A Generic Strategy Framework for Policy-directed Autonomous Trading Agents

Steffen Lamparter, Silvio Becher, Michael Pirker

Corporate Technology, Siemens AG, Munich
{Steffen.Lampeter|Silvio.Becher|Michael.Pirker}@simens.com

Abstract: In this paper, a strategy framework for coordinating decentralized autonomous agents is presented and applied within a smart energy grid. The framework brings together the concept of policy-based computing and market-based coordination. In this context, agents can be seen as self-interested entities that are governed by their local policies. Efficient coordination between these self-interested agents is realized through a market mechanism that incentivizes the agents to reveal their policies to the market. By knowing the agent's policies, an efficient solution for the overall system can be determined. Leveraging a declarative, policy-based approach facilities the specification of highly customizable strategies that can be easily adapted to various resources and markets. As an example for a real-world application, the framework is used for efficient balancing of decentralized energy supply (e.g. photovoltaic, wind power, CHP) and demand (e.g. households, businesses) in a power grid.

1 Introduction

One of the fundamental questions within multi-agent systems research is how autonomous, self-interested agents can be coordinated in a way that the global performance of the system is maximized. In recent years, market mechanisms have become popular as efficient means for coordinating self-interesting agents competing with each other about scarce resources/tasks. A wide range of different negotiation or auction mechanisms have been proposed in this context – many of them exhibiting favorable properties such as efficiency or/and incentive compatibility. The latter becomes an important feature in competitive multi-agent systems as markets incentivize the autonomous agents to reveal their private information truthfully to the market. Having full information from the participating agents the market could determine a global optimal solution that cannot easily manipulated by malicious agents.

When using a market-based coordination framework in a multi-agent system, the design of the agent strategies for interacting with the market mechanisms becomes a crucial element. In this context, two major streams of work can be identified: (i)

On the one hand there are some widely used agent platforms such as JADE (o. J.) (Java Agent Development Framework) or Cougar (o. J.). While they provide some basic coordination (including some market) mechanisms such as negotiation and auction protocols, they do not support the agent developers in specifying domain-specific agent strategies for participating in the coordination process. This makes the development of agents often very cumbersome and complicated since for each resource and market mechanism different strategies are needed and no design time support for strategy development is provided. (ii) On the other hand, there are a lot of powerful systems implementing (often domain specific) marketplaces, which also provide some means for developing the corresponding agent strategies. The Trading Agent Competition (Wellman et al. 2001) provides a testbed for non-cooperative agent strategies. Commercially highly relevant application examples can be found within the financial domain, where the area of algorithmic trading has become increasingly important over the last years (Parkes and Huberman 2001). However, these strategies are specific for a concrete market mechanism and domain. Therefore, they are not geared towards highly configurable strategies that provide the flexibility to add resources at runtime. For example, when developing an agent-based energy market, agents representing households must adapt their strategy in a plug'n'play fashion when adding or removing appliances in the household. While there is work that addresses agent strategy design using more general setting (Giménez-Funes et al. 1998, Vytelingum et al. 2005), these approaches still lack the flexibility and configurability required to support highly configurable strategies.

To address this issue, the paper extends the generic strategy framework presented in Vytelingum et al. (2005) to support developers in specifying device specific agent strategies. It can be used to implement widely autonomous bidding agents that are able to interact with different market mechanisms in various domains. The architecture leverages the idea of *policies* (Kephart and Walsh 2004) for realizing a high-degree of autonomy while making sure that the agents behave within a predefined action space. As these policies are declarative descriptions they can be added and removed at runtime which allows to adapt the strategies dynamically. For example, in the energy market scenario new appliances in the household may come with their policies how they can be regulated. These policies can be used by the energy trading agent to adapt its strategy to the new setting.

The paper is structured as follows. In Section 2, we first introduce the generic agent strategy framework. Subsequently, in Section 3 we describe a realization of the architecture within the energy domain and conclude in Section 4 with a short outlook.

