

***Research Report 12/98***



# **Pragmatics of 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—98/12—SE

## **Pragmatics of Agent Communication**

by Hans Akkermans, Rune Gustavsson, Fredrik Ygge,

ISSN 1103-1581

ISRN HK/R-RES—98/12—SE

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

All rights reserved

Printed by Psilander Grafiska, Karlskrona 1998

# Pragmatics of Agent Communications

**Hans Akkermans** §      **Rune Gustavsson** ‡      **Fredrik Ygge** ‡

§AKMC Knowledge Management and Free University Amsterdam  
Department of Computer Science  
De Boelelaan 1081a, NL-1081 HV Amsterdam, The Netherlands  
HansAkkermans@compuserve.com

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

## Abstract

The process of agent communication modeling 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. In general, a much better integration between the respective achievements of multi-agent and knowledge-based systems modeling is an important research goal. In this paper, we describe how agent communications can be specified as an extension of well-known knowledge modeling techniques. The emphasis is on showing how a structured process of communication requirements analysis proceeds, based on existing results from agent communication languages. The guidelines proposed are illustrated by and based on a large-scale industrial multi-agent application for distributed energy load management in industries and households, called *Homebots*. Homebots enable cost savings in energy consumption by coordinating their actions through an auction mechanism.

## 1 Introduction

The process of agent communication modeling 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.

In knowledge-based approaches, communication modeling has a bias towards the traditional single system–single user situations [Schreiber et al, 1994; Waern et al, 1993]. In multi-agent commu-

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

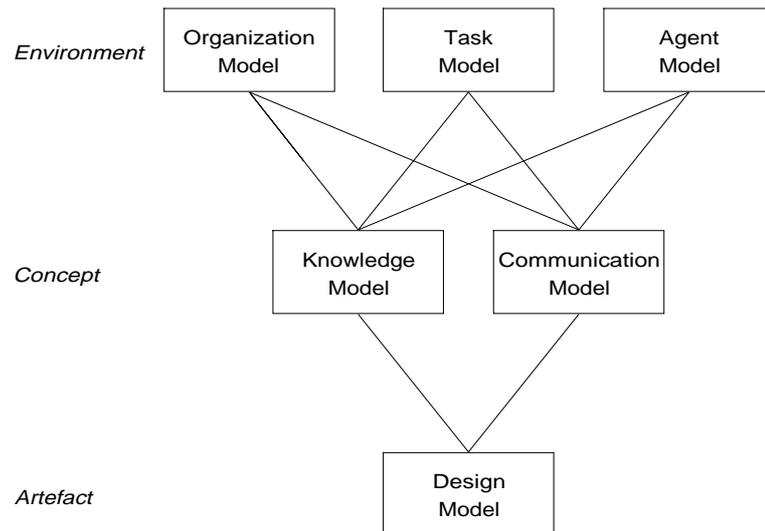


Figure 1: *The CommonKADS Model Suite.*

In this paper, we describe how agent communications can be specified as an extension of well-known knowledge modeling techniques. We do so in the framework of the COMMONKADS methodology [Schreiber et al, 1994, 1997] which provides a comprehensive conceptual modeling approach, ranging from organizational analysis to system design and implementation. Figure 1 shows the associated CommonKADS model suite. Our work emphasizes the semi-formal and pragmatic side, and extends earlier communication modeling for knowledge systems [Waern et al, 1993]. The guidelines proposed are illustrated by and based on a large-scale industrial multi-agent application for distributed energy load management in industries and households, called HOMEBOTS [Akkermans and Ygge, 1996, 1997]. Homebots enable cost savings in energy consumption by coordinating their actions through an auction mechanism.

Apart from the highly relevant application itself, the main contribution of our paper is to show how a structured *process* of communication requirements analysis proceeds, based on existing results from agent communication languages. There is a clear need for this if intelligent agent systems are to be taken up by industry on a large scale.

