

Research Report 11/98



An Integrated Structured Analysis Approach to Intelligent Agent Communication

by

**Hans Akkermans,
Rune Gustavsson, Fredrik Ygge**

Department of
Computer Science and Business Administration
University of Karlskrona/Ronneby
S-372 25 Ronneby
Sweden

ISSN 1103-1581
ISRN HK/R-RES—98/11—SE

An Integrated Structured Analysis Approach to Intelligent Agent Communication

by Hans Akkermans, Rune Gustavsson, Fredrik Ygge,

ISSN 1103-1581

ISRN HK/R-RES—98/11—SE

Copyright © 1998 by Hans Akkermans, Rune Gustavsson, Fredrik Ygge

All rights reserved

Printed by Psilander Grafiska, Karlskrona 1998

An Integrated Structured Analysis Approach to Intelligent Agent Communication

Hans Akkermans ^{§‡} Rune Gustavsson [‡] Fredrik Ygge [‡]

[§]AKMC and Free University Amsterdam
Department of Information and Software Engineering
De Boelelaan 1081a, NL-1081 HV Amsterdam, The Netherlands
HansAkkermans@cs.vu.nl or @compuserve.com

[‡]University of Karlskrona/Ronneby and EnerSearch AB
Department of Computer Science (IDE)
372 25 Ronneby, Sweden
Rune.Gustavsson, Fredrik.Ygge@ide.hk-r.se

Abstract

Intelligent multi-agent systems offer promising approaches for knowledge-intensive distributed applications. Now that such systems are becoming applied on a wider industrial scale, there is a practical need for structured analysis and design methods, similarly as exist for more conventional information and knowledge systems. This is still lacking for intelligent agent software. In this paper, we describe how the process of agent communication specification can be carried out through a structured analysis approach. The structured analysis approach we propose is an integrated extension of the CommonKADS methodology, a widely used standard for knowledge analysis and systems development. Our approach is based on and illustrated by a large-scale multi-agent application for distributed energy load management in industries and households, called HOMEBOOTS, which is discussed as an extensive industrial case study.

1 Introduction

Intelligent multi-agent systems offer promising approaches for knowledge-intensive distributed applications. Now that such systems are becoming applied on a wider industrial scale, there is a practical need for structured analysis and design methods, similarly as exist for more conventional information and knowledge systems. This is still lacking for intelligent agent software. The modeling of agent communication has not yet received much attention in the knowledge systems area. Conventional knowledge systems are rather simple with respect to their communication structure: often it is a straightforward question-and-answer sequence between system and end user. However, this is different in recent intelligent multi-agent systems. Therefore, agent communication aspects are now in need of a much more advanced treatment in knowledge management, acquisition and modeling. Conversely, intelligent multi-agents systems development may learn a lot from the knowledge systems field, as industry-quality systems modeling and analysis in this field is much more mature.

In this paper, we describe how the process of agent communication specification can be carried out through a structured analysis approach. Our work emphasizes, like in structured analysis more generally, the usefulness of semi-formal (e.g. graphical) and pragmatic techniques, and bringing structure into the specification *process*, a necessity in the take-up of intelligent and multi-agent systems on a wider industrial scale. The structured analysis approach we propose is an integrated extension of the CommonKADS methodology. This is a widely used standard for knowledge analysis and systems development, ranging from organizational analysis to system design and implementation. Figure 1 shows the associated CommonKADS model suite [Schreiber et al., 1994, 1998].

Our approach is based on and illustrated by a large-scale multi-agent application for distributed energy load management in industries and households, called HOMEBOTS which is discussed as an extensive case study [Akkermans and Ygge, 1996, 1997, 1998].

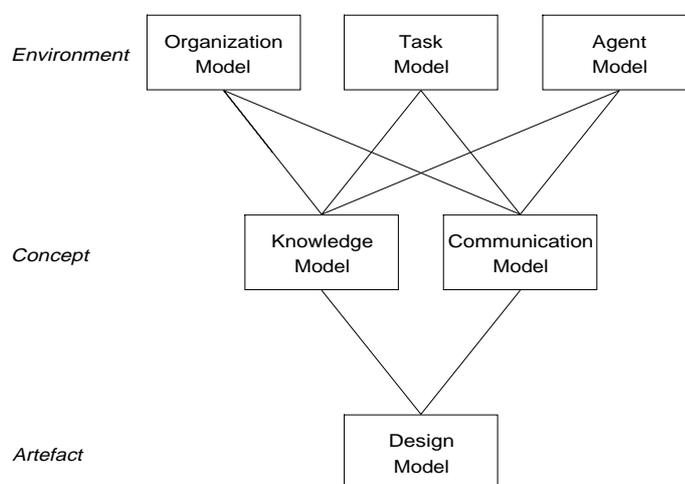


Figure 1: *The CommonKADS Model Suite.*

2 The CommonKADS Communication Model

To become effective, knowledge must, once produced, be transferred to the various parties who use it to perform their own tasks. It is the purpose of the CommonKADS Communication Model to specify the information exchange procedures to realize the knowledge transfer between agents. Figure 2 gives an overview of the main components of the Communication Model and how they relate to the other CommonKADS models.

