Towards Hybrid Experiments on reputation mechanisms: BDI Agents and Humans in Electronic Institutions

Jordi Sabater-Mir, Isaac Pinyol, Daniel Villatoro, and Guifré Cuní

IIIA - Artificial Intelligence Research Institute
CSIC - Spanish Scientific Research Council
Bellaterra, Barcelona, Spain

Abstract. In this paper we present a technological framework that allows virtual cognitive agents and humans using a web interface to participate in the same electronic institution (eI). Given that the main objective of this framework is to perform experiments on the use of reputation, we also introduce the notion of reputation within the eI to incorporate the circulation of opinions about other agents towards certain actions, and we extend the actual eI infrastructure to permit BDI agents to participate in an eI.

1 Introduction

Disciplines like sociology, psychology, anthropology, economy etc... have based part of their research on observing, monitoring and analyzing individual's actions in pre-designed and controlled scenarios. Some of them require some interaction between the participants, for instance, in experiments or games where there is a competition for the same products. Nowadays, a lot of these experiments are designed using computerized models offering the participants nice interfaces to play with. In these kind of experiments humans sometimes are substituted by artificial agents, both to simplify the complexity associated with human experiments with a big number of participants and also to study the reaction of humans in front of these autonomous artificial entities. In these hybrid experiments, humans and autonomous agents are put together in the same environment to interact.

In the context of the eRep project [1] we have to perform these kind of experiments to study the use of reputation in e-commerce environments. In order to perform these experiments it is necessary a theoretical and technological framework that can support both the execution of the experiments and the gathering of the results for a subsequent analysis. Given the nature of the problems for which the use of reputation is more useful, a framework that seems to fit with our needs is that of the electronic institutions.

However, the set of tools already provided to specify, develop and run electronic institutions do not take into account the use of reputation systems as an element fully integrated into the electronic institution or into the participating agents. Moreover, there is no way to incorporate humans to participate remotely either alone or together with autonomous agents in the electronic institution. Given that, we have extended the current technological framework to incorporate all these elements.

This work then has two parts. First, it presents the background technology behind electronic institutions (section 2) and, second, presents the extensions to this technology in order to facilitate the use of reputation mechanisms and the participation of humans. These extensions, that are explained and justified in section 3, can be summarized as follows:

– Ways to integrate reputation mechanisms in electronic institutions (section 3.1). How to integrate centralized reputation mechanisms as services of an electronic institution as well as how to integrate decentralized systems.
– Integration of a cognitive agent architecture in the context of an electronic institution (section 4)
– Set of tools to allow humans to participate remotely into an electronic institution with the same facilities that virtual agents have.(section 5)

2 E-Institutions

The concept of *electronic institution* ([2],[3],[4]) is inspired in human institutions. In open multi-agent systems you have also autonomous entities that interact to achieve individual goals. The behaviour of these entities cannot be guaranteed. Therefore, and similarly to what happens in human societies, you need mechanisms to guarantee the good functioning of the system in spite of the local behaviours. The use of an electronic institution that regulates the behaviour of agents the same way human institutions regulate the behaviour of people is one of this mechanisms that can be complemented by other mechanisms like, for example, the use of reputation. To fully understand the electronic institution machinery we refer the reader to [5].

2.1 e-Institutions Development Tools

IDE-eli [1], the Integrated Development Environment for Electronic Institutions, is a set of tools developped at the IIIA-CSIC aimed at supporting the engineering of multiagent systems as electronic institutions. These tools are the base of our work presented in this paper. Software agents appear as the key enabling technology behind the electronic institution vision. Thus, electronic institutions encapsulate the coordination mechanisms that mediate the interactions amongst software agents representing different parties. IDE-eli is composed of the following tools:

– **ISLANDER[6]:** A graphical tool that supports the specification of the rules and protocols in an electronic institution.
– **AMELI[7]:** A software platform to run electronic institutions; electronic institutions specified with ISLANDER are run by AMELI.

[1] All the software presented in this section and further documentation can be found at http://e-institutions.iiia.csic.es/

– **aBUILDER:** An agent development tool.
– **SIMDEI:** A simulation tool to animate and analyze electronic institutions.

3 Using reputation within electronic institutions

In this section we will present the elements for the use of reputation and why we think they are needed.

