An Architecture for Simulating Internet-of-Services Economies

Stefan König[1], Isaac Pinyol[2], Daniel Villatoro[2], Jordi Sabater-Mir[2], and Torsten Eymann[1]

[1] University of Bayreuth
Chair of Information Systems Management
Bayreuth, Germany
stefan.koenig@uni-bayreuth.de
[2] Artificial Intelligence Research Institute (IIIA)
Spanish National Research Council (CSIC) Bellatera, Barcelona, Spain

Abstract. The Internet-of-Services describes a general paradigm of distributed computing with transparent service selection in a shared infrastructure. One particular question to be solved is how to match service supply and demand dynamically, while information is asymmetrically distributed between buyers and sellers (represented by agents). As buyers can not investigate the computation service before use, sellers can behave strategically. Including trust-enhancing market concepts or reputation mechanisms can help to lower this information gap. Researching into this effect requires the setup of simulation environments, that allow to change policies (e.g. market structure or reputation parameters). In this paper, we present an architecture of a simulation environment integrating electronic institutions from multi-agent research to simulate Internet-of-Services systems.

1 Introduction

Businesses have to encounter several different challenges, when it comes to using Information Technology (IT). The increasing dynamism of markets leads to a continuous need for IT-Business-Alignment and the control of IT investments and resources. The Internet-of-Services (IoS) describes a general computational paradigm, which allows companies to procure computational resources externally and thus to save both, internal capital expenditures and operational costs. For the provider of Internet-of-Services, the business model lies on the economies of scales.

From a technical point of view, Internet-of-Services virtualizes physical resources to logical units, which can be assigned to different users. Thus, using resources in parallel becomes possible, leading to overall better utilization and the execution of computationally intensive jobs within shorter time. One important characteristic is the distributed and perhaps redundant provision of storage, processing power, or more abstract services that extend over different organizations [1,2]. The heterogeneity of services and resources is opaque to the end user.

L. Braubach et al. (Eds.): MATES 2009, LNAI 5774, pp. 258–267, 2009.

1.1 Markets to Allocate Resources and Services

An efficient allocation mechanism between service demand and supply is needed to get such an environment running – a market. But introducing markets will lead to other problems, e.g. asymmetrically distributed information between service providers and consumers. Service providers usually have more information about quality or availability of the services, than their potential users (consumers). This case of asymmetrically distributed information usually leads to suboptimal results due to the uncertainty on the consumer side, and thus to an inferior usage of the service environment in total. Symmetrically, consumers have more information about their liquidity. In addition, both transaction participants deal with uncertainty caused by environmental factors (e.g. network failures). Effects, which are observed in physical markets, can be found in service environments as well [3,4]. Without regulation and coordination mechanisms, the Internet-of-Services can thus suffer from low quality of service (QoS), like in the proverbial "market of lemons" [5].

Because of these economic issues, policies as kind of rules need to be defined to overcome these shortcomings. In this sense, finding strategies and defining policies to ensure certain QoS in the Internet-of-Services must rely on (1) predefined negotiation protocols, following institutional approaches and (2) subjective distributed mechanism, following social approaches (for instance, by the use of reputation and trust mechanisms). Because of the context-dependence of both conditions, simulating tools seem a good solution to test theoretical hypotheses if they allow to change these policies. Therefore we recommend to make use of electronic institutions.

1.2 The Usage of Electronic Institutions in the IoS

On the one hand, negotiation protocols must provide general policies that all participants have to follow. On the other hand, since not all policies are enforceable in all system physiologies, trust and reputation mechanisms, similar to those used in e-Commerce applications, arise as a good distributed solution. Especially systems involving different organizations or open systems hamper the introduction of consistent policies [4]. However, our understanding of trust extends the prevalent technical-oriented views in service environments. Secure communication and digital certificates are necessary, but not sufficient to generate trust both to the system and to other participants. However, trust has also to be seen as a social paradigm, which can be built dynamically between agents with regard to its past behaviour. This social approach offers as well an interaction control that escapes from the security approach and becomes critical for achieving a well-fare market with asymmetrically distributed information.

