contracts with) those other systems. It can respond more intelligently to messages explaining why a request does not have an answer or

* Author's address: Infolab, Tilburg University, POBox 90153, 5000 LE Tilburg, The Netherlands, tel. +31 13 4662767, fax. +31 13 4663069, email: E.M.Verharen@kub.nl

proposing alternatives. And it can negotiate about which requests it is willing to respond to and which requests will have no effect. For this purpose the ICS should contain a task module that plans the tasks the ICS has to fulfil. We refer to the autonomous ICS with tasks and contracts as a *Cooperative Information Agent* (CIA).

This CIA approach has consequences for what is called database semantics and knowledge representation. The traditional focus on logic and algebra to describe the semantics of database or knowledge base content needs to be widened to include the semantics of the communication. Since interfaces often have to reconcile different conflicting viewpoints, and have to be established by different, autonomous, parties, and because they are therefore more difficult to adapt, there is an increasing need for standards, such as reference ontologies. Because the abilities of the CIA to communicate and negotiate take an important place we claim that the influence of *linguistics* for these systems should go beyond that of a natural language interface.

Recent years have shown a growing awareness that linguistic theories are relevant for IS design. The so-called *Language/Action approach* gives content to a new generation of Business Process Models [Teufel & Teufel, 1995], [Goldkuhl, 1995] and communication modelling [Johannesson, 1995]. Other examples are DEMO [Dietz, 1994] and Action Workflow [Medina-Mora et al., 1992]. In the Netherlands, research has been performed within the framework of the Dutch SION-funded LIKE (Linguistic Instruments in Knowledge Engineering) project. The use of linguistics for information systems was described in [Dignum, 1989], [Weigand, 1990], [Weigand, 1991] and [Weigand et al., 1996]. Communication aspects have been described in [Weigand, 1990], [Weigand, 1993], [Dignum and Weigand, 1995a], [Dignum and Weigand, 1995b], and [Weigand et al., 1995]. Semantics and multimodal logics have been described in [Wieringa et al., 1989], [Weigand et al., 1995], [Dignum et al., 1996a], and [Dignum et al., 1996b], while modelling methods for modelling ICS's based on linguistics were described in [Verharen et al., 1994], [Verharen and Weigand, 1994], and [Burg and vdRiet, 1995].

In this paper, we introduce a framework for CIA's that integrates much of our previous work. Our starting-point remains the linguistic theory of Functional Grammar [Dik, 1989], including its theory of the Lexicon. We also use the theory of speech acts as developed by Searle ([Searle, 1969], [Searle and Vanderveken, 1985]) and Habermas ([Habermas, 1984]). We argue that the Language/Action perspective can be fruitfully applied to CIA design.

The structure of this paper is as follows: in section 2, we present the general framework of a Cooperative Information Agent. In section 3, we present the Information and Communication component of the agent. Section 4 takes a closer look at some components of a CIA. Section 5 compares our perspective with other agent approaches, and section 6 gives some conclusions and areas for further research.

2 CIA Framework

Recent years have shown a growing interest in cooperative systems (see e.g. [Power, 1993]). In our context such a system is constituted by a set of intelligent agents (CIA).

An agent is typically a piece of software that has certain capabilities, actions that it can perform, and certain tasks and responsibilities (delegated to it by a human agent). The actions can be material, such as opening a window, database actions such as updates, or communicative, such as providing some piece of information, or sending a message to another agent. Each agent also has an *agenda* containing the actions to be performed by the agent, instantly or at some designated time. In a normative system this agenda consists of the *obligations* of the agent. An obligation is the result of a commitment to perform a certain act and authorizations restrain or allow the commitment to and operation of an act (including doing other communicative acts). We assume that the agenda is not fixed but can be manipulated by the agent. The agent can add new obligations to the agenda (typically done on the request of another agent) and can reason about them. Obligations can be the result of the (sub)task of the agent, but can also follow from the contract (see below). He can also remove items of the agenda by performing actions or by violating the obligation. In the latter case he usually has to perform some compensatory action.

An agent has a certain *task* that can be split up in subtasks that the agent tries to fulfil. In doing so the agent initiates the necessary transactions. A transaction is a (logical) unit of work oriented towards a goal, consisting of one or more actions together with a specification of what should happen in the case of a failure.

Transactions are part of a *contract* describing the communication behaviour between two agents concerning some business relation and process. The contract also specifies what should happen in case of violation of one of the obligations or cancellation by one of the agents, possibly leading to other obligations described by another transaction in the contract and triggering a “contingency” plan, describing what should happen in order to get the subtask fulfilled.

The Lexicon is the system that stores and manages the terminology of a certain domain, it describes the information of the discourse including the message types, and corresponds to the Conceptual Model describing the static and dynamic structure of the Universe of Discourse. The population of this model (possibly including inference rules) is stored in the database.

Figure 1 gives a schema in which the different CIA concepts are related.

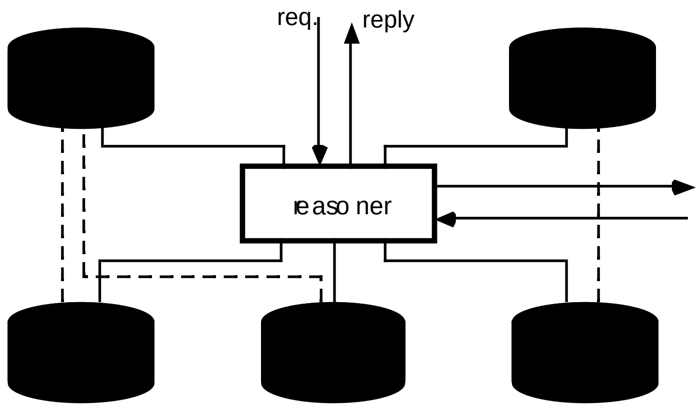

figure 1. CIA architecture

Section 4 describes tasks, contracts, and the lexicon in more detail.

