Modelling a market using software agents –pedagogical experience in a Distributed Artificial Intelligence course.

Hugo Gamboa[*] **, Bruno Silva,** and **Joaquim Filpe**

Escola Superior de Tecnologia de Setúbal, Campus do IPS, Estefanilha 2910-761 Setúbal, Portugal

The implementation of a market based on software agents was the main goal of a curricular innovation experience in an advanced course on Distributed Artificial Intelligence. The task was prepared to enhance the learning experience trough collaboration and competition among the students. The project involved the creation of a community of agents capable of modelling a market of goods and competences. This modelling objective was accomplished using the software agents methodologies (JADE) and introducing expert system inferential motors (JESS) to provide the intelligence layer to the agents. The implementation methodologies, technologies and the resulting modelling capabilities proved to produce a near scientific research environment that produced an additional learning experience.

Keywords Distributed Artificial Intelligence; expert systems; educational experience.

1. Introduction

The Distributed Artificial Intelligence (DAI) course lectured in Escola Superior de Tecnologia de Setúbal present advanced topics in a area of knowledge very young and fertile. As a general overview, the field of DAI[1-2] studies the creation of multi-agents system as a new approach to solve problems. The agents paradigm applies concepts from artificial intelligence and speech act theory to the distributed object technology. This paradigm is based on the agent abstraction, a software component that has three major characteristics: **autonomous** – agents own their thread of control, being able to control its actions and take decisions; **proactive** – they do not only react to external events but are also capable of taking initiative driven by a goal-directed behaviour; **social** – agents are able to, and need to, interact with other agents in order to achieve its goals.

The research ideas and tools involved are relatively new, combining the integration of several technologies and computational methods, conducing to a set of expertise needs that should be covered by the pedagogical experience. To provide the means for an intense learning experience, it was decided to create a gaming environment to be played by agents developed in each student's group project. The pedagogical experience was supported in previous studies conducted by the teachers' team in concerning e-learning environments [3], and collected ideas from constructivism[4-5], social constructivism[6], and project based learning theories[7], that inspired the design options of the project. The group had also previous research experience with the development of market of goods based on software agents[8].

In the next section we will enunciate the problem proposed to the student. In section 3 the technological options will be presented and in section 4 the implemented system is presented. We finalize this document presenting some conclusions from this pedagogical experience.

2. The problem

The project scenario is based on a market of goods where a set of agents interacts. The game works in a turn based scenario, where the time evolves discretely and where each agent has specific tasks to perform. There is a set of abstract goods that have distinct production costs and consumption demands. The

[*] Corresponding author: e-mail: hgamboa@est.ips.pt, Phone: +351265790000

market is composed of six types of agents: Facilitator, School, Bank, Market Information Service, Consumers and Producers.

The market has created with a set of strict rules for the interoperation of the distinct agents in their roles. We describe the market internal workings to give a general ideal of the set of mechanisms existent in the market. The Facilitator receives the list of needs from the Consumers and the list of prices from each Producer. The Facilitator has to find the diverse matches between Consumers and Producers according to the price, past sells history rules and the advertisement efforts. The Bank agent offers services to lend money with respective costs and to receive money as a deposit and give the correspondent dividends. The School agent receives a learning request from the Producers, and, in a nondeterministic way, give credentials to modify the skills of the Producer so that it's production costs, of a particular product, will decrease. The Consumers buy the products from the producers according to a defined set of needs. These Consumers agent inform the Facilitator agent by stating what are their needs. The Producers decide what to produce, what to learn, what to advertise and how to sell (prices) in each turn and inform the Facilitator agent. These last type of agents are the active agents developed by the students and represent the intelligent part of the system. The other agents have rules of interaction that are immutable.