2 Agent Architecture

In this section, we introduce the agent architecture for the agents that allow automated trading on the energy markets. The automation involves autonomous acquisition, storage and processing of information by the agent which is also displayed by the steps perception, cognition and action in Figure 1. As also depicted in the figure, these steps can be assigned to the three layers of agent strategy design as defined in Vytelingum et al. (2005). In the following we discuss each layer (step) in more detail.

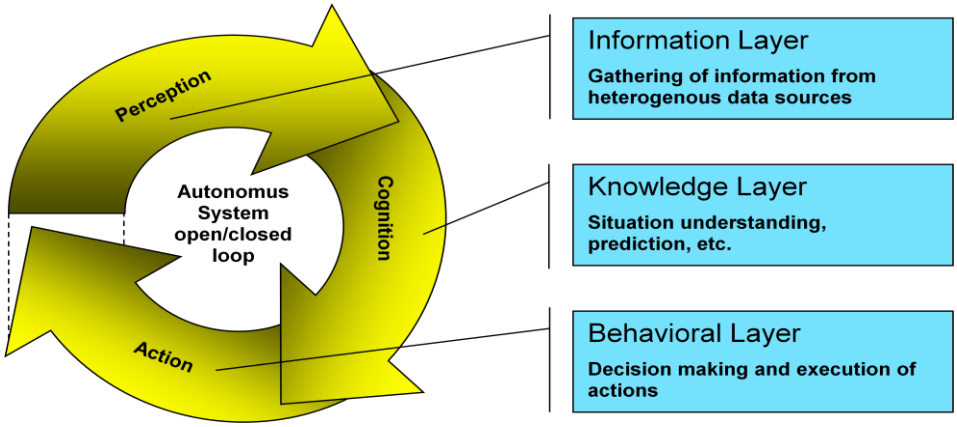

Figure 1: Agent Strategy Framework

2.1 Information Layer

The information layer contains information which an agent $i \in I$ has gathered from the market, the environment or its own private information at time t_k with $k \in \mathbb{N}$. Much in line with Vytelingum et al. (2005), we can define the market and agent state as follows.

Definition 1 (Market State) A market state captures the public information that is available at a certain point in time t_k. Is is defined via a vector $\theta_{\mathcal{M}}(t_k) = (x, B_{t_k}, price_{t_k}, q_{t_k})$ where x represents the trading object, $price_{t_k}$ the clearing price and q_{t_k} the overall traded quantity at time t_k, and B_{t_k} the orders to buy or sell energy which are present in the order book at time t_k.[1] The expressivity of a bid b is defined through the bidding language in the market (c.f. Section 3.3).

[1] Note that full disclosure of B_{t_k} is only available for markets with public order book. Markets with sealed bids usually publish only the highest bid and ask. We thus define the market state as $p_{\mathcal{M}}(t_k) = (x, ask_{t_k}, bid_{t_k}, price_{t_k}, q_{t_k})$.

Given the set of publicly available information on the market, the agents internal state containing private information such as preferences is defined below.

Definition 2 (Agent State) An agent i's state at time t_k is defined by the vector $\theta_i(t_k) = (id_i, q_{i,t_k}, v_{i,t_k}, comp_{i,t_k})$ where id_i specifies whether the agent acts as buyer or seller, q_{i,t_k} defines the quantity of energy required (or provided) by the agent at time t_k, v_{i,t_k} the reservation price of an agent, and $comp_{i,t_k}$ the computational resources available at the given point in time.

In addition to the Market and Agent State there might be additional information necessary dependent on the application scenario. For example, assuming a smart grid scenario where decentralized energy demand and supply is allocated using a market mechanism the state of the power grid might also be relevant for the calculation of the optimal allocation. As such additional information is not yet perceived by the agent or market, we add an additional *Environment State* which captures this application specific information.