Little has been said about how these cooperative systems influence each other's behaviour. Behaviour of CIA’s can be described by means of *interoperable transactions* defined as a logical unit of work (on communication and authorization level, not on technical database transaction level), involving different autonomous systems. Interoperable systems cooperate by means of *message* passing; a specification of an interoperable process consists essentially of a set of communicating subjects (often called agents), the possible message types provided by each agent to support its role, and constraints on the synchronisation of messages.

The best way to design a CIA network , is by starting from the communication structures in the domain. These communication structures can be modelled by communication models such as DEMO ([Dietz and Wiederhoven, 1992], [Verharen et al., 1994]) or Action Workflow ([Medina-Mora et al., 1992]). The model describes the human agents and the communication lines (messages and authorizations) between them. The communication lines are specified on a conceptual level only. They do not prescribe whether the communication is synchronous or not, and which technical intermediaries are used.

The automation of a communication model assigns a CIA to each agent (assuming here that the communication is automated totally, which need not be the case of course). Each of these CIA's falls under the responsibility of the respective agent. The original communication structure can then be replaced by communication between the CIA's. The CIA communicates with other CIA's and executes actions on its objects on the basis of the incoming messages.

In the case that a traditional database exists, and we don't want to reengineer the business processes, it is possible to "agentify" the database by encapsulating it in a CIA. In that case, the CIA falls under the responsibility of a Data Administrator agent, and other agents can access it according to their authorizations. So the technical architecture can be the same, but with a different division of tasks.

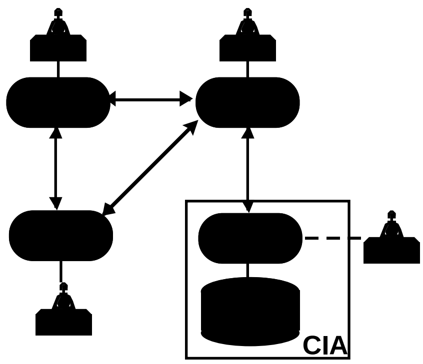

figure 2. Communicating agents

The framework can be related to Habermas' theory of communicative action [Habermas, 1984]. When involved in communicative action (as opposed to strategic action). the participants are oriented towards mutual agreement. It is considered essential that the participants achieve a common definition of the situation in which they find themselves. The CIA framework materializes this common definition in the form of the Contract (agreement on authorizations and obligations) and the Lexicon (agreement on concepts). With regards to the contents of a speech act, Habermas distinguishes between three worlds of reference: the objective world, the social world and the subjective world. Although CIAs do not take part in the human discourse and we don't want to blur the distinction between the human and the artificial, the same principles of rationality apply. A CIA also refers to an objective world (Lexicon), a social world (Contracts) and a subjective world (Tasks).

3 The Language/Action perspective

The interaction between agents can most easily be seen in the cases where a particular service is rendered from one agent to the other. A (prototypical) example of such a service is illustrated in the following picture:

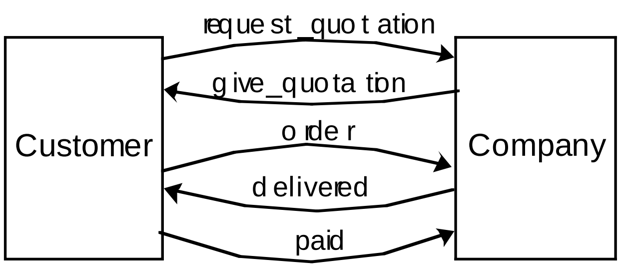

figure 3. Communication in a (prototypical) ordering procedure

In general we can distinguish four phases of communication. The first phase is the inquiry and negotiation about the terms of the contract. In this phase authorizations can be established on the basis of which some actions can be performed in the following phases. In the above figure this corresponds to the requesting and sending of a quotation. Here, sending a quotation implies authorizing the customer to order some products on some specified conditions. The second phase consists of the acceptance of the contract, i.e. the authorizations and obligations that follow from the contract. In the above example this is included in the order which implies an acceptance of the quotation. The third phase is the fulfilment of the contract, i.e. following a protocol according to the terms agreed upon in the contract. In the example this corresponds to the actual order, delivery and payment sequence. The last phase is the satisfaction phase, which has no specific messages in the model. This is perhaps because satisfaction is often implicit, and only dissatisfaction leads to communicative acts, such as appeals to warrant.

To describe the semantics of communication processes, we need to specify the meaning of the *information* and *communication* aspects of the messages.