## 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.

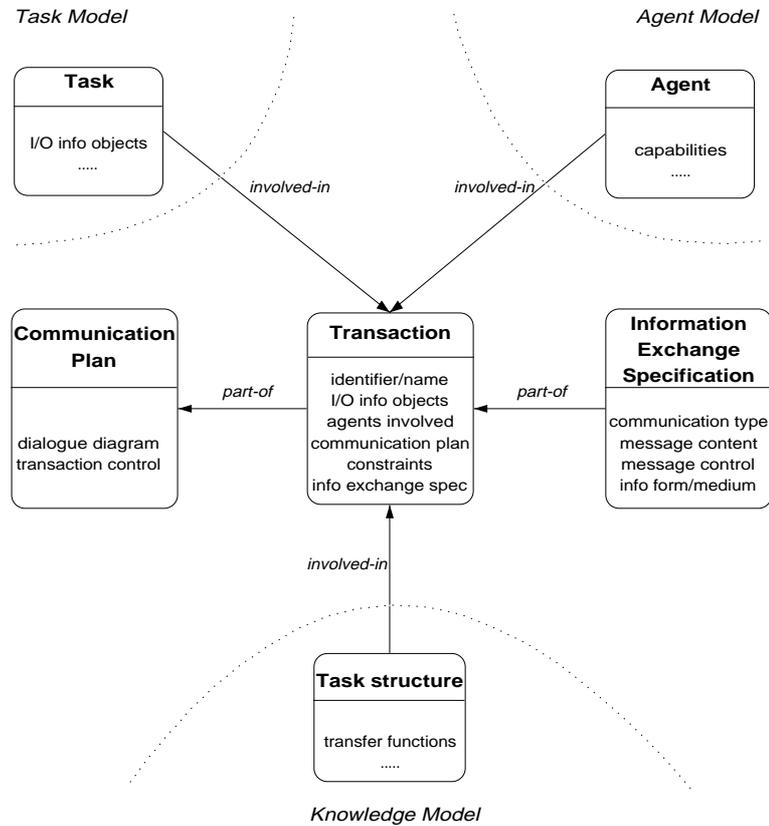


Figure 2: Overview of the Communication Model and how it relates to the other COMMONKADS models.

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.

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 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 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.

#### 3.1 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 crucially depends on information from other COMMONKADS models. In order to start with communication modeling, the following information needs to be available:

- From the Task Model, the list of tasks carried out by the considered agent. For the Communication Model, we are interested in the leaf tasks, i.e., those that are not decomposed further, together with their input/output information objects.
- From the Knowledge Model, the set of so-called transfer functions, that is, leaf nodes in the task/inference structure that depend on data from or deliver reasoning results to the outside world. (Recall that a task/inference structure in the Knowledge Model is a refinement of a non-leaf, knowledge-intensive task stemming from the Task Model).
- From the Agent Model, the description of relevant agents, with their knowledge (or more generally: capabilities), responsibilities and constraints. The Communication Model must

be constructed such that it satisfies the ensuing agent requirements, but on its turn it may also add requirements for communicative capabilities of an agent.

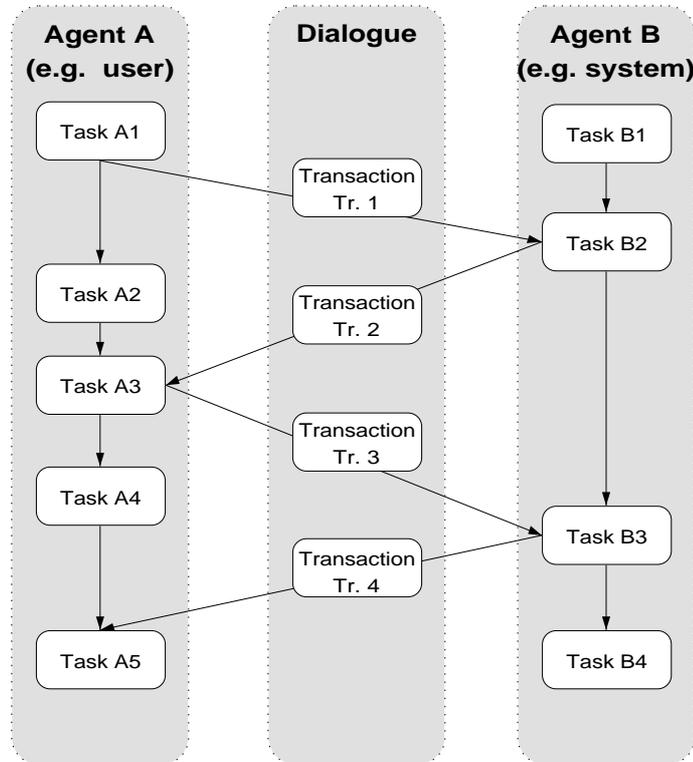


Figure 3: *The general lay-out of a dialogue diagram. It forms the central part of the communication plan, as it shows the overall information flow related to agent communication.*

This is depicted in Figure 2. Thus, normally the knowledge engineer will have already done a significant part of task, agent and knowledge analysis, before starting with communicating modeling. This also follows by looking at the main steps in constructing the communication plan.

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.

### 3.2 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.

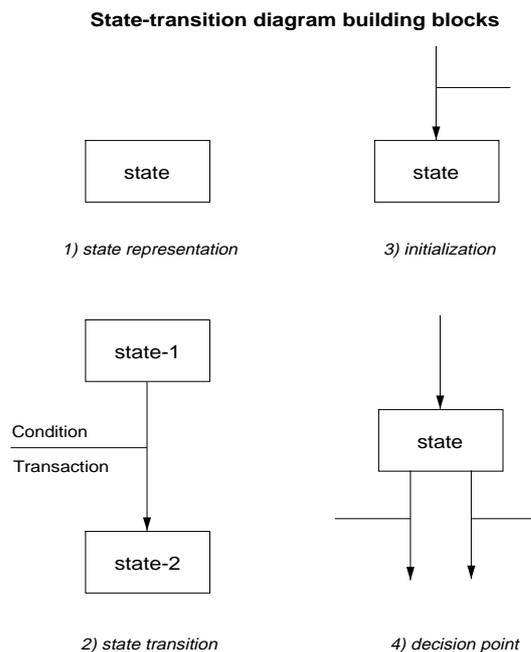


Figure 4: *Legenda of state-transition diagram constructs, used in the Communication Model to specify control over transactions and messages.*

As these are well-known techniques from software and information engineering, there is no need to give a long-winded elaboration of them. In Figure 4 we give a legenda of state-transition diagram constructs as we use them in the Communication Model. Likewise, Table 1 contains the constructs, i.e. basic communication operators and control constructs, specialized to the Communication