3. Technologies

The market was developed using the Java Agent DEvelopment Framework [9], also known as JADE. This platform enables the use of a distributed system topology with peer-to-peer networking and software component architecture with agent paradigm. The software interoperability is assured by compliance with FIPA [10] standards, which defines the reference model of an agent platform and a set of services that should be provided. An agent-based system is intrinsically peer-to-peer as each agent is a peer that needs to initiate a communication with other agents as well as being capable to provide services to the rest of the agents. In this kind of distributed environment the system must offer proper services that allow peers to enter, join or leave the network at any time as well as to search and discover other peers. The JADE platform offers these functionalities by implementing the normative FIPA standard services: life cycle management, white page service, yellow page service and message transport service. These services allow each agent to control its own life cycle, dynamically discover other agents and communicate with them. It can register and modify its services and/or search for agents providing given services.
The communication between agents uses the FIPA Agent Communication Language (ACL), based on the speech act theory[11], which includes several communicative acts that allow the representation of different communicative intentions, such as requesting, informing, querying, etc.
JADE offers an API, that allows the agent development, and the run-time environment that provides the basic services used by the agents. Each instance of the JADE run-time is called a container, since it "contains" the agents, and the set of containers is called platform, forming an abstraction layer to the underlying mechanisms, in which the agents "live".
As said before, the agents are capable of reacting to external events and making decisions. For this, the agents must have reasoning capabilities. These capabilities were incorporated into the agents using JESS [12], a tool widely known for creating Expert Systems. JESS operates on a Knowledge Base (KB), containing a set of rules that can be repeatedly applied to a collection of facts about the world. Rules are fired, or executed, if a set of preconditions (or antecedents) is true. JESS allows agents to make decisions, define strategies and to infer new knowledge based on known facts about the world (market). For example, the rule in the next paragraph states that if the agent price for a particular product is higher than the market price, then a product price redefinition is necessary. If the agent knows by a set of facts that *(market-price product1 78)* and *(my-price product1 80)*, then the rule will fire and the agent will infer one more fact: *(redefine-price product1)*, alerting him to take the necessary measures, which may consist in lowering the manufacturing costs, by subscribing itself on the School agent for learning. JESS provides a Rete engine that is used in out agents.

```
(defrule redefine-price
   (market-price ?prod ?price1)
   (my-price ?prod ?price2)
   =>
   (if (> ?price2 ?price1) then
   (assert (redefine-price ?prod))
   )
)
```

3. Implementation

The current implementation establishes a hierarchy of Jess agents, drawn in Figure 1: Facilitator agents, Market agents and Producer agents are derived directly from the Basic Jess Agent. They differ primarily in the user interface and, of course, in purpose. There is only a Facilitator agent per platform, and it must be activated before all others. All other agents register themselves with the Facilitator and he is responsible for re-routing messages between Market agents and Producers agents, as well as controlling the market evolution in time. The Market agents consist in the Consumer agents, the Bank, the School and the MIS, Market Information Service agent. Aside from the Consumer agents, which can vary in number, the others types of agents are also limited to one agent per platform, for simplicity. Finally the Producer agents are those who compete each other for higher profits.

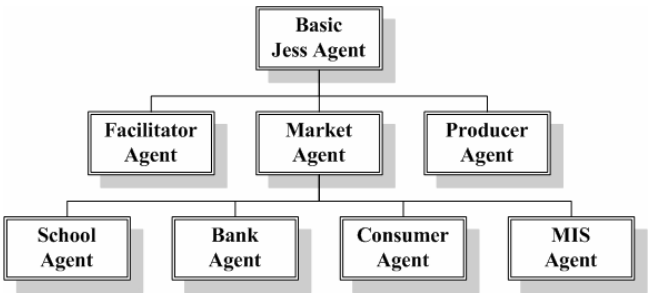

Figure 1 – Agent Class Hierarchy

All agents have their own jess processor object. This is constructed during the agent's setup phase, starting a new JESS inference engine and loading all the appropriate rules for that agent. The Producer loads a custom KB constructed by the competing students. For fraud detection purposes the Producer agent also loads some rules and facts that track the agent capital throughout the market evolution. When the market closes, this value is compared with the value announced by the producer.
The basic Jess Agent exhibits a cyclic behaviour with three parts:

1. Receive ACL messages; these messages contain the sender, the receiver and the content which is a Jess fact that is asserted to the agent's Knowledge Base.
2. Run the Jess inference engine.
3. Check the Jess facts and execute any intentions that can be directly executed (for example an intention to perform a speech act); retract the intention.