In brief, a task that is carried out by one agent may produce results in the form of information objects that need to be communicated to other agents. A simple example is the basic system-user interaction, where the knowledge system presents reasoning results to the user, or, alternatively, the user provides input data to the knowledge system. The description of the agents involved, together with their capabilities, stems from the Agent Model. The tasks, as well as their (input/output) information objects and their assignment to the various agents, originate from the Task Model. If tasks are knowledge-intensive, they are usually refined in the Knowledge Model. The latter has a special leaf subtask type called a transfer function, indicating that input or output reasoning information has to be obtained from or delivered to another agent.

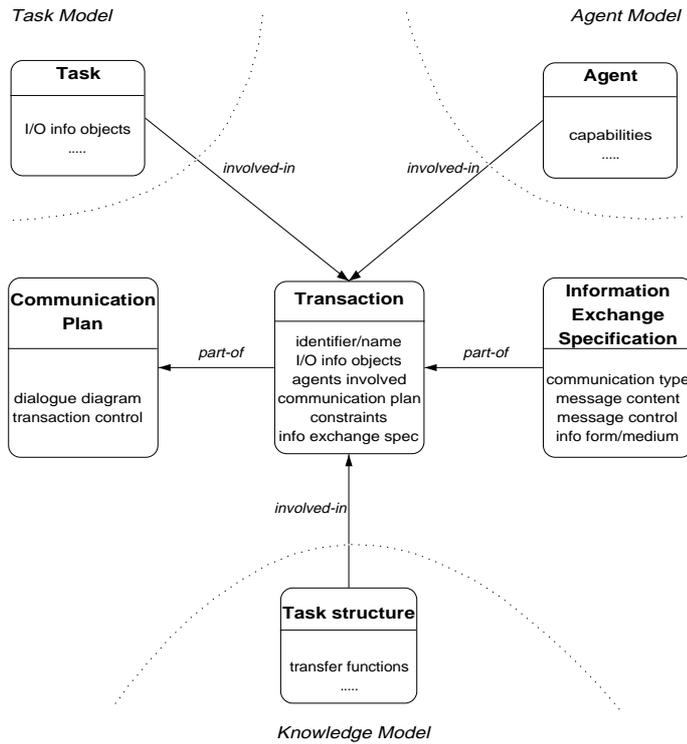


Figure 2: Overview of the Communication Model and how it relates to the other CommonKADS models.

The key Communication Model component describing such communicative acts is called a *transaction*. A transaction tells what information objects are exchanged between what agents and what tasks. It is, so to speak, the go-between of two tasks carried out by different agents. Transactions are the building blocks for the full dialogue between two agents, which is described in the communication plan. Transactions themselves may consist of several messages, which are then detailed in the information exchange specification. This specification is based on predefined communication types and patterns, which make it easy to build up message protocols in a structured way.

Accordingly, the process of constructing of the COMMONKADS Communication Model goes in terms of three subsequent layers, from global to detailed specifications, as follows (see also Figure 2).

1. The overall communication plan, which governs the full dialogue between the agents.
2. The individual transactions that link two (leaf) tasks carried out by two different agents.
3. The information exchange specification that details the internal message structure of a transaction.

This paper explains this *stepwise structured construction process*, and offers a number of specific techniques and guidelines. We will illustrate the main points through an application example, coming from the energy utility industry.

3 The Communication Plan

In constructing the CommonKADS Communication Model, it is easiest to begin with the overall communication plan. The entry point of the communication analysis is: consider two agents that carry out a shared or distributed top task. For successful completion, they need to communicate and exchange information. The communication plan aims to give an overview of all the needed exchanges. Thus, it covers the *full top-level dialogue* corresponding to performing this shared top task. For example, if an expert system and its human user are the considered two agents, the communication plan gives the human-computer dialogue — in this case typically consisting of data input, asking or answering questions and presentation of reasoning results — associated to a single but complete session with the system.

Constructing the dialogue diagram. Since the entry point of the analysis is a top task distributed over more than one agent, it is evident that constructing the Communication Model depends on information from other CommonKADS models. This is depicted in Figure 2. Based on this information, we have a straightforward *structured procedure* to develop the top level of the agent communication model.

1. Go through all Task Model leaf tasks and Knowledge Model transfer functions. Make a list of all tasks that have input or output information objects that must be exchanged with another agent. Do this for each agent.
2. From this list, identify the set of associated agent-agent transactions. Here, a transaction is simply defined as the communication link needed between two leaf tasks (including transfer functions) carried out by two different agents. Transactions are the basic building blocks of the communication plan. Give each transaction an understandable name. Typically, this is a verb+noun combination indicating the communicative act performed with the information object (e.g., present diagnostic conclusions to the user).
3. The results of the previous two steps can be conveniently combined in a so-called dialogue diagram, where in a single picture we see an overview of all transactions and the tasks they are linking, for every agent. The general form of a dialogue diagram is shown in Figure 3. The dialogue diagram (which resembles, by the way, role-activity diagrams known from organizational analysis) presents the complete information flow part of the communication plan.
4. Finally, the communication plan is completed by adding control over the transactions. This may be done in pseudo-code or state-transition diagram form. In basic practical cases it is often a simple sequence that straightforwardly follows from the information flow. But when, for example, exception or outside event handling is involved, a control specification is usually needed.