3.1 Functional Grammar

As stated in the introduction the specification of the structure of the *information* is is based on the linguistic theory of Functional Grammar (FG) ([Dik, 1978], [Dik, 1989], [Weigand, 1990]). The use of FG for knowledge representation has been described before in [Weigand, 1992]. The theory of FG contains two main formal components. The first component describes the structure and semantics of the 'underlying structure' of utterances. FG can be used as a complete knowledge representation language where the central knowledge unit is called the proposition. Propositions are put forward in messages and contain predications. The second component of FG is formed by the various kinds of expression rules which map underlying clause structures onto linguistic expressions. The expression rules must determine the form of each part of the expression, their relative ordering, and, in case of speech output, the prosodic contour (not relevant for this paper). The levels of the FG underlying clause are:

Predicate	Predicates designate object types, relationship types, action types or attribute types. They are stored in the lexicon. Predicates are contained in predicate frames which specify among others the semantic functions applicable to the predicate, such as Agent, Goal, Recipient.
Predication	A predication is a structure created by inserting terms into predicate frames, and instantiating the frame at a spatio-temporal location. It denotes a State of Affairs in the Universe of Discourse.
Proposition	A structure made up of one or more predications (connected by means of conjunction, implication, embedding, etc.). A proposition may also contain modal operators. It designates a possible fact. It can be true or false with respect to a certain Situation or set of States of Affairs.
Message	A message (or clause) consists of a frame of a certain illocutionary type that contains a proposition. Besides, the illocutionary frame has entries for Speaker, Addressee, Speech Time, Theme, etc.

The bottom line of the (semantic) specification is the predicate. Predicates are used to name object types, attribute types, action types, but also illocutionary types. In the traditional view of semantics, these predicates are arbitrary strings. However, as argued at length in [Weigand, 1990] the use of these predicates draws heavily on the lexical meaning associated with their natural language counterparts. This observation gains in importance in the communication-oriented view we take here. When two parties set up a formal communication, they should not only agree on the syntactic format of the messages, and the formal meaning of the actions, but also on the conceptual meaning. By this we mean an (in context) unambiguous definition of the concept, either in natural language or in a formal language. For example, in the context of car registration, a car can be given the definition “a closed road vehicle with at least three wheels driven by a motor engine”. Note that the definition may refer to attributes (e.g. wheels, engine) that might play no role in the object type definition. Hence, although object type definition and conceptual definition are related, one can not be reduced to the other. The concepts are stored in the lexicon (see section 4).

3.2 Communication modelling

The specification of the structure of the *communication* is based on the linguistic theory of speech acts as developed by Searle [Searle, 1969]. Illocutionary logic [Searle and Vanderveken, 1985] is a logical formalisation of the theory of speech acts and is used to formally describe the communication structure itself, that is, the types and effects of the messages. The illocution (=illocutionary force) of a speech act is what the contents of that speech act indicates that the speaker intends the hearer to recognise him to be doing in uttering the speech act. E.g. the speaker might be asserting, denying, predicting, confirming, greeting, baptising, etc. The illocution of the message first of all defines the illocutionary type, and a propositional context (both of which can be expressed in FG structures).

The basic illocutionary types are ASSERT, DIRECT, COMMIT and DECLARE ([Austin, 1962], [Searle, 1969], [Searle and Vanderveken, 1985], [Lehtinen and Lyytinen, 1986]), but many more can be distinguished (cf. [Balmer and Brennenstuhl, 1981]., [Chang and Woo, 1994]. For a motivation of the set of basic speech acts we refer to [Dignum and Weigand, 1995a]). An example of a message is: ‘the assertion that a certain part has been delivered’, or ‘the request that a certain part be delivered (from agent A to agent B)’. In these two cases, the proposition is the same, but because of the different illocutions, the meaning of the two messages is quite different as well.

It is a characteristic of messages that they seldom stand on their own. For example, a request is typically followed by an acknowledgement, commitment or refuse message. Therefore messages can be organized in transactions. One can go even a step further and argue that the transaction, and not the message, is the unit of communication (cf. [Weigand, 1989] for linguistic arguments and DEMO for a method that explores this idea).

3.3 Deontic Dynamic Logic

For the semantics of communication models, we have made use of Deontic Logic. ([Meyer, 1988], [Wieringa et al., 1989], [Weigand et al., 1995]) The fundamental reason for the use of deontic concepts in communication modelling is that coordination of behaviour always requires some form of mutual commitment. If an agent does not execute an action, he has committed himself to, this causes a violation of the contract. Because the action should be executed in the future, it cannot be guaranteed, so the interpretation “it will happen in all future courses of events” is too strong, but the interpretation “it will happen in some course of events” is too weak. Interpreting the formula “α is obligatory” as: “not doing α violates a commitment” we get a more precise meaning of what it is that something is on an agent's agenda. Violations do not cause logical inconsistency, but can be the trigger for sanctions or repair actions.

4 Tasks, Contracts and the Lexicon

In this section the concepts of tasks and contracts are described in more detail. Furthermore we take a closer look at the Lexicon. We start by giving a detailed view at the working of the Task and Contract Manager (see figure 4 below). These are specific roles of the Reasoner in figure 1.