Definition 3 (Environment State) The environment state $\theta_\varepsilon(t_k)$ captures the values of a set of applications specific variables $(e_{1,t_k}, \dots, e_{n,t_k})$ over time t_k. The variables are not part of the agent itself nor can they be observed on the market directly. They can be rather perceived by the agent when observing its direct environment.

Typically, information about environment states is perceived via sensors (e.g. measurement of frequency or voltage in a electrical grid) and is aggregated to a higher level of abstraction that can be interpreted by the agents.

2.2 Knowledge Layer

On this layer previously defined information is combined with polices given at design time. These polices capture general rules that define admissible actions and thereby constrains the strategy space of an agent. In the following, we adopt a rather general approach for defining the strategy space S of a market agent.

Definition 4 (Strategy Space) The strategy space S_i available to an agent i at time t_k is defined by a cartesian product $S_i(t_k) = M \times \Theta_i(t_k) \times A_i$ covering the agent i's action space A_i, the possible states $\Theta_i(t_k)$ and the market mechanism descriptions M. Consequently, a strategy $s \in S_i$ available to agent i defines which action $a \in A$ should be executed for a given market mechanism $m \in M$ in a given state $\left(\theta_i(t_k), \theta_{\mathcal{M}}(t_k), \theta_\varepsilon(t_k)\right) \in \Theta_i(t_k)$.

The description of a market mechanism is important if more than one mechanism (e.g. onesided mechanisms like the english or dutch auctions, or double auctions) should be supported. There are several approaches how market processes can be formalized and described (Lomuscio et al. 2001, Bartolini et al. 2005, Love et al. 2008). For example, the Game Description Language GDL (Love et al. 2008) formalizes games – which are also general formulation of auction protocols – using Datalog and thereby also formally describes the action space for the agents that can be reused in our strategy definition.

By constraining the strategy space S_i policies define whether a certain action is allowed for a market mechanism in a given state. In general policies can be seen as a set of *constraints* that have to be met by a solution to a certain problem. In literature solving a problem specified by a set of constraints is denoted as *constraint satisfaction problems (CSP)* (Dechter 2003). A CSP is described by a set of attribute identifiers L – each representing one aspect of the problem – and the domains of these attributes D. As our goal is to specify constraints over the strategy space we assume $D = S$.

Definition 5 (Constraint Satisfaction Problem) A CSP within the scope of this paper is a tuple (L, S, Φ), where L represent the involved attributes of the problem, D the domains of these attributes and Φ a set of constraints that defines whether a given configuration $c \in C = D_1 \times \ldots \ldots \times D_n$ is allowed or not. A constraint $\phi \in \Phi$ consists of a *scope* and a *relation*, i.e. $\phi = (scp, rel)$. The scope $scp$of a constraint is a k-tuple of attribute labels $(l_1, \ldots, l_k) \in L^+$ and the relation rel of a constraint the set of k-tuples defining the allowed attribute values $rel \subseteq D_1 \times \ldots \times D_k$ for the given scope.[2] As enumeration of all possible relations is often not feasible (e.g. for infinite domains) we allow relations to be defined via predicates $p_\phi: D_1 \times \ldots \times D_k \rightarrow rel_\phi$.

A strategy $s \in S$ is evaluated with respect to a k-ary constraint with $scp_\phi = (l_1, \ldots, l_k)$ and $rel_\phi = \{(d_{11}^{rel}, \ldots, d_{k1}^{rel}), \ldots, (d_{1q}^{rel}, \ldots, d_{kq}^{rel})\}$ as defined below:

$$G_\phi(s) = \begin{cases} 1 \; if \; \exists j \in [1,q], \forall i \in [1,k] : match\left(d_{i,j}^{rel}, d_i^c\right) = true \\ 0 \; else \end{cases} \quad (1)$$

The Equation 2 is evaluated to 1 for a given constraint ϕ and a given strategy s if there is a tuple in the relation rel_ϕ , for which each attribute value d_{ij}^{rel} matches the corresponding attribute value d_i^s in the configuration. The predicate *match* is used to compare two attribute values. In the most simple case, where attribute

[2] Note that the definition assumes that the constraint is defined on the first k attributes.

values represent "flat" datatypes, such as integers or strings, this could be realized by a simple syntacticcomparison, e.g. match $\left(d_{ij}^{rel}, d_i^s\right) = true$ if $d_{ij}^{rel} = d_i^s$.