The use of eI in the simulation tool provides us several advantages in the context of simulating policies for the IoS:

– A completely integrated and widely used framework to graphically design and deploy eI through EIDE[1]. This provides us with tools to easily design

[1] http://e-institutor.iiia.csic.es

negotiation protocols, improving their conceptualization and possible modifications. Further, it provides tools for monitoring at run-time agent's performances in the negotiation protocols.
- An extensible agent architecture (*E-Agent*) to allow software agents to participate in eI. Then, software agents using complex architectures like BDI (*Belief, Desire, Intention*) can easily participate in eI.
- Reputation and trust models management. The E-Agent architecture ensures a completely interoperability of agents using different reputation models in the same eI[6].
- Direct applicability for real applications. In fact, EIDE provides a set of tools to allow humans to participate in eI. Therefore, after an experiment has been simulated using the platform, exactly the same design of eI could be used as negotiation protocols involving humans and virtual agents.

The paper is structured as follows: In the next section we state some related work. Afterwards, in section 3 we expose a brief background on electronic institutions concepts and tools, together with the conceptualization of Internet-of-Services economies as electronic institutions. Also in this section we explain how eIs deal with reputation mechanisms. In section 4 we describe the abstract architecture and put the simulation to work by presenting an instantiation of a possible simulation and showing some results. Finally, we conclude the paper and present the future work in section 5.

2 Related Work

Using intelligent agents for trading resources in the Internet-of-Services is not really new. Foster et al. [7] state that both, Grid computing and software agents, are about the design of distributed systems. Whereas the Grid community focused on the development of the "infrastructure and tools for secure and reliable resource sharing within dynamic and geographically distributed virtual organizations (VOs)" [7, p. 1], the agent community focused on the development of intelligent agents being able to act in uncertain and dynamic environments.

In order to model different characteristics of services and resources, we use a two-layer market, which has been evaluated through a prototype in the CATNETS project (see figure 1). The project defines a market for services (SaaS-market) and infrastructures (IaaS-market). The main differences of this paper to this approach are the assumptions regarding agents' behaviour: the agents are not assumed to act cooperatively.

A set of additional simulation toolkits for distributed systems has been designed. One of the most promising ones is the OptorSim toolkit [8]. However, this system is not further developed as of 2006 and therefore lacks user support and adoption to future network settings. GridSim [9] presents a quite comprehensive simulation framework; it focuses strongly on Grid Computing applications, thus stressing technical details such as scheduling or generation of virtual organizations and advance reservation of resources. Instead, our aim is to build an architecture combined with some existing tools that enables an easy-to-deploy

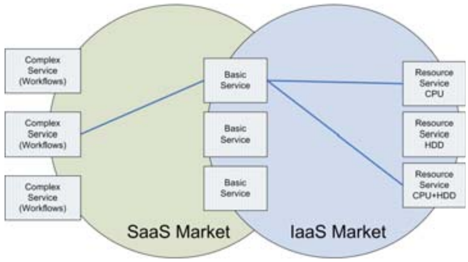

Fig. 1. SaaS and IaaS Markets

simulation environment, even for researchers that are not really familiared with programming simulations.

In the next section we give a brief description of the main concepts and characteristics of electronic institution.

3 Electronic Institutions for Internet-of-Services Economies

3.1 Electronic Institutions: Basic Concepts and Tools

In everyday life, we, as individuals, deal with many other people in order to achieve our goals. Many of these interactions are regulated by an institution that takes care we are following a set of norms and protocols. The concept of *electronic institution* [10] is inspired from these human institutions. Open multi-agent systems are composed of autonomous entities that interact to achieve individual or collective goals. The behaviour of these entities cannot be guaranteed. Therefore, and similar to what happens in human societies, we need mechanisms to guarantee the well working of the system in spite of local behaviours. The use of an electronic institution that regulates the behaviour of agents is one of this mechanisms, and can be complemented by other mechanisms like, for example, the use of reputation. You can find a running example [11] of the usage of electronic institutions for the usage of simulating the behavior of humans in hybrid experiments.