3.1 Integration of centralized and non centralized models into an electronic institution

We have two different ways to integrate reputation mechanisms in an electronic institution depending on the type of reputation model. For centralized reputation models, the reputation mechanism has to be a service provided by the eI. For decentralized models the reputation mechanism is part of the agent.

Incorporating centralized reputation models: A centralized reputation model (like for example the model used in e-markets like eBay) needs a considerable amount of information to be reliable. Therefore, the most sensible thing is that they could be denoted as a service of an eI. A service in an eI is a facility that the eI makes accessible to the agents in that eI. A centralized reputation system can be incorporated as a service that collects the different experiences and opinions of the agents and that provides reputation values under request.

Having the reputation system as a service gives us also another important advantage. The service, as shown in Figure 1, provides an interface (EInstitutionService) to access to its functionality. However it is possible to provide different profiles (EInstitutionProfile) that are linked to the role the agent is playing in the eI.

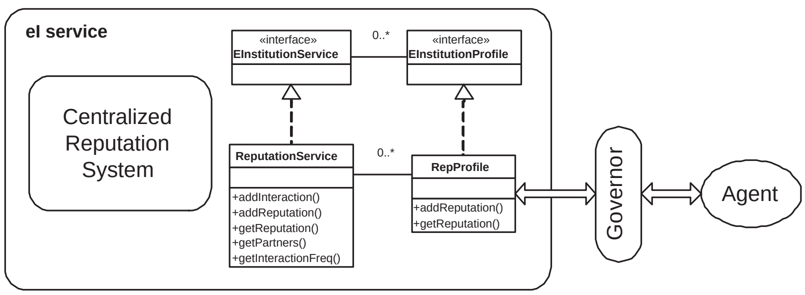

Fig. 1. Centralized reputation system as an eI service

Incorporating decentralized reputation models: In this case, there is no centralized reputation management. Each agent has its own reputation system that is fed by its own experiences and information received from third party agents. At this level, the aBUILDER (sec 2) has to provide facilities to build agents that not only are able to participate in the eI but also to use a reputation model.

3.2 Agent architecture

Although it is possible with the aBUILDER to build agents that are ready to participate in an eI designed using ISLANDER, the architecture of these agents is somehow limited to exploit the capabilities of the most complex reputation systems like for instance RepAge [8]. Therefore we want to include, as an alternative, the possibility to use a more deliberative architecture for the agent. In our case we have decided to use a BDI (Beliefs-Desires-Intentions) architecture because it is one of the most frequently used deliberative architectures in the area of MAS with several implementations available. In section 4 we explain the current architecture and the integration of a BDI architecture in an agent that participates in an eI.

3.3 A common ontology to talk about reputation

There is a great diversity of reputation models around, each one using their own concepts, terminology and ways to represent evaluations. This situation makes an hypothetical transmission of social evaluations between agents that are using different reputation systems impossible. It is necessary a common ontology as well as an ontology mapping mechanism that can be used for these dialogs. We have used the work described in [9] to solve this problem. In our case, virtual agents using some reputation model will reason over the concepts defined in this ontology. Due to space limitations we cannot give a more extensive explanation. However, more details can be found at [9].

3.4 Allowing humans to interact with electronic institutions

The experiments planned in the eRep project are designed to have autonomous agents but also human agents. Therefore we need a mechanism to allow humans to participate in an eI with the same capabilities their artificial partners have. Both because we cannot guarantee that all the human participants can be physically at the same place and because we plan to perform experiments in Internet, the best option seems to be a web based access to the eI. Using the latest Internet technologies we have designed a mechanism that allows humans to participate in an eI remotely. This mechanism is explained in section 5.

4 The agents architecture

As we said, the aBUILDER provides an easy mechanism to build agents that are ready to evolve in an eI designed using the ISLANDER tool. Starting from an eI specification, the aBUILDER generates a skeleton that contains the code that the agent needs to the eI. The engineer only has to add the decision making procedures to make the agent fully functional in that eI. The agents generated using this procedure are based on an architecture called EIAgent. This architecture, although enough for a lot of agent types, is a low level architecture that is cumbersome to use together with cognitive models like for example RepAge [8]. For these kind of models it is easier to program the agent using a more deliberative architecture. Among the different deliberative architectures, we have decided to use a BDI (*Belief-Desire-Intention*) architecture because it has a strong theoretical background and it is the most extended in the MAS community when you design cognitive agents.