In order to judge a strategy $s \in S$ as admissible, Equation 2 has to hold for all constraints $\phi \in \Phi$. This is ensured by the following formula:

$$G_\Phi(s) = \prod_{\phi \in \Phi} G_\phi(s) \tag{2}$$

Based on the evaluation of constraints we are able to define the set of acceptable strategies $\hat{S}_i \subseteq S_i$ for agent i by removing the strategies that violate at least one constraint:

$$\hat{S}_i = \{s \in S | G_\Phi(s) = 1\} \tag{3}$$

The set $\hat{S}_i$ is therefore the strategy space that has to be considered in the behavioral layer where the best strategy is selected and executed.

2.3 Behavioral Layer

The behavioral layer is responsible for deciding on the best action to take at each point in time. In order to select an action, we have to rank the strategies according to preferences of the agent. This can be done by defining a utility function $u_i : \hat{s} \rightarrow \mathbb{R}^+$ over the relevant strategy space $\hat{S}$. Then the best strategy is determined by solving the following maximization problem:

$$s' = \arg\max_{s \in \hat{S}} \mathrm{u_i}(s) \tag{3}$$

Given the best strategy, we execute the action a' contained in the tuple s' which typically involves sending one or more bids to the market. Note that the utility function and strategy space typically depends on the application scenario as well as market mechanism used. In this context, the available set of actions A is determined by the bidding language of the market mechanism used. For example, in an one-shot auction only one bid can be send to mechanism while in sequential auction protocols more complex actions might be involved. Another important application-dependency is the selection of the best price that should be sent to the market. While for incentive compatible mechanisms bidding the real reservation price v_{i,t_k} is the dominant equilibrium strategy for all agents (independent of the strategy of the other agents), for other market mechanisms this is not the case as strategic over- or underbidding might increase the expected return for individual agents. Due to this scenario-dependency we outline the application of the crete Smart Grid scenario in Section 3.

2.4 Summary

Before coming to a concrete application scenario of the strategy framework we shortly summarize the agent reasoning process as shown in Figure 2. In the first step, market, environment and agent state information are perceived and passed to the cognition step. Here the information is interpreted using the agents policies. This leads to a set of acceptable strategies that are evaluated using a given utility function. The action contained in the utility maximizing strategy is finally executed and the corresponding bid(s) is/are send to the market.

