

E4MAS Through Electronic Institutions

Josep Lluís Arcos, Pablo Noriega, Juan A. Rodríguez-Aguilar,
and Carles Sierra

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

Abstract. Today, the concept of an environment for multi-agent systems is in its pioneering phase. Consequently, the development of supporting software technologies is still rather primitive and environment technologies reflecting a specific world-of-interest to the agent systems are yet to be developed in full. In contrast, environment technologies that focus on the agent system itself have been in the agenda of MAS research from its very start. Electronic institutions are prominent in this respect for they have been conceived as a type of restricted MAS environment and have had an engineering technology developed around them. In this paper we explore how the restrictions currently imposed by electronic institutions may be overcome when they are seen as a part of a larger environment where agents act. In particular, we focus on *situating* electronic institutions by connecting them to a world-of-interest and how this process can facilitate full-fledged environment engineering.

1 Introduction

It has become increasingly clear that the applicability of agent technologies requires not only appropriate software agents but also taking into account the environment where those agents interact. In fact, in many cases the design and implementation of the environment is the crucial aspect of an application. The motivation for such focus is readily seen when the purpose of the MAS is to specify the conventions that structure or organize the interactions of participants—as when defining an electronic marketplace or what economists do with mechanism design—when one intends to use MAS technologies either to model social phenomena—as, for example, traffic behaviour—or when testing or experimenting with the uses just mentioned.

In this paper we are concerned with a particular type of environment, electronic institutions (e-Institutions), that can be used for all these purposes. We claim that e-Institutions are well-suited for applications involving “open” multi agent systems and we have three main developments around the idea of e-Institutions that allow us to sustain that claim. First a conceptual model that makes explicit the type of multi-agent systems that may be implemented as EI, second a language to specify arbitrary e-Institutions and, third, the tools

to implement and run e-Institutions specified with that language. This paper presents these three developments and explores how they may be put to work in order to design and use artificial environments where human or software agents may interact according to an explicit set of conventions. The paper, however, is not limited to discussing e-Institutions. It goes a step further and explores the possibility of operationalising the environment where the e-Institution exists and the relation between the institution and that world. Although we have discussed some empirical aspects of establishing the links between an e-Institution and the world (cf. [10]) and presented some details on how the linking was implemented in specific examples ([20,19]), in this paper we want to focus on two aspects that we have not discussed in print before: on the one hand we will explain how we can use environment technologies to “situate” an e-Institution within its “world-of-interest”, and on the other hand we will present the tools and means to build simulation environments for situated e-Institutions.

For these purposes the rest of the paper is structured as follows: First we present our three main developments on e-Institutions, namely, Sect. 2 discusses our conceptual model, Sect. 3 the way of specifying e-Institutions and, in Sect. 4, the way we make such e-Institutions operate. We then present, in Sect. 5, our ideas on situating e-Institutions in a world-of-interest, how those environments may be characterized and simulated (Sect. 6). We close the paper contrasting our proposal with some of the notions about environments discussed elsewhere in this volume.

2 A Conceptual Model for Electronic Institutions

Loosely speaking, e-Institutions are computational realizations of traditional institutions (cf. North [12] 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, as Fig. 1 illustrates.

The notion of e-Institution may be expressed more precisely by stating a conceptual model of e-Institutions that gives ground for their computational implementation. In order to make the conceptual model operational we hold the following assumptions:

1. *Agent neutrality.* Participating entities are agents, in the accepted sense of being persistent, identifiable, communication-capable humans or software programs. We do not assume anything about the rationality, capabilities or intentions but we do assume they are able to communicate with other agents.
2. *Dialogical Stance.* All interactions are construable as speech acts. We therefore assume that there is a shared language whose semantic and pragmatic content is somehow fixed by the institution and adopted by the participants.
3. *Agent-mediated commitment making.* We assume that when participants communicate with other agents they are able and entitled to establish and fulfill commitments, and eventually abide by their consequences.

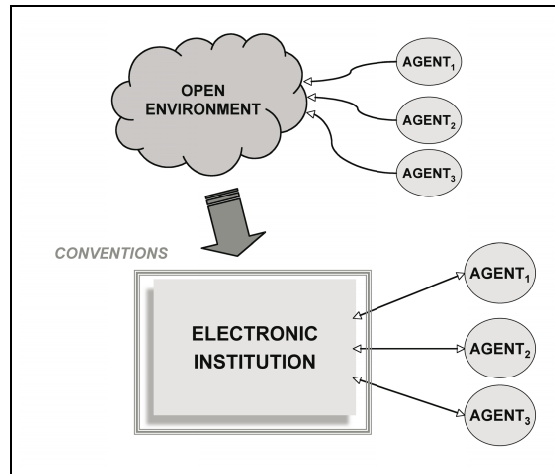


Fig. 1. e-Institutions create a virtual environment where interactions among agents in the real world correspond with illocutions exchanged by agents within the restricted environment. Legitimate illocutions —uttered within the institution— change the state of the institution and count as actions in the world.

4. *Institutional commitments.* Only illocutions uttered by participating agents have effect on the shared environment. The institution is the trustee of the intended conditions for illocution utterance and effects, hence of the commitments established through agent interactions within the institution.
5. *Repetitive Interactions.* We assume that it makes sense to institute interaction conventions when there are interactions that happen not once but many times following a regular pattern. Furthermore, we assume that such patterns of interaction apply not to specific individuals but to any agent that performs a given role during those interactions. We further assume that those repetitive interactions may be organized into some hierarchical system composed by sets of speech acts and relations among these sets.