4.1 EIAgent: The agent architecture for electronic institutions

EIAgent is an architecture that allows *java* agents to participate in an eI. As explained in the Section 2, an electronic institution provides to the agents, through a performative structure, a sorted and normative *way* to act and communicate to each other. This strict environment requires a control mechanism. This is where the *governor* plays a crucial role. The *governor* is the link between a particular agent and the whole electronic institution. It is the individual view the agent has of the eI in which it is participating. There is one single *governor* for each agent. Each *governor* receives all the actions that its agent wants to perform, and checks whether they are following the correspondent rules and norms before notifying them to the eI.

An agent with an EIAgent architecture interacts with the *governor* through an API[2] that offers a set of asynchronous functions and procedures to interact with the eI. In the same way, the *governor* communicates messages and notifications from the eI and trough an API that acts as a bridge with the agent. This asynchronism provokes that both, agent and *governor*, be continuously waiting for messages.

The EIAgent architecture (see Figure 2) is based on a *Task Planner* that gathers messages coming from the eI and executes the programmed task that usually will end with a message that needs to be sent. Notice that messages may arrive at any time. From a technological point of view, due to this asynchronism, each new message will generate a new running thread that will end up running the associated task. This characteristic allows agents to participate at the same time in different scenes of the electronic institution, a capability that have always been a prerequisite in eI.

[2] Standard acronym for *Application Program Interface*

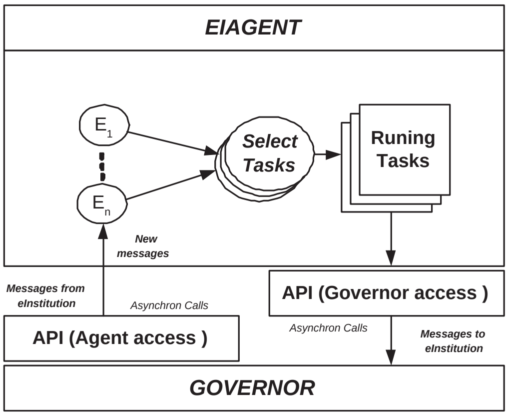

Fig. 2. EIAgent's abstract architecture

4.2 Cognitive agents: The BDI agent architecture

A BDI agent architecture is based on what is known as *practical reasoning*, the process of deciding at every step which action to perform in order to achieve the agent goals. A complete theoretical description can be found at [10]

It is clear that a possible extension of the *EIAgent* architecture to allow a BDI approach would require an implementation of a BDI engine. Nevertheless, there is already the well-known JADEX [11] platform that offers us all the flexibility we need to design our cognitive agents, based on BDI, and that support *java* agents. This BDI reasoning engine follows most of the principles of the simplified BDI theory, but includes the explicit representation of goals. It also provides a mechanism (the authors call them *capabilities*) that allows, among other things, the connectivity with external systems.

As we stated, we decided to use JADEX for our cognitive agents to allow them to participate in an electronic institution. For this interoperability we need an extension of the actual eI architecture. In the next subsection we present this extension and how we finally connect all these pieces to allow BDI agents to participate in eI.

4.3 Connecting JADEX agents to electronic institutions

As we said, the connection of an agent with the eI is done through the *governor*. This connection is using an API that acts as a bridge in both directions. When the *governor* needs to notify something to the agent, generates an asynchronous call to this API and vice-versa. For example, a petition of the agent for changing from one scene to another requires a set of calls to the *governor*, asking first for this willing. After this call, the *governor* informs about all available transactions (elements that act as connectors between scenes), and finally the agent chooses the desired transaction. But still, it has to wait until a confirmation from the *governor* arrives. All these calls are asynchronous, and an EIAgent requires a complex mechanism to control all possible income and outcome petitions that can be generated.

This asynchrony between the action and the result is the main obstacle in order to integrate a JADEX agent in the eI framework. Although it can be done, the result is that the agents are really complex to design and program. On top of that, we wanted to provide a high level access to the eI to avoid the micro-management that is currently necessary when programming an EIAgent and at the same time to separate completely the code necessary to manage the interaction with the eI and the reasoning processes of the agent.

Both issues are linked and the proposed solution is the addition of a new layer between the *governor* API and the JADEX agent. This layer, that has been integrated into the EIDE (the Electronic Institution Development Environment), transforms in synchronous the interaction with the *governor* basically by providing a set of elements that are responsible for managing the asynchronous calls and that only notify the agent when all the micro-management actions have been performed. At the same time, this layer also provides the agent with a more abstract view of the eI hiding all the low level details.

