AGENCY SERVICES
An Agent-based and Services-oriented Model for Building Large Virtual Communities

I. Lopez-Rodriguez and M. Hernandez-Tejera
Institute for Intelligent Systems (SIANI), University of Las Palmas de Gran Canaria, Las Palmas, Spain

Keywords: Intelligent agents, Middle agents, Cloud computing, Utility computing, Matchmaking & Brokering.

Abstract: Despite the current importance of internet and the increasing appearance of virtual societies, there is still no widespread adoption of intelligent agents. Building solutions based on intelligent agents is not a simple task because of the complex nature of the problems that they face, which are mainly related to planning, cooperation and negotiation tasks. In order to overcome those difficulties, and in line with the recent success of the Cloud Computing model, this paper proposes a new model in which brokers able to represent users in virtual societies are offered as one more service of the cloud. The paper also details the technologies necessary to construct a realistic solution which, apart from abstracting the user from all the details of the implementation, loses none of the characteristic advantages of solutions based on agents and Cloud Computing.

1 INTRODUCTION

Although the presence of software agents has significantly increased in various fields, they have not yet been widely adopted as a strategy. However, that is in contrast to the success of electronic commerce and the emergence of virtual societies, to which they can provide enormous advantages. To a great extent, the reason for that absence lies in the complexity surrounding solutions that are based on intelligent agents and which still represent an insurmountable technological barrier to many users.

Moreover, the new means of access and the ubiquitous character of communications are transforming the traditional business models. The classic conception of software consumed as a product is being substituted by one of a service housed in third party installations. This is corroborated by the latest strategic movements of companies as representative as IBM, Google, Yahoo!, Microsoft or Amazon, all of which are undertaking projects in which the user no longer figures as the owner of a software license but as a consumer of their services. This new way of conceiving the software market is known as Cloud Computing.

This paper proposes the creation of Agency Services in the cloud in order to facilitate the implementation of virtual societies based on software agents. They are proposed as intermediary agents that are instantiated in the cloud by third parties and, after being contracted as services, represent the users in virtual societies by conducting brokering tasks. The paper presents a realistic solution that manages to abstract the user from all the details of the technology without losing any of the advantages of the agents.

The remainder of this article is structured as follows. This section is completed with both a presentation of some important challenges faced by software agents, and a brief description of the cloud computing model. Section 2 gives an overview of middle agents. Section 3 describes the Agency Services model, the technologies needed for building a realistic solution based on it, and finally its contribution to the current models. Section 4 comments the implementation of an auction web site based on the new model, and Section 5 is devoted to the conclusions.

1.1 Challenges of Software Agents

A review of the works related to the study of the relationship between intelligent agents and electronic commerce reveals that, in the quality of software modules, agents are rarely instantiated on the client side (He et al., 2003). On the same line, the community devoted to the agency theory was recently invited

Lopez-Rodriguez I. and Hernandez-Tejera M. (2010).
AGENCY SERVICES - An Agent-based and Services-oriented Model for Building Large Virtual Communities.
In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence - Agents*, pages 309-314
DOI: 10.5220/0002722903090314

to discuss the question “Where are all the Intelligent Agents?” (Hendler, 2007). That author believes that there are currently no really important systems based on agents and noted that, despite the apparent activity in the research area, intelligent agents are far from being the widely adopted technology predicted more than ten years ago.

The origin of the problem lies in the fact that it is a difficult technological challenge for the users to incorporate their own agent-based solutions to problems that would normally be related to planning, cooperation and negotiation in distributed, shared and complex environments.

With the aim of stimulating the creation of agent-based solutions, some works have opted to release development frameworks . They follow a strategy that transfers to the users the responsibility of programming their solutions and, therefore, are not aligned with the dominant class of software that is currently demanded by the market; precisely that which requires no effort from the users. Other solutions have also opted to offer users the opportunity to instantiate agents on the server (Wurman et al., 1998; Sandholm, 2002). However, those agents always belong to the virtual site, are only accessible through the web, subtract autonomy from the users, do not allow to track local resources, and do not hide the details of the technology.

For intelligent agents to become a widely accepted technology, new models must be proposed and the ideas to create them may be based on the Cloud Computing model and service orientation, as will be detailed later.