Communication Model	Specifying control over transactions and messages	
<i>Construct</i>	<i>Arguments</i>	<i>Description</i>
SEND	(transaction or message)	<i>Elementary communication operator.</i>
RECEIVE	(transaction or message)	<i>Elementary communication operator.</i>
CARRY-OUT	(transaction)	<i>SEND/RECEIVE combination.</i>
WAIT-until/while	(condition)	<i>Represents the null action in communication.</i>
PROCESS	(task)	<i>Part of agent tasks outside the Communication Model.</i>
;	(transaction-1; transaction-2)	<i>SEQUENCE operator, elementary control construct. (Similarly for messages).</i>
REPEAT-until/while	(condition)	<i>Elementary control construct.</i>
IF THEN ELSE	(condition) (transaction-1) (transaction-2)	<i>Elementary control construct. Similarly for messages.</i>
&	(transaction-1 & transaction-2)	<i>AND operator. (Similarly for messages).</i>
	(transaction-1   transaction-2)	<i>CHOICE operator, denoting exclusive either/or operation. (Similarly for messages).</i>
∨	(transaction-1 ∨ transaction-2)	<i>OR operator, denoting non-exclusive either/or operation. (Similarly for messages).</i>
, .		<i>Syntactic separators for the control specification.</i>

Table 1: Specifying control over transactions and messages in the Communication Model, by means of basic communication operators and control constructs in pseudo-code form.

Model. We note that this approach to control will be used both for specifying the control over transactions, and for specifying control within transactions, the internal structure of which may contain different messages of different types (as we will see later on in discussing the third layer of the Communication Model). Below we discuss a practical application.

## 4 Case: 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 5). 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. This paradigm shift will open up new opportunities, but will also necessitate new ways of thinking for most utilities, as it requires two-way communication between the utility and the customer. Here, utilities are facing the fact that proper utilization of information and knowledge is a key component in a competitive market. The traditional power distribution net must be supplemented with an information network allowing

for extensive two-way communication between customers and the utility, in order to provide the new services as mentioned above. Information and communication technologies (ICT) are crucial enablers here.



Figure 5: *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. There are quite a number of different advanced information technologies that jointly act as enablers here, such as: (i) cheap programmable chips that can be built in into many types of equipment; (ii) advanced telecommunications technology; (iii) knowledge and software engineering: object and knowledge technology and multi-agent systems; (iv) emerging facilities and standards for using the power grid (also) as an integrated information infrastructure.

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.

This will provide the supplying utility with new opportunities for power load management and demand saving in the distribution grid. Better load management and demand saving has a significant impact on reducing and postponing investments by utility industries. At the same time, it serves the customer’s interest, since it allows for cost reduction by taking advantage of tailored and more flexible tariffs and client contracts.

**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.

Our approach is to achieve dynamic and automatic load balancing by means of software agent technology. Devices can now be equipped with communication and information-processing capabilities, by supplying them with networked, communicating microprocessors together with smart software running on top of them, as depicted in Figure 6. In everyday language, it is now technologically possible that software-equipped communicating devices ‘talk to’, ‘negotiate’, ‘make

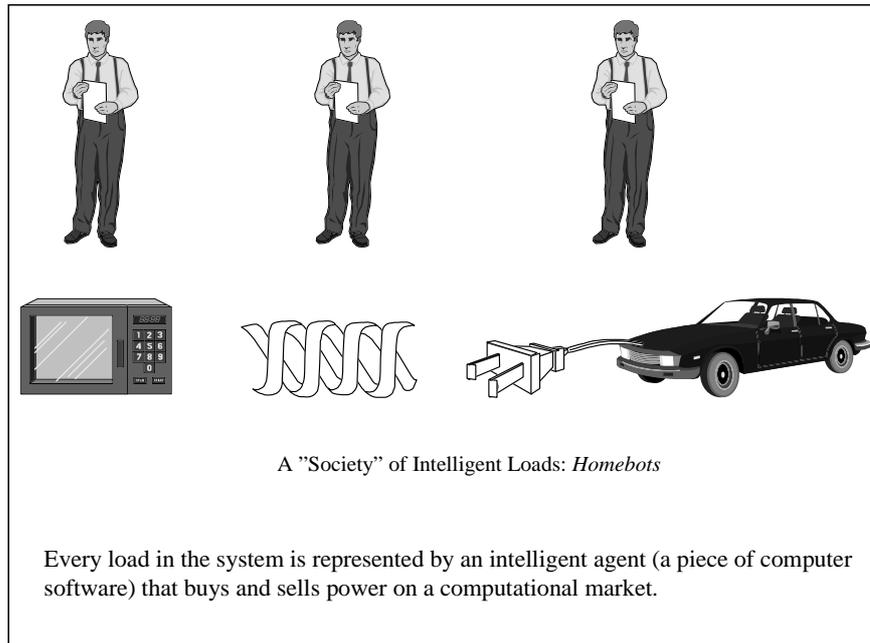


Figure 6: *Devices and loads are equipped with smart small software programs. These software agents communicate, act and cooperate as representatives assisting the customer, to achieve given goals such as power load management.*

decisions' and 'cooperate' with each other, over the low-voltage grid and other media. This enables radically new approaches to utility applications. We use this concept to achieve distributed load management in a novel fashion: by a cooperating '*society of intelligent devices*'. Knowledge plus communication are the ingredients of intelligence in systems.