The second element necessary to complete the JADEX integration is the definition of a JADEX *capability*. A *capability* is a set of specific plans that are associated to a given functionality, in this case the interaction with the eI. In our case, the *capability* is a single plan called *EIConnectionPlan*, that is running in background. It is able to generate JADEX application events as a response from calls coming from the middle layer described before. These events will be enqueued and probably will generate the execution of some predefined plans. The advantage of this approach comes forward when the designer of JADEX agents is able to treat the eI events in the same way that regular events, offering a powerful flexibility and avoiding technical issues that do not belong to the reasoning process. Furthermore, this capability allows JADEX agents to use global eI services (like the reputation service described in section 3.1).

The *EIConnectionPlan* is specific for a concrete eI, since different electronic institutions might have different scenes with different names, as well as a different ontology for communication. In order to facilitate even more the programming task of agents, a wizard-alike tool is being designed as an extension of the aBUILDER to generate skeletons of JADEX agents, including both ADF files and *java* plan templates, for the participation in a concrete electronic institution.

5 Providing a computer interface between humans and the e-Institution

As we have seen, the eI is the kernel of our system. However till now we have shown only how a cognitive virtual agent can connect to this e-institution and interact with the environment and with other virtual agents. The same way a virtual agent can evolve in an e-institution we need a mechanism that allows humans to participate in the same environment.

5.1 Requirements

To simulate any mixed experiment participating eAgents together with humans, we need the following characteristics to be fulfilled in order to humans to be able to interact with the eI: Remote access, several humans in the same experiment, friendly interface, and data tracking. All these requirements suggest that the best technology regarding the human interaction with the e-institution is one based on the WWW. Clearly it allows remote access and parallelism to allow several humans at the same time. Finally the interfaces can simulate those interfaces used in current e-commerce applications that run over the net and that users have seen several times or even have had the opportunity to use. About tracing the actions of a user, the specific WWW technology we have selected (and that will be commented in the following sections) makes this also possible.

5.2 The AJAX approach

AJAX[3] is not a technology by itself but a combination of other three technologies that work together. In fact it is considered a web design technique that provides a high level of interactivity, avoiding the undesirable reloading of the web pages after each user action. The AJAX approach allows the exchange of small pieces of information between the client side (the web browser) and the web server through background asynchronous calls, and therefore, it offers the possibility to update or change parts of the web page without having to reload it completely.

The three technologies that use the AJAX approach are HTML(or XHTML) as a markup language for the web pages, DOM (Acronym of *Document Object Model*), and the *XMLHttpRequest* object, for the exchanging of XML text with the web server. The AJAX approach helps to transform web interfaces to something more interactive, fast and the most important, usable. An extended explanation can be found at [12].

5.3 Integrating AJAX in e-institutions

Figure 3 shows the elements that allow a human to participate in a simulated e-institution using the AJAX approach. These elements are:

- Virtual agents (E-Agents). Agents that are connected to the e-institution through the *governors*, together with the humans, represented by interface agents (I-Agents),are the individuals that participate in the experiment.
- Staff agents. These are agents that belong to the eI and that take care of different aspects related to the good functioning of the eI in the scenes.
- Data base (MySQL DB). For both data storage and analysis.
- Web server. This is the connection between a human user and the e-institution.
- Client application. As stated before, we have adopted the AJAX technology approach. The user only needs a relatively recent web browser and therefore it is not necessary to install special software.

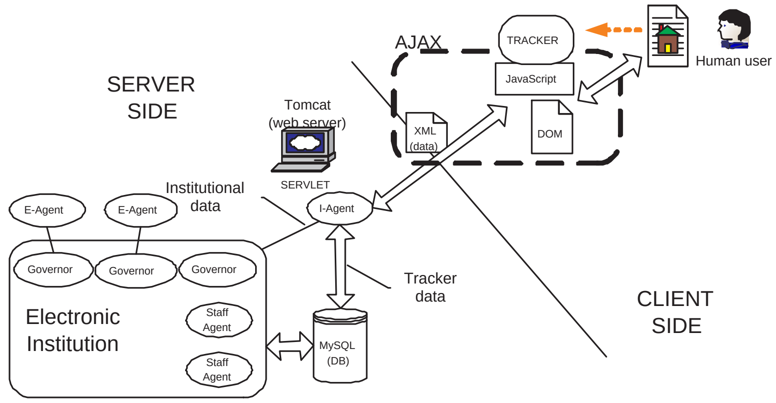

Fig. 3. Human - eI interaction

The central element in the link between the user and the e-institution is the servlet that is activated once the user (client side), using a web browser, establishes the connection with the server side. This piece of software (I-Agent) is seen as a servlet from the point of view of the web server but at the same time as a normal agent from the point of view of the eI. The difference with a normal E-Agents is that the I-Agent is only an interface that acts as a bridge between the eI and the web server.

