

Engineering Autonomic Electronic Institutions

Josep Lluís Arcos, Juan A. Rodríguez-Aguilar, and Bruno Rosell

IIIA, Artificial Intelligence Research Institute
CSIC, Spanish National Research Council
08193 Bellaterra, Spain
{arcos,pablo,jar,sierra}@iiia.csic.es

Abstract. There is a growing interest in the study and development of self-* systems motivated by the need for information systems capable of self-management in distributed, open, and dynamic scenarios. Unfortunately, there is a lack of frameworks that support the intricate task of developing self-* systems. We try to make headway along this direction by introducing a framework, EIDE-*, to support the engineering of a particular type of self-* systems, namely *autonomic electronic institutions*: regulated environments capable of adapting their norms to comply with institutional goals despite the varying behaviours of their participating agents.

1 Introduction

There is a growing interest in the study and development of self-* systems [12] (where the * sign indicates a variety of properties: self-organization, self-configuration, self-diagnosis, self-repair, etc) motivated by the need for information systems capable of self-management in distributed, open, and dynamic scenarios. A particular approximation to the construction of self-* systems is represented by the vision of autonomic computing [10], which constitutes an approximation to computing systems with a minimal human interference. Unfortunately, there is a lack of frameworks that support the intricate task of developing systems with autonomic capabilities. As an exception we can consider the Living Systems framework [17]. Nonetheless, it is hard to conceive a general-purpose development framework for self-* systems. Therefore, our endeavour can be eased if we depart from a particular model of open system [9] that can eventually be endowed with self-management capabilities. A review of the literature indicates that electronic institutions (EIs) [5], regulated environments wherein the relevant interactions among participating agents take place, have proved to be valuable to develop open agent systems. Indeed, EIs do even count on a development environment (EIDE) to ease their engineering [1]. However, the challenges of building open systems as EIs are still considerable, not only because of the inherent complexity involved in having adequate interoperation of heterogeneous agents, but also because the need for adapting regulations to comply with institutional goals despite varying agents' behaviours. Particularly, when dealing with self-interested agents as noticed in [3].

In this paper we try to make headway in the engineering of self-* systems by introducing a framework to support the development of a particular type of these systems, namely *autonomic electronic institutions*: EIs with self capabilities. The framework we introduce, EIDE-*, must be regarded as an extension of EIDE, the current development framework for EIs. Specifically, the new engineering requirements imposed by the autonomic capabilities brought about a new approach in the agent development tool (*aBUILDER*) and in the simulation tool (*SIMDEI*).

Furthermore, we illustrate the capabilities of the framework through the analysis of a power electricity market inspired on the actual operation of the Spanish electricity market. The main goal of an electricity market is to provide a set of rules for conciliating the demand of electricity and its generation. There are two issues that must be avoided: a lack of production that can leave customers without supply and an unwanted overproduction. Moreover, these goals have to be achieved while maintaining a reasonable electricity price. We show how EIDE-* can support self-configuration policies in such setting.

The paper is organized as follows: Section 2 introduces the formal concepts around autonomic electronic institutions. Section 3 describes the set of tools we provide for helping in the engineering of autonomic electronic institutions. Section 4 presents the electricity market problem and shows how all the concepts and tools are used to design a specific institution. Finally, conclusions and future work are presented in section 5.

2 Autonomic Electronic Institutions

Loosely speaking, EIs are computational realizations of traditional institutions (cf. North [14] pp. 3 ss.); that is, coordination artifacts that establish an environment where agents interact according to stated conventions, and in such a way that interactions within the (electronic) institution would *count as* interactions in the actual world.

According to the basic definition of an electronic institution (see [5]), an EI is composed of three components: a dialogical framework that establishes the social structure, the ontology, and a communication language to be used by participating agents; a performative structure defining the activities along with their relationships; and a set of norms defining the consequences of agents' actions.

MAS applications are usually concerned with some external environment. The environment is application-specific and refers to the part of the world that is relevant to the MAS application. For instance, in the electricity market example that will be presented in section 4, the power demand is modeled by an equation-based tool that simulates real electrical consumption patterns.