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. 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]. Individual equipment agents communicate and negotiate, in a free-market bidding like manner, to achieve energy and cost savings both for the utility and the customer. The market models adapted from business such as auctions, offer promising concepts to automatically manage large distributed technical systems. 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 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

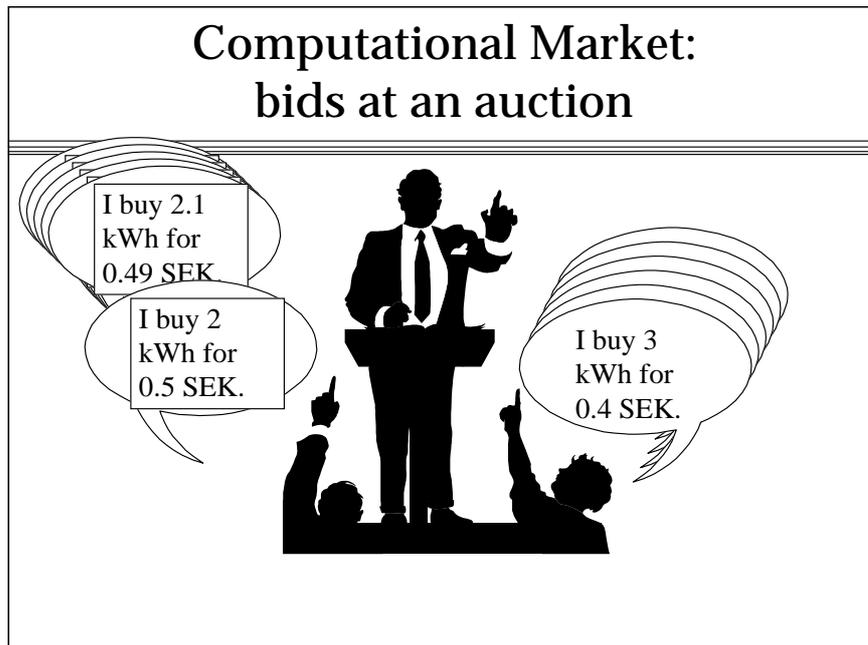


Figure 7: *Distributed load management is implemented in terms of an auction, whereby software agents representing the utility and the customers bid and negotiate to buy and sell power demand.*

power (that is: to postpone or reduce energy use) in return for a financial rebate as offered by the utility, cf. Figure 7.

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).

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 8. The important transactions, with their input/output information objects, in this announce-bid-award computational market scheme are the following.

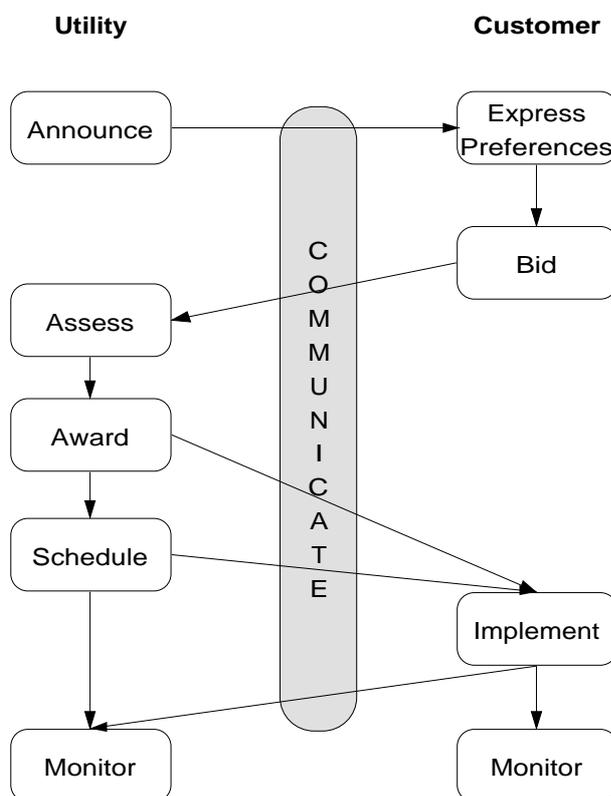


Figure 8: Dialogue diagram of the HOMEBOOTS system: tasks in the power auction for electricity load management, with their communication links.

- *Kick-off the auction:* sends a trigger signal to the customer agents to commence a load management action.

- *Submit the bids*: transmits the bids from the customer agents to the auctioneer for further processing.
- *Present the awarded power allocation*: informs the customer agents about the results of the auction.
- *Present the associated real-time schedule*: provides the customer agents with the calculated schedule that implements the awarded allocation.
- *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 and scenarios are very well possible. For example, it is probably preferable to separate the utility agent (representing the interests of the utility) from the actual auctioneer agent supervising the bidding process. In large-scale applications, customer agents will be hierarchically ordered. The initiative of various tasks can also be different. In so-called direct load management, the initiative to an auction lies with the utility agent, but in indirect load management the customer may take the initiative, though within a preset contractual framework. Also, the scheduling task can be allocated to agents in different ways. Our computational market approach is very flexible in this respect. Figure 8 thus only intends to show the basics of a load management scenario.

