A Multi-Agent System Performing One-to-Many Negotiation for Load Balancing of Electricity Use

Frances Brazier[1], Frank Cornelissen[1], Rune Gustavsson[2], Catholijn M. Jonker[1], Olle Lindeberg[2], Bianca Polak[1], Jan Treur[1]

[1]Vrije Universiteit Amsterdam
Department of Mathematics and Computer Science, Artificial Intelligence Group
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
URL: http://www.cs.vu.nl /~{frances,frankc,jonker,treur}. Email: {frances,frankc,jonker,treur}@cs.vu.nl

[2]University of Karlskrona/Ronneby (HK/R)
Department of Computer Science and Business Administration, Research Laboratory SIKT
URL: http:// www.sikt.hk-r.se. Email: {Rune.Gustavsson, Olle.Lindeberg}@ide.hk-r.se

Abstract

Emerging technologies allowing two-way communication between utility companies and their customers are changing the rules of the energy market. Deregulation makes it even more demanding for utility companies to create new business processes for the mutual benefit of the companies and their customers. Dynamic load management of the power grid is essential to make better and more cost-effective use of electricity production capabilities, and to increase customer satisfaction. In this paper, methods from Agent Technology and Knowledge Technology have been used to analyse, design, and implement a component-based multi-agent system capable of negotiation for load management. The proof-of-concept prototype system NALM (Negotiating Agents for Load Management) developed shows how under certain assumptions peaks in power load can be reduced effectively based on a negotiation process.

Keywords:

Negotiation, electronic market place, intelligent agent, load management

1. Introduction

In most European countries utility companies soon face competition: they may no longer be the sole providers of resources to both industry and the consumer market. Consumers will have the opportunity to choose between different providers. Price differences are bound to influence consumer preferences. A Swedish based consortium consisting of the utilities Sydkraft, PreussenElektra and Electricite de France have, together with the companies ABB and IBM and several universities, explored possibilities and impacts of information technologies in the energy market; e.g., [11]. Resource management, such as load balancing, is one of the first applications where a multi-agent approach (cf. [14]) has shown to be very promising, as presented in this paper. Agent Technology and Knowledge Technology have been used to address the problem. A model has been designed with a transparent component-based structure, based on explicit and formal specification of knowledge of negotiation process at a conceptual level. Due to these characteristics this model supports reusability in other, comparable problem domains. Based on this component-based model, the proof-of-concept prototype system NALM (Negotiating Agents for Load Management) has been implemented and evaluated.

The domain of load management is briefly described in Section 2. Different protocols for negotiation are discussed and the models of the negotiating agents used in the system are presented in Section 3. Section 4 provides more details of the design and implementation of the prototype system. In Section 5 the behaviour of the system is evaluated; in Section 6 some of the results are discussed.

2. Load Management: Domain Description

Consumers can be divided into three different kinds: industrial, commercial (companies, institutes, trade) and domestic consumers. The focus of this paper is on domestic consumers. Although domestic consumers, as such, differ significantly, they all have devices within their homes that consume electricity to various degrees.

A typical demand curve of electricity is depicted in Figure 1. The purpose of load management is to smoothen the total peak load by managing a more appropriate distribution of the electricity usage among consumers. Flexible pricing schemes can be an effective means to influence consumer behaviour. The assumption behind the model presented in this paper is that, to acquire a more even distribution of electricity usage in time, consumer behaviour can be influenced by financial gain. Consumers are autonomous in the process of negotiation: each individual consumer determines which price/risk he/she is willing to take and when. Consumers are all individuals with their own characteristics and needs (partially defined by the type of devices they use within their homes), that vary over time. Therefore, adaptive and flexible models of consumers are required in systems to support the consumer (cf. [1]).

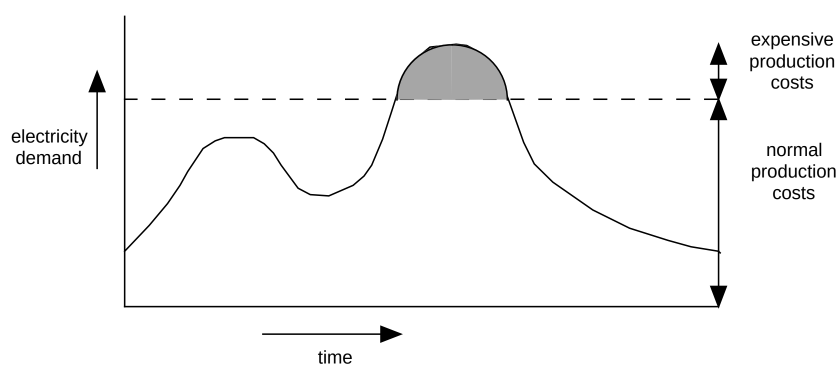

Figure 1, Demand curve with peak

Utility companies negotiate price with each and every individual separately, unaware of the models behind such systems. Individual consumers, consumer resources, utility companies and production companies can all be modelled as autonomous agents that interact on the basis of a shared understanding of a negotiation protocol. In this paper the negotiation process is modelled for one utility company and a number of consumers, each with their own respective agent to support them in the negotiation process: one Utility Agent and a number of Customer Agents. Neither negotiation between the Utility Agent and Production Agents (the agents that represent production companies), nor negotiation between Customer Agents and their Resource Consumer Agents (the agents that represent the devices within a home), are addressed in this paper.

3. Modelling Negotiating Agents

In [12] and [13] Rosenschein and Zlotkin describe a number of mechanisms for co-operation and co-ordination between agents. They set out, and evaluate, how groups of agents can co-ordinate their efforts and resolve certain types of conflicts according to different types of encounters. One important class of encounters is governed by negotiation processes. To control a certain process of negotiation between a group of agents, Rosenschein and Zlotkin describe a protocol with well-defined properties, called the *monotonic concession protocol*: during a negotiation process all proposed deals must be equally or more acceptable to the counter party than all previous deals proposed. Agreement is reached when one of the agents proposes a deal that coincides or exceeds the deal proposed by the other agent.

The strength of this protocol is that the negotiation process always converges. The monotonic concession protocol can be applied to the load management problem: both utility companies and consumers stand to benefit from the negotiation process. Utility companies are willing to decrease the price of electricity if customers are willing to decrease peak usage. Consumers are willing to give up some luxury in return for financial compensation (lower electricity bill). The bidding process between an individual utility company and an individual consumer, as performed by a Utility Agent and a Customer Agent, can be seen as a process in which both agents need to succeed to make a good deal.

3.1. Methods of Negotiation in the Load Management Domain

In this section the interaction between one Utility Agent and a number of Customer Agents is described. The Utility Agent always starts the negotiation process, as soon as a peak in the electricity consumption is predicted. The Utility Agent communicates an announcement to all Customer Agents to which they can respond by making a bid. The Utility Agent may then need to put forward another announcement, depending on the bids the Customer Agents have made and this goes on until finally an agreement is established: a situation which is acceptable for all agents. For the negotiation between the Utility Agent and the Customer Agents three announcement methods are distinguished, namely: offer, request for bids, announcing reward tables.