Control over transactions. The dialogue diagram shown in Figure 3 is useful for displaying the flow of discourse between two agents. But it does not show control. In strongly reasoning-oriented tasks, control over transactions is often a quite simple sequence that follows the flow of information objects. However, this is not good enough in situations, where for example external events occur that conditionally trigger tasks or transactions. In such cases, we need some way to describe control over transactions. In the CommonKADS Communication Model we do this in a rather standard way, either by means of some kind of structured control language or pseudo-code, or by means of a state-transition diagram. As these are well-known techniques from software and information engineering, for reasons of space we skip a further description here and refer to the structured analysis literature, see e.g. [Yourdon, 1989]. To illustrate our approach, we discuss

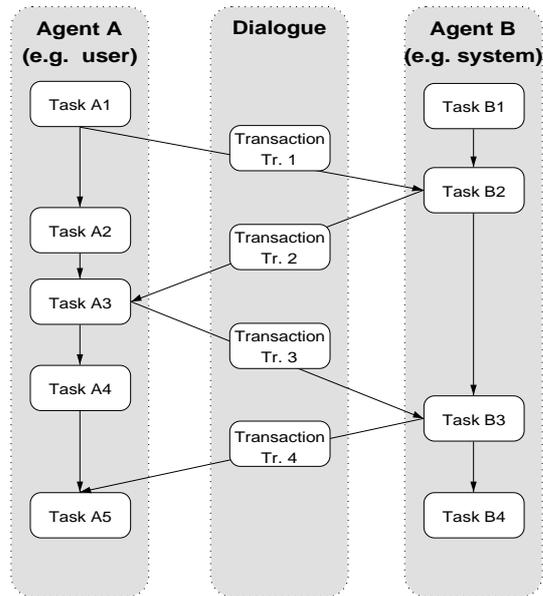


Figure 3: The general lay-out of a dialogue diagram. It forms the central part of the communication plan, as it shows the overall information flow related to agent communication.

below a real-life application as a case study in communication modelling.

4 HOMEBOTS — A Multi-Agent System for Energy Management

Industrial application context. Due to the deregulation of the European energy market, the electric utility industry is in a transition from being a regulated and rather monopolistic power generation industry, to a business operating in a dynamic and competitive free market environment. For the utility industry a new business paradigm is therefore emerging. The usual business of generating, distributing and billing customers for kWh —essentially a pure product-oriented delivery concept— is being transformed into offering different kinds of new value-added customer services (Figure 4). These vary from automated metering and billing at-a-distance, advice on optimized energy use, tailored rates and contracts, to home automation, home energy management and demand-side management at the customer’s premises.



Figure 4: Paradigm shift in energy utilities due to the new information society: from a pure product delivery concept to two-way customer-oriented services.

Recent advances in ICT have made it technologically and financially possible to equip many different types of nodes in the electrical network (including 230V and other substations, industrial loads and even household equipment) with significant communication (230V power grid, fiber optics, GSM, etc.) as well as computing capabilities of their own. In this way, nodes in the electrical network will obtain the capabilities to act as intelligent and interacting agents on behalf of the customer and the utility.

In Sweden, a large project called ISES (Information - Society - Energy - System) is now underway to perform research and develop new services based on these recent advances in ICT. One of the new service applications that are foreseen is that the electric network nodes themselves act as intelligent agents in order to take care of power load management. Such intelligent agents we call 'homebots'. This load management would lead: (i) for the utility, to a better utilization of the power grid as a result of reduction of peak (valley) loads of the power net; and (ii) for the customer, to a minimization of the overall energy cost, while maintaining a specified (individual) comfort level.

Multi-agent system solution. Existing forms of load management are limited to a low number of large facilities since manual control still plays an important part. The benefits of multi-agent systems for load management are a higher level of automation, a much larger scale, and more flexibility and distributedness.

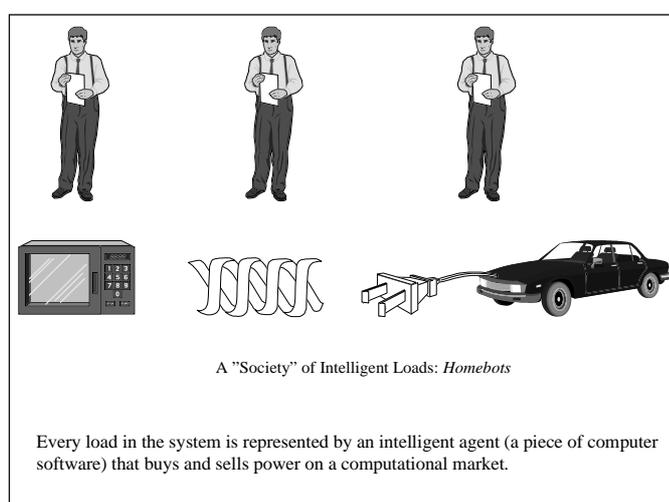


Figure 5: *Devices and loads are equipped with smart small software programs. These software agents communicate, act and cooperate as representatives assisting the customer, to achieve given goals such as power load management.*