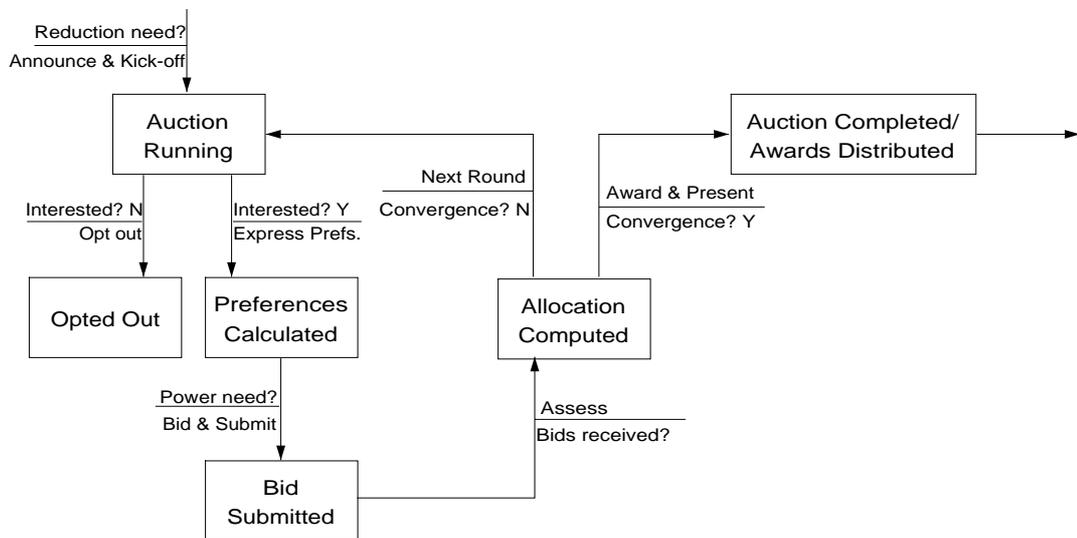


Figure 9: *Communication plan control in the auction process of the HOMEBOOTS system in state-transition diagram form.*

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 9 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 is based upon agent states.

## 5 Transactions between Agents

When we have 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.

### 5.1 Specification of individual transactions

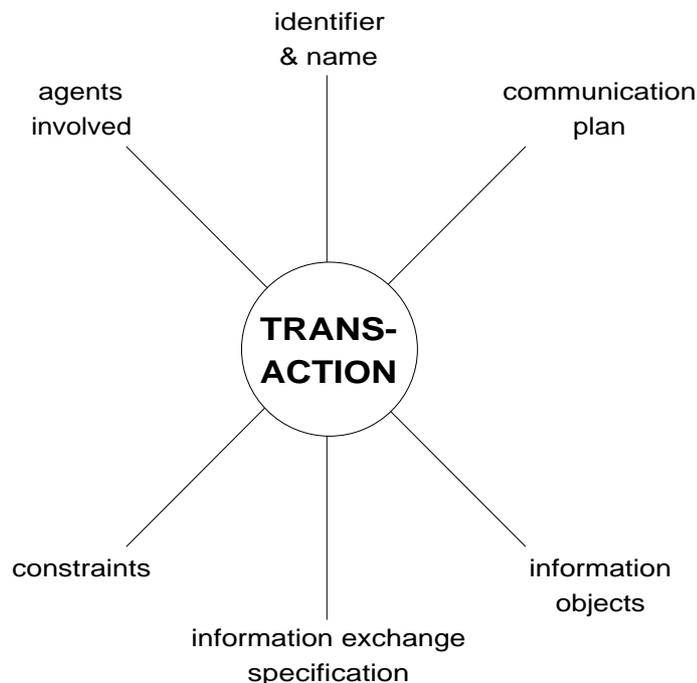


Figure 10: *The components that together specify a transaction in the COMMONKADS Communication Model.*

The elements needed to specify an individual transaction are shown in Figure 10. For the specification itself we can use the Transaction Description *Worksheet CM-1* displayed in Table 2. Most of it is rather self-explanatory. Collecting all this information within a single worksheet helps to make the transaction description self-contained, and thus more suitable for inspection and review.

If the communication plan has been properly constructed, some components of the transaction description easily follow from the dialogue diagram: in particular the name of the transaction, and the agents and tasks it links. In addition to a name, it is helpful to give a brief explanation of purpose and context of the transaction. The heart of a transaction, of course, is transmitting some information object, and this is noted in the Worksheet as the core information object (there may

Communication Model		Transaction Description Worksheet CM-1
Transaction Identifier/Name	Identifier	<i>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.</i>
Information object		<i>Indicate the (core) information object, and between which two tasks it is to be transmitted.</i>
Agents involved		<i>Indicate the agent that is sender of the information object, and the agent that is receiving it.</i>
Communication plan		<i>Indicate the communication plan of which this transaction is a component.</i>
Constraints		<i>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.</i>
Information Exchange Specification	Exchange	<i>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.</i>

Table 2: Worksheet CM-1: Specifying the transactions that make up the dialogue between two agents in the Communication Model.

also be auxiliary, non-core information items that have a facilitating role; explanations and help items are a common example). Furthermore, we indicate the communication plan of which this transaction is a component. Note that this is of course trivial when there is only one communication plan. However, in multi-agent systems, for example, there might be several communication plans covering different groups and types of agents.