Environments are plugged into EIs as institutional services [2]. In our approach, agents cannot directly sense and act over the environment. Instead, and likewise all interactions of external agents in the realm of an EI, they are *medi-*

ated by the institution wherein they interact. The link of an institution with an environment enriches the functionality of the EI components.

2.1 Self-Organizing Capabilities

From this basic definition of an EI we have extended the model to support self-configuration [3]. The notion of Autonomic Electronic Institutions (AEIs) has been proposed as a model for providing self-configuration capabilities to EIs. AEIs incorporate three new main components: an explicit set of *institutional goals* G , an *information model* I , and a *normative transition function* δ that allows to transform interaction conventions.

The main objective of an AEI is to accomplish its goals. For this purpose, an AEI has to be able to both dynamically observe/analyze the performance of the institution and to adapt its interaction conventions. We assume that an institution can observe its environment, the institutional state of the agents participating in the institution, and its own state to assess whether its goals are accomplished or not. Thus, from the observation of environmental properties, institutional properties, and agents institutional properties, an AEI maintains the information model I required to determine the fulfillment of goals.

Formally, we define the goals of an AEI as a tuple $G = \langle V, C \rangle$ composed of : (i) a set of reference values $V = \langle v_1, \dots, v_q \rangle$ where each v_j results from applying an evaluation function h_j upon the information model; $v = h(I), 1 \leq j \leq q$; and (ii) a finite set of constraints $C = \{c_1, \dots, c_p\}$ where each c_i is defined as an expression $g_i(V) \triangleleft [m_i, M_i]$ where $m_i, M_i \in \mathbb{R}$, \triangleleft stands for either \in or \notin , and g_i is a function over the reference values. In this manner, each goal is a constraint upon the reference values where each pair m_i and M_i defines an interval associated to the constraint. Thus, the institution achieves its goals if all $g_i(V)$ values satisfy their corresponding constraints of being within (or not) their associated intervals.

Finally, the normative transition function δ defines the set of actions allowed for re-configuring the institution at runtime. The re-configuration is performed by changing the interaction conventions. Specifically, δ actions will have effects over the performative structure and the normative rules. For instance, the role flow policy among activities can be modified by δ .

Nowadays, we are not dealing with the re-configuration of the dialogical framework (i.e. the social structure, the domain ontology, and the communication language are invariant).

Because staff agents are those in charge of the institutional activities, only staff agents will be allowed to observe the fulfillment of the institutional goals and will be able to change the interaction conventions.

3 Development and Simulation Framework

In order to facilitate the engineering of AEIs we have developed a set of software tools that give support to all the design and execution phases. These tools are integrated in the Development Environment for Autonomic Electronic Institutions

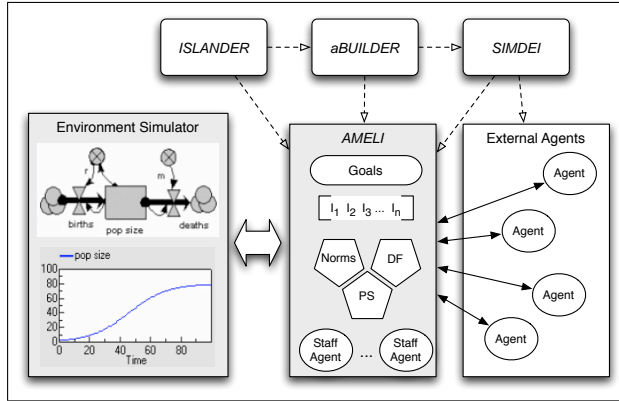


Fig. 1. The EIDE-* Framework.

(EIDE-*). EIDE-* allows for engineering both the institutional rules and the participating agents. Figure 1 depicts the EIDE-* framework. The tools provided by the EIDE-* framework are: a graphical tool that supports the specification and static verification of institutional rules (*ISLANDER*); an agent development tool (*aBUILDER*); a simulation tool to animate and analyse *ISLANDER* specifications (*SIMDEI*); and a software platform to run EIs (*AMELI*). All these tools have been enhanced to provide the new requirements of autonomic electronic institutions.