3.1.1. The Offer Method

The *offer* method is the easiest of the three, because only one step is made in the negotiation and then the negotiation ends. The offer the Utility Agent proposes to its Customer Agents is that if they only use x_{max} % of a given amount of electricity, they will receive that electricity for a lower price. If, however, they use more electricity than this given amount, they will have to pay a higher price for the extra electricity they use. This x_{max} is the same for all consumers and Customer Agents know the values for the lower, normal and higher prices for the electricity. This is an example of a 'take-it-or-leave-it' deal: Customer Agents may only answer 'yes' or 'no' to this offer. If they say 'no', they pay the normal electricity price in the peak period.

A major advantage of this announcement method is that it is very fast, because only one round of negotiation is required. Some time is needed to determine the announcement, but a prediction about the outcome of the announcement is relatively easily determined (e.g., the Utility Agent knows that normally about 70% of the Customer Agents will respond positively to the announcement). Another property of the 'offer'-method is that all customers are treated in the same way. In one respect this is an advantage, yet when considering the fact that a one person household uses less electricity than a four persons household, it is a disadvantage. A possible solution to this problem is to divide the customers into different categories (for example according to the number of persons in the household) and treat all customers in a certain category in the same way. However, this solution would make this method more complicated. Another possibility would be to make a 'private' announcement for each customer, but that too would make the method more complicated. In addition, it is illegal in Sweden to treat the same kind of customers differently. A considerable disadvantage of this method is that the customers almost have no influence on the negotiation process, they can only say 'yes' or 'no'.

3.1.2. The Request for Bids Method

Another method, in which the customers have significant influence on the negotiation process, is the *request for bids* method. When a peak in the electricity load is expected, the Utility Agent communicates a request to its Customer Agents. Each Customer Agent responds to this request by saying how much electricity it will want, given the reward promised: y_{min}. If a Customer Agent does not respond, the normal price holds. If, however, a Customer Agent makes a bid that is awarded, it will pay the lower price for the y_{min} electricity and a higher price for the extra electricity it needs. When all (or an acceptable number of; acceptable in the sense that the Utility Agent is willing to base its predictions on the bids acquired) bids have been collected, the new balance between the consumption and production of electricity is predicted by the Utility Agent. If this new prediction is satisfactory the Utility Agent stops the negotiation process. Otherwise, a new request for bids is communicated to the Customer Agents and they respond by doing either the same bid again ('stand still') or by doing a (slightly) better bid ('one step forward'). In this position, the Customer Agents know the values of the lower, normal and higher electricity prices. This method solves the problem of the former method that the customers have almost no influence on the bidding procedure, yet this type of announcement may entail a more complex and time consuming negotiation process and therefore cannot be made shortly before a peak is expected.

3.1.3. The Announce Reward Tables Method

The *announce reward tables* method can be seen as a structured combination of the two methods described above. The basic idea is the same as in the request for bid method, letting the Customer Agents state how much they are prepared, or able, to save. But instead of giving Customer Agents complete freedom to communicate a bid of their choice, there are some discrete values from which they can choose.

In this method the Utility Agent constructs a so-called *reward table* and communicates this table to the Customer Agents. A reward table consists of possible cut-down values, a reward value assigned to each cut-down value, together with a time interval. The cut-down value specifies an amount of electricity that can be saved (either in percentages or in kWh's) and the reward value specifies the amount of reward the Customer Agent will receive from the Utility Agent if it lowers its electricity consumption by the cut-down value in a specific time interval. Based on information received from its Resource Consumer Agents (the agents that represent a consumerís devices in the home) on the amount of electricity that can be saved in a given time interval, a Customer Agent examines and evaluates the rewards for the different cut-down values in the reward tables. If the reward value offered for the specific cut-down is acceptable to the Customer Agent, it informs the Utility Agent that it is prepared to make a cut-down x during interval I. If the reward value for the specific cut-down is not acceptable (for example, not worth the effort) Customer Agent can also decide to agree to a smaller cut-down x.

As soon as a sufficient number of Customer Agents have responded to the announcement of a reward table (sufficient in the sense that the Utility Agent is willing to base its predictions on the bids acquired), the Utility Agent predicts the new balance between consumption and production of electricity for the stated time interval. If the Utility Agent is satisfied by the responses, i.e. a peak can be avoided if all Customer Agents implement their bids, the Utility Agent confirms to the Customer Agents that their bids have been accepted. If the Utility Agent is not satisfied by the responses communicated by the Customer Agents, it announces a new reward table (according to the monotonic concession protocol) to the Customer Agents in which the reward values are at least as high, and for some cut-down values higher than in the former reward table (determined on the basis of, for example, the formulae described in Section 4). The Customer Agents react to this new announcement by responding with a new bid or the same bid again (in line with the rules of the monotonic concession protocol). This process continues until (1) the peak is satisfactorily low for the Utility Agent (at most the maximal allowed overuse), or (2) the reward values in the new reward table have (almost) reached the maximum value the Utility Agent can offer. This value has been determined in advance. In this case the utility company may have to pay excessive costs, for example, to buy additional electricity from a competitor.

Customers are assumed to keep their committed reductions. This can be enforced by letting them pay an excessive penalty if they exceed their promised use.

3.1.4. Evaluation of the Methods

The methods described above have their advantages and disadvantages and it is not trivial to determine which method is the best method for load management of electricity use in all situations. One solution is to allow agents to use all three methods (and maybe even more) as different strategies. The agents can then decide themselves which strategy to use and when. In some cases strategy number one is preferable, while in other cases strategy number two or three are most appropriate. This depends, for example, on the amount of time available for the negotiation process. The model on which the prototype described in Section 4 is based, uses the third method.

3.2. Utility Agent Model

The focus of this paper is on the negotiation process between a Utility Agent and a (large) number of Customer Agents. To model the Utility Agent and the Customer Agents, a generic agent model described in [5] is (re)used. In this agent model, an agent is composed of the following generic agent components: own process control, agent specific task, cooperation management, agent interaction management, world interaction management, maintenance of world information, maintenance of agent information. The specialisations of the generic agent components for the agents in a negotiation process are described below, together with the process abstraction levels distinguished within these components.

The Utility Agent (or UA for short) needs to be able to perform a number of tasks, the most relevant of which are:

- to acquire information from Production Agent (e.g., availability of electricity and cost)
- to acquire information from the External World (e.g., weather conditions, but also electricity consumption)
- to determine which negotiation strategy is most appropriate
- to monitor a negotiation process as it progresses,
- to predict the balance between consumption and production
- to assess when to negotiate with Customer Agents
- to determine content of negotiation
- to interact with Customer Agents.

The following sections show how these tasks are modelled in relation to the generic agent components distinguished in the generic agent model.