The worksheet component concerning constraints is used to specify the preconditions that must be fulfilled before the transaction can be carried out. This might be various things, such as the availability of certain data measurements needed as input, a needed agent capability (e.g. sensory capabilities related to sound or vision) when relevant information comes in a certain form or medium, or the occurrence of an outside triggering event as is often the case in real-time embedded systems. Sometimes, it is also useful to state postconditions that are assumed to be valid after the transaction. In state administration and legal matters, it is usually assumed that ‘every citizen knows the law’ in all its often intricate detail, whether this is actually true or not. Likewise, a transaction may simply suppose that a transmitted information object is correctly received and processed by the receiving agent, without actually asking for an acknowledgement. That this is a non-trivial postcondition, and therefore worth reflecting about in communication modeling, is something we all are familiar with from lost letters in both regular and electronic mail.

The final component of a transaction description is called the information exchange specification. Basically, it gives the type, content and form of the message that ‘packages’ the information object that is transmitted. In very simple cases, for example when only data strings are exchanged between two systems, it can be sufficient to give the content of the message as a proposition or predicate statement here. However, it is well possible that a single transaction contains more than one message. For example, this occurs in a buying/selling negotiation task running between two

parties. Then, there is one transaction linking the buy and sell tasks, but this transaction has an internal structure consisting of a bid-react-rebid pattern of messages. Moreover, it is sometimes necessary to be able to state in what form or through which medium information in a transaction is conveyed. For all these reasons, the information exchange specification is usually not given directly as a basic component in Worksheet CM-1, but a reference is given instead to a separate (worksheet) description. How to describe this detailed information exchange specification is discussed in the next section.

## 5.2 Detailing the Information Exchange: Communication Types and Patterns

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, concerning agent communication protocols, messages, and human-computer interaction and interfacing. An information exchange specification refines the transaction description discussed previously, in two ways. First, it gives the internal message typing and structure of the transaction. Second, it gives syntactic form and medium information about the messages. This is done by means of a separate Information Exchange Specification *Worksheet CM-2*, given in Table 3.

Some of the information in this worksheet is already available in the transaction description, such as transaction name and involved agents. It is mainly incorporated to make the specification self-contained, and easier for review and inspection. Also, the core information object that is to be transferred is already found in the transaction description. This is, however, not the case with the so-called supporting information items. In many cases it is helpful to not only transmit the core information object that is needed in the agents' task structure, but also to facilitate better understanding of the transaction. This can be achieved by means of additional explanations, help texts, and the like.

The most characteristic element of our information exchange specification, however, is the way in which the messages that make up the transaction are described. The reason is that transaction sentences are often composite and convey, in one shot, different types of information. As an everyday example, let us consider the following sentence, viewed as a transaction between two agents: *"I'm getting cold, so could you please shut the window?"*. If we think a little about it, it becomes clear that this sentence actually consists of several messages that, moreover, have a different intent. Specifically:

1. The first part, *"I'm getting cold"*, is strictly speaking no more than a bare *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.
2. In contrast, the second part, *"Could you please shut the window?"*, is directly aimed at eliciting activity by the other agent, here in the form of a *request* for action.

So, within one transaction sentence, we have two messages here, differing in content as well as intent. That this is a rather general situation, will be clear upon reflecting about variations of the

Communication Model	Information Exchange Specification Worksheet CM-2
Transaction Identifier/Name	Give the transaction identifier and name of which this information exchange specification is a part.
Agents involved	Indicate the agent that is sender of the information items, and the agent that is receiving them. 1. <b>Sender</b> (agent-1); 2. <b>Receiver</b> (agent-2).
Information items	List all information items that are to be transmitted in this transaction. This includes the ('core') information object which transfer is the purpose of the transaction. However, it may contain other, supporting, information items, that for example provide help or explanation. For each information item, describe the following: 1. <b>Role</b> : whether it is a core object, or a support item. 2. <b>Form</b> : the syntactic form in which it transmitted to another agent , e.g. data string, canned text, a certain type of diagram, 2D or 3D plot, etc. 3. <b>Medium</b> : the medium through which it is handled in the agent-agent interaction, e.g. a pop-up window, navigation and selection within a menu, command-line interface, human intervention, etc.
Message Specifications	Describe all messages that make up the transaction. For each individual message, describe: 1. <b>Communication Type</b> : the communication type of the message describing its intention ('illocutionary force', in speech-act terminology), according to Table 4 and Figure 11. 2. <b>Content</b> : the statement or proposition contained in the message. 3. <b>Reference</b> : in certain cases, it may be useful to add a reference to, for example, what domain knowledge model or agent capability is required to be able to send or process the message.
Control over messages	Give, if necessary, a control specification over the messages within the transaction. This can be done in pseudo-code format or in a state-transition diagram, similar to how the control over transaction within the communication plan is specified. See for this Figure 4 and Table 1. The difference is just the level of detail.