To design an AEI we have a tool, *ISLANDER* [6], that allows us to make a graphical specification of the AEI components and produces an XML file with the specification. That specification is used to enact instances of the institution, by agent designers to build agents that conform to the institutional conventions, and to design and run experiments with different agent populations.

The core of EIDE-* is *AMELI* [7], an institutional engine that provides a run-time middleware for the agents that participate in the enactment of a given institution. The middleware is deployed to *guarantee* the correct evolution of each scene, to *warrant* legal movements between scenes, and to *control* the obligations or commitments that participating agents acquire and fulfill. Furthermore, the middleware *handles* the information agents need within the institution. The *AMELI* generated middleware *mediates* between agents in order to facilitate agent communication within scenes. Broadly speaking, *AMELI* achieves those functions because on the one hand it generates the staff agents and the institutional *governors* that mediate all communications with external agents and, on the other hand, it handles all the institutional communication traffic by wrapping illocutions as messages that are handled by a standard agent-communication layer. *AMELI* has been extended so that staff agents can observe the fulfillment of the institutional goals and change the interaction conventions at run-time.

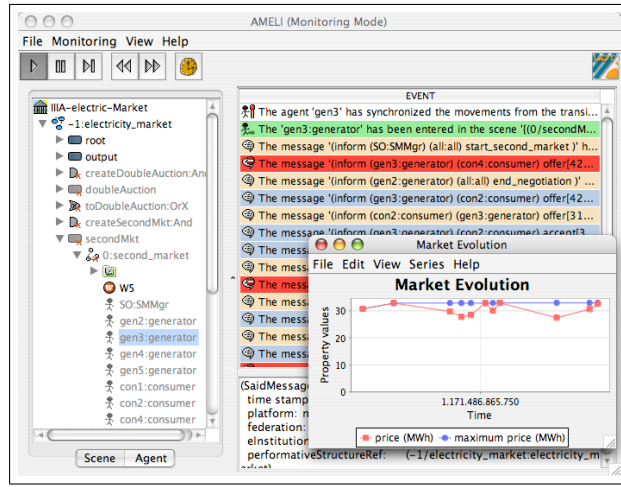


Fig. 2. Monitoring the Electricity Market.

Additionally, *AMELI* provides a set of new monitoring facilities that allow a graphical depiction of all the events that occur during the enactment of an AEI. Fairness, trust and accountability are the main motivations for the development of a monitoring tool that registers all interactions in a given enactment of an electronic institution [13, 18]. Giving accountability information to the participants increases their trust in the institution. This is specially important for electronic institutions where people delegate their tasks to agents. Furthermore, the tool permits them to analyse their agent(s) behaviour within the institution in order to improve it.

Figure 2 shows some of the monitoring facilities activated for the electricity market. The left frame contains a list of the institution's scenes and transitions along with their executions. In the monitoring snapshot shown in the figure, the execution of the secondary market is monitored at state *W5*. The right frame depicts the events occurring during scene execution: agents' entrance (second event); the utterance of valid (third event) and wrong (fourth event) illocutions; transitions caused by timeouts; and agents' exit. Furthermore, the monitoring tool allows the tracking of the institution information model and the tracking of the institution goals achievement. For instance, figure 2 depicts the tracking of the energy cost parameter. The chart allows the tracking of the evolution of the energy cost along the time together with the maximum cost (calculated using the monitoring facilities of the tool).

External environments are plugged into *AMELI* by implementing a required Java interface, the so-called *EInstitutionService*, providing all methods for observing and acting with them. Thereafter, different interfaces to access the service can be incorporated into *AMELI* as implementations of the *ServiceProfile* interface. These service profiles can be regarded as different views to an environ-

ment. The motivation to consider different profiles is that an AEI may require that external agents have different views to the environment depending on their roles. An example of a market forecast service for the electricity market is described in the next section.

EIDE-* provides a software tool, *aBUILDER*, for agent development based on *ISLANDER* specifications. Specifically, *aBUILDER* takes an *ISLANDER* specification and produces for each role that may be played in the institution an "agent skeleton". Those skeletons comply with all the conventions of the specified institution, in particular with its dialogical framework and the performative structure. The previous version of *aBUILDER* presented in [1] has been extended to support the graphical specification of agent skeletons. Hence, staff agents may be easily built —on top of the *aBUILDER* skeletons— by concentrating the programming efforts on the decision policies and having the skeleton take care of navigation and communication within the AEI. Additionally, external agents may be modeled as parametric skeletons and used in the simulation environment to validate the institution goals.