Our approach is to achieve dynamic and automatic load balancing by means of software agent technology embedded in 'smart' equipment. Every device or load, such as heaters, radiators and boilers in a household, is represented by a software agent responsible for efficient and optimal use of energy, while taking the customer preferences into account (cf. Figure 5). In our view, *knowledge plus communication are the ingredients of intelligence in devices.*

We call these agents HOMEBOOTS. A key idea is that the communication and cooperation between devices for the purpose of load management takes the form of a computational market where they can buy and sell power demand [Ygge and Akkermans, 1996; 1997, 1998]. Individual equipment agents communicate and negotiate, in a free-market bidding like manner, to achieve energy and cost savings (Figure 6). This is a decentralized way to reduce unwanted peak loads.

Informally, the task distribution over agents is as follows. To begin with, a software agent representing the utility (say, at the level of a secondary substation) announces the start of a load management action to the customer agents (which may represent a smart electricity meter in a household or plant or equipment beyond that such as radiators). For example, if its goal is to

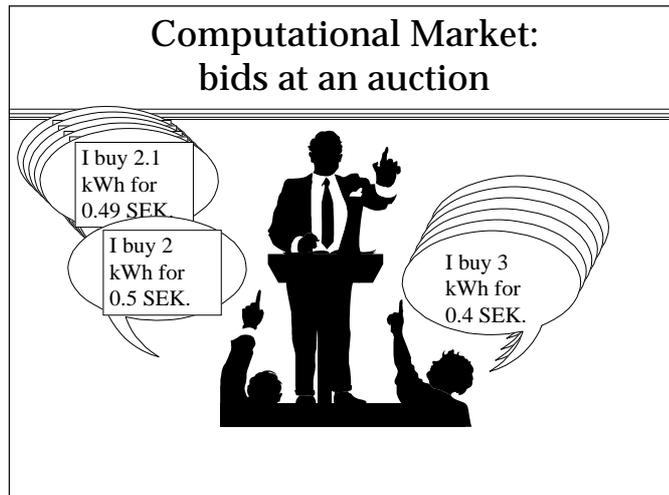


Figure 6: *Distributed load management is implemented in terms of an auction, whereby software agents representing the utility and the customers bid and negotiate to buy and sell power demand.*

reduce current energy consumption, it may offer a price or tariff different from the usual one. The customer agents then determine to what extent they are interested in participating in the load management action. This is based on the customer's preferences and is, of course, also changeable and programmable by the customer. On this basis, the customer agents prepare bids to sell some power (that is: to postpone or reduce energy use) in return for a financial rebate as offered by the utility, cf. Figure 6.

The totality of bids is then assessed in an auction as in a free, competitive market. The auction results in a reallocation of the available power. In our system, power is treated as any resource or commodity that is traded on a market. In a load management action there is a certain (limited) supply of it. Both the utility and the customer agents also have a certain demand for it, for which they are willing to pay a certain price. How much everyone gets, and at what price, is determined automatically in the auction.

In realizing an auction on the computer, we employ long-established formal theory on the functioning of competitive markets, which is available from economic science (especially from the field known as micro-economic theory). Customer preferences are in this framework expressed in terms of so-called utility functions. They represent in a numerical way the value for the customer for getting a certain amount of power: the higher the number, the higher the demand. Due to its rigorous mathematical form, this theory is readily adaptable for implementation on a computer. The corresponding algorithms that calculate the market equilibrium have been adapted from numerical analysis and optimization (since market mechanisms can be reformulated as a kind of optimum search problems). For discussions of these technical issues, see e.g. [Ygge and Akkermans, 1996, 1997, 1998].

Market negotiation and computation continues until a market equilibrium is established. This is the case when supply becomes equal to demand in the auction process. Then, each participating agent achieves the best possible deal in terms of obtaining power use versus spending financial budget. Economic market equilibrium can be proven to correspond to the optimum allocation of available power over all involved equipment agents. No agent will then gain any further by buying or selling power, and so the load management action as a market process is completed.

After the auction has been completed, its outcomes — that is, the allocation of power corresponding to the market equilibrium — are awarded and communicated to all agents involved. Next, the loads are scheduled in accordance with the awarded power over some agreed period (say, the next hour). This is implemented through appropriate on/off switching of the involved loads, whereby power-line communication will typically play an important role. Finally, agreed results as well as implemented outcomes are monitored by all parties, providing a data base of the facts needed in the contracts between utility and customer. This whole process is carried out automatically.

Homebots agent communication plan. This informal task description leads us in a straightforward way to a top view of the communication plan in the HOMEBOOTS system, as seen in the dialogue diagram of Figure 7. The important transactions, with their input/output information objects, in this announce-bid-award computational market scheme are the following.

