

Agents Negotiating for Load Balancing of Electricity Use

Frances Brazier¹, Frank Cornelissen¹, Rune Gustavsson², Catholijn M. Jonker¹,
Olle Lindeberg², Bianca Polak¹, Jan Treur¹

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, as well as with smart equipment, are changing the rules of the energy market. Deregulation makes it even more demanding for utility companies to create new business processes for mutual benefit. 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. The compositional development method DESIRE has been used to analyse, design, implement and verify a multi-agent system capable of negotiation for 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 Electricité de France are, together with the companies ABB and IBM and several universities, exploring possibilities and impacts of information technologies in the energy market; cf. [5], [6], [8]. Resource management, such as load balancing, is one of the first applications where a multi-agent approach (cf. [11]) has shown to be very promising, as shown in this paper.

In this paper, first the domain of load management is described in Section 2. Different protocols for negotiation are discussed in Section 3. The compositional development method for multi-agent systems DESIRE used is briefly presented in Section 4. In Section 5 the model of the negotiating agents used in the system is discussed. Section 6 presents more details of the prototype system developed. In Section 7 some of the results are discussed.

2. Load Management

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 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. As consumers are all individuals with their own characteristics and needs (partially defined by the type of equipment they use within their homes), that vary over time, models of consumers used to design systems to support the consumer, need to be adaptive and

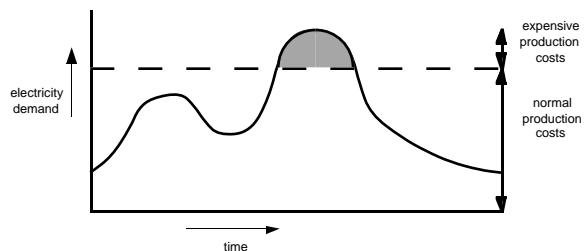


Figure 1. Demand curve with peak

flexible (cf. [1]). 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. Negotiation between the Utility Agent and Production Agents that represent production companies, and the negotiation between Customer Agents and their Resource Consumer Agents, are both not addressed in this paper.

3. Modelling Negotiation Processes

In [9] and [10] a number of mechanisms for co-operation and co-ordination between agents are described. 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.

3.1. 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.

3.2. 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. In this model the Utility Agent always starts the negotiation process, as soon as a coming 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.

3.2.1. 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 would make it far too 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'.

3.2.2. The Request for Bids Method

Another method, in which the customers have quite a large influence on the negotiation process, is the *request for bids* method. If a peak in the electricity load is expected the Utility Agent communicates a request to its Customer Agents. Each Customer Agent is obliged to respond by saying how much electricity it really needs when a reward is promised: y_{\min} . If a Customer Agent does not respond, it pays the normal price. But if 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) 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'). Also in this case, 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.

3.2.3. 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 they 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 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 l . 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), 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 mentioned in Section 3.1) 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 6). 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.

3.2.4. 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 6 is based, uses the third method.

4. Compositional Development of Multi-Agent Systems

The example multi-agent system described in this paper has been developed using the compositional development method DESIRE for multi-agent systems (framework for DEsign and Specification of Interacting REasoning components; cf. [3]). The development of a multi-agent system is supported by graphical design tools within the DESIRE software environment. Translation to an operational system is straightforward; the software environment includes implementation generators with which formal specifications can be translated into executable code of a prototype system. In DESIRE, a design consists of knowledge of the following three types: process composition, knowledge composition, the relation between process composition and knowledge composition. These three types of knowledge are discussed in more detail below.

4.1. Process Composition

Process composition identifies the relevant processes at different levels of (process) abstraction, and describes how a process can be defined in terms of (is composed of) lower level processes.

4.1.1. Identification of Processes at Different Levels of Abstraction

Processes can be described at different levels of abstraction; for example, the process of the multi-agent system as a whole, processes defined by individual agents and the external world, and processes defined by task-related components of individual agents. The identified processes are modelled as *components*. For each process the *input and output information types* are modelled. The identified levels of process abstraction are modelled as *abstraction/specialisation relations* between components: components may be *composed* of other components or they may be *primitive*. Primitive components may be either reasoning components (i.e., based on a knowledge base), or, components capable of performing tasks such as calculation, information retrieval, optimisation. These levels of process abstraction provide process hiding at each level.

4.1.2. Composition of Processes

The way in which processes at one level of abstraction are composed of processes at the adjacent lower abstraction level is called *composition*. This composition of processes is described by a specification of the possibilities for *information exchange* between processes (*static view* on the composition), and a specification of *task control*

knowledge used to control processes and information exchange (*dynamic view* on the composition).