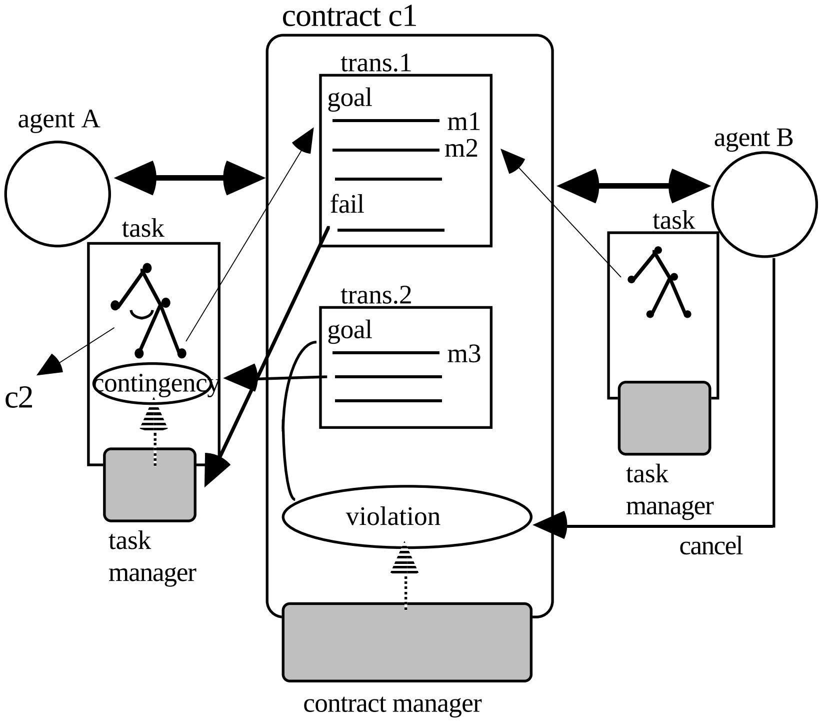

figure 4. Tasks, contracts and transactions (from [Weigand et al., 1996])

4.1 Tasks

A *task* is a meaningful unit of work assigned to an agent. Performing the task often involves communication transactions. However, its specification and updates thereof do concern the agent in question only, whereas changes in the possible transactions (involving other agents) can only be made by consent of the other agent. For that reason, we have made a distinction between task and contract, where the contract corresponds to the agreements between the two agents and the task draws on this potential for fulfilling an agent's goal.

Tasks are typically described in some task language (e.g. TasL [Nodine et al., 1994]). It typically allows the specification of tasks and subtasks, alternatives and temporal constraints. Transactions are prone to many types of failures, including traditional ones like system failures such as system crashes (in case of interoperable transactions also including network failures) and possible deadlocks caused by concurrent processes ([Ngu et al., 1994]). More interesting a failure occurs when a transaction (as one subtask) is initiated that does not commit (e.g. order product), but also when the transaction does commit first, and the resulting deontic state is violated later (e.g. the company doesn’t deliver), or because of cancellation, whereby the other party undoes an achieved effect. These features of transactions directly influence the task specification. For a CIA, it is important that tasks are not embedded in application code, but made explicit so that the Reasoner can use them in scheduling or rescheduling the work. Rescheduling is necessary when a subtask fails or a planned subtask becomes superfluous.

The execution model of the Task Manager is as follows:

- when a task is called, the Task Manager collects the subtasks, puts them into a right order according to precedence relationships and puts them on the agenda The subtasks are then called one by one. This step continues recursively.

- when a subtask cannot be fulfilled, the next alternative is chosen. If no alternative is available, it means that the subtask fails. It does not mean directly that the parent is aborted, since it is possible that the subtask can still be fulfilled by choosing an alternative in the preceding subtasks. This is tried first (backtracking). Only when no alternative is left, the parent fails.
- when an exception occurs , the procedure is basically the same. A next alternative is taken, or, if no alternative is left, backtracking is attempted. To keep the task specification simple, we give the designer the opportunity to specify a contingency plan separately (see below). If no alternatives for backtracking are left, the task fails, but its parent may have alternatives or contingency plans of its own. Every time a task fails, the Task Manager goes up one level in the task hierarchy.

Note that our execution model enforces a "structured approach" in the task specification. We do not allow for arbitrary abort or commit dependencies between subtasks over the boundaries of the parent tasks. This modularity is enforced to keep the specification transparent and maintainable.
In [Dignum and Linder, 1996] a first attempt in implementing tasks is made by giving preferences to goals. This also gives us the opportunity to reason over preference relations on (sub)tasks and how they influence each other.

A notorious problem with contingencies is that later (dependent) subtasks may already have completed, but their result may have become obsolete. Whether they have to be retried or not depends on what kinds of results they have produced. We therefore give the designer the opportunity to specify a separate contingency plan. The contingency plan consists basically of a set of *results* (such as supplier list, order-set). Results have an internal object structure (not specified here) and come about by certain subtasks and can become invalidated by other subtasks. When such occurs, a task can be triggered to repair the damage. This task can make use of the fact that all the essential results obtained so far (and not invalidated) are explicit.

4.2 Contracts

In our framework, contracts conceptually specify obligations between different parties about services provided to each other. If a particular service is not being fulfilled, it is possible to reason over this violation and take a remedial action without forcing the whole task to abort. A contract describes the authorized communication behaviour among providers and receivers of services. If the provider does not adhere to the obligation, it is the job of the Contract Manager to impose the violation policies. It would complicate the task specifications and lower the reusability if this communication is included in the task. As stated before the Task Manager should only be responsible in ensuring that a task is brought to its goal, not how violation of commitments is being dealt with. In this way, the process is more reactive to failures.

From a technical point of view, a contract is nothing but a protocol binding different parties to their commitments by explicitly specifying the type of services agreed upon, the obligations and the failure recovery methods. From an organizational point of view, a contract between interoperable systems stemming from different organizations also has the purpose of laying down some agreement. By grounding it in other contracts or business law, it may have legal status. Contracts, in the general sense of agreements between commercial partners, can be more or less elaborate. For instance, in Goldkuhl's model [Goldkuhl, 1995] a contract is the result of the negotiation, and boils down to a commitment to deliver and a commitment to pay. We assume that such a contract is set up by the two partners only once and then frequently used.

Since contracts can be more or less elaborate, we should require from any communication modelling approach that it can account for different levels of contracts. We want to say a few words about how our approach can do this.