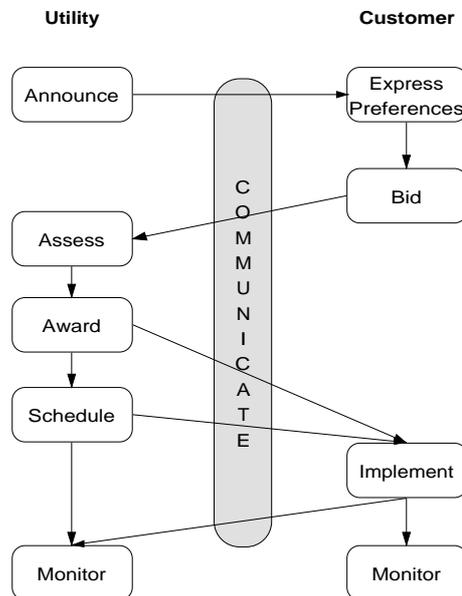


Figure 7: Dialogue diagram of the HOMEBOOTS system: tasks in the power auction for electricity load management, with their communication links.

1. *Kick-off the auction*: sends a trigger signal to the customer agents to commence a load management action.
2. *Submit the bids*: transmits the bids from the customer agents to the auctioneer for further processing.
3. *Present the awarded power allocation*: informs the customer agents about the end results of the auction.
4. *Present the associated real-time schedule*: provides the customer agents with the calculated schedule that implements the awarded allocation.
5. *Receive the resulting real-time implementation data*: transmits the actual metering data. This is needed for billing as well as for assessment of the need for further load management actions.

For simplicity, we have given the simplest possible task distribution and agent architecture. Other architectures, based on other scenarios and energy service set-ups, are very well possible but fall outside the scope of this paper. Figure 7 thus only intends to show the basics of a load management scenario. In this basic scenario, control within the communication plan is straightforward, as it follows the information flow from the subtasks. The top-level control is shown in Figure 8 with the help of a state-transition diagram. Agent task/transaction pairs are indicated by an ampersand (Announce & Kick-off, Bid & Submit, Award & Present). Only the auction part of the load management action has been given in the Figure. Generally, a state-based representation is convenient, as the formal semantics of communication languages such as KQML [Finin et al., 1997] is based upon agent states.

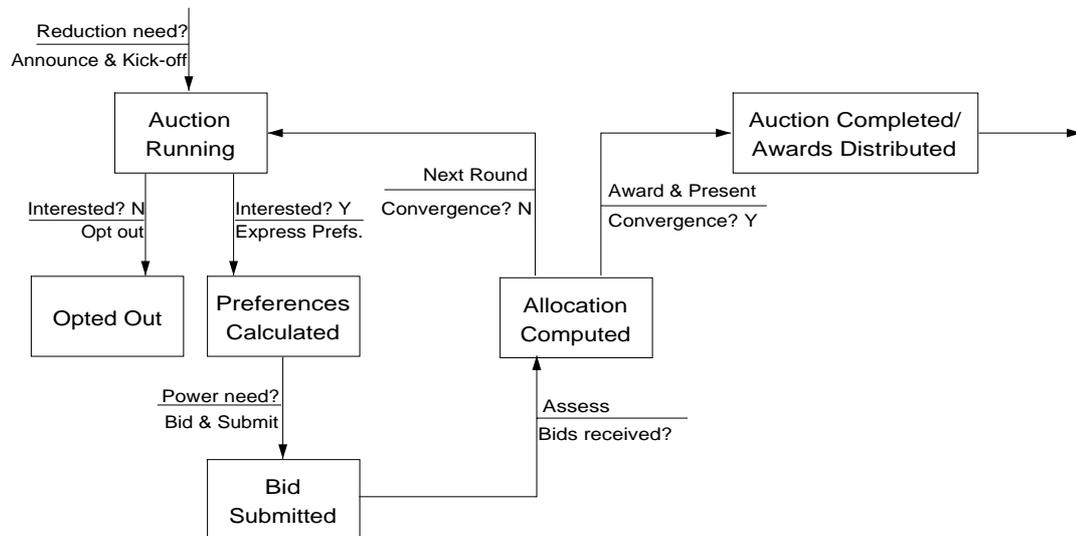


Figure 8: Communication plan control in the auction process of the HOMEBOOTS system in state-transition diagram form.

5 Transactions between Agents

Having finished the communication plan, we have the agent-to-agent dialogue, in terms of the transactions that form the communicative go-betweens linking two tasks. At this point, we do have the set of transactions, the related communication flow of information objects, and the control over the transactions. However, we have not yet specified much about the individual transactions themselves, other than that they have an identifier and name stating what their communicative purpose is. Hence, the second layer of the Communication Model contains a description of the properties of individual transactions.

Specification of individual transactions. The elements needed to specify an individual transaction are shown in the Transaction Description *Worksheet CM-1* displayed in Table 1. Most of it is rather self-explanatory; in the next section we will see a practical example. Collecting all this information within a single worksheet helps to make the transaction description self-contained, and thus more suitable for inspection and review. More generally, the current version of the CommonKADS methodology employs worksheets at several points: they are simple to use and provide

a standardized information base for specifications and decisions.