3.2.1. Own Process Control within UA

For the Utility Agent the task of controlling an agent's own internal processes entails determining a general negotiation strategy and evaluating the negotiation process during the process itself. Within the agent component own process control two sub-components, namely determine general negotiation strategy and evaluate negotiation process, are distinguished for this purpose (see Figure 2).

The component determine general negotiation strategy determines the general strategy for negotiation with the Customer Agents. This strategy determines the announcement method to be used and the strategy for accepting the bids. The different possibilities for making an announcement and the different strategies for accepting the bids were discussed in Section 3.1. The component evaluate negotiation process evaluates the overall negotiation process of the Utility Agent with its customers once a process of negotiation has ended.

3.2.2. Agent Specific Tasks within UA

The Utility Agent has two specific tasks: to predict the balance between consumption and production, and to assess the need to start negotiations with Customer Agents: determine predicted balance consumption/production and evaluate prediction. To predict the balance between

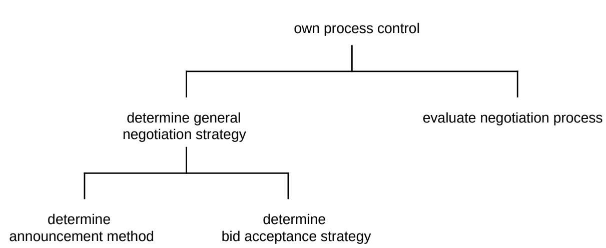

Figure 2, Process abstraction levels within own process control of UA

consumption and production, available information is analysed and predictions are calculated on the basis of statistical models. The decision to start a negotiation process is based on a predicted balance. In a stable situation no peak usage is expected and the situation can be left unchanged. In a peak situation, the decision (the component evaluate prediction) to start a negotiation process depends on level of predicted overuse: whether the predicted overuse is high enough to warrant the effort involved.

3.2.3. Co-operation Management within UA

Different tactics can be used to determine which announcements should be initiated. Depending on an agent's preferences, statistical analysis and optimisation, for example, can be used, or a more qualitative approach can be used. A *computational market model* of bidding is described in [1], [15]. An example of a more qualitative approach is the *generate and select* approach, in which all possible announcements are generated and one is selected (see Figure 3). This selection process can be randomly determined, or it can be based on, for example, predictions of the results. To determine which bids to accept, bid receipt is monitored and bids are assessed. Based on bid assessment, bids are either accepted or rejected.

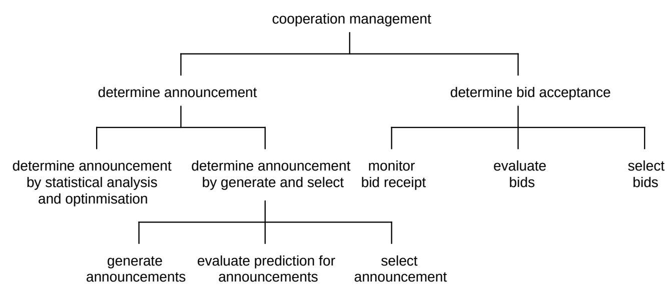

Figure 3, Process abstraction levels within cooperation management of UA

3.2.4. Other Agent Tasks within UA

The Utility Agent interacts with both the Production Agent and all Customer Agents. Interaction with the Production Agent is essential to acquire information about the availability of electricity and the cost involved. Interaction with Customer Agents is required to communicate announcements when peaks in electricity consumption are expected, to receive their bids in reply, and to award bids: the component agent interaction management.

The only interaction between the Utility Agent and the External World is the interaction required to acquire (1) general information about the external world itself, for example weather conditions, and (2) information about electricity consumption. The component world interaction management is responsible for the acquisition of this information.

The Utility Agent has models of other agents, including for example, information on how often Customer Agents have positively responded to announcements. The component maintenance of agent information is responsible for not only storing this information, but also updating this information on the basis of interaction with the agents. The component maintence of world information is responsible for the storage and maintenance of all information about the External World (including information acquired by the component world interaction management).

3.3. Customer Agent Model

A Customer Agent (or CA) also needs to be able to perform a number of tasks, the most relevant of which are:

- to determine which negotiation strategy is most appropriate for negotiation with its own Resource Consumer Agents,
- to monitor the negotiation with its own Resource Consumer Agents
- to determine which negotiation strategy is most appropriate for negotiation with a Utility Agent
- to monitor the negotiation with a Utility Agent
- to determine the content of negotiation

The following sections show how these tasks are modelled in relation to the generic agent components distinguished in the generic agent model. Only the most important generic agent components are described: Own Process Control and Cooperation Management.

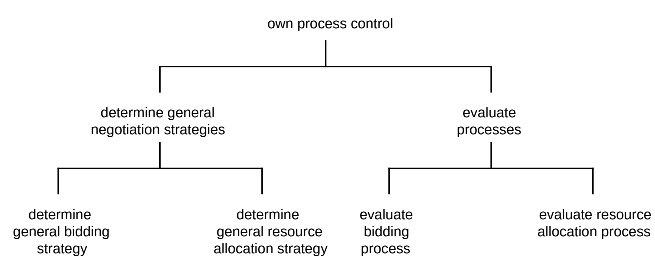

Figure 4, Process abstraction levels within own process control of CA

3.3.1. Own Process Control within a CA

As described in earlier sections, each Customer Agent needs to negotiate not only with a Utility Agent but also with its own Resource Consumer Agents. To control these negotiation processes each Customer Agent needs to be able to (1) determine which negotiation strategies to use to guide their interaction with each of the two types of Agents and (2) analyse these negotiation processes. This is modelled by two sub-components within the component own process control: determine general negotiation strategies and evaluate processes (see Figure 4).

3.3.2. Co-operation Management within a CA

The component cooperation management within the Customer Agent is similar to the component cooperation management within the Utility Agent. Determining the content of negotiation is part of the process of managing co-operation between agents as shown below in Figure 5, and entails determining both the content of negotiation with a Customer agentís own Resource Consumer

Agents and negotiation with a Utility Agent. Negotiation with a Utility Agent entails determination of appropriate bids (generation of possible bids, selection of a bid on the basis of expected gain and evaluation of the bid in the light of the Customer Agentís bidding strategy) on the basis of interpretation of available information (on results of bids and results of resource allocation, and customer preferences). Negotiation with Resource Consumer Agents entails determining needs and consequences of bids for individual Resource Consumer Agents.

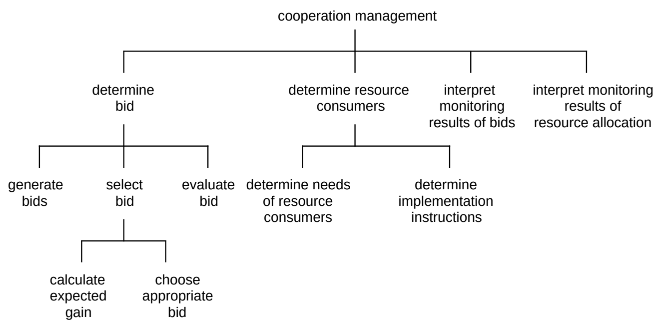

Figure 5, Process abstraction levels within cooperation management of CA

4. Design and Implementation of the Prototype System

To illustrate how negotiation between a Utility Agent and Customer Agents has been modelled using the reward table approach described in Section 3, the proof-of-concept prototype multi-agent system NALM has been developed using the component-based design method DESIRE, and (fully) specified and (automatically) implemented in the DESIRE software environment. The top level process composition of the system NALM is shown in Figure 6 (picture taken from the graphical design tool within the DESIRE software environment).