We can consider that the agent's mental state uses the BDI architecture [13] since the agent has a set of beliefs, desires and intentions expressed in the KB. They reflect the knowledge that the agent has about the world, goals and necessary actions to achieve them. Each agent makes decisions according to the state of their beliefs, which are updated continuously as the market evolves.

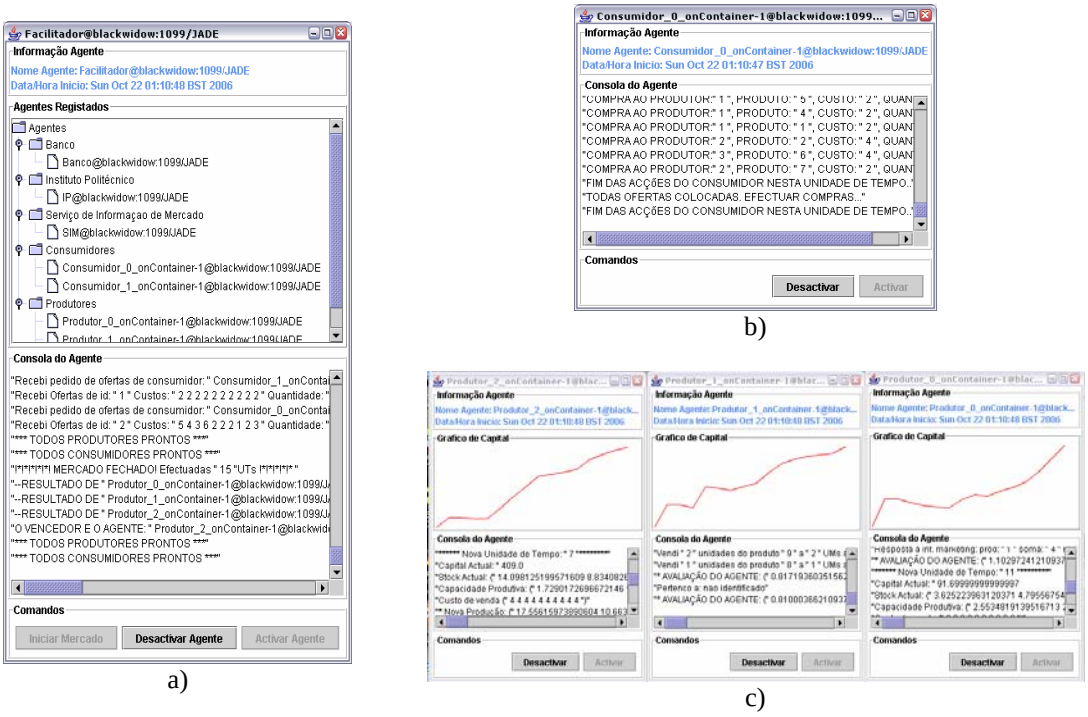

Figure 2 – Agents user interfaces (text in Portuguese). a) Facilitator agent b) Consumer agent c) Three Producer agents displaying graphically their capital gains over time.

The market can be configured in a global definitions file. This file contains information such as the duration of the market, in time units, number of different products, number of Consumers, minimum and maximum needs of products for each Consumer, interest rates applied by the Bank, learning and information costs and initial capital, stock, production capabilities and production costs. The market evolves discretely in time, being the Facilitator responsible for issuing the next time unit message for all agents, thus synchronizing all agents to the market's timeline. We can describe the market's evolution in the steps listed in table 1.

Table 1 – Market evolution.