These assumptions reflect our intuition about what institutions are and how we intend to implement them. Thus, because of assumption 1, we deal with a sort of open multi agent systems, i.e., those that are populated by independent, heterogeneous, self-interested agents whose ownership and accountability may be other than the environment's. As a matter of fact we want the e-Institutions to be an interface that separates the internal behaviour of agents from their external interactions.¹ Assumptions 2, 3 and 4 make our lives simple by reducing the world to what happens inside the e-Institution and restricting to structured conversations (speech acts and scene transitions) all that may happen there.

¹ Note that e-Institutions will be MAS that are open in as much as they admit agents of unknown origin and internals, but once these agents are inside the EI the behaviour of the agents is constrained —and to that extent— “closed” by the e-Institution.

Finally, assumption 5 is there to allow us to “structure” interactions into a network of regulated activities whose conventions apply to individuals performing a role. Although —as we shall see in Sect. 5— we may want to relax these assumptions in order to get a handle on some extra features, the conceptual model we are able to build from the five assumptions is practical and general enough to deal with a large class of MAS environments.

The model we propose ([9,21,5]) makes it possible to specify an e-Institution through the following components:

- A *dialogical* framework that defines ontology, social structure and language conventions.
- A *deontological component* that establishes the pragmatics of admissible illocutory actions. This is a set of norms that constrains possible illocutionary exchanges and manages the obligations established within the institution.

e-Institution is currently operationalized as EI_0 . In particular, its deontological component is specified with two constructs:

- A *performative structure* that includes a network of scenes linked by transitions between scenes. Scenes are role-based interaction protocols specified as finite state machines, arcs labelled by illocutions and nodes corresponding to an institutional state. Transitions describe the role-flow policies between scenes.
- *Rules of behavior* that establish role-based conventions regulating commitments. These are expressed as pre and post-conditions of the illocutions that are admissible in the performative structure of the *e-Institution*.

Thus a typical e-commerce application like public procurement or auctioning may be implemented as an e-Institution that incarnates the conventions that regulate the (verbal) exchanges between buyers and sellers, supervisors, banks, etc. In such type of applications, the overall activity; for example, contracting the construction of a school house can be thought of as a play that is organized as a *performative structure* by network of sub-activities or *scenes*: call for bids, selection of best offer, contract agreement, etc.

The dialogical framework makes explicit those elements that the institution “speaks about” (chairs, doors, checks, contract, roles (supervisor, auctioneer, buyer,...), time, etc.) their intended semantics and the other conventions needed to express those illocutions that will be legitimate in the institution. Each scene describes an interaction protocol that states what can be said by whom and under what circumstances, for instance that before a supplier is entitled to make a bid for building a school house, it has to prove that it is a certified construction company. The high-level layout of the performative structure indicates scenes and transitions between scenes. These transitions state the conditions that agents must fulfill in order to move from one scene to another. Those conditions may involve synchronizations, changing roles, unfolding the actions of the same agent inside more than one scene (i.e., spawning *alteroids*) and other analogous situations, hence the depiction of the performative structure may be a rather complex picture, as shown for example in Fig. 3.

From an agent's point of view, two major benefits stem from the use of this conceptual model of e-Institutions for modelling environments. On the one hand, *e-Institutions help reduce the frame problem for agents*. As noted above, e-Institutions establish conventions on behavior, language, and protocols that force agents to behave in particular and restrictive ways. In a sense, the environment is given structure, so that the agents have an easy comprehension of its working laws. Think for instance on how auctions or parliaments work: buyers or MPs know when they can talk, what consequences their acts will have, and what actions are possible at each moment in time. These restrictions facilitate the programming of agents, since by restricting the set of actions that agents have to consider at each moment in time one can address the frame problem by limiting the set of options that agents have to think about.

In the next two sections we will show how this conceptual model gives rise to actual e-Institutions through the tools that we have built to specify and generate e-Institutions, and a middleware that activates the corresponding runtime e-Institutions to be enacted by actual agents.

3 Specifying Electronic Institutions

To specify an EI we need to deal expressly with three components mentioned before: dialogical framework, scenes and performative structures. We have a tool, ISLANDER [6], that allows us to make a graphical specification of those components and produces an XML file with the specification. That specification may then be used to build the actual e-Institution or by agent designers to build agents that conform to the institutional conventions.

3.1 Dialogical Framework

The dialogical framework is specified by enumerating all the roles that intervene in the institution, all the constants that may appear in an illocution and all the illocutionary particles that may be used. The right hand side of Fig. 2 enumerates the illocutions used in a Vickrey auction. These involve two roles (`auctioneer` and `buyer`), some verbs (`startauction`, `offer`, ...), some variables (`good`, `price`, ...) a function (`all`), the illocutory particles `inform` and `request` and a "silence" particle (in expression 5). Illocutions, as shown in the same figure, may also involve variables that may or may not be bound to a given value; that is indicated by the "!" sign, meaning that the variable is bound to the last instance or "?", unbound.

In EI_0 dialogical frameworks capture only simple social structures by the specification of roles and the relationships among them, if any. These relationships may be involved in the description of scenes and transitions. For example a role "dancer" in a ball room institution may be specialized into "female" and "male" dancers and some scene transitions may allow any "dancer" to pass, while the receiving scene may require only one of the two special types in a given illocution. In EI_0 only static separation of roles is permitted, that is, an agent may change subsumed roles during a transition but not inside a scene.

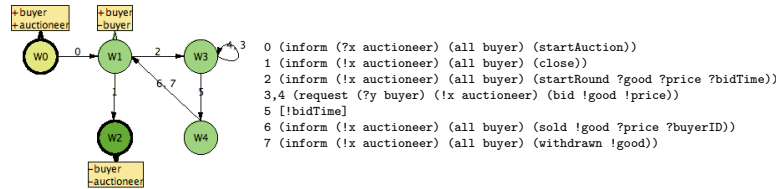


Fig. 2. An ISLANDER specification of a Vickrey bidding protocol. The left part shows the transition network whose arcs are labeled by the illocutory expressions on the right.