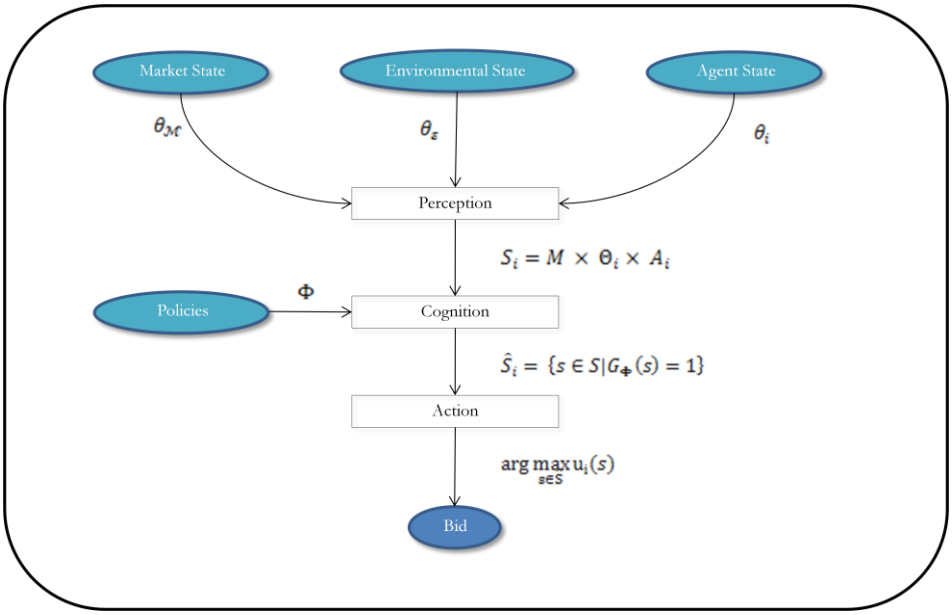

Figure 2: Bidding process according to strategy framework. Note that for sake of reading time-dependency is omitted in the figure.

3 Application: Smart Energy Grid

In this section, the agent framework defined above is applied within the energy domain. The power network will be penetrated more and more with decentralized energy supplier like wind power, photovoltaic and CHPs (Combined Heat and Power) connected to the distribution grid (low and medium voltage). Some of them are also fluctuating and only limited controllable. For a economical, ecological and network stable power generation and consumption a intelligent balancing of supply and demand is needed.

For that reason producers and consumers can coordinate each other instantiating and using local energy markets. Introducing intelligent coordination mechanisms is to assure an optimal balancing of the energy supply and demand while gua-

ranteeing grid capacity constraints and additional quality criteria. The rational behind using electronic markets as a coordination mechanism is the decentral nature of our scenario without a fully informed central agent. In such scenarios where self-interested provider, consumer or prosumer agents try to optimize their personal utility in cooperation or competition with other agents, a coordination mechanism must incentivize the individual agents to reveal their goals to reach a global optimum for the overall system. As discussed above, markets can provide efficient mechanisms in the presence of selfish agent that optimize social welfare in the market. As each of the appliances and decentralized energy supplier require needs its own adapted bidding strategies, a strategy framework based on policies that can be seamlessly combined provides thus the right means to realize customizable agent bidding strategies.

In order to fully specify a market mechanism, we have to define two aspects: a bidding language for communicating the agents' preferences to the market and the mechanism itself consisting of the allocation function $\mathcal{X}$ and pricing function $\mathcal{P}$.

3.1 Bidding language

Generally, a bidding language defines the preferences that an agent wants to reveal to the market, i.e. bidding is about reporting the preference function v_i. When designing a bidding language, there is a trade-off between the expressivity of the language, the privacy loss of users and the complexity of the market mechanism. For example, a bidding language could support expressing how valuation changes depending on time, on available units, etc. For the energy scenario we therefore decided to use a quite restricted bidding language. This has the advantages that we are able to implement a quite efficient mechanism and the agents do not have to reveal too much private data to the mechanism. However, note that this could lead to less efficient markets if dependencies between bids cannot be compensated with local agent intelligence (e.g. smart splitting of originally complex into simple bids). A general overview of bidding language with different expressivity can be found in Nisan (2000).

Based on these considerations, we define the set of *requests* to buy energy B^R and the set of *offers* to sell energy B^O where $B^O \cap B^R = \emptyset$ and $B^O \cup B^R = B$ holds. A *bid* each $b_i \in B$ represents a tuple $b = (v_i, q_i)$ where v_i defines the reservation price for a single unit of the good x (i.e. maximal price for requests and minimal prices for offers) and $q_i : X \to \mathbb{R}^+$ defines how many units of the good are desired/provided. As for the good energy it is reasonable to assume divisibility, the overall reservation price for a good x is given by $v_i(x)\ q_i(x)$ or simply $v_i q_i$.

3.2 Mechanism Design

Having defined how agents submit their bids and asks to the market, we are able to define the choice and payment functions. As we have multiple producer and consumer agents in the energy market, our goal in this section is to design a two-sided market mechanism – often called *double auction* or *exchange*. In addition, for energy markets we can assume divisible bids (i.e. partial execution of bids), a call market (i.e. accumulation of bids over a period of time), buy-side and sell-side aggregation of bids, and risk-neutral agents with quasi-linear preferences.