Figure 6, Process composition at the top level of the system

The process composition within the Utility Agent and the Customer Agents in NALM and the knowledge used by the agents is briefly described below.

4.1. The Utility Agent in the Prototype System

In the prototype system NALM the Utility Agent communicates the same announcements to all Customer Agents, in compliance with Swedish law.

4.1.1. Process Composition within the Utility Agent

The process composition at the highest process abstraction level within the Utility Agent is depicted in Figure 7.

This EPS image does not contain a screen preview.
It will print correctly to a PostScript printer.
File Name : ua_links.ps
Creator : PCE 4.8.18, June 1996
CreationDate : (Thu Oct 10 19:03:21 1996)
Pages : 1

Figure 7, Process composition at the top level within the Utility Agent

4.1.2. Knowledge Used within the Utility Agent

The predicted balance between the consumption and the production of electricity, is determined by the following formulae.

The first of the formulae determines the prediction of a consumer C's electricity use, after this consumer has committed to a reduction by a cutdown. Here the predicted use pu_c is the expected electricity use of customer C during the considered period if no reduction (cutdown) is decided by C. Morover, cd_c denotes the reduction fraction of C, and au_c is the maximal allowed use as agreed in the general contract with the customer.

$$\text{predicted\_use\_with\_cutdown}_c = \begin{cases} pu_c & \text{if } (1 - cd_c).\, au_c \geq pu_c \\ (1 - cd_c).\, au_c & \text{otherwise} \end{cases}$$

The second formula takes the sum over all consumers of the difference between predicted use (assuming the reduction to which they committed) and normal use (the overall use that is considered to be optimal by the Utility Agent) to determine the predicted overuse. This predicted overuse is the number that needs to be reduced to zero by the negotiation process. Note that this formula is linear in the contribution of customers. Therefore predictions can be based on statistical averages: there is no need to have reliable predictions per individual customer.

$$\text{predicted\_overuse} = \Sigma_{C \in CA} \text{predicted\_use\_with\_cutdown}_C - \text{normal\_use}$$

The last formula normalises this overuse by normalising it with respect to normal use.

$$\text{relative\_overuse} = \text{predicted\_overuse/normal\_use}$$

In the prototype system the increase of rewards in announcements during the negotiation process is based on the following formula:

$$\text{newr}_{cd} = r_{cd} + \beta \,.\, \text{relative\_overuse} \,.\, (1 - r_{cd}/\text{rmax}_{cd}).\, r_{cd}$$

Here is r_{cd} the reward for cutdown cd in the previous negotition round, and newr_{cd} the reward determined for the current round. Note that the increase of rewards is proportional to the relative overuse. Therefore, if the overuse decreases, also the increases in rewards decrease during the negotiation process. The factor β determines how steeply the reward values increase; in the current system it has a constant value. As said, the reward value increases more when the predicted overuse is higher (in the beginning of the negotiation process) and less if the predicted overuse is lower. However, the rewards never exceed the maximal reward rmax_{cd}, due to the logistic factor

$$(1 - r_{cd}/\text{rmax}_{cd})$$

The negotiation process ends when the difference between the new reward values and the (old) reward values is less than or equal to 1. Note that for the predicted use of a customer there is no need to use an individual value: an average value based on available customer statistics is sufficient, since in the formula for predicted over-use the sum is taken over all customers. Furthermore, the predictions assume that a customer commits to the reduction as promised. To assure that customers indeed live up to these commitments, for example, high financial penalties can be used if commitments are violated.

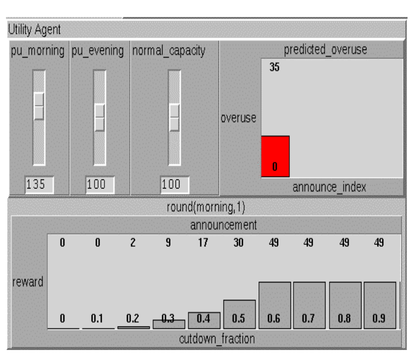

Figure 8, The Utility Agent during a negotiation process: initial phase

4.1.3. The Graphical Interface of the Utility Agent

The graphical interface of the Utility Agent in the prototype is depicted in Figure 8 (this is part of the user interface implemented specifically for this prototype system, as an extension of the DESIRE software environment). In the example shown in Figure 8, normal capacity is 100, and the predicted usage is 135, indicating a predicted overuse of 35 (depicted in the right upper part), in the initial situation. In the lower part of Figure 8, the reward offered in the first round of negotiation (e.g., a reward of 17 for a cut-down of 0.4) is depicted for each cut-down fraction (0, 0.1, 0.2, ...). In Figure 9, the predicted overuse depicted for the third round of negotiation has been reduced to 13. In the lower part of Figure 9, the reward announced (e.g., a reward of 24.8 for a cut-down of 0.4) is depicted for each cut-down fraction (0, 0.1, 0.2, ...).

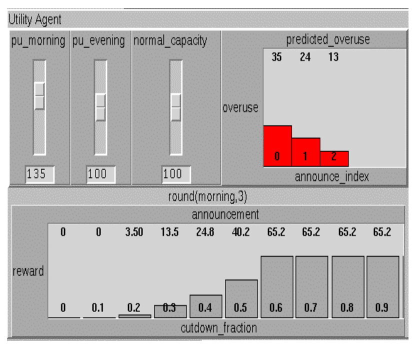

Figure 9, The Utility Agent during a negotiation process: final phase

4.2. A Customer Agent in the Prototype System

The process composition of the highest process abstraction level within a Customer Agent is depicted in Figure 10.

4.2.2 Knowledge Used within a Customer Agent

Within a Customer Agent, knowledge of the customer's preferences is represented in the form of a cut-down-required-reward table. The cut-down-required-reward table specifies the percentage with which a customer is willing to decrease (cut-down) its electricity usage, given a specific level of financial compensation.