Starting from an empty cyberspace in which agents can just send messages, agents must have the possibility to offer services (described by their individual tasks). A supplier agent can then provide the service of giving an offer. When a customer agent requests for this service, the supplier can refuse or accept. If he accepts, he sends an offer, that is, an authorization for the customer to order a product with the obligation to pay for it. If the customer agent uses this authorization, an agreement has been accomplished. All that is required in this case is a reliable and standardised way of requesting a service, as well as the "umbrella" protection of some international business law that guarantees the legal status of the obligations. Nowadays it is often considered an economic advantage when business relations can be tied closer with some preferred supplier or customer. In that case, the two parties want more agreements beforehand, for example about a guaranteed delivery time. Such a contract has to be set up. This can be supported by another service of the agent that offers contracts rather than specific products. If the customer accepts the offer, a mutually agreed contract is accomplished.

The contract can be "symmetric" in the sense of Goldkuhl, but also establish an asymmetric power relationship, in the same way as an employer can contract a new employee. We then arrive in a situation in which Goldkuhl's critique of Action Workflow - that it is rather one-sided in its customer[initiator] emphasis - is no longer valid. The one-sided Action Workflow loops fit well in an organizational setting with existing power and authorization relationships, whereas the symmetric business process model fits well in the free market context of autonomous negotiating agents. So we feel that both models have their value, depending on the context.

4.3 Lexicon

In this subsection we take a closer look at the Lexicon as the place to store (linguistically) specified organization knowledge.

The Lexicon is the system that stores and manages the terminology of a certain domain. It has two major functions. Firstly, it explicitly represents the *mutual* understanding about a certain concept of all agents involved. This can be used in the initial phase (negotiation) of setting up the contract (besides this there is a 'private' part that contains the agent's knowledge about the domain). The second major function of a Lexicon is the input it can provide to the software development process. The more formal structure the Lexicon exhibits, the more can be derived from it in terms of entity types, relationships, actions etc. One research project in Tilburg University investigates to what extent NIAM object models can be derived automatically from a lexical specification.

In [Weigand, 1990] a linguistic approach is presented in which the lexical definition as well as the lexical structures are based on linguistic primitives. The Lexicon defines the possible predicate frames describing the domain. That is, grammatical information, such as gender, word category, stem, etc.; structural information, in particular the taxonomic relation, designating subsumption or subtyping, but also semantic sets and pre- and postconditions for dynamic predicates; and conceptual information (predicate schemata, i.e. 'stereotypes', including essential but not necessary characteristics of a concept), specified in Functional Grammar.

Note that lexical definitions should not be considered as exhaustive characterizations of a concept. A definition is made relative to a certain context. Within the context, the definitions must differentiate different concepts, and provide a *basis* for mutual understanding.

In such a Lexicon, no formal distinction is made between entities, relationships, actions, attributes, identifiers, in so far all of them have a lexical entry. Following linguistic theory, the concepts are organized in a taxonomy and around prototypes. The higher levels of this taxonomy form a basic ontology, including primitive concepts like "entity", "event", "state", "sign". The lexical entry abstracts from the various word forms and inflections of a certain word, as in normal dictionaries. In the case of complex concepts, such as actions, the lexical entry includes a frame containing the roles of the participants, such as "agent", "recipient", etc. The definition of a concept is in principle simply the (natural language) dictionary entry. However, the definition can be parsed and stored internally in a linguistic representation formalism like FG, which allows formal treatment. Moreover, lexical research has shown that definitions typically make use of a small array of basic structures, such as the "isa" and "partof" relation and "purpose". This allows for even more formalization and hence computational processing, while the definition can still be expressed in natural language.

An example of some lexicon definitions is the following. The italicised noun phrase indicates a taxonomic link to a superconcept. The arguments are given between parentheses:

car: a closed *road vehicle* on more than three wheels driven by a motor engine
model(car): the *name* of a registered car design
registration number(car): a *string* of 6 alphanumeric characters uniquely identifying a car
serial number(car): a *string* of 12 digits uniquely identifying a car of a certain manufacturer
year of production(car): the *year* in which the car has been given a serial number

An example in which the semantic structure of a concept is explicit, is the following. It defines *sell* as a transfer action with three semantic roles (the object has an empty label, ag stands for agent and rec for recipient). It describes pre- and postconditions using predicates from the same Lexicon such as *own*. The incondition says that the action includes a payment action of the recipient.

sell(ag X:human)(P:thing)(rec Y:human)
`isa` transfer
`pre` = own(ag X)(P), price(P) = D
`post` own(ag Y)(P)
`inc` pay(ag Y)(D)(rec X)

In circumscribing the terminology for a particular application domain, the knowledge engineer might draw on available terminologies for the generic domain. Such generic terminologies might draw in turn on more general dictionaries. Hence, it seems useful to organize the Lexicon as a collection of related sublexicons. A *Lexicon Management System* is a system that can handle multiple lexicons. In this way, it is possible to set up application architectures in a consistent way.

We assume that the Lexicon is filled first by the knowledge engineer (backed by the domain expert). Dynamic negotiations between CIAs about concepts is still a road ahead. However, when more ontologies, ISO standards and domain lexicons become available, it is possible for negotiating CIAs to set up a mutual understanding by making a reference to such a given set of concepts. It could also use a bilingual (certified) Lexicon that translates Dutch business concepts to French, for example. It should be clear to which definitions both CIAs commit themselves .

5 Comparison with other agent frameworks

In this paper we introduce an agent-oriented framework for CIAs. Agent theories and architectures have gained much interest lately, both from the software engineering and AI discipline.