Communication Model		Transaction Description Worksheet CM-1
Transaction Identifier/Name	Identifi-	A transaction is to be defined for each information object that is output from some leaf task in the Task Model or in the Knowledge Model (i.e. a transfer function), and that must be communicated to another agent for use in its own tasks. The name must reflect, in a user-understandable way, what is done with this information object by the transaction. In addition to the name, give a brief explanation here of the purpose of the transaction.
Information object		Indicate the (core) information object, and between which two tasks it is transmitted.
Agents involved		Indicate the agent that is sender of the information object, and the agent receiving it.
Communication plan		Indicate the communication plan of which this transaction is a component.
Constraints		Specify the requirements and (pre)conditions that must be fulfilled so that the transaction can be carried out. Sometimes, it is also useful to state postconditions that are assumed to be valid after the transaction.
Information Exchange Specification	Exchange	Transactions can have an internal structure, in that they consist of several messages of different types, and/or handle additional supporting information objects such as explanation or help items. This is detailed in Worksheet CM-2. At this point, only a reference or pointer needs to be given to a later info exchange specification.

Table 1: Worksheet CM-1: Specifying the transactions that make up the dialogue between two agents in the Communication Model.

Detailing the Information Exchange: Communication Types. After the communication plan and the transaction description, the information exchange specification constitutes the third layer of the CommonKADS Communication Model. It contains the lowest level of detail. As such, it provides several important inputs for the Design Model and associated knowledge system implementation. This is also done in *worksheet* form. An example is given in the next section.

The most important aspect here concerns the *communication type*. Transaction sentences are often composite and convey, in one shot, different types of information. As an everyday example, consider the following sentence, viewed as a transaction between two agents: “*I’m getting cold, so could you please shut the window?*”. Such a sentence consists of several messages that, moreover, have a different intent. The first part is strictly speaking just an *information or notification* message, stating that the speaking agent apparently does not find the current temperature comfortable anymore. Note that this notification message does not necessarily imply any action, on the part of either agent. In contrast, the second part is directly aimed at eliciting activity by the other agent, here in the form of a *request* for action.

This aspect can be practically specified, as is the case in many agent communication models and systems, by associating each message with a set of *predefined communication types*. The communication types distinguished in the CommonKADS Communication Model are presented in Table 2. They have been adapted from work by the Daimler-Benz company in this area [Haddadi, 1996]. A more extensive, but somewhat less structured set of performatives is used in KQML [Finin et al, 1997]. In organizing the communication types, we use two dimensions: the first dimension is the purpose of a message — task delegation, task adoption, or pure information exchange —, while the second dimension indicates the degree of commitment or strength one is striving for this purpose. So, in a nutshell one may say that for typing the intention of a message, we have that $intention = purpose \times commitment$. In the Table, this leads to $3 \times 4 = 12$ predefined communication types. The semantics of the communication types in Table 2 is quite straightforward, and discussed elsewhere [Haddadi, 1996; Schreiber et al., 1998].

Communication Model	Predefined Communication Types		
Task Delegation			
Request	Require	Order	Reject-td
Task Adoption			
Propose	Offer	Agree	Reject-ta
Pure Information Exchange			
Ask	Reply	Report	Inform

Table 2: Predefined communication types, used in specifying the intention (intended effect on the receiving agent) with which a message is sent.

We now have at our disposal a rather rich vocabulary to specify the intention of messages. (Other, even richer proposals exist in the agent software literature, but the present one does cover a wide range of practical knowledge systems.) It is also seen that only giving the (propositional) content of messages is very limited, and that additional, explicit specification of intention greatly improves the understanding of communicative acts. And this is what the Communication Model aims at.

6 The HOMEBOTS system example continued

The HOMEBOTS system contains several transactions, as discussed above. Some are rather simple, especially the transaction linked to the *Announce* task serving to kick-off the auction (cf. Figures 7 and 8). The second transaction, whereby the bids are calculated (*Bid* task carried out by the customer agents) and subsequently are submitted to the auctioneer, is much more interesting. For reasons of space, we only treat this transaction in some detail, see Table 3.

Communication Model	Transaction Description Worksheet CM-1: Homebots System	
Transaction Identifier/Name	Identifi-	Transaction 2: Submit-the-bid: <i>transmits the bids from the customer agents to the auctioneer for further processing. The auctioneer then does intermediate calculations (of the going price and/or of the going allocation), and on its turn transmits these to the customer agents to enable bid revision.</i>
Information object		<i>Linking the Bid and Assess tasks: (1) customer bids; (2) intermediate auctioneer calculations (going price or going allocation).</i>
Agents involved		<i>Customer agents, sending bids and receiving the going market data; the auctioneer agent (in base version identified with the utility agent) receiving bids and sending out the going market data.</i>
Communication plan		<i>Homebots (base version).</i>
Constraints		<i>During the transaction a decision procedure (e.g. how often to retry and how long to wait) is needed telling what to do when agents do not submit bids, as this might be due to a communication failure. A postcondition derives from the convergence condition for market equilibrium (e.g. all agent bid prices must have become the same).</i>
Information Exchange Specification		<i>see the Worksheet CM-2 below.</i>

Table 3: Worksheet CM-1: The submit-the-bid transaction in the HOMEBOTS system.