Validating the desired behavior of an AEI is a highly intricate and computationally expensive task, as illustrated by [22, 21, 11, 8]. Such validation becomes even more complicated when we incorporate into the AEI an environment with a partially observable behavior. We have developed an extended version of *SIMDEI* (formerly introduced in [1]). *SIMDEI* allows to run discrete event simulations of *AMELI* along the lines of multi-agent simulations produced with the aid of libraries like Repast [16]. As to environment simulations, we must choose the modelling simulation tool (e.g. Simile [19], Simulink [20], EJS [4]) that best fits the domain features. Chosen a simulation tool, it is necessary to glue it with *AMELI* so that agents in an AEI can sense and act upon the simulated environment. This required *simulation bridge* (see the arrow connecting the simulation environment with *AMELI* in figure 1), is a software component whose main purpose is: (i) to synchronise both simulators; (ii) to forward environment variables' values to *SIMDEI*; and (iii) to translate actions within the simulated AEI into environment actions. At present, we do offer implementations of the simulation bridge to connect *SIMDEI* simulations to either Simulink [20] or EJS [4].

SIMDEI can exploit parametrised agent skeletons to generate agent populations by setting the number of agents to create from a given skeleton along with the means to set up values for their parameters. An agent's action can be parametrised in two ways: (i) by defining whether an action is carried out or not as a parameter; (ii) by defining (some of) the actual values of each action as parameters. Figure 3 illustrates how to generate a population of energy producers whose production capacity will be randomly generated by a normal distribution.

In summary, we have extended the original EIDE development framework providing a set of tools for engineering (specify and test) autonomic electronic institutions (EIDE-*). EIDE-* has been used for designing and testing the electricity market problem that is described below.

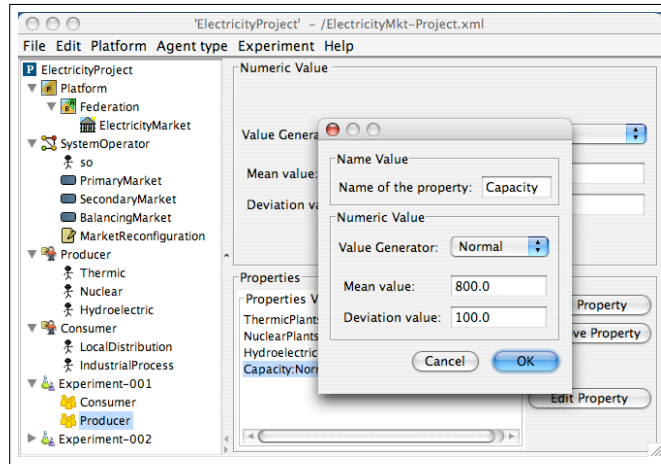


Fig. 3. Generating Agent Populations for the Electricity Market with SIMDEI.

4 Electricity Market

We will illustrate the capabilities of the framework through the *Power Electricity Market* problem. The main goal of an electricity market is to provide a set of rules to conciliate the demand of electricity and its generation. There are two issues that must be avoided: a lack of production that can leave some customers without electricity and an unwanted overproduction. Moreover, these goals have to be achieved while maintaining a reasonable electricity price.

We will model an electricity market as an electronic institution where the power demand is the environment where the institution is situated and the market is only able to partially observe the impact of their decisions in the environment.

4.1 Market Goals

As we mentioned above, the first goal of the electricity market AEI is to guarantee that the energy demand is always satisfied and that the overproduction is minimized. Because each producer is obliged to guarantee a safety power that is a 10% of its production, we are interested in minimizing the amount of safety power required.

The second goal of the electricity market AEI is to keep the power cost in a reasonable interval. For instance, the power cost in a working winter day oscillates from a minimum of 30 Euros/MWh to a maximum of 70 Euros/MWh.