This EPS image does not contain a screen preview.
It will print correctly to a PostScript printer.
File Name : ua.eps
Creator : PCE 4.9.7, December 1997
CreationDate : (Thu Apr 9 14:08:42 1998)
Pages : 1

Figure 10, Process composition at the top level within a Customer Agent

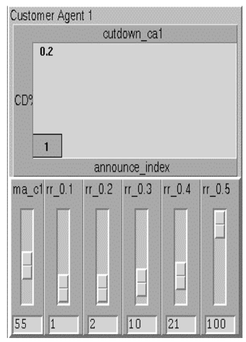

Figure 11, The Customer Agent during a negotiation process: initial phase

4.2.3 The Graphical Interface of a Customer Agent

The graphical interface of a Customer Agent in the prototype system is depicted in Figures 11 and 12, for different phases in the negotiation process. As shown in Figures 11 and 12 this specific customer requires a reward of at least 10 for a cut-down of 0.3, at least 21 for a cut-down of 0.4, and so on. When the Customer Agent receives an announcement (i.e., an announced reward table) from the Utility Agent, it compares the Utility Agent's table to its own cut-down-reward table. Each cut-down for which the required reward value of the customer is lower than the reward offered by the Utility Agent, is an acceptable cut-down. In the first round of negotiation, as depicted in Figure 11, the Customer Agent chooses the highest acceptable cut-down as its preferred cut-down and informs the Utility Agent of this choice, namely a cut-down of 0.2. In the second and thirds round of negotiation, as depicted in Figure 12, the Customer Agent again chooses the highest acceptable cut-down as its preferred cut-down, and informs the Utility Agent of this choice, in this case a cut-down of 0.4.

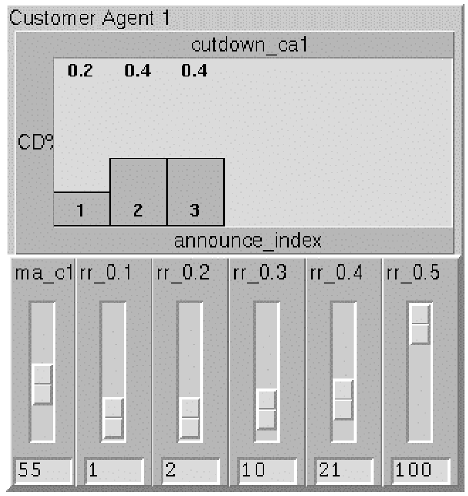

Figure 12, The Customer Agent during a negotiation process: final phase

5. Evaluation of the Behaviour of the Prototype System

The behaviour of the system NALM has been tested extensively. In Section 5 some of the negotiation patterns encountered are shown. Moreover, the behaviour of the system has been formally analysed, i.e., verified by mathematical proof that under certain assumptions the system is able to reduce the over-use to zero. This is briefly discussed in Section 6. A large number of possible negotiation patterns have been generated to examine system behaviour. Some typical patterns are briefly discussed in this section.

5.1 Example negotiation pattern 1: a successful negotiation process

A relatively standard pattern is one in which after a number of 3 negotiation rounds the negotiation process is terminated successfully with over-use close to zero as shown in the table below (recall the knowledge used by the Utility Agent presented in Section 4.1.2).

negotiation	predicted	UA announcement: reward table					bid: cutdown	
round	overuse	0.1	0.2	0.3	0.4	0.5	CA1	CA2
1	35	0	2	9	17	30	0.2	0
2	22	0	2.8	11.7	21.8	36.6	0.3	0
3	15.5	0	3.53	13.6	24.9	40.4	0.4	0.2
4	-5							

The profiles of the Customer Agents in this example are

required rewards	0.1	0.2	0.3	0.4	0.5
CA1	1	2	11	23	45

CA2	1	3	15	20	58

In the first round, the reward offered for a cut-down of 0.2 is 2: precisely the minimum cut-down required by Customer Agent 1. As can be seen in the above table, CA1 makes the bid for a cut-down of 0.2. The minimum cut-down values required by Customer Agent 2 are all higher than the aanounced cut-down values in the first round. In the second round the announced values are all slightly higher. The reward for 0.3 cut-down has been raised to 11.7. This reward is more than the minimum of 11 required by Customer Agent 1. Customer Agent 1ís bid for a cut-down is, therefore raised to 0.3. The minimum cut-down values required by Customer Agent 2 are still all higher than the announced cut-down values. In the third round the values are again all slightly higher, resulting in a bid for a cut-down of 0.4 for Customer Agent 1, and a bid for a 0.2 cut-down for Customer Agent 2. This reduces the overuse to below zero.

5.2 Example negotiation pattern 2: an unsuccessful negotiation process

Next a negotiation pattern where the Customer Agents are so demanding that the Utility Agent reaches its limit and breaks off the negotiation after four rounds without complete reduction.

negotiation	predicted	UA announcement: reward table					bid: cutdown	
round	overuse	0.1	0.2	0.3	0.4	0.5	CA1	CA2
1	35	0	2	9	17	30	0	0
2	35	0	3.3	13.3	24.6	40.5	0	0
3	35	0	5.0	17.2	31.0	47.2	0.2	0
4	22	0	6.0	18.5	32.9	48.6	0.2	0.2
5	8							

The profiles of the Customer Agents in this example are the following:

required rewards	0.1	0.2	0.3	0.4	0.5
CA1	1	4	24	57	92
CA2	1	6	32	65	98

The rewards offered by the Utility Agent are lower than the rewards required by both Customer Agents in the first 2 rounds. In the third round Customer Agent 1 makes a bid for a cut-down of 0.2. Customer Agent 2 does the same in the fourth round. At this point the Utility Agent decides to no longer negotiate but to stick to the cut-down percentages received (and, for example, he buys additional electricity from a competitor) .

5.3 Example negotiation pattern 3: nonmonotonic concession

A different negotiation pattern is acquired when the monotonic concession protocol is violated: one of the Customer Agents withdraws its previous offered reduction and the predicted over-use, as a result, increases. As the trace shows, although the prototype system was designed under the assumption of a monotonic concession protocol, the prototype system is also robust in situations like this.

negotiation round	predicted overuse	UA announcement: reward table					bid: cutdown	
		0.1	0.2	0.3	0.4	0.5	CA1	CA2
1	35	0	2	9	17	30	0.2	0
2	22	0	2.8	11.7	21.8	36.6	0	0
3	35	0	4.4	15.9	29.0	45.1	0	0.2
4	21	0	5.5	17.6	31.6	47.4	0	0.2
5	21	0	6.4	18.7	33.2	48.7	0	0.2
6	21							

The profiles of the Customer Agents in this example are depicted below. Customer Agent 1ís profile changes after the first round as shown: e.g., a reward of 10 is required for a cut-down of 0.2, instead of a reward of 2.

required rewards	0.1	0.2	0.3	0.4	0.5
CA1	1	2 ->10	11 -> 35	23 -> 56	45 -> 84
CA2	1	4	22	41	75

At the end of the fifth round the only bid that still holds is the bid done by Customer Agent 2 in the third round. The increased values of the cut-down percentages announced by the Utility Agent were not sufficient to increase the bid by either Customer Agent 2, nor to allow for a bid by Customer Agent 1 with its new profile.