As was (or, is) the case with Object-Orientation almost every researcher has a different definition of agent. [Wooldridge and Jennings, 1995] describe how Carl Hewitt recently remarked (at the 13th international workshop on Distributed AI) that the question “what is an agent ?” is embarrassing for the agent-based computing community in just the same way that the question “what is intelligence ?” is embarrassing for the mainstream AI community.

Although there is no consensus in the AI community about precisely which combination of information attitudes (like knowledge and belief) and pro-attitudes (like desire, intention, commitment, choice) are best suited to characterize agents, a number of approaches have gained much support. Perhaps the best known agent theories are the intention theory of Cohen and Levesque ([Cohen and Levesque, 1990a] and the BDI framework of Rao and Georgeff ([Rao and Georgeff, 1991]).

Cohen and Levesque developed a theory of intention which the authors required as a pre-requisite for a theory of speech-acts ([Cohen and Levesque, 1990b]). They described two basic attitudes: beliefs and goals; and intention was defined in terms of these. In related work, Rao and Georgeff developed a logical framework on three primitive modalities: Belief, Desires and Intention. And although in other work ([Rao and Georgeff, 1992]) they consider the potential for adding (social) plans to their formalism, the BDI framework considers agents in isolation; it ignores the communicative aspects that are central in our approach.

Another way of looking at such frameworks is through the Information-Motivation-Action-Social (IMAS) model ([Dignum and Linder, 1996]). In this framework the Information corresponds to knowledge and belief attitudes, the Motivation to the desires and intentions of the agent (represented by goals and plans), in the Action component a description of the capabilities and actions an agent can perform is given (usually specified in a dynamic logic), and finally the Social component which describes the communication and cooperation attitudes. Our framework corresponds closely to the IMAS framework in that the Information corresponds to the knowledge and beliefs recorded in the database; the Motivation corresponds to the obligations on the agenda, brought about by the Task and Contract; and the Action corresponds to the actions performed by the Reasoner.

Another influential approach is the one taken by Shoham ([Shoham, 1993]) in which he uses the mentalistic notions knowledge, belief, intention, and obligation to characterize an agent. He also describes a new programming paradigm based on this notion of agents: agent-oriented programming. The CIA framework takes a similar stance to agent characterization, as described in section 2 and above, in which the obligations play a central role. In [Verharen et al., 1994] we described how we can use this paradigm to experiment with an implementation of the CIA framework.

6 Concluding Remarks

In this paper, we have described an framework and integrated semantics for modern Cooperative Information Agents, making use of a Language/Action approach. We proposed the use of CIAs in different application settings, such as Business Computing, EDI, and Workflow Management.

A first attempt to model communication using speech acts is made in the method SAMPO [Lehtinen and Lyytinen, 1986], but this does not include the deontic aspects (authorization and obligation). In [Lee, 1988], Lee presents a language and modelling technique for deontic rules, but these model procedures rather than communication structures. The techniques described here also aim at providing a formal way of specifying business process models such as DEMO, Goldkuhl's BPR and Action Workflow. These approaches however have different scopes. For example, DEMO offers a rigid methodology, but only on the transaction level. Goldkuhl's BPR is a generic task model that is particularly aimed at free market exchanges, whereas Action Workflow is a generic task model (and perhaps transaction model) that is more oriented towards an organizational context. All models do not pay much attention yet to failure handling.

The conceptual framework we introduced can be summarized as follows: the *Communication Model* is the starting point. It contains agents (with tasks) and messages (organized in contracts); the messages in the Communication Model can be decomposed in elementary communicative actions, also called speech acts. Speech acts make reference to actions and objects, described in the *Information Model*. All predicates in all models draw on the restricted vocabulary contained in the *Lexicon.*

The models used are not very different from those found in other current approaches, e.g. OO design, where one usually has Object Models, Data Flow Diagrams (for the actions) and Functional Models (that overlap with the Communication Model). The main difference, in our view, is that our approach gives more proper weight to the peculiar problems occurring in CIA design. These problems are not the formal problems in the first place, but (1) problems of conflicting conceptual frameworks, (2) the establishing of authorizations, and (3) the use of standards and application architectures. Therefore, we require that the predicates are well-defined in a semantic Lexicon, which can be used also for the specification of generic models. The Lexicon is not found in the other approaches.

Secondly, our Communication Model highlights who is responsible for a certain database, and it also contains primitives for assigning and retracting authorizations dynamically. In other approaches, authorizations are not taken into account, and the communication is modelled as a mechanical process. Being based on speech act theory, our communication language contains higher-level primitives than other approaches based on the concept of communication as data flow, or communication as synchronization.

Contracts were presented as a way of modularizing the normative specifications. Both the negotiation phase to establish the contract as well as the contract itself can be modelled using our formalism. Contracts were specified by using interoperable transactions, together with deontic constraints and rules for appropriate action when violated. The Interactions model [Nodine et al., 1994] uses "weak constraints" and backtracking when they are violated. Our failure handling uses backtracking as well, but it goes further in two respects. First, we do not abort later (dependent) subtasks immediately, but only when the committed service cannot be maintained (by the Contract Manager). The contract may provide specific "contingency plans" as alternatives; if these succeed, no abort or backtracking is necessary. Secondly, we have sorted out the complex concept of "compensation" into two more focused notions: compensation of the other party, as specified in the contract, and "compensation", or contingency handling, of the task. For contingency handling, we propose a separate specification part that monitors the results obtained and invalidated so far. Although exception handling is a complex issue and will remain so, the least we can do is try to manage the complexity. This is the most important but modest goal of the task/contract distinction. We also consider it an advantage that task management can be turned into a local issue, rather than a concern of the (global) multidatabase. In this way the global control is kept to a minimum, which makes the specification and implementation more flexible.