The I-Agent, acting as a servlet, receives the messages in XML format from the application in JavaScript that is running locally in the web browser of the user. The XML messages can be of two types: Tracker messages and Institutional messages. The first one is the information that can be used later to analyze the actions performed by the user in the web pages. The servlet stores these messages directly in the DB. The second one is associated to the e-institution. These are actions that the user wants to perform (in form of illocutions) and that can have some influence in the state of the scene (or scenes) of the eI the user is participating at that moment.

At the same time, the client application receives XML messages from the servlet with the changes that have been produced in the eI so they can be shown in the browser. The advantage of the AJAX technology is that only the data that has changed has to be sent to the client. All the information about the visualization of this data is already in the client side. This reduces a lot the amount of information the client and the server have to exchange, which is very important in our context.

6 Conclusions

In this paper we have presented a technological framework which allows both humans and virtual agents participate in the same experimental environment

[3] Acronym of Asynchronous JavaScript And XML

in order to perform experiments on reputation mechanisms. This framework is based on the notion of electronic institution and uses a set of tools currently available to design, run and simuate these kind of environments. The extensions proposed to these tools is the main contribution of these paper.

Using these extensions we have implemented a proof of concept platform where users can log in using a web browser to participate in the task being simulated by the eI behind, interacting meanwhile with other humans and agents. In this case, we simulate an electronic auction market, eBay-like, where users are given a list of products to buy. They compete with several agents, and thanks to the anonymity that the web provides, users do not know whether the other agents are humans or electronic agents.

7 Acknowledgments

This work was supported by the European Community under the FP6 programme (eRep project CIT5-028575 and OpenKnowledge project FP6-027253). Jordi Sabater-Mir enjoys a Ramon y Cajal contract from the Spanish Government.

References

1. eRep: http://megatron.iiia.csic.es/eRep. (2006)
2. Noriega, P.: Agent-Mediated Auctions: The Fishmarket Metaphor. IIIA Phd Monography. Vol. 8 (1997)
3. Rodríguez-Aguilar, J.A.: On the design and construction of agent-mediated electronic institutions. IIIA Phd Monography. Vol. 14 (2003)
4. Esteva, M.: Electronic Institutions: from specification to development. IIIA PhD Monography. Vol. 19 (2003)
5. Sierra, C., Rodríguez-Aguilar, J., Noriega, P., Arcos, J., Esteva, M.: Engineering multi-agent systems as electronic institutions. Upgrade **5** (2004) 33–38
6. Esteva, M., de la Cruz, D., Sierra, C.: Islander: an electronic institutions editor. In: Proceedings of AAMAS'02, Bologna, Italy. (2002) 1045—1052
7. Esteva, M., Rodríguez-Aguilar, J., Rosell, B., Arcos, J.: Ameli: an agent-based middleware for electronic institutions. In: Proceedings of AAMAS'04, New York, USA. (2004)
8. Sabater, J., Paolucci, M., Conte, R.: Repage: Reputation and image among limited autonomous partners. Journal of Artificial Societies and Social Simulation **9**(2) (2006)
9. Pinyol, I., Sabater-Mir, J., Cuni, G.: How to talk about reputation using a common ontology: From definition to implementation. In: Proceedings of the Ninth Workshop on Trust in Agent Societies. Hawaii, USA. (2007)
10. Rao, A., Georgeff, M.: Bdi agents: From theory to practice. In: Proceedings of the First International Conference on Multi-Agent Systems, San Francisco, USA. (1995)
11. Jadex: http://vsis-www.informatik.uni-hamburg.de/projects/jadex/. (2006)
12. Garrett, J.: Ajax: A New Approach to Web Applications. Technical report, Adaptive Path, LLC. San Francisco, CA, USA. (2005)