4.2. Knowledge Composition

Knowledge composition identifies the knowledge structures at different levels of (knowledge) abstraction, and describes how a knowledge structure can be defined in terms of lower level knowledge structures. The knowledge abstraction levels may correspond to the process abstraction levels, but this is often not the case.

4.2.1. Identification of knowledge structures at different abstraction levels

The two main structures used as building blocks to model knowledge are: *information types* and *knowledge bases*. Knowledge structures can be identified and described at different levels of abstraction. At higher levels details can be hidden. An *information type* defines an ontology (lexicon, vocabulary) to describe objects or terms, their sorts, and the relations or functions that can be defined on these objects. Information types can logically be represented in order-sorted predicate logic. A *knowledge base* defines a part of the knowledge that is used in one or more of the processes. Knowledge is represented by formulae in order-sorted predicate logic, which can be normalised by a standard transformation into rules.

4.2.2. Composition of Knowledge Structures

Information types can be composed of more specific information types, following the principle of compositionality discussed above. Similarly, knowledge bases can be composed of more specific knowledge bases. The compositional structure is based on the different levels of knowledge abstraction distinguished, and results in information and knowledge hiding.

4.3. Relation between Process Composition and Knowledge Composition

Each process in a process composition uses knowledge structures. Which knowledge structures are used for which processes is defined by the relation between process composition and knowledge composition.

5. Models of Negotiating Agents

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 [4] is (re)used. In this agent model, an agent performs the following generic agent tasks: 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 tasks for the agents in a negotiation process are described below, together with the process abstraction levels distinguished within these tasks.

5.1. Utility Agent

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 Producer Agent (e.g., availability of electricity and cost)
- to acquire information from 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 subsections show how these tasks are modelled in relation to the generic agent tasks distinguished in the generic agent model.

5.1.1. 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 task own process control two sub-tasks, namely determine general negotiation strategy and evaluate negotiation process, are distinguished for this purpose (see Figure 2).

The task 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. The component evaluate negotiation process evaluates the overall negotiation process of the Utility Agent with its customers once a process of negotiation has ended.

5.1.2. 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

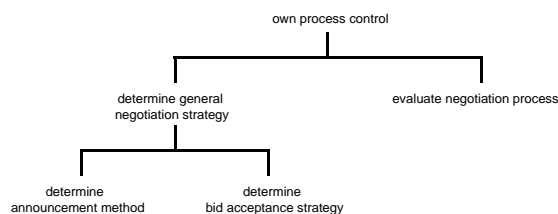


Figure 2. Process abstraction levels within own process control of UA

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 task 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.

5.1.3. 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], [12]. 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 they are not.

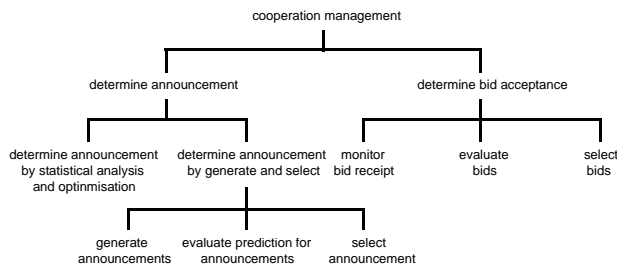


Figure 3. Process abstraction levels within cooperation management of UA

5.1.4. Other agent tasks within UA

The Utility Agent interacts with both the Producer Agent and all Customer Agents. Interaction with the Producer 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 task 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 task 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 task 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 task maintenance of world information is

responsible for the storage and maintenance of all information about the External World (including information acquired by the task world interaction management).

5.2. Customer Agent

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

5.2.1 Own Process Control within 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-tasks within the task own process control: determine general negotiation strategies and evaluate processes (see Figure 4).

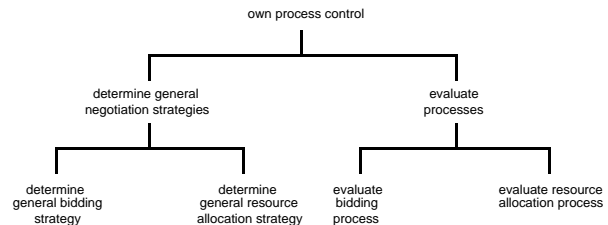


Figure 4. Process abstraction levels within own process control of CA

5.2.2. Co-operation Management within CA

The task cooperation management within the Customer Agent is similar to the task 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 agent's 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 Agent's 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.