6. Formal Analysis of the Negotiation System

The purpose of verification is to prove, under a certain set of assumptions, that a system will adhere to a certain set of properties, for example the design requirements. In our approach, this is done by a mathematical proof (i.e., a proof in the form to which mathematicians are accustomed) that the specification of the system, together with the assumptions, logically implies the properties that the system needs to fulfil. In [2] it is shown how the prototype system NALM has been verified using the compositional verification method for (component-based) agent systems introduced in [8]. In this section a brief survey is given of some of the properties involved in this compositional verification process. For more properties and for proofs, see [2].

6.1 Formal Analysis by Compositional Verification

During the verification process the properties can be derived from properties of agents (one process abstraction level lower) and these agent properties, in turn, can be derived from properties of the agent components (again one abstraction level lower). The compositional multi-agent system verification method described in this section is based on the component-based structure of the system.

Primitive components can be verified using more traditional verification methods for knowledge-based systems (if they are specified by means of a knowledge base) or other verification methods tuned to the type of specification used; e.g., see [10].

Verification of a (composed) component at a given process abstraction level is done using

- properties of the sub-components it embeds
- the composition relation, defined by the information exchange and task control knowledge
- environmental properties of the component (depending on the rest of the system, including the world).

Compositionality provides the basis for the verification process: given a set of environmental properties, the proof that a certain component adheres to a set of behavioural properties depends on the (assumed) properties of its sub-components, and the composition relation: properties of the interactions between those sub-components, and the manner in which they are controlled. The assumptions under which the component functions properly, are the properties to be proven for its sub-components. This implies that properties at different levels of process abstraction play their own role in the verification process.

If the load balancing negotiation is to converge with an over-use below the maximal allowed over-use, several assumptions have to be made about the behaviour of the utility agent and the customer agents. The Utility Agent should make announcements that are high enough to persuade the Customer Agents to perform a cut-down that will lower their use, and the Customer Agents should be responsive to such announcements. Some of the properties and assumptions of the prototype system and the agents within the system that have been formalised and used in the proof are discussed below.

6.2 Language and Semantics used

To obtain a formalisation of behavioural properties of agent systems a temporal trace language is used. An *information state* I of a system or system component D (e.g., the overall system, or an input or output interface of an agent) is an assignment of truth values {true, false, unknown} to the set of ground atoms describing the information within D. The set of all possible information states of D is denoted by IS(D). A *trace* $\mathcal{M}$ of D is a sequence (over the natural numbers) of information states $(I^t)_{t \in \mathbf{N}}$ in IS(D). Given a trace $\mathcal{M}$ of D, the information state of the input interface of an agent A at time point t is denoted by

$$\text{state}_D(\mathcal{M}, t, \text{input}(A)),$$

where state_D and input are function symbols. Analogously,

$$\text{state}_D(\mathcal{M}, t, \text{output}(A))$$

denotes the information state of the output interface of agent A at time point t within system (component) D. The information states can be related to statements via the formally defined satisfaction relation |=, comparable to the Holds-predicate in the Situation Calculus. Differences from the Situation Calculus approach are, however, that we

(1) use an infix notation for the |= predicate instead of a prefix notation,

(2) refer to a trace and time point instead of a single state, and

(3) can focus on part of the system.

Based on these statements, behavioural properties can be formulated in a formal manner in a sorted first-order predicate logic with sorts **T** for time points, **Traces** for traces and **F** for state formulae, using quantifiers over time and the usual first order logical connectives such as $\neg, \wedge, \vee, \Rightarrow, \forall, \exists$. An example of such a statement is the following other examples of can be found in Sections 6 and 7 below). Consider the following informally expressed property for the dynamics of a multi-agent system as a whole:

> *Each service request of agent A to agent B must be followed by a satisfactory service proposal of agent B after a certain time.*

In a structured, semiformal manner, this property can be reformulated (and detailed) as follows:

if at some point in time
agent A outputs: *a service request for B,*
then at a later point in time
agent B outputs: *a service proposal for the request for A*
and at a still later point in time
agent A outputs: *the proposal is accepted to B*

Using the formal language introduced above the following temporal formalisation is made of this example property:

$$\forall \mathcal{M}, t, r\ [\ \text{state}(\mathcal{M}, t, \text{output}(A)) \models \text{request\_for\_from}(r, B, A)$$
$$\Rightarrow [\ \exists p, t1 > t\quad \text{state}(\mathcal{M}, t1, \text{output}(B)) \models \text{proposal\_for\_from}(p, r, A, B)$$
$$\wedge\ \exists t2 > t1\ \ \text{state}(\mathcal{M}, t2, \text{output}(A)) \models \text{accepted\_proposal\_for\_from}(p, r, A, B)\ ]\ ]$$

Here the statement state($\mathcal{M}$, t, output(A)) $\models$ request_for_from(r, B, A) means that within trace $\mathcal{M}$ at time point t a statement request_for_from(r, B, A) occurs in the output interface of agent A, i.e. has truth value true in the output state of A.

6.3 Overview of some of the properties

In this section an overview is given of the most relevant properties that have been proven for the system NALM.

succesfulness of negotiation

The Utility Agent satisfies *successfulness of negotiation* within the system as a whole if at some point in time the predicted use will be lower than the required maximal use. More precisely: at some point in time t and for some negotiation round N the predicted overuse is less than or equal to 0:

$$\forall \mathcal{M} \in \text{Traces}(S)\ \exists t, N\ \exists U \leq 0\quad \text{state}_S(\mathcal{M}, t, \text{output}(UA)) \models \text{predicted\_overuse}(U, N)$$

Here predicted_overuse(U, N) denotes that for round N the predicted overuse is U.

This property has been mathematically proven making use of intermediate properties shown below (which by themselves also have been proven for the prototype system).

negotiation round generation effectiveness

If the predicted over-use is higher than the maximal allowed over-use, then a next negotiation round is initiated. The Utility Agent satisfies *negotiation round generation effectiveness* if the following holds: if and when predicted overuse is higher than the maximal overuse, a next negotiation round is initiated:

$\forall \mathcal{M} \in$ Traces(S) $\forall$t , N, U, CD, R
[$state_{UA}(\mathcal{M}$, t, output(UA)) |= round(N)
& $state_{UA}(\mathcal{M}$, t, output(UA)) |= predicted_overuse(U, N)
& U > 0
& $state_{UA}(\mathcal{M}$, t, output(UA)) |= announcement(CD, R, N)
& R < mr_{UA}(CD)]
$\Rightarrow$ $\exists$t' > t $state_{UA}(\mathcal{M}$, t', output(UA)) |= round(N+1)

Here round(N+1) denotes that the Utility Agent has declared round N+1 active.
for each negotiation round an announcement will be generated.

monotonicity of announcement

For each announcement and each cut-down percentage the reward is at least the reward for the same cut-down percentage in the previous announcement (monotonicity). The Utility Agent satisfies *monotonicity of announcement* if for each announcement and each cut-down percentage the offered reward is at least the reward for the same cut-down percentage offered in the previous announcements:

$\forall \mathcal{M} \in$ Traces(UA) $\forall$t, t', N, N' $\forall$CD, R, R'
$state_{UA}(\mathcal{M}$, t, output(UA)) |= announcement(CD, R, N)
& $state_{UA}(\mathcal{M}$, t', output(UA)) |= announcement(CD, R', N')
& N ≤ N'
$\Rightarrow$ R ≤ R'

progress in announcement

For at least one cut-down percentage the difference between the currently announced reward and the previous announced reward is at least the constant c (announce margin).
The Utility Agent satisfies *progress in announcement* if for at least one cut-down percentage the difference between the currently announced reward and the previously announced reward is at least the positive constant m (announce margin):

$\forall \mathcal{M} \in$ Traces(UA) $\forall$t, t', N $\exists$CD $\forall$R, R'
$state_{UA}(\mathcal{M}$, t, output(UA)) |= announcement(CD, R, N)
& $state_{UA}(\mathcal{M}$, t', output(UA)) |= announcement(CD, R', N+1)
$\Rightarrow$ R + m ≤ R'

announcement rationality

No announced reward will be higher than the maximal reward. The Utility Agent satisfies *announcement rationality* if no announced reward is higher than the maximal reward:

$\forall \mathcal{M} \in$ Traces(UA) $\forall$t, N $\forall$CD, R
$state_{UA}(\mathcal{M}$, t, output(UA)) |= announcement(CD, R, N) $\Rightarrow$ R ≤ mr_{UA} (CD)

negotiation round generation groundednes

If the predicted over-use is at most the maximal allowed over-use, then no new negotiation round is initiated. The Utility Agent satisfies *negotiation round generation groundednes* if the following holds: if the predicted overuse is at most the maximal overuse, then no new negotiation round is initiated:

$$\forall \mathcal{M} \in \text{Traces(UA)}\ \forall t,\ N,\ U$$
$$\text{state}_{UA}(\mathcal{M},\ t,\ \text{output(UA)}) \models \text{predicted\_overuse}(U,\ N)\ \ \&\ \ U \leq 0$$
$$\Rightarrow \forall t',\ N'>N \quad \text{state}_{UA}(\mathcal{M},\ t',\ \text{output(UA)}) \not\models \text{round}(N')$$

bid generation effectiveness

Each customer responds to each announcement (possibly with a bid for reduction zero).
A Customer Agent CA satisfies *bid generation effectiveness* if for each announced negotiation round at least one bid is generated (possibly a bid for reduction zero):

$$\forall \mathcal{M} \in \text{Traces(CA)}\ \forall t,\ N$$
$$\text{state}_{CA}(\mathcal{M},\ t,\ \text{input(CA)}) \models \text{round}(N) \quad \Rightarrow \exists CD,\ t' \geq t \ \ \text{state}_{CA}(\mathcal{M},\ t',\ \text{output(CA)}) \models \text{cutdown}(CD,\ N)$$

monotonicity of bids

Each bid is at least as high (a cut-down percentage) as the previous bid. A Customer Agent CA satisfies *monotonicity of bids* if each bid is at least as high (a cut-down percentage) as the bids for the previous rounds:

$$\forall \mathcal{M} \in \text{Traces(S)}\ \forall t,\ t',\ N,\ N'\ \forall CD,\ CD'$$
$$\text{state}_{S}(\mathcal{M},\ t,\ \text{output(CA)}) \models \text{cutdown}(CD,\ N)\ \ \&\ \ \text{state}_{S}(\mathcal{M},\ t',\ \text{output(CA)}) \models \text{cutdown}(CD',\ N')\ \ \&\ \ N \leq N'$$
$$\Rightarrow\ CD \leq CD'$$

Assumption for the population of agents as a whole

For each Customer Agent and each cut-down percentage, the required reward of the customer agent is at most the maximal reward that can be offered by the Utility Agent.

required reward limitation

The system S satisfies *required reward limitation* if for each Customer Agent and each cut-down fraction CD, the required reward of the Customer Agent $rr_{CA}(CD)$ is at most the maximal reward $mr_{UA}(CD)$ that can be offered by the Utility Agent:

$$\forall CA\ \forall CD \quad rr_{CA}(CD) \leq mr_{UA}(CD)$$

The latter assumption is rather simple, but stronger than is required. A weaker assumption on the population can be formulated; this weaker assumption has a more complicated formulation, however. Further properties, and proofs involved in this compositional verification process can be found in [2].

6. Discussion

A multi-agent approach to the design and implementation of large open distributed industrial systems has shown to be promising. To come to a clearer understanding of strengths and weaknesses of such approaches it is, however, important to address real world problems where size and/or complexity challenge system design methodologies, and to evaluate the results. The load balancing problem of power, as stated in this paper, belongs to the class of real world problems. Furthermore, load balancing can be seen as a subclass of the general class of resource management problems.

A multi-agent approach to load management in which different negotiation strategies can be modelled and analysed has a great potential. An endeavour as such requires intensive interaction between researchers from different disciplines: power distribution, customer service, computer science, with different types of knowledge. Knowledge about utility companies and their customers is essential.

The approach proposed in this paper combines elements of Agent Technology and Knowledge Technology. The model presented has a component-based structure and has been designed and specified in detail at a conceptual level, and has been implemented as the proof-of-concept prototype system NALM. The component-based structure supports transparency and reuse; replacement of components, or changes in the explicitly specified (strategic) negotiation knowledge, are straightforward operations. More information on the component-based design method for multi-agent systems DESIRE used can be found in [4] (the underlying principles) or [3] (an extensive case study). In [2] the compositional verification method for multi-agent systems introduced in [8] was used to verify the multi-agent system discussed in this paper.

This paper focuses on negotiation between a Utility Agent and its Customer Agents. Agent models have been designed in which explicit knowledge of negotiation strategies and their applicability is represented. One (monotonic) negotiation strategy, based on announcing reward tables, has been fully specified and implemented. Initial evaluation has shown the approach to be promising. More extensive evaluation of the parameters and their effect is, however, required. For example, in the prototype implementation NALM the factor beta which determines the speed of negotiation has a constant value. The effects of dynamically varying the value of beta on the basis of experience, should be examined.

Compared to the work of Huberman and Clearwater [7], in which the temperature of the air in a building is regulated, the following can be remarked. Huberman and Clearwaterís approach is based on an auction principle: there is one auctioneer agent, and all other agents are either sellers of buyers (role switching is possible). In the approach discussed in this paper there is no separate auctioneer agent, and agents have fixed roles (either seller of buyer). In the building environment described by Huberman and Clearwater a double-blind auction is used: none of the buyers/sellers know the value of any other bids (besides their own). In our approach, only the sellers are blind (the customers): only the buyer (utility agent) knows the bids from the customers (there is no auctioneer agent).

The agents in the building environment receive a fixed amount of money per auction round (depending on the variable air volume) which they can use to buy cold air and which can be increased by selling cold air. The authors indicate that the effectiveness of their model can suffer from human users that abuse the fact that the agents use virtual money, that take a “free ride” by setting their thermostats to unrealistic values. This problem cannot occur in our approach because the financial consequences are real.