Given these goals, we defined four reference values in the AEI: the power deficit percentage (PDP); the overproduction percentage (OPP); the power cost average (AvgC); and the power cost deviation (DevC).

Because we are interested in experimenting with different scenarios, the constraints associated to the reference values (the maximum and minimum ranges) will be parameters to be filled when enacting specific institutions.

4.2 Market Players

The players of the market are the producers, the consumers, and the system operator. Producers and consumers are external roles in the institution whereas the system operator is a staff role.

Producers: The producers use different technologies for electricity generation in order to satisfy the demand. The three main types of power stations modeled are: Thermic (coal-fired, gas fired and fuel-fired) stations, Nuclear stations, and Hydroelectric stations. Each type of power station has its own production features. For instance, nuclear and hydroelectric are cheap and come on stream rapidly. However, if nuclear plants are backed-off significantly, recovery time is slow. Thermic-based generation is relatively expensive and slow to come on stream.

Consumers: The consumers that participate in an electricity market are large industrial companies and local energy wholesalers that sell the energy to smaller or domestic consumers. The main goal of the consumers is to buy energy for half an hour periods according to the information provided by the demand model.

System Operator: The task of the system operator is to guarantee the voltage level and the dynamic security of the electricity network. Specifically, the system operator controls that the power deficit is never greater than a 10% of the total production, which is the obliged safety power that each power station must fulfill. Notice that, in our example, producers are autonomous about deciding their own production and the system operator is only responsible for the distribution of the demand.

4.3 Market Activities

The electricity market is organized in three different markets: the primary market, the secondary market, and the balancing market.

Primary Market: The primary market performs periodic auctions of transmission rights, in the form of tickets valid for the injection or extraction of energy over the next half an hour period. We have modeled the primary market with a double auction protocol. Every half an hour a new auction is launched.

Secondary Market: Once the auction has taken place, the goal of the secondary market is to provide an additional round for the trading of transmission tickets. The market allows the trading of a ticket until half an hour before the ticket time. This time is known as “gate closure”.

Balancing Market: This market exists to permit the system operator to adapt the plans of production to the quality and security restrictions. Based on the analysis of the tickets held in the previous markets, the system operator is able to identify shortfalls or excesses of energy that will arise during the ticket window.

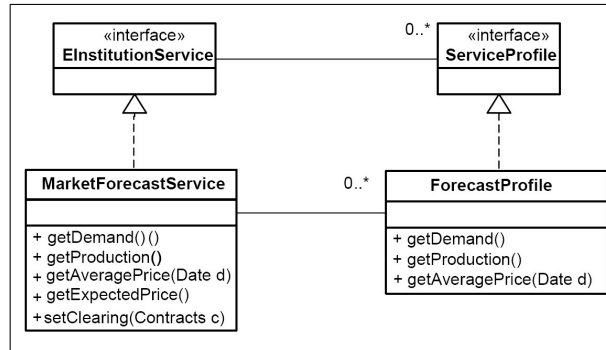


Fig. 4. Plugging a Forecast Service to the market.

The only actions available are: the dispatching of additional generation and the back-off of scheduled generation.

4.4 Simulation Environment

The power demand has been modeled following the electrical consumption in Spain every hour. The information has been taken from the “Red Eléctrica Española” [15] which controls the electrical power distribution in Spain. The power demand has been simulated using the EJS tool [4]. We have modeled four different consumption patterns: working days, Saturday, Sunday, and holidays. Moreover, some perturbations can be introduced arbitrarily into the simulated patterns.

We have developed the `MarketForecast` service that offers the forecast methods—namely expected demand (`getDemand`); expected energy production (`getProduction`); and expected MWh price (`getExpectedPrice`)—as well as a method to retrieve past market price on a particular date `getPrice(Date d)`. Furthermore, it provides a method for acting into forecast calculi: the method `setClearing(Contracts c)` sets the contract information corresponding to a market cleared by the system operator. The `setClearing` method is employed by affecting the demand simulation and, consequently, the subsequent forecasts. The idea behind this method is to disturb the estimation of the next expected price by means of analyzing the production and consumption mismatches.