3.2 Scenes

Scenes are specified as transition networks with one initial state, one or more final states. Additionally, one or more entry states for each role and one or more exit states for each role. Illocutions connect states as a way of indicating that illocutions are the actions that happen inside the institution and that only what is said –if and when it is a legal illocution– changes the “state” of the institution as a whole.

For example, Fig. 2 depicts an ISLANDER specification of the Vickrey protocol for auctioning. In the left part of the figure, w_0 is the initial state and w_2 the end state. Boxes indicate that agents enacting a given role (**auctioneer** or **buyer**) may enter (+) or leave (-) certain states. Arcs are labeled by the illocutory expressions listed on the right part of the figure. Thus, for example, expression 4 states that an agent (y) acting as a buyer requests the agent (x) who acts as the auctioneer to accept a bid for the good being offered.

When specifying a scene in ISLANDER, each illocution may have preconditions and postconditions associated, so that the effects of illocutory actions — commitments— are properly governed by the institution and may be observed by participants. In practice, these conditions correspond to the rules of behaviour that individual agents are bound to obey. Scenes may also have some global conditions associated, like the minimal number of participants needed for the scene to be enacted or the average clearing price in a double auction.

In addition to well-formedness, ISLANDER takes care of some syntactic scene consistency checks thus preventing the most frequent miss-specifications such as incomplete dialogical frameworks, spurious roles, lack of access and exit states, etc.

3.3 Performative Structure

In its most abstract conception, the performative structure (PS) captures the conventions that regulate the flow of commitments in an institution. In more concrete terms, the PS describes the way agents may engage in different activities. Therefore the PS is made up by scenes and the transitions that connect those scenes. Recall that scenes already had a scene protocol AND incoming and outgoing states for the roles involved; well, scene transitions are used to control how an agent, performing a given role, may pass from a proper exit state in one scene to a proper entry state in another scene.

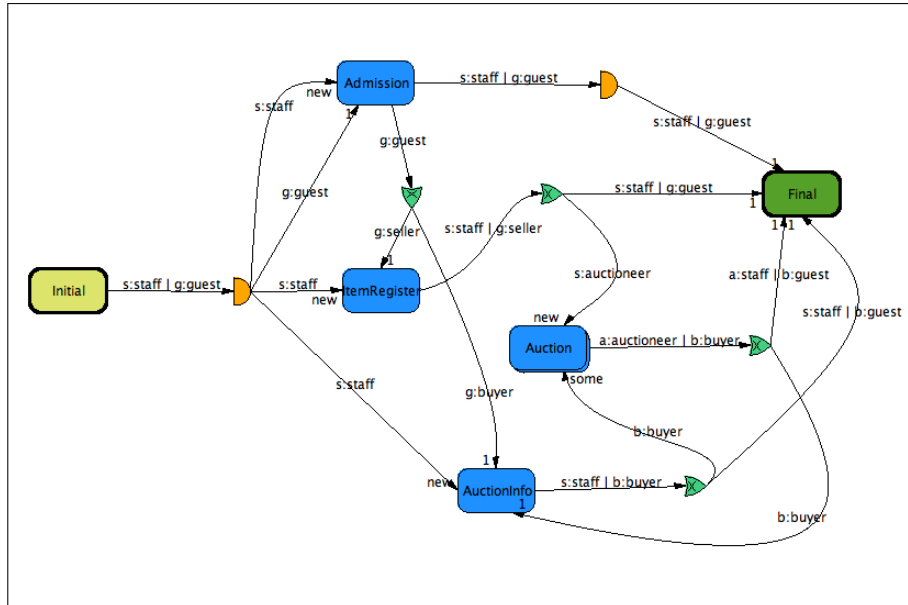


Fig. 3. An ISLANDER specification of the Performative Structure of an institution for simultaneous auctions. Boxes represent scenes and directed arcs inter-connect scenes through transition gates (of two types in this example: exclusive —crescents— and inclusive —half-circles— OR). Arcs are labeled by agent-variables and the roles these are to play. Moreover, arcs entering a scene are also labeled with a legend that indicate if transient agents may enter one scene, or one or more scenes of that type, or if a staff member may create new scenes of that type.

The management of transitions is rich in EI_0 because we want to be able to implement institutions where agents may be in two places at the same time (bidding in two auction houses that open the same hours), that institutional staff open scenes whenever needed, or that a given scene becomes closed once every participant has left. ISLANDER is able to express all these functionalities and the designer needs only point and click to enable them in a PS.

For example the performative structure depicted in Fig. 3 defines the relationships between the usual activities during the enactment of an auction. In this case —in addition to the initial and final scenes that are necessary in all e-Institutions— there are four basic scenes: `Admission`, `ItemRegister`, `Auctioninfo`, `Auction`, all depicted as boxes. Scenes are connected with directed arcs that come out of a scene, reach a transition (crescent-like figures) and then leave a transition into another scene. Staff members are present in all scenes, and therefore all scenes have in-arrows and out-arrows labeled with a `staff` or `auctioneer` (one kind of staff member). Likewise guests may either be buyers or sellers and although they enter the `Admission` scene as `guest`, they leave it either as `seller` (into the `ItemRegister` scene) or as `buyer` (into `auctioninfo`). Because this separation of roles is strict, the transition between

the **Admission** scene and the other two is an “exclusive” transition, while the transition between the **Admission** and **Final** scenes is a direct (inclusive or) transition through which all **staff** and **guest** agents may go through. Since, by design, in this institution there may be simultaneous auctions taking place in different rooms and the same agent may be present in more than one room, the performative structure shows a transition from the **ItemRegister** scene that allows a staff member to become an **auctioneer** and open **new Auction** scenes, and a **buyer** may leave the **Auctioninfo** scene and enter one or more **Auction** scenes or proceed directly to **Final**.