1.2 Cloud Computing

The classic conception of software consumed as a product is being shifted by that of a service housed in third party installations. Thus, the software is contracted in accordance with needs of the user, who pays for the specific time, quality and functionalities that they need. This new way to marketing computer product and services is known as Cloud Computing (Armbrust et al., 2009).

The spreading of this emerging paradigm is attributed to: (1) the strong ubiquitous and flat character of communications; (2) the progressive orientation to software solution services; (3) and the maturity of virtualization tools.

The experts coincide in emphasizing that the property that best describes this new paradigm is the fact the clients only have to concern themselves with what they can do with the technology and not with how to install and configure it. They also coincide in highlighting the following aspects as the most significant advantages of the Cloud Computing model (Plummer, 2008):

- It enables firms to scale up rapidly in accordance with their requirements by eliminating the need to add new equipment, software or personnel.
- It significantly reduces the costs of maintenance and energy consumption by moving the scalability to the cloud.
- It allows solutions to be implemented in a much more streamlined fashion since it eliminates much of the investment stage.
- It expands the technological options of businesses by smoothing the path for them to embark on projects of greater importance without the costs that they would normally entail.
- It makes it possible to delegate the execution of critical tasks to specialized services.
- Informatics as a service can give way to movements of cooperation in the cloud. There is not doubt that it predisposes the net to adopt standards of communication and interoperation.

2 AGENTS AS INTERMEDIARIES

The objective of middle agents is to assist in communication tasks in order to facilitate exchanges between requesters and providers. Requesters are those agents with objectives they wanted to be achieved by other agents, whereas providers are those agents that fulfill objectives on behalf of other agents. The presence of middle agents is especially valuable in distributed, open environments, where they constitute a mechanism to overcome the heterogeneity of the partners. Although there are several roles for middle agents, three of them are mainly recognized (Klusch and Sycara, 2001):

- **Matchmaker.** Its functionality corresponds to that of the Yellow Pages. Providers register their skills in the matchmaker agent. Requesters consult it in order to identify those providers that are capable of fulfilling their objectives. If the activity of the matchmaker is successful, the requester and the provider then enter into a new dialogue.
- **Blackboard.** Requesters send their petitions to the blackboard agent, and providers look up petitions that they can fulfill in the blackboard. The blackboard agent keeps track of the requests and their respective answers so that other agents can easily extract information later.

- **Brokering.** The broker agent registers the skills of the providers and negotiates the requesters' petitions with them, and finally gives the results of that process to the requesters. In the models dominated by a broker, there is no direct interaction between the requester and the provider.

For its part, FIPA, defines the protocol for the interactions mediated by brokering agents (FIPA, 2002). In short, the agent that initiates the interaction (initiator) delegates the accomplishment of a task to a broker. After sending the request, the initiator plays no further part in the process. In general, the FIPA specification proposes neither client autonomy nor the need to relieve the client of the technical details.

3 AGENCY SERVICES

3.1 The Model

The Agency Services can be defined as middle agents that are instantiated in the cloud and, after being contracted as services, represent the users in virtual societies by performing brokering and management tasks.

The objective of the Agency Services model is to transfer the complexity of the agents' activity to the cloud. However, that does not mean that the presence of an agent in the client node should be eliminated. In this sense, the agency service agent is an abstraction of two agents: a broker agent instantiated in the cloud as a service, and a local agent that is light but sufficient to represent the interests of the users and to give them autonomy.

In one hand, the tasks that should be delegated to the broker agent in the cloud are all those that are complex, such as the planning, coordination, cooperation and negotiation processes, and the algorithms necessary to solve special problems. In the other hand, the client is a software agent who is responsible for the simple, easily automated tasks, which includes: monitoring of local resources, response to events, meeting the broker agent's requests, and sending it information such as the state of the resources or new user's directives.

Basically, the model proposes a functional breakdown of the tasks that a single agent usually tackles. Now, the client will be represented by two intercommunicating agents who together are able to accomplish the user's objectives. In order not to lose the simplicity that characterizes Cloud Computing, both agents are supplied by a provider of Agency Services located in the cloud. In that respect, it is particularly important that the installation of the local agent is automated in a way that requires no intervention by the user.