The `ForecastProfile` profile only allows external agents to obtain information about past market prices on particular dates, and the expected energy demand and production. The `ForecastProfile` has been further split so that only consumers can access the production forecast, whereas only producers can access the demand forecast. Figure 4 summarizes the `MarketForecast` service.

4.5 Self-Configuration Policies

The system operator is the agent in charge of tracking the fulfillment of the institutional goals and the one responsible for re-configuring them when necessary.

The interest of the institution is the market autonomy, i.e. that producers and consumers would reach all the required agreements in the primary and secondary markets with the minimum mismatch between offer and demand. The intervention of the system operator in the balancing market has to be minimized and the task of the system operator is to dynamically adapt the institutional rules for enforcing this result.

After each execution round in the balancing market, the institutional goals are automatically updated by *AMELI*. First at all, the result of a balancing market round fires the updating of the reference values: the power deficit percentage (PDP); the overproduction percentage (OPP); the power cost average (AvgC); and the power cost deviation (DevC). Then, the fulfillment of the goals is updated by checking the constraints related to each goal, i.e. by contrasting position of the reference values into the desired intervals.

The most important goal of the institution is to minimize the amount of reserve power consumed (PDP). Because the guaranteed reserve power is only a 10% of the production, the priority of the system operator must be to avoid the usage of this reserve. The system operator uses the **MarketForecast** service for assessing whether a usage of the reserve power is the product of a punctual demand peak (the power demand usually has two maximum peaks per day) or reflects a problem between offer and demand. Only this second phenomenon is considered as an indicator to re-configure the institutional rules. We assume that producers and consumers follow a rational behavior. Producers are interested in offering all the energy they are able to produce when demand peaks arise because the price in those situations is usually high. On the counter part, consumers are aware that they have to pay an extra price when the global demand is high. Thus, the main reason of this market mismatch is the partial awareness that each consumer or producer has about the global market behavior. The scope for action of the system operator focuses the secondary and balancing markets. The system operator may change the role flow policies for enforcing the participation of producers in the secondary market (left side transitions in figure ??) and for re-configuring the protocol parameters in the secondary market providing more flexibility to the consumers.

The overproduction is preferable to the lack of production but also has to be minimized. Assuming again a rational behavior in producers and consumers, the system operator will change the role flow policies for inhibiting the participation of producers in the secondary market. Furthermore, the system operator may change the window of the demand forecast the producers are able to access, i.e re-configuring the **ForecastProfile** for helping the producers in the planning of their optimal production.

Finally, maintaining the energy cost in a reasonable interval should be a natural consequence of any balanced market. Because of the openness of participants this hypothesis cannot be assumed and the system operator has to

prevent also unexpected low/high prices. The way a system operator may enforce reasonable prices is by modifying the normative rules of the institution by increasing/decreasing punishments.

5 Conclusions

In this paper we have tried to make headway in the engineering of self-* systems by introducing a framework, EIDE-*, to support the development of a particular type of these systems, namely *autonomic electronic institutions* (AEIs). We have introduced the formal concepts around autonomic electronic institutions and described the set of tools we provide for helping in the engineering of autonomic electronic institutions. Furthermore, we have illustrated the capabilities of the framework through the analysis of self-configuration policies in a power electricity market.

As future work, we plan to deal with the reasoning capabilities required by a participating agent in order to cope with institutional changes.

Acknowledgements

This work was partially funded by projects AT (CONSOLIDER CSD2007-0022), IEA (TIN2006-15662-C02-01), OK (IST-4-027253-STP), EU-FEDER funds, and by the Generalitat de Catalunya under the grant 2005-SGR-00093.

and 2006 5 OI 099. The authors would like to thank to the UDT-IA development team for their help in the deployment of the EIDE-* framework.