Summarizing, we identify the following main components regarding eI: *Agents* are the players in an electronic institution, interacting by the exchange of illocutions, whereas *roles* are defined as standard patterns of behaviour. EIs establish the acceptable speech acts by defining the ontology and the common language for communication and knowledge representation, which are bundled in what is called a *Dialogical Framework*. Interactions between agents are articulated through agent group meetings (which are called *scenes*) with a well-defined protocol. Protocols in a scene are considered to be the specification of the possible dialogues the participating agents may have. Scenes can be connected, composing a work flow, in a so-called performative structure. The specification of a *performative structure* contains a description of how agents can legally move from scene to scene. The purpose of the normative rules is to modify the behaviour of agents by imposing obligations or prohibitions. Institutional agents are committed to undertake the required actions so as to ensure that non-institutional agents abide by *institutional rules*.

The *Electronic Institutions Development Environment* (EIDE), an integrated development environment for Electronic Institutions, is a set of tools developed at the IIIA-CSIC aimed at engineering multiagent systems as electronic institutions. ISLANDER [12] is the graphical tool to specify eI, and AMELI [13] is the agent-based middleware to run eI. In the next subsection we use these concepts to specify an eI that captures the nature of Internet-of-Services economies in the context of a multiagent system.

3.2 Using Electronic Institutions in the Context of Service Economies

To use an eI, it is necessary to define agents and roles, a dialogical framework, scenes, a performative structure and normative rules in accordance to the identified service markets (see figure 1) and agent roles:

Complex Service agents buy high level services on the service market. *Resource agents*, on the other side, sell their resources (e.g. disk space or CPU power) on the resource market. *Basic Service agents* are the only agent type acting on both markets. They offer their services on the service market and buy and recombine resource services on the resource market. Beside these agents we have to define staff agents, which offer auction services. The participating agents can optional exhibit the role of an staff agent. That means, they can sell their own products or demands. Ontological problems that occur in real distributed systems are not addressed by this paper.

A bit more interesting seems to be the performative structure of our Internet-of-Services use case. Figure 2 illustrates the definition made with the ISLANDER tool [12].

We will now consider the most important scenes. Some of them have just "technical" reasons. After the mandatory initial state all agents have to register at the eI. Additionally, Resource agents are able to register their resources. Once the resource agent has successfully sold its resource for a certain time it can decide how to proceed: leaving the e-Institution (as kind of marketplace) or providing another resource service again. With the latter case, we are able to simulate one kind of cheating behaviour of single agents. The Complex Service agents' possibilities are analogous to the resource agents: after buying a service they can choose between leaving the system or buying another service. A bit more complicated is the route of the Basic Service agents through the different scenes: a Basic Service agent has to buy resources on the resource market. The resource auction is implemented through the scene *AuctionRM*. Only after succeeding on the resource market the Basic Service agent is able to provide its basic service on the service market. To do this, it has to join the *ItemRegisterSM* scene. After finishing the service auction there, the Basic Service agent can decide if it leaves the e-Institution or runs through this cycle again.