The new model (Figure 1) is necessarily made up of providers with sufficient resources to deploy an infrastructure of Agency Services in the cloud. These Agency Services Providers (thereafter called ASPs) are contracted by users to participate in Business Sites, i.e. virtual environments. Thus, the principal model is defined by three elements:

- **Agency Services Providers (ASPs).** Technological firms with the ability to deploy services architectures on internet. They offer clients brokering services based on agents capable of participating in Business Sites.
- **Business Sites.** A virtual environment on internet in which all the ASPs that adhere to the standards of communication and its prescribed social conventions can participate. Common examples of Business Sites are the auction and buy-and-sell sites on internet, grid computing organizations and applications based on Cloud Computing.
- **Client.** A user or company that contracts the Agency Services in order to participate in Business Sites from which it expects to make profit. Basically, clients do not have the necessary resources to develop their own solutions, or to do so profitably. In the client node, a light agent capable of communicating with the contracted Agency Service is instantiated.

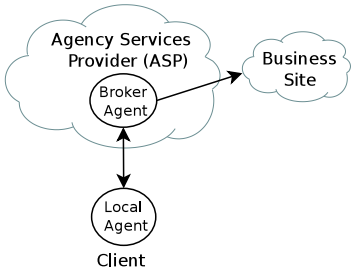

Figure 1: Agency Services model.

This conception provides all the features of the service orientation to the agency services model. In general, a firm can provide Agency Services for many types of problem and Business Sites while, due to the definition of interfaces and social conventions, a Business Site is able to accept the participation of agents instantiated by different ASPs.

Figure 2 presents an example of the proposed model. It includes three ASPs (P_x, P_y, P_z), two Business Sites (B_m, B_n) and four clients (C_a, C_b, C_c, C_d). Clients C_a and C_b are interested in participating in Site

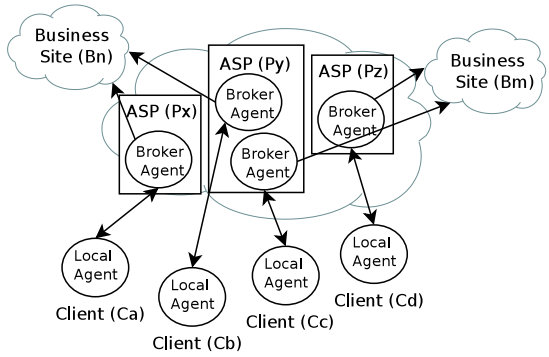

Figure 2: Scenario based on Agency Services.

B_m. To that end, the first of those clients uses the services of P_x, and the second those of P_y. Moreover, Clients C_c and C_d wish to participate in Site B_n and, with that aim, C_c uses the services of P_y while C_d uses those of P_z. In the diagram, P_y represents the class of provider able to offer services for more than one environment. The figure also shows the fact that, as a general rule, Business Sites must release communications interfaces based on standards to which more than one provider can subscribe.

More specifically, the life cycle of the proposed model comprises the following steps:

1. In order to participate in a Business Site, the client sends a request to an ASP that is suitable for the task and with which the client has previously registered.
2. The ASP deploys a light software agent in the client that is able to communicate with the user (Thanks to the technologies mentioned in Section 3.2). At the same time, it requests a broker agent in its servers, who will subsequently be responsible for representing the user in the Business Sites.
3. If necessary, the user defines his/her directives and preferences on the interface provided by the local agent, who then sends that information to the remote broker agent.
4. The broker agent contacts with the Business Site and initiates its activity on that site. Depending on the context and the activity to be undertaken, the broker agent may need information from the local agent, such as the state of the local resources, or inform it of the state of his participation in the environment. The local agent may also send new directives to the broker agent or respond to the events and information that the broker has communicated.
5. After the broker agent's participation has ended, it informs the local agent of the result. That information may also be registered in the ASP and will be useful in terms of the contract linking it to the user.

Nothing prevents the user from having more than one local agent assigned at the same time. Thus, a local agent can automatically inform the broker about the state of the resources, meanwhile the user can use another agent installed in his/her mobile phone for both receiving information from the broker and, if necessary, sending it new directives. Therefore, the broker agent, apart from participating in Business Sites, can also work as a proxy agent able to communicate the local agents among themselves.

3.2 Software Technologies

For the proposal to be transparent, realistic and effective, the technology required by the client must be simple. Thus, removing the brokering tasks from the context of the user would not make sense if the client later encounters difficulties in establishing or using a module capable of interacting with the broker agent.