Given the set of requests B^R and offers B^O, the winner determination problem is defined as allocation function that maximizes the social welfare in the market. The corresponding linear program is defined as follows.

$$\max_{z_{ij}} \quad \sum_{b_i \in B^R} \sum_{b_j \in B^O} (v_i(x) - v_j(x)) q_j z_{ij} \tag{4}$$

$$s.t. \quad \sum_{b_j \in B^O} q_j(x) z_{ij} \leq q_i(x) \qquad \forall b_i \in B^R \tag{5}$$

$$\sum_{b_i \in B^R} z_{ij} \leq 1 \qquad \forall b_j \in B^O \tag{6}$$

$$0 \leq z_{ij} \leq 1 \qquad \forall b_i \in B^R, \forall b_j \in B^O \tag{7}$$

Unfortunately, defining the payment function (and the mechanism as a whole) in a way that the resulting double auction is efficient, incentive compatible and budget balanced mechanism is generally impossible as stated by the seminal impossibility theorem of Myerson and Satterthwaite (1983). However, it is possible to design a mechanism that meets two of the three properties. In literature, several different variants have been proposed. Using the well-known Vickrey-Clark-Groves (VCG) mechanism we get an efficient and incentive compatible auction, however, budget-balance cannot be guaranteed any more. To calculate prices the offers B^O have to be arranged in descending order $(b_1, \ldots, b_j, \ldots, b_n)$ and requests B^R in ascending order $(a_1, \ldots, a_i, \ldots, a_m)$ w.r.t. their prices. We then determine the index l where $v_l^b \leq v_l^a$ with v_l^b in b_j and v_l^a in a_i. Given this index l we set the price for buyers to $max(a^l, b^{l+1})$ and for sellers $min(a^{l+1}, b^l)$. Other approaches which implement a budget balanced mechanism are – for instance – presented in McAfee (1992).

3.3 Bidding Strategy

Given the market mechanism previously specified in this section, we are able to further define a bidding strategy for energy markets using the agent strategy framework introduced in Section 2. As a wide range of different systems are connected to an electricity grid (ranging from appliances of private households to complex industrial machines) and each of these systems has to implement different bidding strategies, a generic strategy framework as specified in Section 2 greatly facilitates the system implementation task.

In the following we give examples how the framework can be used to define strategies for some typical smart grid agents.

3.3.1 Defining the information layer

First, we have to adapt the information layer to the the smart grid market scenario. This requires to adapt the market state to the market mechanism $\theta_{\mathcal{M}}(t_k) = (x, B_{t_k}, price_{t_k}, q_{t_k})$ defined in Section 3.2. As electricity is a highly homogenous good, the trading object x represents simply electricity according to the IEC Norm 60038:1983 with a predefined set of quality criteria, such as frequency between $50\sim hz$ and a voltage level of $230V$ with a tolerance of $\pm 10V$. As the market mechanism does not reveal the bids of other participants we assume $B = \emptyset$. The $price_{t_k}$ is a tuple $(\max(a^l, b^{l+1}), \min(a^{l+1}, b^l))$ representing the bid/ask-spread in the market and q_{t_k} is the overall amount of electricity traded at time t_k measured in *kwh*.

Second, the agent's private state $\theta_i(t_k) = (id_i, q_{i,t_k}, v_{i,t_k}, comp_{i,t_k})$ is adapted as follows: the agent is either a buyer, seller or prosumer, i.e. $id_i = \{seller, buyer, prosumer\}$, q_{i,t_k} represent the maximal amount of electricity that can be presumed by agent i at time t_k, v_{i,t_k} is the maximal/minimal valuation of a single *kwh* electricity, and $comp_{i,t_k}$ is currently not used within the smart grid scenario.