The result of the Huberman and Clearwater auction is that every party attains approximately the same level of comfort. The result in our approach is that every party is as comfortable as he/she has chosen to be. Apart from the differences in approach the difference in results is also due to the nature of the domains studied. To control the building environment only a limited supply of cold air is available: this implies that a distribution is necessary and that the result is aimed at keeping the discomfort as minimal as possible. In the electricity domain every customer has a right (by way of comfort) to a fixed maximal amount of electricity (enough to satisfy the practical needs); in principle there is an unlimited supply available.

Another problem in the building environment is that the agents do not receive what they pay for due to inefficiency of air distribution (influencing temperature and the amount of cool air) and the normal external factors with respect to temperature. In our approach all parties receive the amount of energy requested. Current research focuses on verification of the agents' behaviour using this strategy. The potential of other negotiation strategies, such as computational markets (see, for example, [15]) are also currently being explored. In addition, interaction between Producer Agents and the Utility Agents, and between Customer Agents and Resource Consumer Agents, is of importance. Negotiation strategies comparable to those employed between a Utility Agent and its Customer Agents may be applicable, but also other strategies may have potential. These different types of negotiation strategies are also subject of further research.

Compared to auction-based approaches, the approach proposed here has the advantage of direct one-to-one communication, without the need for a centrally coordinated auction organisation.

Another research question for further research is the question how the proposed approach will work if more than one Utility Agent is involved in the negotiation process with the same customers: the situation where there is competition between providers. If monotonicity is imposed as a requirement on the negotiation protocol, it may be expected that similar techniques would be feasible. A difference will be the following. For the one-to-many case in NALM, the successfulness of the negotiation depends on the collective willingness of the customers (e.g., The required reward limitation in Section 6.3). In a situation with competition, assumptions made on this collective willingness, will also involve the competitor's Utility Agents, so the statement will be slightly more complex. However, once this has been formulated, it can be expected that the further proofs will carry over to this situation.

Another question that may be posed is why only peaks were addressed and not selling overcapacity. Indeed, by exactly the same methods also selling overcapacity can be addressed. As the industry cooperating in this project put their interest on peak avoidance, the prototype NALM (meant as proof of concept prototype) was kept simple and restricted to the peaks. There was however, no technical reason for this limitation.

A more general question is whether techniques for multi-attribute negotiation can have advantages over the single-attribute approach put forward above. The restriction to the price attribuut was made, since the industry involved expected that customers are mainly interested in the price attribute. However, it is clear that multi-attribute techniques in general provide a more sincere fit to the profiles of customers. Some recent research in this area (in another application domain) by some of the authors has shown that the approach can be generalised to the area of multi-attribute profiling and negotiation; see [6], [9].

References

[1] Akkermans, H., Ygge, F. and Gustavsson, R., HOMEBOTS: Intelligent Decentralized Services for Energy Management. In Proc. of the Fourth International Symposium on the Management of Industrial and Corporate Knowledge, ISMICK'96, 1996.

[2] Brazier, F.M.T., Cornelissen, F., Gustavsson, R., Jonker, C.M., Lindeberg, O., Polak, B. and Treur, J., Compositional Design and Verification of a Multi-Agent System for Load Balancing. In: Y. Demazeau (ed.), Proc. of the Third International Conference on Multi-Agent Systems, ICMAS'98. IEEE Computer Society Press, 1998. pp. 49-56.

[3] Brazier, F.M.T. , Dunin-Keplicz, B., Jennings, N.R. and Treur, J., Formal specification of Multi-Agent Systems: a real-world case. In: V. Lesser (ed.), Proc. of the First International Conference on Multi-Agent Systems, ICMAS'95, MIT Press, Cambridge, MA, 1995, pp. 25-32. Extended version in: International Journal of Cooperative Information Systems, M. Huhns, M. Singh, (eds.), special issue on Formal Methods in Cooperative Information Systems: Multi-Agent Systems, vol. 6, 1997, pp. 67-94.

[4] Brazier, F.M.T., Jonker, C.M., and Treur, J., Component-Based Design of Intelligent Agents. Data and Knowledge Engineering, vol. 41, 2002, pp. 1-28. Preliminary, shorter version in: J. Cuena (ed.), Proceedings of the 15th IFIP World Computer Congress, WCC'98, Conference on Information Technology and Knowledge Systems, IT&KNOWS'98, 1998, pp. 347-360. [5] Brazier, F.M.T., Jonker, C.M., and Treur, J. , Compositional Design and Reuse of a Generic Agent Model. Applied Artificial Intelligence Journal vol. 14, 2000, pp. 491-538.

[6] Dastani, M., Jacobs, N., Jonker, C.M, and Treur, J., Modeling User Preferences and Mediating Agents in Electronic Commerce, In: F. Dignum, C. Sierra (eds.), *Agent-Mediated Electronic Commerce.* Lecture Notes in AI, vol. 1991, Springer Verlag, 2001, pp. 164-196.

[7] Huberman, B., and Clearwater, S.H., (1995), A Multi-Agent System for Controlling Building Environments, In: V. Lesser (ed.) Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95), AAAI Press / The MIT Press, pp. 171-176.

[8] Jonker, C.M., and Treur, J. , Compositional verification of multi-agent systems: a formal analysis of pro-activeness and reactiveness. International Journal of Cooperative Information Systems, vol. 11, 2002, pp. 51-92. Preliminary, shorter version in: W.P. de Roever, H. Langmaack, A. Pnueli (eds.), Proceedings of the International Workshop on Compositionality, COMPOS'97. Lecture Notes in Computer Science, vol. 1536, Springer Verlag, 1998, pp. 350-380.

[9] Jonker, C.M., Treur, J., An Agent Architecture for Multi-Attribute Negotiation. In: B. Nebel (ed.), Proc. of the 17th International Joint Conference on AI, IJCAI'01. Morgan Kaufman, 2001, pp. 1195 - 1201.

[10] Leemans, N.E.M., Treur, J., and Willems, M., A Semantical Perspective on Verification of Knowledge. Data and Knowledge Engineering, vol. 40, 2002, pp. 33-70.

[11] Ottosson, H., Gustavsson, R. and Bosch, J., Utility Aspects on Strategic IT- Integration Issues. Proc. II CEPSI'96, 1996.

[12] Rosenschein, J.S. and Zlotkin, G., Rules of Encounter: Designing Conventions for Automated Negotiation among Computers, MIT Press, 1994.

[13] Rosenschein, J.S. and Zlotkin, G., Designing Conventions for Automated Negotiation, AI Magazine, Vol. 15-3, Fall 1994, pp. 29-46.

[14] Wooldridge, M. and Jennings, N.R., Intelligent Agents (Proc. ATAL'94), Lecture Notes in Artificial Intelligence, Vol. 890, Springer Verlag, 1995.

[15] Ygge, F. and Akkermans H., Power Load Management as a Computational Market. Proc. of the Second International Conference on Multi-Agent Systems, ICMAS'96, 1996, pp 393-400.