4 Running Electronic Institutions

How can we implement and run an ISLANDER specified EI?

Implementation can be seen as a three-stage process that includes reinterpreting the conceptual model so that it may become operational through agents and an actual, functioning, computational realization of the particular e-Institution specified using ISLANDER. For the deployment of the actual e-Institution and its activation we rely on the software we have developed and is describe below.

The implementation process is the following:

1. Translate our specification model into the corresponding execution model as follows:
 - e-Institutions are populated at run-time by heterogenous, self-interested agents.
 - Agents interact within scenes via speech acts.
 - Agents move from scene (activity) to scene (activity).
 - The execution of an institution can be regarded as the execution of its different scenes (activities).
2. Deploy two types of internal (or institutional). First, we will need staff agents that are intended to perform all the actions that institutional staff is entitled and enabled to perform; second, we will attach to each external agent an institutional “governor” that controls all the information flow between the (external) agent and the institution and is thus able to enforce the institutional conventions that the external agent is bound to observe in addition to insulating the institutional environment from potential misbehaviour of the external agent (see Fig. 4).
3. Insert an “institutional environment” between the agents and their (communicative) interactions to fulfill three essential functions: i) mediation, ii) coordination and enforcement, and iii) information management.

We have assembled an “Electronic Institution Development Environment”, EIDE [1] that includes all the software tools needed for deploying an ISLANDER-specified e-Institution.

The core of EIDE is an institutional engine, AMELI ([6]), that generates 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

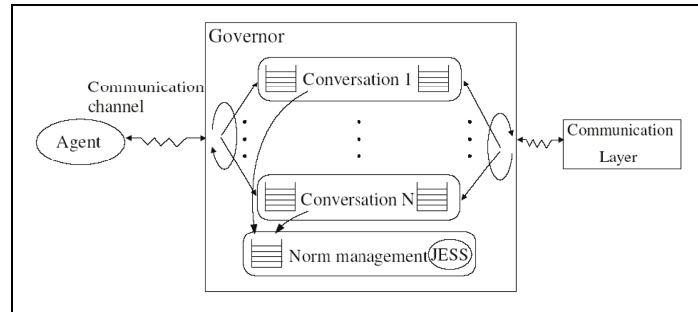


Fig. 4. Governors wrap around external agents and control information flows between these and the e-Institution. Governors keep track of the institutional state and update it when a valid illocution passes through them from an agent to the institution or back.

each scene to *warrant* legal movements between scenes and to *control* the obligations or commitments that participating agents acquire and fulfil and, finally, 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, handles all the institutional communication traffic by wrapping illocutions as messages that are handled by a standard agent-communication layer (e.g., JADE) as illustrated in Fig. 5.

Another EIDE tool, 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, and are compatible with the governors that are automatically produced at run time by the AMELI middleware. Hence, external agents may be built form scratch—based on the XML specification of the e-Institution— but they may also be readily built—on top of the aBuilder skeletons— by programming the decision means associated with illocutions and having the skeleton take care of navigation and communication within the e-Institution. The aBuilder tool is convenient for prototyping agents and (as will be seen in Sect. 6) to use agent skeletons to produce parametrized agents for modeling and testing institutions. In addition to AMELI and aBuilder, two more tools are part of EIDE: A simulation tool, SimDei, is used for animation and analysis of ISLANDER specifications and a Monitorig tool provides a graphical depiction of all the events that happen during the enactment of an e-Institution.

5 Situating Electronic Institutions

Today, the concept of an environment for multi-agent systems is in its pioneering phase. Consequently, the development of supporting software technologies

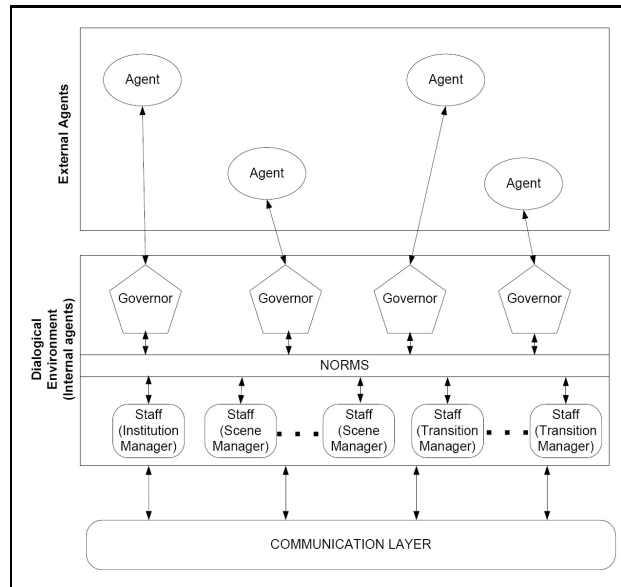


Fig. 5. An agent middleware for e-Institutions (AMELI) lays between participating agents and an agent communication infrastructure (e.g. JADE) composed of internal agents of two types: governors and staff agents

is in an early stage. Existing applications typically incorporate ad hoc implementations while analysis reveals how they may benefit from a more mature environment technology – e.g. from its time and resource management services. Environment technologies reflecting a specific world-of-interest (WoI) – e.g. a transport or manufacturing system – to the agent systems are yet to be developed in full [27]. In contrast, the environment technology that focuses on the agent system itself is being addressed by research already. e-Institutions are prominent in this respect [1].