Third, the environment state observable by all agents comprises information about the status of the electricity network that can be measured via sensors, such as frequency e_{f,t_k}, voltage e_{v,t_k}, or current e_{c,t_k} and time t_{k,t_k}. Consequently, $\theta_i(t_k) = (e_{f,t_k}, e_{v,t_k}, e_{c,t_k})$. In addition, specific sensor data might be available to some of the agents which could include the current temperature of within a fridge, the current load of a manufacturing machine, etc.

3.3.2 Defining the knowledge layer

In the knowledge layer the general guidelines are specified how a specific agent should behave. This is done by specifying a set of policies constraining the allowed

strategy space. In order to get an intuition about the policy driven approach, we give some simple examples for such policies in the smart grid domain.

Demand Profile: A customer (or prosumer) has to be able to specify his preferences with respect to the electricity demand. Typically, the overall required amount of electricity $q_{i,t_k}^{overall}$ is split in an amount $\propto_{i,t_k}^{overall}$ essentially required by agent i and the sheddable load $(1-\propto)q_{i,t_k}$ that is negotiable according to the market price. In this context, $\propto \in [0,1]$ is the share of inflexible demand. Thus, the minimal required load can be expressed by constraining q_{i,t_k} (part of the agent state) using the constraint $\phi_{minQ} = (q_{i,t_k}, rel_{minQ})$ with $rel_{minQ} = \{q | q > \alpha q_{i,t_k}^{overall}$.

Appliances Specification: As the share of inflexible and negotiable energy depends on the appliances of the customer (i.e. $q_{i,t_k}^{overall} = q_{i,t_k}^{App1} + q_{i,t_k}^{App2} + \ldots$), the demand profile can be constructed from individual policies coming with this appliances. This also means that ϕ_{minQ} could also be defined for individual appliances separately. In addition, policies can regulate whether an appliance such as a fridge, industrial manufacturing machine, etc. can either (i) reduce total load in time t_k to some extend or (ii) shift load from time t_k with high energy prices to a later point in time $t_{k+\epsilon}$ where energy prices are cheaper (i.e. load shedding). An example for a constraint defining that a certain quantity of load can be shifted within a timeframe ϵ can be done with the following policy: $\phi_{shedding} = (\{q_{i,t_k}^{fridge}\}, rel_{shedding})$ with $rel_{shedding} = \{q_{i,t_k} | \sum_{t \in [t_s,t_k]} q_{i,t} > q_i^{fridge} \wedge tk \leq ts+\epsilon$ stating that aggregated electricity within the period $ts, ts+\epsilon$ has to be above a given threshold q_i^{fridge}.

Pricing: While for the inflexible load a unlimited price is offered by the customer, the maximal price for the shaddable load depends on the customers attitude. A very simple strategy to define the extend to which electricity is bought above the $\alpha q_{i,t_k}$-level could be defined using the historical market prices. Let $\eta_i \in \mathbb{R}$ be the price elasticity of a customer with $\eta_i = (\Delta q_i / q_i)/(\Delta p_i / p_i)$. Therefore, a highly negative η_i represents a "savaholic" whereas a _i near zero represents a rather convenient customer with a low responsiveness w.r.t. the market price. Therefore, we can define a constraint on the customer's reservation price $\phi_{maxPrice} = (\{v_{i,t_k}\}, rel_{maxPrice})$ with $rel_{maxPrice} = \{v \in \mathbb{R} | v < \eta_i v_{i,t_k}\}$.

Analogously to the policies on customer side, policies for electricity producers could be defined. For example, each type of energy plant such as solar plants, wind turbines, or combined heat and power plants come with common policy sets that

regulate whether/how production schedules can be changed dynamically, define the the marginal costs, etc. In addition, policies may specify regulatory constraints important for the security of energy supplies or antitrust guidelines.

As policy specification are purely declarative, policies from different appliances ϕ can be combined to policy sets Φ which are evaluated using 2 leading to the set of strategic. This is a huge advantage of the framework as appliances are constantly added or removed and this should be supported in a plug'n'play fashion.