In that respect, by exploiting the latest technologies, it is possible to achieve solutions that, for most scenarios, do not involve the user in the configuration or installation processes. This paper emphasizes the instantiation of remotely transmitted software in the client. More specifically, the paper proposes the adoption of:

- **JNLP.** For downloading and instantiation of the local agent in the client node. Java Network Launching Protocol (JNLP) permits the download and launch of Java applications located in remote servers (Zukowski, 2002). The following features stand out: (1) it permits applications to be activated by using a link; (2) it ensures that the latest version of the application is always run; (3) it eliminates complex procedures of installation and updating; (4) it only requires the Java Runtime Environment (JRE); (5) it permits digitically sign the application.
- **JXTA or XMPP.** For communication between the local agent and the broker. Both JXTA (JXTA, 2007) and XMPP (XMPP, 2009) are P2P protocols. The following characteristics stand out in both technologies: (1) they are protocols on which light clients can be simply developed; (2) the content of the messages is coded in XML; (3) the only infrastructure required is the installation of a messenger server, which can be installed in the ASP.

JNLP and XMPP/JXTA constitute a solution that meets the conditions of simplicity, remote instantiation and bidirectional communications described above. Moreover, they have characteristics that make

them specially attractive for open and distributed environments:

- In the Cloud Computing model, it is essential not to ignore mobile devices, which, in the near future, might become an important mean to access the web. In that respect, the presence of Java is practically indispensable nowadays.
- In an environment that is open and consequently inhabited by heterogeneous sources, standard versatile codes like XML are even more important.
- In open environments it is essential to use technologies that ensure that the communications are secure, reliable and confidential. In this regard, XML presents mature solutions.

Moreover, the technology used in the communication between the broker agent and the Business Site must be chosen by the latter. It is the Business Site that determines the communications interface to which the ASPs wishing to participate in its life cycle must subscribe. Here it is not required to be guided by the principal of simplicity: it is now supposed that both parties have sufficient technology to develop elaborate and efficient solutions.

3.3 Advantages of Agency Services

In general, the advantages of the concept of Agency Services are:

- **Horizontality.** A solution that, with all its virtues, can be applied to a wide range of domains.
- **Participation.** The transfer of the most tedious tasks to the cloud, together with the simplicity of the technologies instantiated in the client, makes it easier for users to automate their participation in virtual societies.
- **Scalability.** Since the most complex part of the technology is located in the cloud, the solutions can increase with no effort required from the client.
- **Autonomy.** Bidirectional communications offer both of them, the broker and the client, the possibility of being able to exchange dialogue while the activity is developed in the Business Site. This guarantees the autonomy of the users since they can create scenarios in which they are kept informed of events and, if necessary, can send new directives.
- **Flexibility.** Without making any type of extra effort, the client can participate in more than one virtual society at a time, and always with the provider that best fits its needs.
- **Transparency.** To participate in a virtual society, the clients do not have to develop their own agents.
- **Reliability.** The model permits criteria to be established for the ASPs in order to guarantee that the activity of the provided broker agents does not endanger the stability of the system with selfish or anti-social behaviors.
- **Security.** Access to the Business Sites is restricted to a controlled set of ASPs on whose interfaces all efforts related to the security of the system can be concentrated.
- **Business.** The nature of the interactions, clearly oriented to a competitive model, helps create business activities: a recognized advantage of the Cloud Computing model in which, this time, the agency theory can play a role.

4 A PRACTICAL CASE

To illustrate the advantages of the new model and verify the viability of the proposed technology, a practical case based on an auction site similar to eBay is implemented.

The infrastructure developed for the example comprises two ASPs and one Business Site (the auction site). To attain an environment that is as similar as possible to the cloud, each ASP was installed in a different server and each server was located in a different physical location. Communication between the ASPs and the Business Site took place by means of a set of web services that were released by the latter and basically permitted the broker agents to consult the catalogue and make bids and offers. The web services that the broker can invoke are:

- **GetItemsList.** This returns the list of items for which the broker agents can currently bid.
- **GetItemInfo.** This provides the identifier and description of the item, the sale price, the highest bid made for it, and the starting time of the next round of bidding.
- **SetBid.** This adds a bid for a specific item.
- **SetItem.** This adds a new item to the catalogue or modifies the price of an existing item.
- **SetAccept.** This specifies that the last bid for an item has been accepted by the vendor.