Research into e-Institutions has been pioneering MAS environments for several years [11]. As shown in preceding sections, e-Institutions do not address the environment services and functionalities. Instead, as stated in Sect. 2, the research has focused on the norms and laws that apply to the agent society in a given *dialogical* environment. Thus, E-Institutions are a technology to enforce, monitor and encourage these norms and laws. Typically, they constrain the trajectories in the environment to the set of trajectories that are considered desirable, safe, acceptable, and/or manageable. Thus, the technology makes the agents in their agent society behave according to the needs and requirements of their environment.

And yet, is this enough? The answer is "not quite". Notice that, as pointed out above, MAS applications are usually concerned with some external WoI in addition to the agent society issues. The WoI is application-specific and refers to the part of the world that is relevant to the MAS application. For instance,

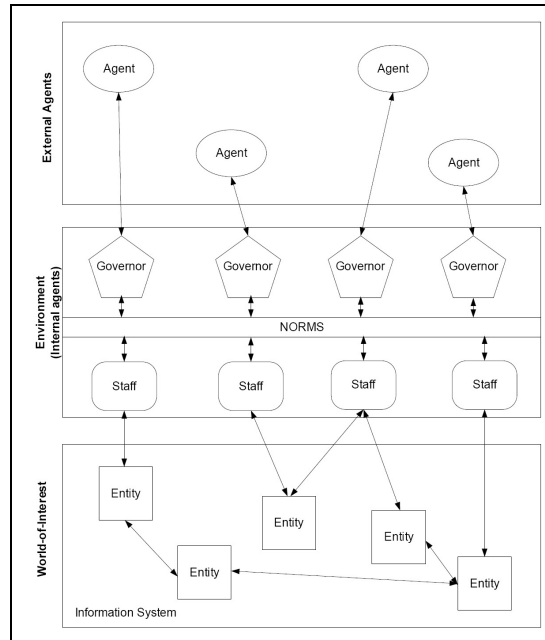


Fig. 6. Linking EIs with a world-of-interest

for a climate control application, the WoI comprises rooms, doors, heaters, etc. Therefore, it is necessary to extend the notion of e-Institution in order to link it with the notion of WoI. In this manner, participating agents will be able to *sense* and *act over* the WoI. Notice though that external agents cannot directly sense and act over the WoI. Instead, and likewise all interactions of external agents in the realm of an e-Institution, sensing and acting over the WoI is also *mediated* by the e-Institution wherein they interact. To summarise, what we propose is: (i) to directly *situate*² e-Institutions; and (ii) to indirectly (via mediation) situate external agents.

Figure 6 depicts how an e-Institution controls the interactions between the external agents and the WoI. The E-Institution is regarded as part of the environment and is still realised, as already shown in Fig. 5, by a collection of so-called staff agents. External agents only interact through the environment via dialogical actions that are filtered in or out by their governors. Thus, external agents can only sense and act over the entities in the WoI through their governors. Figure 7 details how to plug an entity into AMELI so that agents can subsequently sense/act over it. AMELI requires the implementation of a Java interface, the so-called `EInstitutionService`, per entity to incorporate into the platform all methods to operate on a given entity. Thereafter,

² We understand by *situatedness* the property of an AI program being located in an environment that it senses. Via its actions, the program can select its input, as well as change its environment.

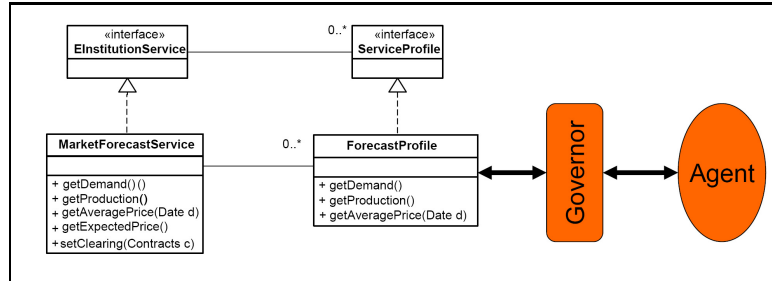


Fig. 7. Linking EIs with a world-of-interest

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 the service. In Fig. 7, we provide a service example based on the electricity market in [23]. The service `MarketForecast` offers several forecast methods—expected demand (`getDemand`), expected energy production (`getProduction`), expected Kw price (`getExpectedPrice`)—as well as a method to retrieve past market price on a particular date (`getPrice(Date d)`), set the contract information corresponding to a market cleared by the market operator (`setClearing(Contracts c)`) to be employed by subsequent forecasts. 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 motivation to consider different profiles is that the very same e-Institution may require that external agents have different views to the very same service depending on their roles. For instance, profile `ForecastProfile` can be further split so that only consumers can access the production forecast, whereas only producers can access the demand forecast.

The flexibility of the notion of e-Institution comes from its clear separation of concerns between the internal behavior of agents and their external interactions (environment modeling). The environment modeling outlined in Sect. 3 and 4 does change once the situatedness of e-Institutions is taken into account. Hence, from the perspective of an agent, its environment is modeled as the result of composing the following elements:

- *A number of agents* (usually called staff agents) that model/expand their human counterparts in the real world or that simply behave according to an internal model.
- *A number of norms* that restrict the behaviour of agents preventing them to behave in unacceptable/impossible ways. In this respect norms can be thought as physical laws or as social conventions that shape/constraint the evolution of interactions that may or may not take into account the WoI. For instance, in an electricity market the market operator may either oblige each power station to supply its spare production if the produced power is less or equal than the market demand, or it may prohibit energy producers to operate on the network when the operator detects thermal overloads.