Table 3: Worksheet CM-2: Specifying the messages and information items that make up an individual transaction within the Communication Model.

sketched transaction. Take for example the alternative sentence: “I’m getting cold, why were you so stupid to open the window?”. Or consider alternative answers to the original question such as “Of course, dear” versus “I’m watching this world champion football match on TV, so why don’t you do it yourself?”. Such statements have to be broken down into more than one message. To do otherwise even feels very artificial. It is easy to come up with other illustrations that are quite interesting from a Communication Model viewpoint. (As an example, we leave it to the reader to make a communication analysis of the following message: “It’s the economy, stupid!”).

It goes without saying that the pragmatics of human communication is often quite delicate. We believe, however, that considerations like those above are also relevant in the modeling of communication where information and knowledge systems are concerned. Nowadays, systems that involve multi-agent communication, such as information systems based on the Internet or the World Wide Web, have to confront such issues. In such systems, agent communication is often inspired by the so-called speech act theory, which distinguishes between the actual content (‘locutionary nature’)

of a speech act or message — what is actually being said — and its intended effect (‘illocutionary force’) upon the other agent.

This distinction is employed in many agent communication models and systems, and also in COMMONKADS. This can be practically done by associating each message with a set of *predefined communication types*, which must be filled in as indicated in Worksheet CM-2, cf. Table 3.

The communication types distinguished in the COMMONKADS Communication Model are presented in Table 4. 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, 1994]. 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.

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

Table 4: Predefined communication types, used in specifying the intention (intended effect on the receiving agent) with which a message is sent.

The semantics of the communication types in Table 4 is as follows.

- *Request/Propose*: refers to a message sent by an agent that sees a potential for cooperation, but wishes to negotiate on the terms of this cooperation. Loosely: ‘I have an interest, but not yet a commitment’.
- *Require/Offer*: refers to a message indicating that the sending agent already has made a pre-commitment, and intending to prompt the receiving agent for its commitment. This type thus denotes a conditional commitment.
- *Order/Agree*: the message types indicating that the agent has made a commitment, and thus will act accordingly in carrying out its tasks.
- *Reject-td/ta*: denotes that the agent does not want to commit or cooperate in task delegation or adoption.
- *Ask/Reply*: evidently refer to messages that have as intent a query for information from another agent, and delivery of such information in return.
- *Report*: types a message sent after an agent has acted towards a (previously) agreed task goal, with the intention to let the other agent know the status of achievement (e.g. success, failure, outcome of the action).

- *Inform*: refers to a message type, that just delivers, provides or presents information objects to another agent. It indicates an independent informative action, where no previous request or agreement is involved (in contrast to reply or report messages).

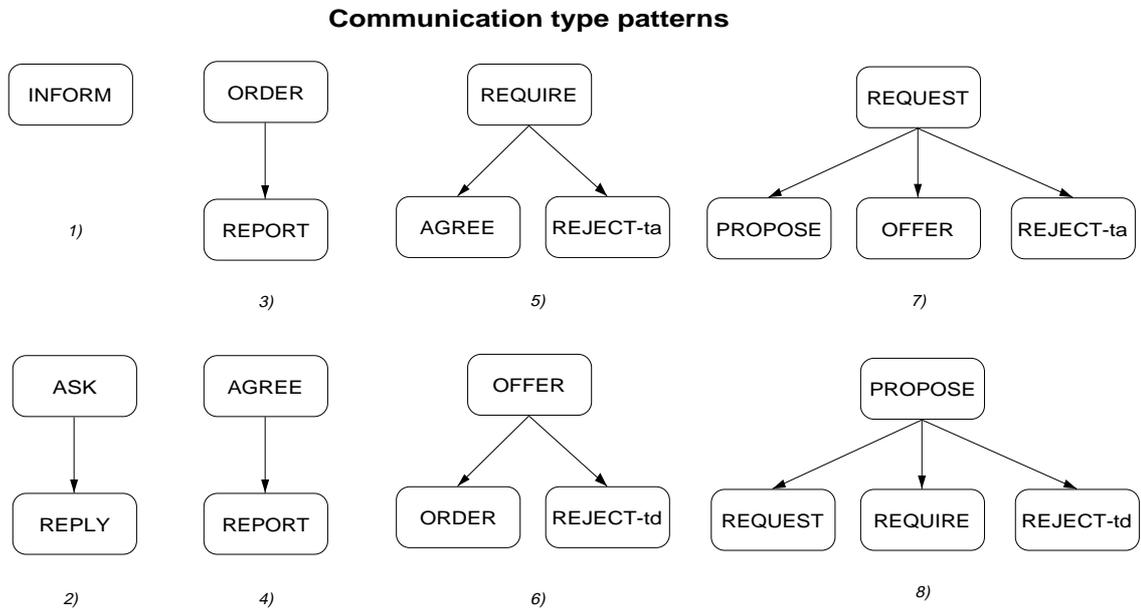


Figure 11: *Library of message patterns, built from the predefined communication types. Branching arrows indicate (exclusive) alternatives.*

We see that 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. This is further magnified by realizing that the above communication types not only are suitable for characterizing separate messages. The communication types lend themselves very well to construct typed patterns or chains of messages that naturally belong together. A library of such patterns, again based on work done at Daimler-Benz [Haddadi, 1996], is presented in Figure 11.

Question/answer patterns are a straightforward example occurring in many knowledge systems. Negotiation tasks and associated bidding protocols provide another, more complex, pattern. In the following section we will show an example of this, relating to our market-based program for energy management.