The Agent Communication Language (ACL) over XMPP is used for communication between the agents. More specifically, the broker sends ACL Inform messages to the local agent, whereas the latter uses ACL

Request messages to send new directives to the broker agent. Both agents are able to initiate interactions thanks to the bidirectional nature of XMPP.

The client uses a JNLP link for automatically instantiating the local agent. The user interface deployed displays the list of available items, as well as information about the actions of the broker in the auction site. Moreover, when necessary, the client can initiate a dialog window for sending directives to the broker agent, which is finally responsible for bargaining (Sandholm, 1999) in the auction site according to the latest preferences specified by the user.

The specific technologies used for the example were (Figure 3):

- **Agency Services Provider:**
  - **JRE.** Runtime environment of the broker agents instantiated in the ASP.
  - **OpenFire.** Server for XMPP communications between the broker and local agent.
  - **Jetty.** Web application server from which the local agents can be downloaded by using JNLP.
  - **Smack API.** Java library that implements the XMPP protocol.
- **Business Site (auction site):**
  - **Jetty.** Web application server where the web services to access the auction site are deployed.
- **Client:**
  - **JRE.** Runtime environment of the local agent.
  - **Smack API.** Java library that implements the XMPP protocol.

The client can be any device able to run Java applications, from mobile phones to desktop computers.

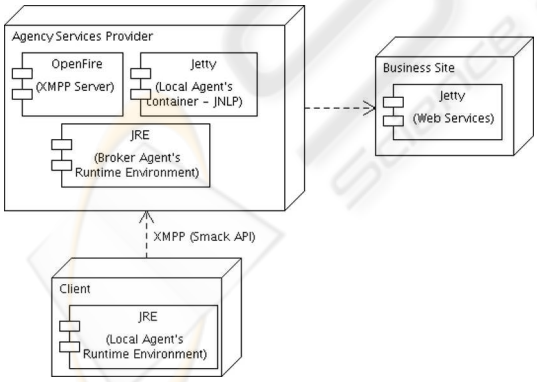

Figure 3: Technologies used in the practical case.

5 CONCLUSIONS

This paper introduces the concept of Agency Services, which is presented as an effective means of transferring to the cloud the complexity that the participation of intelligent agents in virtual societies currently entails. The new approach gives the agency theory all the virtues of the Cloud Computing model and of service orientation in general. Moreover, with the aim of defining a realistic solution, this paper has detailed the technologies necessary to implement a complete solution based on Agency Services, including the remote instantiation of a light agent in the client in order to guarantee the autonomy of the user. Also, in order to illustrate the use of the model, an ASP based solution for an auction web site is described.

REFERENCES

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I., and Zaharia, M. (2009). Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley.

FIPA (2002). FIPA Brokering Interaction Protocol Specification.

He, M., Jennings, N., and Leung (2003). On Agent-Mediated Electronic Commerce. *IEEE Transactions on Knowledge and Data Engineering*, 15:985–1003.

Hendler, J. (2007). Where Are All the Intelligent Agents? *IEEE Intelligent Systems*, 22:2–3.

JXTA (2007). JXTA Community. https://jxta.dev.java.net. Accessed 28 August 2009.

Klusch, M. and Sycara, K. (2001). Brokering and matchmaking for coordination of agent societies: a survey. *Coordination of Internet agents: models, technologies, and applications*, pages 197–224.

Plummer, D. C. (2008). Cloud Computing Myths, Magic and Mayhem. In *Gartner Symposium ITxpo*.

Sandholm, T. (2002). eMediator: A Next Generation Electronic Commerce. *Computational Intelligence*, 18:656–676.

Sandholm, T. W. (1999). *Multiagent Systems: A modern approach to Distributed Artificial Intelligence*, chapter Distributed Rational Decision Making, pages 201–258. MIT Press.

Wurman, P. R., Wellman, M. P., and Walsh, W. E. (1998). The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents. In *Proceedings of the Second International Conference on Autonomous Agents*, pages 301–308.

XMPP (2009). XMPP Standards Foundation. http://xmpp.org/. Accessed 28 August 2009.

Zukowski, J. (2002). Deploying Software with JNLP and Java Web Start. http://java.sun.com/developer/technicalArticles/Programming/jnlp. Accessed 28 August 2009.