- *An explicit agreement on language and ontology.* Since illocutionary acts are the only actions permitted within an e-Institution, it fits well with the MAS configurations where real entities are represented/expanded by an agent that, in the case of e-Institutions, will be using illocutions as its action repertoire. Notice that the very same language is used for sensing/acting over the WoI.
- *An explicit set of activities.* Scenes represent tasks solved by groups of agents and are nodes within the performative structure that models the flow of agents. Actions within scenes are further fixed as a protocol that will only permit certain dialogues among the agents. Agents are restricted (by social conventions or physical laws) in what they can do at a particular moment in time. Such restrictions may now take into account the WoI. Furthermore, an agent's actions within a scene may now have consequences as: (i) changes to an agent's institutional state (e.g. a consumer agent may have its credit diminished after winning an auction); (ii) changes to the institutional state (e.g. the average market price does change after a market clearing occurs); and (iii) actions over the WoI (e.g. the thermal load of an electricity network changes after an energy producer agent delivers its supply to the network).

On the other hand, the fact that external agents have governors as their unique means of sensing and acting over the environment makes agents *neutral* to the WoI. In other words, external agents are unaware of *how* entities in the WoI are sensed by the e-Institution wherein they take part. And thus, changes to the services connecting entities to AMELI have no impact whatsoever on the inner architecture of external agents.

6 Environment Simulation

At this point we are ready to engineer E4MAS based on the notion of situated e-Institution. Nonetheless, as environment engineers, we must wonder whether our MAS application is to behave as expected. Checking the properties of an e-Institution is a highly intricate and computationally expensive task, as illustrated by [29,28,8,7]. Such checking becomes even more complicated when adding a WoI composed not only of *static* entities (e.g. a database), but also of *dynamic* entities endowed with varying behaviours (e.g. a heater, a weather forecast service). Hence, it would be desirable for environment engineers to count on dynamic verification tools that help them analyse the dynamic behaviour of their MAS applications. At this aim, in what follows we detail the tools we have developed to simulate environments created as situated e-Institutions.

We regard an environment simulation as the result of coordinating an e-Institution simulation and a WoI simulation as illustrated by Fig. 8. As to simulating an e-Institution we employ 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 WoI simulations, we must choose the modelling simulation tool (e.g. Simile [24], Simulink [25], EJS [4]) that best fits the WoI features; for instance, the entity connected through the service depicted in Fig. 7 must be

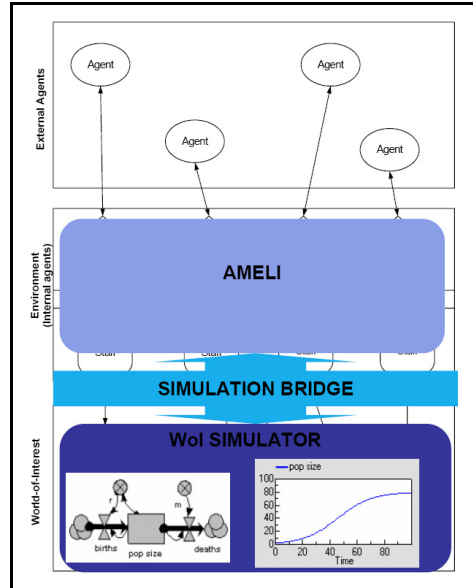


Fig. 8. Simulating situated EIs

modelled and simulated by some system dynamics simulator. Finally, it is necessary to glue the e-Institution simulation with the WoI simulation so that agents in an e-Institution can sense and act upon the simulated WoI. This motivates the introduction of the *simulation bridge* (see Fig. 8), a software component whose main purpose is: (i) to synchronise both simulators; (ii) to forward WoI variables' values to SimDei; and (iii) to translate actions within the simulated e-Institution into WoI actions. Notice that the implementation of the simulation bridge depends on the particular simulator we choose to simulate the WoI. At present, we do offer implementations of the simulation bridge to connect SimDei simulations to either Simulink [25] or EJS [4] simulations.

In order for the environment simulation to properly work, environment engineers are required to design simulations according to the simulation design workflow depicted in Fig. 9. Such workflow requires that an environment engineer performs the following tasks:

- [1] **Islander specification** of an e-Institution as explained in Sect. 3.
- [2] **WoI model** describing the dynamics of the entities in the WoI.
- [3] **Agent skeletons' specification** with the aid of aBuilder [1], the software tool for agent development included in EIDE that supports the graphical specification of agent skeletons based on Islander specifications.
- [4] **SimDei configuration**. It is composed of: (i) parameters to generate populations of agents based on agent skeletons; and (ii) the observation variables, namely the objects to probe in the simulation along with the functions to employ to combine their observed values.

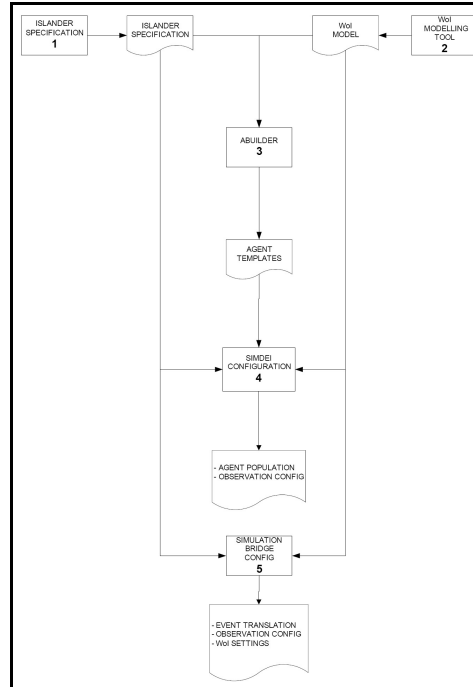


Fig. 9. The simulation design workflow

- [5] **Simulation bridge configuration.** It is composed of: (i) initial settings for the entities in the WoI; (ii) entities to probe in the WoI; and (iii) translation rules from institutional actions into WoI actions.