Our short-term goal is to build a prototype implementation of the task/contract model so that more experience can be gained with contingency handling in an "agent-oriented DBMS" to be built as an extra layer around a traditional DBMS. A design methodology that is in accordance with the framework presented in this paper, is the subject of a forthcoming Ph.D. Another topic for research is the elaboration of and relation between contracts. A contract can be a specialisation of another contract or an extension of that contract. Also an agent can have contracts with different parties at the same time. We should be able to prove that these contracts are mutually consistent.

References

[Austin, 1962]. J.L. Austin, *How to do things with words*, Clarendon Press, Oxford, 1962.

[Balmer and Brennenstuhl, 1981]. T. Balmer and W. Brennenstuhl, *Speech Act Classification: A Study in the Lexical Analysis of English Speech Activity Verbs*, Springer-Verlag, Berlin, 1981.

[Burg and vdRiet, 1995]. J.F.M. Burg and R.P. van de Riet, "COLOR-X: Linguistically-based Event Modeling: A general approach to dynamic modeling", in: Proc. of Advanced Information Systems Engineering (CAISE'95), J. Iivari, K. Lyytinen and M. Rossi (eds.) (LNCS-932), Springer-Verlag, Berlin, pp. 26-39, 1995.

[Chang and Woo, 1994]. M.K. Chang and C.C. Woo, "A Speech Act Based Negotiation Protocol: Design, Implementation and Test Use", in: ACM Transactions on Information Systems (TOIS), vol.12, no.4, pp. 360-382, 1994.

[Cohen and Levesque, 1990a]. P.R. Cohen and H.J. Levesque, "Intention is choice with commitment", in: Artificial Intelligence 42, pp. 213-261, 1990.

[Cohen and Levesque, 1990b]. P.R. Cohen and H.J. Levesque, "Rational interaction as the basis for communication", in: Intentions in Communication, P.R. Cohen, J. Morgan and M.E. Pollack (eds.), MIT Press, Cambridge, pp. 221-256, 1990.

[Date, 1990]. C. Date, *An Introduction to Database Systems, vol. 1, 5th Ed.*, Addison-Wesley, 1990.

[Dietz and Wiederhoven, 1992]. J.L.G. Dietz and G.A.M. Wiederhoven, "A comparison of the linguistic theories of Searle and Habermas as a basis for communication supporting systems", in: Proc. of Linguistic Instruments in Knowledge Engineering, R.P. van de Riet and R.A. Meersman (eds.), North-Holland, 1992.

[Dietz, 1994]. J.L.G. Dietz, "Business Modelling for Business Redesign", in: Proc. of 27th Hawaii Int.l. conf. on System Sciences, IEEE Computer Society Press, 1994.

[Dignum, 1989]. F.P.M. Dignum, *A language for modelling knowledge bases*, Ph.D. Thesis, Vrije Universiteit, Amsterdam, 1989.

[Dignum and Weigand, 1995a]. F. Dignum and H. Weigand, "Communication and Deontic Logic", in: Information Systems, Correctness and Reusability, Proc. of ISCORE-94 Workshop, R. Wieringa and R. Feenstra (eds.), World Scientific, Singapore, pp. 242-260, 1995.

[Dignum and Weigand, 1995b]. F. Dignum and H. Weigand, "Modelling Communication between Cooperative Systems", in: Proc. of Advanced Information Systems Engineering (CAISE'95), J. Iivari, K. Lyytinen, M. Rossi (eds.) (LNCS-932), Springer-Verlag, Berlin, pp. 140-153, 1995.

[Dignum and Linder, 1996]. F.Dignum and B. van Linder, "Modelling Rational Agents in a Dynamic Environment: Putting Humpty Dumpty Together Again", in: Proc. of the 2nd workshop of the ModelAge Project, J.L. Fiadeiro and P.-Y. Schobbens (eds.), Sesimbra, Portugal, January 1996, pp. 81-92, 1996.

[Dignum et al., 1996a]. F. Dignum, H. Weigand and E. Verharen, "Meeting the deadline: on the formal specification of temporal deontic constraints", in: Proc. of the Int.l. Symposium on Methodologies for Intelligent Systems (ISMIS'96), 1996 (to appear).

[Dignum et al., 1996b]. F. Dignum, J.-J.Ch. Meyer, R. Wieringa and R. Kuiper, "A Modal approach to intentions commitments and obligations: Intention plus commitment yields obligation", in: Deontic Logic, Agency and Normative Systems, M. Brown and J. Carmo (eds.), Springer-Verlag, Berlin, pp. 80-97, 1996.

[Dik, 1978]. S.C. Dik, *Functional Grammar*, North-Holland, Amsterdam, 1978.

[Dik, 1989]. S.C. Dik, *Theory of Functional Grammar*, Foris, Dordrecht, 1989.

[Goldkuhl, 1995]. G. Goldkuhl, "Information as Action and Communication", in: The Infological Equation, *Essays in honour of B. Langefors*, B. Dahlbohm (ed.), Gothenburg Studies in Information Systems, Gothenburg Univ., 1995. (Also: Lingkoping Univ. Report LiTH-IDA-R-95-09)

[Habermas, 1984]. J. Habermas, *The Theory of Communicative Action: Reason and the Rationalization of Society, Volume One*, Beacon Press, Boston, 1984.

[Johannesson, 1995]. P. Johannesson, "Representation and communication-a speech act based approach to information systems design", in: Information Systems, vol.20, no.4, pp. 291-303, 1995.

