MASFIT: Multi-Agent System for FIsh Trading

Guifre Cuni, Marc Esteva, Pere Garcia, Eloi Puertas, Carles Sierra and **Teresa Solchaga**[1]

Abstract. Traditional wholesale fresh fish markets carry out their sales by means of the Dutch auction protocol, with the buyers physically present in the market hall. In this paper we present the Multi-Agent System for FIsh Trading (MASFIT) which allows buyers to participate remotely in several fish markets simultaneously with the help of software agents, while maintaining the traditional auction procedures. The system includes all the necessary tools to create, customize and train buyer software agents.

1 Introduction

In the mediterranean Fresh fish has been traditionally sold through downward bidding auctions operating in different auction houses close to the harbors. There, fish is grouped into sets of boxes, called *lots*. These lots are auctioned following the Dutch protocol: price is progressively and quickly lowered -4 prices per second- until a buyer submits a bid or the price descent reaches a withdraw price. The buyer submitting the bid can decide to buy the complete lot or just some boxes. In this later case, the remaining boxes are auctioned at the next round. When the last box is sold, the auction is over.

Some fish auctions are adapting their selling methods to new technologies and most auctions are nowadays automatized by some specific *auction system*. Nonetheless, the presence of human buyers at the auction houses is still necessary. This imposes two main barriers. First, it restricts the potential buyers to those present in the auction house. Second, it makes the participation in several auctions simultaneously costly, as companies have to send a representative to each one. The elimination of such limitations would be very profitable for both buyers and sellers. Increasing the number of buyers makes the market more competitive and thus increases the buying price to the benefit of sellers. It also permits the participation of buyers without intermediaries saving costs to the buyers.

In this paper we show how agent technologies may be used to eliminate these limitations. We introduce the Multi-Agent System for FIsh Trading (MASFIT) [2] which allows buyers to remotely participate in several wholesale fish auctions simultaneously with the help of software agents, while maintaining the traditional auctions.

The participation of buyer agents in auctions is mediated by an electronic institution [8, 12, 4]. Electronic institutions fix the rules of the game within agent societies. They structure agents interactions, and establish what agents are permitted and forbidden to do. In fact, electronic institutions have been successfully used to model fish auctions [11, 8, 9]. Since there are already specific tools to develop electronic institutions [7, 5], they become specially appropiate for our purposes.

[1] IIIA-CSIC, Campus UAB, Bellaterra, Spain email: cuni, marc, pere, eloi, sierra, teresa@iiia.csic.es

[2] http://www.masfit.net

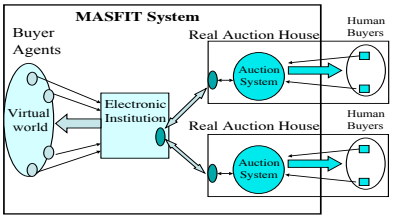

Figure 1. A federation of auction houses via an electronic institution.

The MASFIT's institution controls buyers' access to the auctions, provides them with information, and collects their bids during the auctions. To permit this, the auction system running at the auction houses has been connected to the developed institution. MASFIT interconnects multiple auction houses, and therefore it gives structure to a federation of auction houses. MASFIT guarantees equal conditions for both human buyers present at the auction and buyer agents.

The participation on simultaneous auctions is a complex decision making task [1]. Agents have to manage huge amounts of information -even uncertain information- and their reasoning and processing time must be short enough to react to changes. To support this complex design MASFIT also includes tools to create, customize, manage and train software buyer agents.

The paper is organized as follows. In Section 2 we identify the main requirements and objectives of the MASFIT system. Next, in Section 3 we introduce the specification of the institution. In Section 4 we outline the implementation of the system, describing the auction system, the implementation of the institution, and the agents' server. In Section 5 we present an example of how the system works. Finally, in Section 6 we detail the conclusions and related work.

2 Requirements and Objectives of the system

As already mentioned, the objective of the MASFIT system is to allow buyers to participate remotely in several simultaneous auctions by means of intelligent software agents (see Figure 1) . To do so, an electronic institution is connected to the auction systems of different auction houses (right part of Figure 1). In this way, buyer agents can participate simultaneously in multiple auction houses via the electronic institution (left part of Figure 1). The auction system has been extended in order to allow its connection to the institution. Furthermore, a specific protocol governs the messages that the auction system and the institution interchange.

The MASFIT system guarantees that the buyer agents have access to the same information, and have the same bidding opportunities as human buyers physically present at the auction house. Furthermore, the system does not alter the current operation of the auction houses.

In order to permit each auction house to continue having control over the buyers which are authorized to participate in their auctions,

human buyers must register in each auction house before this auction house authorizes buyer agents to participate on their behalf. As a consequence, buyer agents may not have access to all the federated auction houses. For this reason, we distinguish between the agents' admission to the institution, as the federation of auction houses, and their admission to the different auction houses.

Although it is not implicitly necessary for the system, we have developed the tools to permit human buyers to create, manage, and train their agents before sending them to actually buy.