References

1. Josep Lluís Arcos, Marc Esteva, Pablo Noriega, Juan A. Rodríguez-Aguilar, and Carles Sierra. Engineering open environments with electronic institutions. *Engineering Applications of Artificial Intelligence*, 18(1):191–204, January 2005.
2. Josep Lluís Arcos, Pablo Noriega, Juan A. Rodríguez-Aguilar, and Carles Sierra. E4mas through electronic institutions. In D. Weyns, H.V.D. Parunak, and F. Michel, editors, *Environments for Multiagent Systems III*, volume 4389 of *Lecture Notes in Artificial Intelligence*, pages 184–202. Springer-Verlag, 2007.
3. Eva Bou, Maite López-Sánchez, and Juan A. Rodríguez-Aguilar. Towards self-configuration in autonomic electronic institutions. In *Coordination, Organizations, Institutions and Norms in Multi-Agent systems II*, volume 4386 of *Lecture Notes in Artificial Intelligence*, 2007.
4. Ejs, easy java simulations. <http://www.um.es/fem/Ejs>.
5. M. Esteva. *Electronic Institutions: from specification to development*. PhD Thesis Universitat Politècnica de Catalunya (UPC), 2003. Number 19 in IIIA Monograph Series. IIIA, 2003.
6. Marc Esteva, David de la Cruz, and Carles Sierra. ISLANDER: en electronic institutions editor. In W. Lewis Johnson Cristiano Castelfranchi, editor, *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems, (july 15-19, 2002, Bologna, Italy)*, volume 3, pages 1045–1052. ACM PRESS, 2002.

7. Marc Esteva, Bruno Rosell, Juan A. Rodríguez-Aguilar, and Josep Lluís Arcos. AMELI: An agent-based middleware for electronic institutions. In N. et al. Jennings, editor, *Third international joint conference on autonomous agents and multi-agent systems (AAMAS 2004)*, volume I, pages 236–243, New York, USA, July 19–23 2004. ACM.
8. Marc Esteva, Wamberto Vasconcelos, Carles Sierra, and Juan Antonio Rodríguez-Aguilar. Norm consistency in electronic institutions. In *Proceedings of the XVII Brazilian Symposium on Artificial Intelligence (SBIA'04)*, number 3171 in Lecture Notes in Artificial Intelligence, pages 494–505. 2004.
9. Nicholas R. Jennings, Katia Sycara, and Michael Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-agent Systems*, 1:275–306, 1998.
10. J. O. Kephart and D. M. Chess. The vision of autonomic computing. *IEEE Computer*, 36(1):41–50, 2003.
11. Ismail Khalil-Ibrahim, Gabriele Kotsis, and Reinhard Kronsteiner. Substitution rules for the verification of norm-compliance in electronic institutions. In *Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE04)*, pages 21–26. IEEE Computer Society, 2004.
12. Michael Luck, Peter McBurney, Onn Shehory, and Steve Willmott. *Agentlink Roadmap*. Agenlink.org, 2005.
13. Pablo Noriega. *Agent-Mediated Auctions: The Fishmarket Metaphor*. Number 8 in IIIA Monograph Series. 1997.
14. Douglass C. North. *Institutions, Institutional change and economic performance*. Cambridge University press, 40 west 20th Street, New York, NY 10011-4211, USA, 1990.
15. Red Eléctrica Española. <http://www.ree.es>.
16. Repast. <http://repast.sourceforge.net>.
17. Giovanni Rimassa, Dominic Greenwood, and Martin E. Kernland. The Living Systems Technology Suite: an autonomous middleware for autonomic computing. In W. Lewis Johnson Cristiano Castelfranchi, editor, *Proceedings of the International Conference on Autonomic and Autonomous Systems, (ICAS'06)*, volume 3, page 33. ACM PRESS, 2006.
18. J. A. Rodríguez-Aguilar. *On the Design and Construction of Agent-mediated Electronic Institutions*. Number 14 in IIIA Monograph Series. 2003.
19. Simile. <http://simulistics.com>.
20. Simulink. <http://www.mathworks.com/products/simulink/>.
21. Wamberto Vasconcelos. Norm verification and analysis of electronic institutions. In Joao Leite, Andrea Omicini, Paolo Torroni, and Pinar Yolum, editors, *Declarative Agent Languages and Technologies II: Second International Workshop, DALT*, volume 3476 of *Lecture Notes in Computer Science*, pages 166–182. Springer-Verlag, 2005.
22. F. Viganò. A framework for model checking institutions. In *Proceedings of the ECAI Workshop on Model checking and Artificial Intelligence (MOCHART IV)*, 2006.