[Lee, 1988]. R. Lee, "Bureaucracies as Deontic Systems", in: ACM Transactions of Office Information Systems (TOIS), vol. 6, no. 2, pp.87-108, 1988.

[Lehtinen and Lyytinen, 1986]. E. Lehtinen, K. Lyytinen, "Action Based Model of Information Systems", in: Information Systems, vol. 12, no. 3, pp. 299-317, 1986.

[Medina-Mora et al., 1992]. R. Medina-Mora, T. Winograd, R. Flores and F. Flores, “The Action-Workflow Approach to Workflow Management Technology”, in: Proc. of 4th Conf. on Computer Supported Cooperative Work (CSCW94), R. Furuta and C. Neuwirth (eds.), ACM Press, 1994.

[Meyer, 1988]. J.-J.Ch. Meyer, “A different approach to deontic logic: deontic logic viewed as a variant of dynamic logic”, in: Notre Dame Journal of Formal Logic 29(1), pp. 109-136, 1988.

[Nodine et al., 1994]. M.H. Nodine, N. Nakos and S. Zdonik, “Specifying Flexible Tasks in a Multidatabase”, in: ACM SIGOIS Bulletin 16 (1), p. 13-17, 1994.

[Ngu et al., 1994]. A. Ngu, R.A. Meersman and H. Weigand, “Specification and verification of communication for interoperable transactions”, in: Int.l. Journal of Intelligent and Cooperative Information Systems 3(1), pp.47-65, 1994.

[Power, 1993]. R. Power, *Cooperation among organizations: the potential of computer supported cooperative work*, Springer-Verlag, Berlin, 1993.

[Rao and Georgeff, 1991]. A.S. Rao and M.P. Georgeff, “Modeling rational agents within a BDI-architecture”, in: Proc. of Knowledge Representation and Reasoning (KR&R-91), R. Fikes and E. Sandewall (eds.), Morgan Kaufmann Publishers, San Mateo, pp. 473-484, 1991.

[Rao and Georgeff, 1992]. A.S. Rao and M.P. Georgeff, “Social plans: preliminary report”, in: Proc. of Decentralized AI 3 - 3th European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-91), E. Werner and Y. Demazeau (eds.), Elsevier Science Publishers, Amsterdam, 1991.

[Searle, 1969]. J.R. Searle, *Speech Acts*, Cambridge University Press, 1969.

[Searle and Vanderveken, 1985]. J.R. Searle and D. Vanderveken, *Foundations of illocutionary logic*, Cambridge University Press, 1985.

[Shoham, 1993]. Y. Shoham, “Agent-oriented programming”, in: Artificial Intelligence 60, pp. 51-92, 1993.

[Teufel & Teufel, 1995]. S. Teufel and B. Teufel, “Bridging Information Technology and Business - Some Modelling Aspects”, in: ACM SIGOIS Bulletin 16 (1), pp. 13-17, 1995.

[Verharen et al., 1994]. E. Verharen, H. Weigand, and O. De Troyer, “Agent-oriented information system design”, in: Working Papers. of the Int.l. Workshop on Information Systems, Correctness and Reusability (ISCORE’94), R. Wieringa and R. Feenstra, (eds), Vrije Universiteit Report IR-357, Amsterdam, pp. 378-392, 1994.

[Verharen and Weigand, 1994]. E. Verharen and H. Weigand, “Agent Oriented Information Systems Design”, in: Poster Proc. of the Int.l. Symposium on Methodologies for Intelligent Systems (ISMIS’94), Z. Ras and M. Zemankova (eds.), Oak Ridge Nat.l. Lab., 1994.

[Weigand, 1989]. E. Weigand, *Sprache als Dialog : Sprechakttaxonomie und kommunikative Grammatik*, Niemeyer Verlag, Tubingen, 1989.

[Weigand, 1990]. H. Weigand, *Linguistically Motivated Principles of Knowledge Based Systems*, Foris, Dordrecht, 1990.

[Weigand, 1991]. H. Weigand, “The linguistic turn in information systems”, in: Proc. of Collaborative Work, Social Communications and Information Systems, R. Stamper, P. Kerola, R. Lee and K. Lyytinen (eds.), Elsevier Scientific Publishers, IFIP, 1991.

[Weigand, 1992]. H. Weigand, “Assessing Functional Grammar for Knowledge Representation”, in: Data and Knowledge Engineering 8 (1992), pp. 191-203, 1992.

[Weigand, 1993]. H. Weigand, Deontic aspects of communication, in: *Deontic Logic in Computer Science*, J.-J.Ch. Meyer and R. Wieringa (eds.), Wiley Professional Computing, 1993.

[Weigand et al., 1995]. H. Weigand, E. Verharen, and F. Dignum, "Integrated Semantics for Information and Communication Systems", in: Proc. of IFIP DS-6 "Database Application Semantics", R. Meersman, L. Mark (eds), Stone-Mountain, Georgia, 1995.

[Weigand et al., 1996]. H. Weigand, E. Verharen and F. Dignum, “Interoperable transactions - a structured approach”, submitted to Conf. on Advanced Information Systems Engineering (CAISE'96).

[Wieringa et al., 1989]. R. J. Wieringa, J.-J.Ch. Meyer and H. Weigand, “Specifying dynamic and deontic integrity constraints”, in: Data and Knowledge Engineering 4 (1989), pp. 157-189, 1989.

[Wooldridge and Jennings, 1995]. M. Wooldridge and N.R. Jennings, “Intelligent Agents: Theory and Practice”, submitted to Knowledge Engineering Review.