We believe that the generation of agent populations deserves special attention. The software tool *aBuilder* supports the specification of *parametrised* agent skeletons. Thus, 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. *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. Figure 10 illustrates how to generate a population of buyer agents from the *Buyer* skeleton for an electricity market. *SimDei* will randomly generate between 50 and 100 buyer agents that shall assess the values of their *price* and *kw* parameters using two different Normal distributions.

After the design stage, at run time, *SimDei*, the chosen WoI simulator, and the simulation bridge are concurrently launched. *SimDei* starts by generating agent populations using agent skeletons created with *aBuilder*. Thereafter, *SimDei* feeds AMELI with an Islander specification to run it in simulation mode. *SimDei* also employs AMELI’s monitoring tool to display the observed variables. The simulation bridge synchronises *SimDei* with the WoI simulator using its

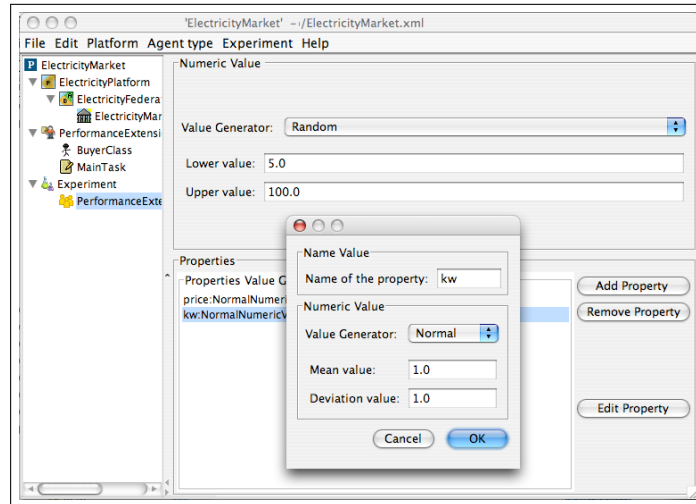


Fig. 10. Generating agent populations with SimDei

translation rules to reflect agents' actions over the WoI and observing the entities to probe in order to convey the values of variables to SimDei.

We argue that several benefits stem from our approach to environment simulation. Firstly, SimDei promotes multi-agent simulation from the programming level to the graphical specification level along the lines of graphical simulation tools (e.g. PowerSim [15], Arena [2], Simulink [25] or Simile [24]) unlike multi-agent simulation tools like Repast [16] or Swarm [26]. Secondly, we observe that it is unusual that multi-agent simulation libraries do offer organisational/institutional patterns that provide higher levels of abstraction to programmers, and if so (like [16]) they are quite limited. SimDei handles institutional patterns at the simulation level unlike state-of-the-art multi-agent simulation tools like [16,26]. Thirdly, a wide range of modelling and simulation tools can be employed together with SimDei whenever the appropriate simulation bridge is available. Notice that in order to plug a simulation bridge to an e-Institution simulation it must implement a generic API defined by SimDei. Lastly, notice that external agents within an e-Institution are neutral to the simulation tool employed for the WoI since all their sensing and acting is mediated by the e-Institution.

7 Discussion

A review of the other papers in this volume gives us ground to contrast some strengths and weaknesses of e-Institutions for engineering E4MAS.

First of all, we are confident that the approach of e-Institutions is appropriate for E4MAS whenever the intended MAS requires social structure and

regulation, as is the case for the approaches reported in [3,14]. In both instances e-Institutions would provide a set of alternative modeling constructs for an alternative design and implementation. For example, the notions of *space* and *mode* in [3] could be mapped into scenes and social roles. The notions of *rules* and *reaction rules* in [14] could also be readily captured in our framework. Namely, *reaction rules* regulating agent softbodies may be expressed as *norms* and as *postconditions* of agents' actions —within scenes— that change agents' institutional states, whereas *rules* may be translated into the specification of scenes and transitions.

Second, although e-Institutions do provide the means to enact a social structure, they do not include any means to structure the WoI. Thus, in particular, our framework offers no means of structuring the entities of the WoI according to a particular topology. Nevertheless, it is still possible to structure the WoI indirectly through an e-Institution: scenes provide locality by regulating the access to some entities in the WoI, while transitions between scenes model the change of locality—for example, in an auction house, the bidding scene *takes place* inside a bidding-hall and buyers *move* from there to a delivery hall (scene). In other words, by relating scenes with entities in the WoI we locally situate these. Therefore, a performative structure can be regarded as a way of building a *social* topology on top of a WoI.

Third, the approach in [22] based on the filtering of perceptions raises a very important issue concerning the regulation of perception in an e-Institution. So far, the focus of our framework has been to regulate agents' actions with the purpose of assessing whether they are institutionally valid or not. Thus, although e-Institutions offer a language to specify such regulations along with a software platform, AMELI, to implement them, we do not provide the same degree of functionality to regulate perceptions, yet.

Fourth, we would like to point out that the work on artifacts [13,17] is the approach closer to e-Institutions. And yet, there are two significant differences among these approaches: (i) e-Institutions are tailored to a particular—even if large— family of applications while artifacts are claimed to be more generic; (ii) e-Institutions are a well established and proven technology that includes a formal foundation, and advanced engineering and tool support. For artifacts, these features are still in a preliminary phase.

Finally, we are decidedly in favour of: (i) designing reference architectures that provide a blueprint for developing software architectures for MAS along the lines of [30]; and (ii) building development tools to engineer computational environments along the lines of CArtAgO [18] and EIDE [1]. We regard both activities as complementary because we are aware that although—as discussed in [27]— a development tool like EIDE can help develop a wide range of MAS configurations, at least for the time being it is hard to envision a general purpose development tool for computational environments.

It is our belief that the approaches to E4MAS reported in this volume outline several promising paths to future research, we would like to issue a call for further joint developments.