We thus see that the submit-the-bid transaction is composite. It is a single transaction, because it is an exchange link between two tasks, but it handles more than one core information object. Both types of agents are problem solving and reasoning agents, both are acting as sender and receiver in this transaction, and both hold part of the overall initiative. This is a prototypical multi-agent

situation that contrasts with the one usually encountered in conventional knowledge systems. A more detailed specification of the information exchange is given in Table 4.

Communication Model	Information Exchange Specification Worksheet CM-2: Homebots system
Transaction Identifier/Name	Transaction 2: Submit-the-bid
Agents involved	<ol style="list-style-type: none"> 1. Sender (customer): bid. 2. Receiver (auctioneer): bid. 3. Sender (auctioneer): going market data. 4. Receiver (customer): going market data.
Information items	<p>We have for bids as well as for going market data (price and/or allocation):</p> <ol style="list-style-type: none"> 1. Role: both are core information objects. No support items are defined in this transaction. 2. Form: both are normally numerical data strings (real). Depending on the problem solving method chosen, they might be scalar, or a pair of reals. Several variants are possible, and this has an impact on computational efficiency and communication speed, see [Ygge and Akkermans, 1996, 1998]. 3. Medium: not of interest here, as the agent-agent interaction is fully electronic within a standard software environment and communication protocol.
Message Specifications 1. BID-MESSAGE 2. OPT-OUT-MESSAGE 3. AUCTION-DATA-MESSAGE 4. NEXT-ROUND-MESSAGE	<p>Type: PROPOSE Content: bid (structure depends on reference theory) Reference: market theory, e.g. [Ygge and Akkermans, 1996, 1998] From: customer agents To: auctioneer</p> <p>Type: REJECT-ta Content: no further participation in bidding From: customer agents To: auctioneer</p> <p>Type: INFORM Content: going market data (see reference theory) Reference: theory underlying market protocol From: auctioneer To: customer agents</p> <p>Type: REQUEST Content: trigger signal for new round of the auction From: auctioneer To: customer agents</p>
Control over messages	See pseudo-code specification in main text.

Table 4: Worksheet CM-2: The submit-the-bid messages and their communication types in the HOMEBOOTS system.

For the control specification of the submit-the-bid transaction in Worksheet CM-2, we may use (as an alternative to state transition diagrams) a pseudo-code form, see Figure 9.

In multi-agent systems, transactions related to task adoption and delegation come into play. In contrast, conventional knowledge systems, for example for diagnosis of technical systems (such as the NESTOR system discussed in [Waern et al., 1993]) are characterized by the fact that most communication refers to basic information exchange. Core information objects are either simply provided or delivered (INFORM message type), or exchanged through a question-and-answer pattern (ASK/REPLY communication types). Supporting information items such as help and explanation texts belong to a different type, however. Since such items are only presented as an open

```

REPEAT
WHILE <market convergence condition not satisfied>
  IF <interest in load management>
    THEN PROCESS(bid-task);
        SEND(BID-MESSAGE)
    ELSE SEND(OPT-OUT-MESSAGE)
  END-IF
  IF <bids received>
    THEN PROCESS(assess-task)
    ELSE PROCESS(decision subprocedure [e.g. WAIT...])
  END-IF
  SEND(AUCTION-DATA-MESSAGE)
  &
  SEND(NEXT-ROUND-MESSAGE)
END-REPEAT

PROCESS(award-task)
(et cetera).

```

Figure 9: Control specification of the submit-the-bid transaction in the HOMEBOTS system.

option to the user, the related support item messages are typically of the OFFER communication type. Hence, for intelligent multi-agent systems we need a richer repertoire in communication modelling.

7 Related Work and Conclusion

Summarizing, we have sketched how intelligent agent communication can be specified in a structured analysis way. A communication model is stepwise constructed, by means of three consecutive layers: overall communication plan, individual transactions, and detailed information exchange specification. This process is easily integrated into a wider framework of knowledge systems analysis [Schreiber et al., 1998]. Our approach extends earlier communication modelling for knowledge systems [Waern et al., 1993].

In knowledge-based approaches, communication modelling has a bias towards the traditional single system–single user situations [Schreiber et al, 1994; Waern et al, 1993]. In multi-agent communication languages, such as KQML [Finin et al., 1997] and COSY [Haddadi, 1996], the emphasis is at the level of individual messages, along with a relative neglect of overall task and knowledge modelling. Likewise, multi-agent applications often focus on the algorithmic more than the conceptual, knowledge-level aspects, e.g. [Ygge and Akkermans, 1996, 1997, 1998]. In general, we do not yet see much integration between the respective achievements of multi-agent and knowledge-based systems modelling. First attempts in this direction [Glaser, 1996; Jonker and Treur, 1997] are only very recent.