3.3.3 Defining the behavior layer

As we rely on an incentive compatible market mechanism in our application example, the behavior layer can be much simplified as already discussed in Section 2.3. In this special case, the action space is simply $A = \mathbb{R} \times \mathbb{R}$ where a tuple represents a bid $b = (v, q)$ defining the maximal valuation of agent v and the required quantity q. As the dominant – and thus u maximizing – strategy of a rational agent is to reveal its true valuation and maximal quantity, the only rational action is to choose strategy $s \in \hat{S}$ with minimal deviation from v_{i,t_k} and q_{i,t_k} defined in θ_i.

4 Conclusion and Outlook

In this paper, a formal bidding strategy framework is proposed that specifies the information available to the agents at runtime, the policies defined at agent design time, and the behavior that selects the optimal action. As the framework follows the idea of a declarative policy based approach, the agent strategies can be seamlessly composed from resource-specific policies (e.g. appliance/plant specific polices) and dynamically changed during runtime. This is essential for applying agent technology in many realworld scenarios such as the introduced smart grid application.

Based on a first implementation of the system for the energy domain, we are currently evaluating whether the agent strategy framework is on the one hand expressive enough to support the specification of the required strategies and on the other hand sufficient intuitive for agent developers to be applied in real world settings. For the latter we are currently working at appropriate APIs and user interfaces. Furthermore, we plan to carry the framework forward to additional scenarios (with different market mechanisms) in order to ensure the generic applicability.

References

Bartolini C, Priest C, Jennings N. R. (2005) A Software Framework for Automated Negotiation. In: Choren R et al. (Hrsg.) SEMMAS 2004, LNCS 3390, 213-235.

Cougar (o. J.)– Cognitive Agent Architecture. http://www.cougaar.org/.

Dechter R (2003) Constraint Processing. Morgan Kaufmann, 2003.

Giménez-Funes E, Godo L, Rodríguez-Aguilar J, Garcia-Calvés P. (1998) Designing Bidding Strategies for Trading Agents in Electronic Auctions. In: ICMAS '98: Proceedings of the 3rd International Conference on Multi Agent Systems, page 136, Washington, DC, USA, 1998. IEEE Computer Society.

Jade (o. J.) JADE – Java Agent Development Framework. http://jade.tilab.com/.

Kephart J O, Walsh W E. (2004) An artificial intelligence perspective on autonomic computing policies. In: Policies for Distributed Systems and Networks, 2004., POLICY 2004., 3-12.

Love N, Hinrichs T, Haley D, Schkufza E, Genesereth M (2008). General Game Playing: Game Description Language Specification. Technical Report LG-2006-01, Stanford University, March 2008.

Lomuscio A, Wooldridge M, Jennings N R (2001). A Classification Scheme for Negotiation in Electronic Commerce. In: Agent Mediated Electronic Commerce, The European AgentLink Perspective. London, UK, Springer-Verlag, 19–33.

McAfee R P (1992). A dominant strategy double auction. Journal of Economic Theory 56(2), 434–450.

Myerson R B, Satterthwaite M A (1983). Efficient mechanisms for bilateral trading. Journal of Economic Theory 29(2):265–281.

Nisan N (2000). Bidding and allocation in combinatorial auctions. In: ACM Conference on Electronic Commerce, 1–12.

Parkes D C, Huberman B A (2001). Multiagent Cooperative Search for Portfolio Selection. Games and Economic Behavior, 35, 124–165.

Vytelingum P, Dash R K, He M, Jennings N R. (2005) A Framework for Designing Strategies for Trading Agents. In IJCAIWorkshop on Trading Agent Design and Analysis, 7–13.

Wellman M P, Wurman P R, O'Malley K, Bangera R, de Lin S, Reeves D M, Walsh W E (2001). Designing the Market Game for a Trading Agent Competition. In: IEEE Internet Computing, 5(2), 43– 51.