### 5.3 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 8 and 9). 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 5.

Communication Model		Transaction Description Worksheet CM-1: Homebots System
Transaction Identifier/Name		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 5: 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 typical 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 6.

For the control specification of the submit-the-bid transaction in Worksheet CM-2, we now use the pseudo-code form as presented in Table 1.

```

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).

```

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"> <li>1. <b>Sender</b> (customer): bid.</li> <li>2. <b>Receiver</b> (auctioneer): bid.</li> <li>3. <b>Sender</b> (auctioneer): going market data.</li> <li>4. <b>Receiver</b> (customer): going market data.</li> </ol>
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"> <li>1. <b>Role</b>: both are core information objects. No support items are defined in this transaction.</li> <li>2. <b>Form</b>: 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].</li> <li>3. <b>Medium</b>: not of interest here, as the agent-agent interaction is fully electronic within a standard software environment and communication protocol.</li> </ol>
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]            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 spec in main text.

Table 6: Worksheet CM-2: The submit-the-bid messages and their communication types in the HOMEBOTS system.

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

## 6 Guidelines for the Communication Modeling Process

Summarizing, we have sketched how the Communication Model specifies the information exchange between tasks carried out by different agents. It is stepwise constructed, by means of three consecutive layers: overall communication plan, individual transactions, and detailed information exchange specification. The communication plan describes the full dialogue between two agents. Transactions are the basic building blocks for a dialogue, and act as the go-between of two tasks carried out by different agents. Transactions on their turn may consist of one or more messages which are detailed in the information exchange specification. Predefined communication types and patterns allow to build up message protocols in a structured way.

**Guidelines for ‘balancing’ the Communication Model.** The relation of the Communication Model with the other models has been explained previously, and is also shown in Figure 2. On this basis, we have the following rules and guidelines on what the boundaries and connections should be of the Communication Model vis-à-vis the other models.

- Leaf tasks from the Task Model as well as the Knowledge Model are key inputs to the Communication Model, insofar as they handle information objects that must be exchanged between agents. (Such leaf tasks in the Knowledge Model are called transfer functions). The foremost rule for communication modeling says that a separate transaction must be defined for each information object exchanged, *and* for each distinct pair of leaf tasks.
- The Agent Model describes the agent capabilities (knowledge), responsibilities, and constraints. Check whether these are compatible with the constraints for the transactions in the Communication Model. It may be that communication requires additional capabilities from an agent. If so, add these by revising the Agent Model.
- As a double-check, verify whether the communication plan is compatible with structure, power/culture, process, and resources in the Organization Model.
- The rule of structure-preserving design in the Design Model also holds for the Communication Model constructs, in the same way as it holds for the Knowledge Model structure.

- A borderline case between the Design and Communication Models are the syntactic form and media aspects of information items in the detail information exchange specification. They might belong to either one. The demarcation criterion is that if there is an *intrinsic* conceptual reason that information items take on a certain form or are carried by a certain medium, this is to be modeled in the Communication Model. Otherwise, it is a matter of implementation choice, as a consequence of which it belongs to the Design Model. For example, some information objects ‘must’ come with a certain form or medium. For example, a signature authorizing a purchase or expense might not be considered legally valid if it is given in electronic form. Such a constraint is part of the Communication Model. The general rule is to model form and media aspects in the Design Model, *unless* there is a good conceptual reason *not* to.
- Decisions as to what supporting information items to introduce belong to the Communication Model, and not to the Design Model, because they are a matter of user task support, not system implementation.

**Guidelines for the construction process of the Communication Model.** In this paper, we have outlined the construction of an agent communication model, and illustrated this with a real-life industrial application example. The ensuing guidelines for the construction process are listed in Table 7.

<b>Main steps in constructing the Communication Model</b>
<i>1. Identify the core information objects to be exchanged between agents. Do this by checking, for each agent, the list of leaf tasks from the Task Model and the Knowledge Model (the transfer functions).</i>
<i>2. Identify the associated list of transactions, as exchange links between two tasks, and give each transaction a suitable, i.e. user-understandable, name.</i>
<i>3. Now, construct the dialogue diagram so that you have a pictorial overview of the overall communication plan. If needed, add a specification of the control over the transactions. This yields a complete communication plan.</i>
<i>4. Describe all individual transactions, following the format given in Figure 10 and Worksheet CM-1.</i>
<i>5. Describe the internal structure of each transaction where necessary, by filling in the information exchange specification according to Worksheet CM-2.</i>
<i>6. Validate and balance the Communication Model according to the techniques and guidelines given.</i>

Table 7: COMMONKADS *Communication Model construction guidelines.*

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*. This aspect of structured process methodology is still lacking also in the agent software area. Here, we have shown how to do this in a stepwise fashion. We have proposed 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.

## 7 References

[Akkermans et al., 1996]

J.M. Akkermans, F. Ygge, and R. Gustavsson: *HOMEBOTS: 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., 1994]

T. Finin, R. Fritzson, D. McKay and R. McEntire: *KQML as an Agent Communication Language*, in Proceedings of the Third International Conference on Information and Knowledge Management CIKM'94, ACM Press, 1994.

[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 0.5, 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.