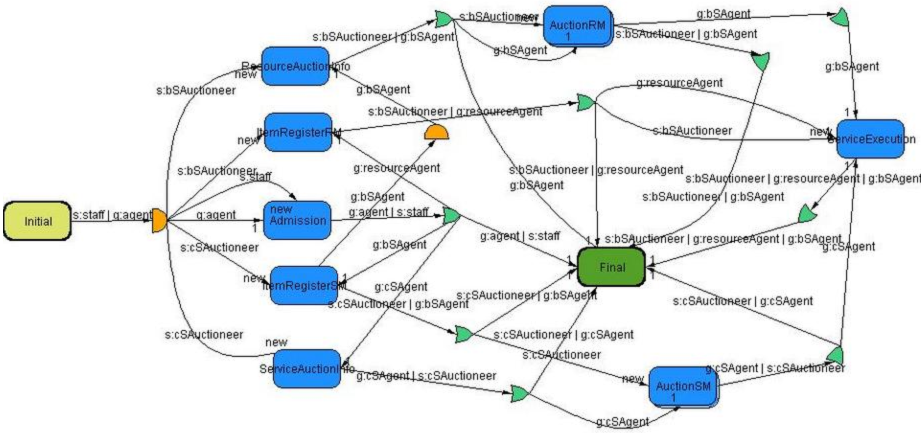

Fig. 2. Service Economy Model implemented as e-Institution. In the scenes *AuctionRM* and *AuctionSM* agents participate in auctions in the resource market and the service market respectively. Previously, all agents and products have been registered in some of the other scenes.

In the *ServiceExecution* scene all agent types meet up after performing successful transactions on the markets. They try to execute the service invocation. If the resource agent for example decides to cheat and not to execute the service, it notifies the other agents in this scene.

The way staff agents take through the performative structure is like follows: they create new instances of their scenes, e.g. the *Service Auctioneer* (and analogous the *Resource Auctioneer*) creates an instance of the *AuctionSM* (i.e. the service market). After that it just waits for products to sell. Note: Currently we have just implemented an electronic Institution market definition for auctions. Changing this to some other (smarter) negotiation protocols will be future work.

3.3 Reputation Mechanisms and Electronic Institutions

As we mention in the introduction, a social vision of trust provides another level of interaction control. The security approach guarantees privacy, authenticity and integrity of information, and the institutional approach guarantees protocols of interaction and performative structures. From a social approach it is desirable that the same customers, as autonomous entities, have access to some social information to dynamically build their trust towards other members of the society. One source of social information is *reputation*.

From a computational point of view, trust and reputation models have attracted increasing interest in the field of multiagent systems. The models that appeared in the literature mainly follow two different and well-distinguished approaches. On the one hand, centralized reputation models consider trust or reputation as a global and public property of the agent that everybody can observe. In this sense, reputation values are kept by a central authority that is accessible for all participants. These kinds of models are widely used in online markets like eBay or Amazon. On the other hand, distributed models consider trust or reputation as a subjective property of the agent. So, each agent has its own vision of the society. Through its experience, observation and maybe communications from other members each agent creates its own opinion, that can be shared, or not.

– **Centralized Models in eI:** These models need 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. Different reputation models are provided through the usage of the eI.
– **Distributed Models in eI:** In this case, each agent has its own reputation model, and because of that, another problem arises: the interoperability between agents using different reputation models. This problem is partially solved in [6], by proposing a common ontology on reputation concepts that all agents have to share and use when communicating with other agents.

4 Integrating eI into an Internet-of-Services Simulator

This section briefly describes the overall architecture. The e-Institutions are used as a kind of marketplace to trade resources or services. In such a marketplace different negotiation protocols can be implemented. On the one hand resources and services (auctions) or demand (reverse auctions) can be offered or both of them (double sided markets like Continuous Double Auctions or through bargaining). On the other hand, agents must be able to use a central or distributed reputation mechanism to consider the reputation impacts on service markets.

While the last section focused on the e-Institutions layer, we will now consider the functionalities of the simulator core and the agent architecture.

In an advanced stage of the prototype implementation different competing marketplaces could be thinkable. Most of the agent process phases (i.e. Selection, Negotiation, Execution and Enforcement phase) are fulfilled by the eI functionality. Only the Execution phase, as a very specific one, has to be added (simulated or real). Further, different scenarios can be simulated, for example some agents defecting all the time, with a certain probability or the smartest one, agents deciding based on their strategy if they cooperate or defect, to allow for strategic agent behaviour.