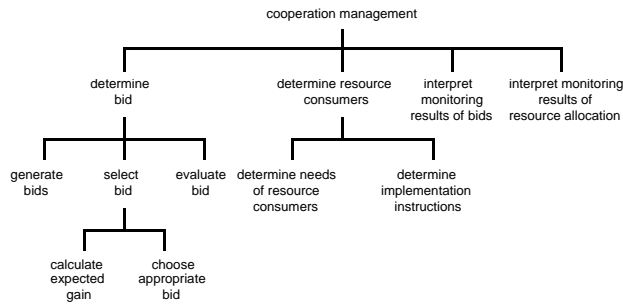


Figure 5. Process abstraction levels within cooperation management

6. 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, a simplified prototype of a Multi-Agent System has been developed using the compositional development method DESIRE, and (fully) specified and (automatically) implemented in the DESIRE software environment. The knowledge used by the Utility Agent and the Customer Agents in the prototype, is briefly described below.

6.1. The Utility Agent in the Prototype System

In this prototype system the Utility Agent communicates all Customer Agents the same announcements, in compliance with Swedish law. The predicted balance between the consumption and the production of electricity, is determined by the following formulae:

$$\text{predicted_use_with_cutdown}(c) = \begin{cases} \text{predicted_use}(c) & \text{if } (1 - \text{cutdown}(c)) \cdot \text{allowed_use}(c) \geq \text{predicted_use}(c) \\ (1 - \text{cutdown}(c)) \cdot \text{allowed_use}(c) & \text{otherwise} \end{cases}$$

$$\text{predicted_overuse} = \sum_{c \text{ in } CA} \text{predicted_use_with_cutdown}(c) - \text{normal_use}$$

$$\text{overuse} = \frac{\text{predicted_overuse}}{\text{normal_use}}$$

$$\text{new_reward} = \text{reward} + \beta \cdot \text{overuse} \cdot (1 - \text{reward}/\text{max_reward}) \cdot \text{reward}$$

In the prototype system, the factor beta determines how steeply the reward values increase; in the current system it has a constant value. 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. It never exceeds the maximal reward, due to the

logistic factor

$$(1 - \text{reward}/\text{max_reward})$$

The negotiation process ends when the difference between the new reward values and the (old) reward values is less than or equal to 1.

The graphical interface of the Utility Agent in the prototype is depicted in Figure 6 (this is part of the user interface implemented specifically for this prototype

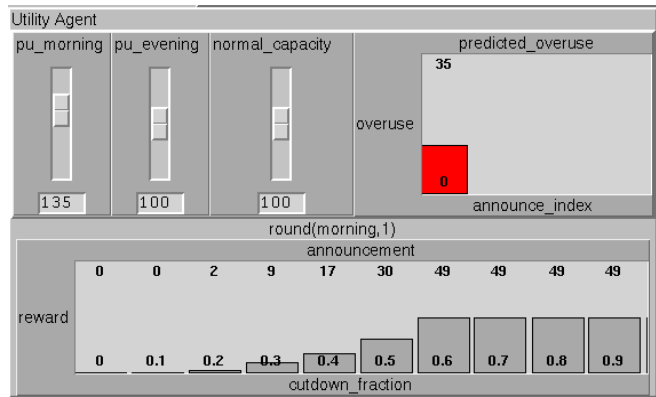


Figure 6. The Utility Agent during a negotiation process: initial phase

system, as an extension of the DESIRE software environment). As shown in Figure 6, normal capacity is 100, whereas the predicted usage is 135, which indicates a predicted overuse of 35 (depicted in the right upper part), in the first round of negotiation. In the lower part of Figure 6, 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 7, as a consequence of the negotiation process, the predicted overuse has been reduced to 13. In the lower part of Figure 7, the reward announced in the third round of negotiation (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, ...).