Acknowledgements

The authors would like to thank our anonymous reviewers for their very pertinent comments. This work has been supported by projects IEA (TIN2006-15662-C02-01), OK (IST-4-027253-STP), eREP(EC-FP6-CIT5-28575) and 2006 5 OI 099.

References

1. Josep Lluís Arcos, Marc Esteva, Pablo Noriega, Juan A. Rodríguez-Aguilar, and Carles Sierra. Environment engineering for multiagent systems. *Engineering Applications of Artificial Intelligence*, 18(1):191–204, January 2005.
2. Arena. <http://www.arenasimulation.com/>.
3. José-Antonio Báez-Barranco, Tiberiu Stratulat, and Jacques Ferber. A unified model for physical and social environments. In Weyns et al. [31]. To appear.
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, Juan A. Rodríguez-Aguilar, Bruno Rosell, and Josep L. Arcos. Ameli: An agent-based middleware for electronic institutions. In *Third International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS'04)*, New York, USA, July 19-23 2004.
7. 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.
8. 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.
9. P. Noriega. *Agent-Mediated Auctions: The Fishmarket Metaphor*. PhD Thesis Universitat Autònoma de Barcelona (UAB), 1997. Number 8 in IIIA Monograph Series. IIIA, 1999.
10. Pablo Noriega. Fencing the open fields, empirical concerns on electronic institutions (invited paper). In Olivier Boissier, Julian Padget, Virginia Dignum, Gabriela Lindemann, Eric Matson, Sascha Ossowski, Jaime Simao Sichman, and Javier Vázquez-Salceda, editors, *Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems AAMAS 2005 International Workshops on Agents, Norms and Institutions for Regulated Multi-Agent Systems, ANIREM 2005, and Organizations in Multi-Agent Systems, OOP 2005, Utrecht, The Netherlands, July 25-26, 2005, Revised Selected Papers*, volume 3913 of *Lecture notes in computer science*, pages 81–98. Springer, 2006.
11. Pablo Noriega and Carles Sierra. Electronic institutions: Future trends and challenges. In Matthias Klusch, Sascha Ossowski, and Onn Shehory, editors, *Proceedings of the 6th International Workshop on Cooperative Information Agents VI*, volume 2446 of *Lecture Notes in Computer Science*, pages 14–17. Springer Verlag, 2002.
12. Douglass C. North. *Institutions, Institutional change and economic performance*. Cambridge University press, 40 west 20th Street, New York, NY 10011-4211, USA, 1990.

13. Andrea Omicini, Alessandro Ricci, Mirko Viroli, Cristiano Castelfranchi, and Luca Tummolini. Coordination artifacts: Environment-based coordination for intelligent agents. In *AAMAS*, pages 286–293. IEEE Computer Society, 2004.
14. Eric Platon, Nicolas Sabouret, and Shinichi Honiden. Tag interactions in multi-agent systems: Environment support. In Weyns et al. [31]. To appear.
15. Powersim. <http://www.powersim.com>.
16. Repast. <http://repast.sourceforge.net>.
17. Alessandro Ricci, Andrea Omicini, Mirko Viroli, Luca Gardelli, and Enrico Oliva. Cognitive stigmergy: A framework based on agents and artifacts. In Weyns et al. [31]. To appear.
18. Alessandro Ricci, Mirko Viroli, and Andrea Omicini. Cartago: An infrastructure for engineering computational environments in MAS. In Weyns et al. [31]. To appear.
19. Armando Robles, Pablo Noriega, Michael Luck, and Francisco Cantú. Using mas technologies for intelligent organizations: a report of bottom-up results. In Carlos Alberto Gelbukh, Alexander ; Reyes-Garcia, editor, *MICAI 2006: Advances in Artificial Intelligence. 5th Mexican International Conference on Artificial Intelligence, Apizaco, Mexico, November 13-17, 2006. Proceedings*, volume 4293 of *Lecture notes in computer science. Lecture notes on Artificial Intelligence*, pages 1116–1127. Springer, 2006.
20. Armando Robles, Pablo Noriega, Marco Julio Robles, Héctor Hernández, Víctor Soto, and Edgar González. A hotel information system implementation using mas technology. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS06 May 8-12 2006, Hakodate, Japan)*, pages 1542–1548. ACM, 2006.
21. J. A. Rodríguez-Aguilar. *On the Design and Construction of Agent-mediated Electronic Institutions, PhD Thesis, Universitat Autònoma de Barcelona (2001), 2001*. Number 14 in IIIA Monograph Series. IIIA, 2003.
22. Julien Saunier, Flavien Balbo, and Fabien Bodeig. Environment as active support of interaction. In Weyns et al. [31]. To appear.
23. Carles Sierra, Jordi Sabater, Jaume Agustí, and Pere Garcia. *Methodologies and Software Engineering for Agent Systems*, chapter The Sadde Methodology, pages 195–214. Kluwer Academic Press, 2004.
24. Simile. <http://simulistics.com>.
25. Simulink. <http://www.mathworks.com/products/simulink/>.
26. Swarm. <http://www.swarm.org>.
27. Paul Valckenaers, John Sauters, Carles Sierra, and Juan Antonio Rodríguez-Aguilar. Applications and environments for multi-agent systems. *Autonomous Agents and Multi-agent Systems*, 14(1):61–85, February 2007.
28. 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.
29. F. Viganò. A framework for model checking institutions. In *Proceedings of the ECAI Workshop on Model checking and Artificial Intelligence (MOCHART IV)*, 2006. To appear. Available from: www.istituti.usilu.net/viganof.
30. Danny Weyns and Tom Holvoet. A reference architecture for situated multiagent systems. In Weyns et al. [31]. To appear.
31. Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors. *Environments for Multi-Agent Systems III*. Springer Verlag, 2007. To appear.