4.1 Simulator Core Functionalities

A Scenario Generator is able to generate a grid network. The Application Layer Network is composed of different interconnected nodes. Each node can host different agents, which represent a certain resource or resource bundle (Resource agent) or a higher abstracted service (Basic Service and Complex Service agents). The network determines the time delay between sending a message and receiving it. So, a service with a very short-term time restriction might fail due to the message time delay between the corresponding nodes.

The network representing an IoS is defined by a connected non-oriented graph, represented by a set of network sites $S = \{1, .., n\}$ and a set of links between the sites $L = \{\langle i_1, j_1\rangle, .., \langle i_m, j_m\rangle\}$. In addition, a failure probability f_{S_i} is defined for each node. When a failure occurs during simulation, the node is not able to answer any request or routing further messages. Which site will fail in each time tick, is chosen randomly. Furthermore, on each node a set of the three different agent types, Complex Service Agents (CSA), Basic Service Agents (BSA) and Resource Agents (RA) is initialized. For each site the number of economic agents is $|CSA_{S_i}| \geq 0, |BSA_{S_i}| \geq 0, |RA_{S_i}| \geq 0$. A node without any associated economic agent is a router. Each link $\langle i, j\rangle$ between two nodes has a certain bandwidth. A higher bandwidth leads to an increased data transfer. In our simulation model, the bandwidth is biased, which means that a link is defined or not. Nodes, which are not directly linked can be addressed through a routing table that is calculated by a common routing algorithm, the Dijkstra algorithm [14].

Further, a network connection between nodes might break down. Agents are not longer connected to each other. This might also cause problems during service execution. In addition to these functionalities, the simulator provides a resource locking model. If resources are sold, they are locked and can not provide their capacity to other agents. This does not avoid the cheating behaviour that agents might sell a service or resource twice. But during the execution phase, the resource can be locked only once, such that one service can not be fulfilled by the agent.

4.2 Experimenting with the Simulation Platform

In the following, we present a simple instantiation of the simulation platform to illustrate the potential of this tool. Even when the objective of this paper is not to test any experimental hypothesis we want to show the flexibility that our simulator offers when testing Internet-of-Services economies.

In this case, taking as a base the eI specified in figure 2 we decided to use an English Auction protocol for both, resource and service markets negotiation. ISLANDER offers graphical tools to specify such protocols. Thus, the modification or even a complete exchange of interaction protocols is quite simple and furthermore, can be used in both the simulation platform or real environments where eI is used. The simulation process works as mentioned in subsection 3.2. The entry point to the simulation is the CSA. The CSA has to fulfill an external generated demand. In our simulation the demand is generated by an uniformly distributed interval span. The kind of Basic Service that the CSA has to buy is also given by demand generation. The BSA on the other hand has to compose different resources by a certain combination. The demanded resource bundle can differ between the BSA-types. After the agents have found an agreement, the settlement phase is simulated through the exchange of messages, which are sent from the invoking agent to the sold agent, together with their corresponding answer, if the invocation has been successful.

Figure 3 shows the market price on the service and on the resource market. This simulation run has been conducted with ten nodes, hosting eleven CSAs,

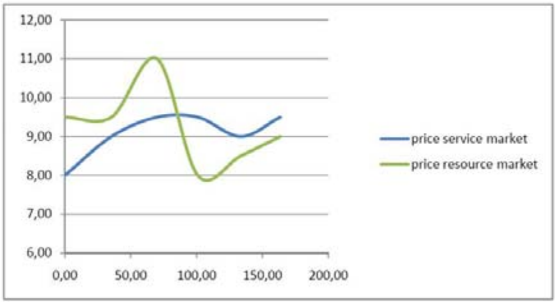

Fig. 3. Price evolution

seven BSAs and four RAs. More than 20 interactions have been fulfilled. The agents follow a very simple strategy and use their reservation price when bidding in an English Auction. If they succeed, they decrease their own reservation price, otherwise they increase it in order to be able to win the next auction.