Clearly, our approach incorporates many advances due to other work. In particular, this holds for notions such as message typing, well-known from agent software [Finin et al., 1997; Haddadi, 1996], but also from CSCW, e.g. [Malone et al., 1993]. The idea of our work, however, has not been so much to advance the *contents* of agent communication specification, but to improve the state of the art in terms of a *structured analysis process*. Here, we have shown how to do this in

a stepwise fashion. Also, we have indicated a number of simple but useful techniques (diagrams, worksheets) and guidelines for communication modelling, and we have shown how they work in a realistic large-scale application.

References

- [Akkermans et al., 1996] J.M. Akkermans, F. Ygge, and R. Gustavsson: *HOMEBOOTS: Intelligent Decentralized Services for Energy Management*, in J.F. Schreinemakers (Ed.): *Knowledge Management: Organization, Competence and Methodology* (Proceedings Fourth International Symposium on the Management of Industrial and Corporate Knowledge ISMICK'96, Rotterdam, NL, 21-22 October 1996), pages 128–142, Ergon Verlag, Würzburg, D, 1996. ISBN 3-932004-26-4.
- [Akkermans and Ygge, 1997] J.M. Akkermans and F. Ygge: *Smart Software as Customer Assistant in Large-Scale Distributed Load Management*, in Proceedings DA/DSM DistribuTech 97 Europe (Amsterdam, The Netherlands, October 14-16, 1997), PennWell Energy and Utility Group, Utrecht, NL, 1997.
- [Finin et al., 1997] T. Finin, Y. Labrou, and J. Mayfield: *KQML as an Agent Communication Language*, in J.M. Bradshaw (Ed.): *Software Agents*, pages 291–316, AAAI Press/MIT Press, Menlo Park, CA, 1997.
- [Glaser, 1996] N. Glaser: *Contribution to Knowledge Acquisition and Modelling in a Multi-Agent Framework — The CoMoMAS Approach*, PhD Thesis Université Henry Poincaré, Nancy, F, December 1996.
- [Haddadi, 1996] A. Haddadi: *Communication and Cooperation in Agent Systems*, Springer-Verlag, Berlin, 1996. Lecture Notes in Computer Science Series No. 1056. ISBN 3-540-61044-8.
- [Jonker and Treur, 1997] C. Jonker and J. Treur: *Modelling an Agent's Mind and Matter*, in M. Boman and W. Van de Velde (Eds.): *Multi-Agent Rationality*, (Proceedings European Conference on Modelling Autonomous Agents in a Multi-Agent World MAAMAW'97, Ronneby, Sweden, May 23–26, 1997), pages 210–233. Springer-Verlag, Berlin, 1997. Lecture Notes in Computer Science Series No. 1237. ISBN 3-540-63077-5.
- [Malone et al., 1993] Th.W. Malone et al.: *The Information Lens: An Intelligent System for Information Sharing and Coordination*, in R.M. Baecker (Ed.): *Readings in Groupware and Computer-Supported Cooperative Work*, pages 461–473, Morgan Kaufmann, San Francisco, CA, 1993.
- [Schreiber et al, 1994] A.Th. Schreiber, B.J. Wielinga, J.M. Akkermans, W. Van De Velde and R. de Hoog: *CommonKADS — A Comprehensive Methodology for KBS Development*, IEEE Expert 9(6) (1994) pages 28–37. ISSN 0885-9000.
- [Schreiber et al., 1998] A.Th. Schreiber, J.M. Akkermans, A.A. Anjewierden, R. de Hoog, W. Van De Velde and B.J. Wielinga: *Engineering and Managing Knowledge — The CommonKADS Methodology*, Draft text book, version 0.6, Amsterdam, NL, 1998.
- [Waern et al., 1993] A. Waern, K. Höök, R. Gustavsson and P. Holm: *The COMMONKADS Communication Model*, Esprit Project P5248 Deliverable D.M.3.1b, SICS, Stockholm, SE, 1993.
- [Ygge and Akkermans, 1996] F. Ygge and J.M. Akkermans: *Power Load Management as a Computational Market*, in M. Tokoro (Ed.): *Proceedings Second International Conference on Multi-Agent Systems ICMAS'96* (Kyoto, Japan, December 10-13, 1996), pages 393–400. AAAI Press, Menlo Park, CA, 1996. ISBN 0-1-57735-013-8.
- [Ygge and Akkermans, 1997] F. Ygge and J.M. Akkermans: *Making A Case for Multi-Agent Systems*, in M. Boman and W. Van de Velde (Eds.): *Multi-Agent Rationality*, (Proceedings European Conference on Modelling Autonomous Agents in a Multi-Agent World MAAMAW'97, Ronneby, Sweden, May 23–26, 1997), pages 156–176. Springer-Verlag, Berlin, 1997. Lecture Notes in Computer Science Series No. 1237. ISBN 3-540-63077-5.
- [Ygge and Akkermans, 1998] F. Ygge and J.M. Akkermans: *On Resource-Oriented Multi-Commodity Market Computations*, in Proceedings Third International Conference on Multi-Agent Systems ICMAS'98 (Paris, France, 4–7 July 1998), in press, 1998.
- [Yourdon, 1989] E. Yourdon: *Modern Structured Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1989.