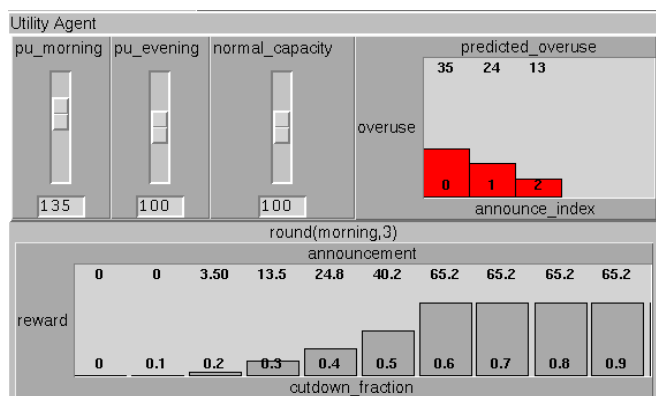


Figure 7. The Utility Agent during a negotiation process: final phase

6.2. A Customer Agent in the Prototype System

Within the Customer Agent, knowledge of the customers preferences is represented in the form of a cut-down-reward table. The cut-down-reward table specifies the percentage with which a Customer Agent is willing to decrease (cut-down) its electricity usage, given a specific level of financial compensation. The graphical interface of a Customer Agent in the prototype system is depicted in Figures 8 and 9, for different phases in the negotiation process.

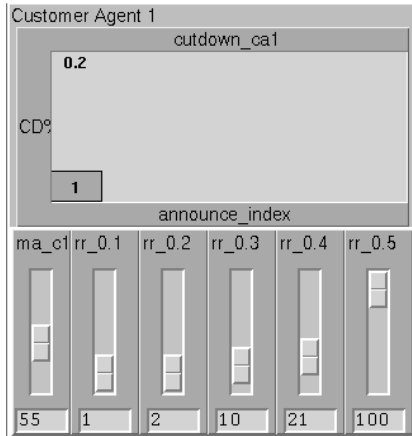


Figure 8. The Customer Agent during a negotiation process: initial phase

As shown in Figures 8 and 9 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

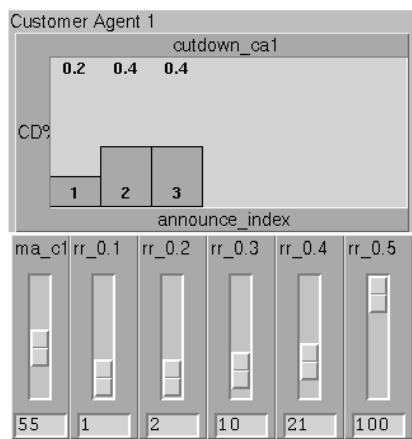


Figure 9. The Customer Agent during a negotiation process: final phase

the Utility Agent, is an acceptable cut-down. In the first round of negotiation, as depicted in Figure 9, 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 8, 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.

7. Discussion

A multi-agent approach to the design and implementation of large open distributed industrial systems has shown to be promising. To come to 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 development 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. More information on the compositional development method for multi-agent systems DESIRE that was used can be found at [URL1]. In [2] the compositional verification method for multi-agent systems introduced in [7] is used to verify the multi-agent system discussed in this paper as part of the design process.

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 ISES (cf. [URL2]) project within EnerSearch (cf. [URL3]) is an attempt to promote a platform for this type of R&D coupled with field tests.

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 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. 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, [12]) 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.

Relevant URL's

[URL1] The DESIRE project and course description at Vrije Universiteit Amsterdam; www.cs.vu.nl/~wai/demas

[URL2] The ISES project. Home page of EnerSearch AB; www.enersearch.se/

[URL3] Research program Societies of Computation (SoC) at HK/R; www.sikt.hk-r.se/~soc/

References

[1] Akkermans, H., Ygge, F. and Gustavsson, R., HOMEBOOTS: 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, Proc. of the Third International Conference on Multi-Agent Systems, ICMAS'98. IEEE Computer Society Press, 1998. To appear.

[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., Treur, J., Formalisation of a cooperation model based on joint intentions. In: J.P.

Müller, M.J. Wooldridge, N.R. Jennings (eds.), Intelligent Agents III (Proc. of the Third Int. Workshop on Agent Theories, Architectures and Languages, ATAL'96), Lecture Notes in AI, vol. 1193, Springer Verlag, 1997, pp. 141-155.

[5] Gustavsson, R.: Requirements on Information Systems as Business Enablers. Invited paper. In Proc. of DA/DSM Europe DistribuTECH'97, PennWell 1997.

[6] Hägg, S., A Senteniel Approach to Fault Handling in Multi-Agent Systems. To appear in Proc. of the Fourth Pacific Rim International Conference on Artificial Intelligence, PRICAI-96, 1996.

[7] Jonker, C.M., and Treur, J., Compositional verification of multi-agent systems: a formal analysis of pro-activeness and reactiveness. In: W.P. de Roeper, H. Langmaack, A. Pnueli (eds.), Proc. Int. Workshop on Compositionality, COMPOS'97, Springer Verlag, 1998. In press.

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

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

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

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

[12] 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.