This results could be easily compared with simulations using other negotiation protocols (for instance a Dutch auction protocol), other reputation models and other agent strategies[2].

5 Conclusion and Future Work

We have presented a simulation platform that uses agents, electronic institutions and the paradigm of the Internet-of-Services. This seems to be a promising combination when talking about service markets where reputation-based trust plays an important role. The main characteristics of an agent participating in both different environments have been extracted, and merged into one that is able to participate in a simulated environment like that one. With our simulation tool based on the Electronic Institutions, it will be possible to vary different parameters regarding policies in the Internet of Services, i.e. the interaction protocols, the reputation models or the network topology of the overall system. Our tool set can contribute to the vision of Internet of Services and service markets becoming reality as it alleviates the side effects, which occur by introducing markets to trade resources and services. Finally, changing the simulation core against a real middleware implementation will provide a proof-of-concept system as future work.

Acknowledgments

This work was supported by the EC under the FP6 programme [eRep project CIT5-028575]; the Spanish Education and Science Ministry [AEI project TIN2006-15662-C02-01, AT project CONSOLIDER CSD2007-0022, INGENIO

[2] The source code of a beta version of the platform together with this particular simulation can be downloaded at http://sourceforge.net/projects/erepsim

2010]; Proyecto Intramural de Frontera MacNorms [PIFCOO-08-00017] and the Generalitat de Catalunya [2005-SGR-00093]. Daniel Villatoro is supported by a CSIC predoctoral fellowship under JAE program.

References

1. Kesselman, C., Foster, I.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, San Francisco (1998)
2. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The physiology of the grid: An open grid services architecture for distributed systems integration (2002)
3. Streitberger, W., Hudert, S., Eymann, T., Schnizler, B., Zini, F., Catalano, M.: On the simulation of grid market coordination approaches. Journal of Grid Computing; Special Issue on Grid Economics 6(3) (2008)
4. Eymann, T., König, S., Matros, R.: A framework for trust and reputation in grid environments. Journal of Grid Computing; Special Issue on Grid Economics 6(3), 225–237 (2008)
5. Akerlof, G.A.: The market for 'lemons': Quality uncertainty and the market mechanism. The Quarterly Journal of Economics 84, 488–500 (1970)
6. 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, pp. 90–101 (2007)
7. Foster, I., Jennings, N., Kesselman, C.: Brain meets brawn: Why grid and agents need each other (2004)
8. Bell, W.H., Cameron, D.G., Millar, A.P., Capozza, L., Stockinger, K., Zini, F.: Optorsim: A grid simulator for studying dynamic data replication strategies. International Journal of High Performance Computing Applications 17(4), 403–416 (2003)
9. Buyya, R., Murshed, M.: Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. Concurrency and Computation: Practice and Experience (CCPE) 14, 1175–1220 (2002)
10. Esteva, M.: Electronic Institutions: from specification to development. IIIA PhD Monography, vol. 19 (2003)
11. Brito, I., Pinyol, I., Villatoro, D., Sabater-Mir, J.: Hiherei: Human interaction within hybrid environments. In: The Eighth International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2009 (2009)
12. Esteva, M., de la Cruz, D., Sierra, C.: Islander: an electronic institutions editor. In: Proceedings of AAMAS 2002, Bologna, Italy, pp. 1045–1052 (2002)
13. Esteva, M., Rodríguez-Aguilar, J., Rosell, B., Arcos, J.: Ameli: an agent-based middleware for electronic institutions. In: Proceedings of AAMAS 2004 (2004)
14. Dijkstra, E.: A note on two problems in connexion with graphs. Numerische Mathematik 1(1), 269–271 (1959)