Initialization	All agents register themselves with the Facilitator; Consumer post their needs to the Facilitator.
Market evolution **For each time unit *t***	Producers are alerted of product sells, adjust their strategies, can query the MIS agent for information, post offers and do marketing on their products, deposit/withdraw money on the Bank agent, and subscribe at School agent for learning;
	The MIS agent replies to the Producer agents with the information asked in t-1, if any;
	The School agent sends the learning grades to the Producer agents, if they have subscribed for learning in t-1;
	The Bank agent calculates the current time unit interest rate and applies it to the money deposited by Producer agents, alerting the Producer agents of the new balance;
	The Facilitator, besides re-routing messages, selects a restricted number of Producers that meet the Consumer needs. The probability of a Producer being selected increases with the marketing that he does on his products. If a Consumer bought a particular product from a Producer in t-1, then he is automatically selected in t for that product (fidelization);
	Consumers inform the Facilitator of their intention to buy products. They choose the Producers, from the list sent by the Facilitator, based on the best offered price;
End	Producer agents inform their results to the Facilitator; The Bank agent communicates the Producers account balances to the Facilitator; The Facilitator Agent checks results for fraud and declares a winner.

In figure 2, we present the user interfaces (UI) of the different types of Agents. All user interfaces show the agent's identifier, date of creation and a console that can be used to output information, reflecting their current internal mental state. The Facilitator agent UI additionally displays information of all agents registered with him, and the Producer agent UI displays a graph of the capital gains over time. The current implementation allows the market to be spanned across an arbitrary number of computers, with each machine running a JADE container that can itself contain several agents.

4. Conclusion

The students constructed the market platform while they were learning the technologies set needed to implement the project. This was the platform used by each group to integrate their intelligent producer agent. All the agents provided by the students were placed in the distributed environment and the platform run for a set of time units. In the final step the platform evaluates the economic performance of the Producer agents and decides for a winner. In all the process the students discuss with their course colleagues some of their ideas, and after the winner was discovered the strategies were shared and some discussions were promoted to learn what were the correct strategies that have conduced an agent to better performance.

The resulting platform demonstrated modelling capabilities of an artificial market. The students have developed diverse production agents with specific behaviours, where some of the agents had better success by using the diverse possibilities of decisions and actions over the market. The work of creation of the market and respective agents had created a pedagogical active environment. The diversity of knowledge and technology integration needed in a relatively young and active research area, created a pedagogical environment that we consider to be a near research project.

References

[1] S. Russel and P. Norving, Artificial Intelligence: A Modern Approach (2nd Edition) Prentice Hall, (2002).
[2] M. Wooldridge, Introduction To Multi-Agent Systems. John Wiley & Sons, (2002).
[3] H. Gamboa, N. V. Ribeiro, A. Holt, A. Fred, An e-learning Platform Concept, 2a International Conference on Information Technologies in Education: Chalanges 2001 (2001).
[4] T. Wulf, Constructivist approaches for teaching computer programming. In Proceedings of the 6th Conference on information Technology Education (Newark, NJ, USA, October 20 - 22, 2005). SIGITE '05. ACM Press, New York, NY (2005).
[5] J. Piaget, The Psychology of Intelligence. New York: Routledge. (1950).
[6] L. Vygotsky, Mind in society. Cambridge, MA: Harvard University Press (1978).
[7] B. Barron. Doing with understanding: Lessons from research on problem- and project-based learning. In Journal of the Learning Sciences. 7(3&4), 271-311. (1998).
[8] Nuno Valero Ribeiro, Hugo Gamboa. An evolutionary multi-agent system for simulating e-commerce markets. 1st Enterprise Networking Applications and Services Conference, EntNet2001, Atlanta USA.
[9] F. Bellifemine, G. Caire, A. Poggi,G. Rimassa, White Paper on Jade, EXP in search of innovation, 3(3) (2003).
[10] F. Bellifemine, A. Poggi, G. Rimassa. Developing multi agent systems with a FIPA-compliant agent framework. in Software - Practice & Experience, John Wiley & Sons, Ltd. vol no. 31 (2001).
[11] J. Searle, Speech Acts, Cambridge University Press (1969).
[12] E. Friedman-Hill, Jess in Action: Java Rule-Based Systems, Manning Publications (2003).
[13] Anand S. Rao, Michael P. Georgeff. Modeling Rational Agents within a BDI-Architecture. In Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)