3 MASFIT electronic institution

The purpose of this section is to present the specification of the electronic institution that we have defined to mediate buyer agents participation in the auction houses. Electronic institutions define a normative environment that shapes agent interactions at execution time. An institution defines a common ontology that allows agents to exchange knowledge, the roles that participating agents can play, the valid interactions that agents may have and the consequences of such interactions (a detailed description of electronic institutions can be found in [8], [12] or [4]). The MASFIT electronic institution has been specified using the ISLANDER editor [7], a specification and verification tool for electronic institutions.

3.1 Institution Roles

Agent actions within the context of an institution are associated to roles. Hence, each role defines a pattern of behavior within the institution, and each participating agent is required to play some of them. The roles that an agent is playing determine the actions that it can do. Furthermore, institutions distinguish between internal and external roles. Since an institution delegates their services and duties to internal roles, these can only be played by staff agents, that is, agents that belong to the institution.

In our particular case, there are the following internal roles: *buyer admitter* (BAD), *data base manager* (DBM), *auction house boss* (AH), *auction admitter* (AAD), *auction broker* (AB), *good register* (GR), and *remote control* (RC). These roles are in charge of controlling buyer agents access to the institution and to the different auctions, providing them with historical and current information about the auctions, and collecting their bids.

Complementary, there is only one external role, the *buyer* role, which is played by the agents coming to buy. We want to remark that buyer agents submit their bids through a staff agent playing the RC role to which they send orders for bidding. This permits buyer agents to concentrate on the general strategy (where to bid and at which price), while RCs are in charge of actually submitting the bids at the appropriate moment (when the descending price is equal or lower than the bid value detemined by the buyer agent).

3.2 Scenes

The activities in an electronic institution are the composition of multiple, distinct, and possibly concurrent, dialogic activities, each one involving different groups of agents playing different roles. For each activity, interactions between agents are articulated through agent group meetings, called *scenes*, that follow well-defined communication protocols. The protocol of each scene models the possible dialogic interactions between roles, and it can be multiply instantiated by different groups of agents.

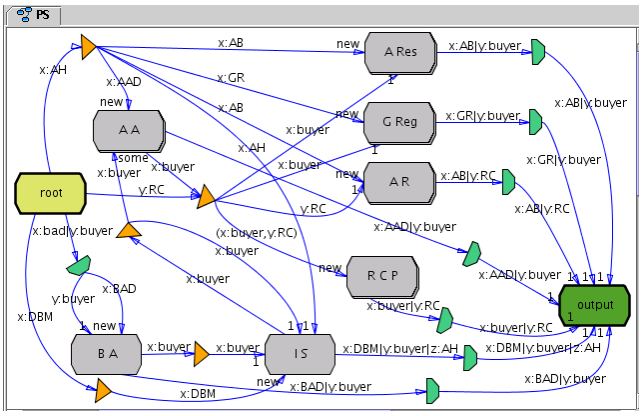

Figure 2. Specification of the MASFIT institution performative structure.

In the case of the MASFIT's institution we have: *buyer admission* (BA), where a *buyer admitter* controls buyer agents' access to the federation; *info-seeking* (IS), where buyer agents are informed about which auction houses are federated and can access historical information about them; *auction admission* (AA), where an *auction admitter* controls buyer agents' access to an auction house; *good registering* (GReg), where the *good register* agent informs the buyers of the lots registered within an auction house; the *auction room* (AR), where an *auction broker* mediates the participation of RCs (on behalf of buyer agents) in an auction house; *auction results* (ARes), where buyer agents are informed about the results of an auction; and the *RCprogramming* (RCP), where a buyer agent programs -through dialogue- a RC to participate in an auction.

3.3 Performative Structure

While a scene models a particular multi-agent dialogic activity, more complex activities can be specified by establishing networks of scenes, the so-called *performative structures*. These define how agents can legally move among different activities depending on their role. Furthermore, a performative structure defines when new *conversations* (scene executions) can be started, and if a scene can be multiply executed at run time.

In order to connect scenes we use different types of *transitions*. Thus, a performative structure must be regarded as a graph whose nodes are both scenes and transitions, linked by different types of directed arcs. The type of transition allows to express choice points (*Or* transitions) for agents to choose which target scenes to enter, or synchronization/parallelisation points (*And* transitions) that force agents to synchronize before progressing to different scenes in parallel. Notice though that scenes and transitions are connected by directed arcs whose labels determine which agents, depending on their roles, can progress from scenes to transitions, and from transitions to scenes. Since the same scene specification can be multiply executed, the arcs connecting transitions to scenes define whether an agent following the arc can join a *new*, *one*, *some* or *all* execution(s) of the target scene. Finally, each performative structure is requiered to have an initial and a final scene which represent the entry and exit points of the institution.

Figure 2 depicts the performative structure of MASFIT's institution. Rectangles represent scenes, triangles *and* transitions (synchronization and parallelisation points), semi-circles *or* transitions (choose points), and the connections determine how agents, depending on their role can move among them (e.g., x:buyer over a connection means that an agent playing role 'buyer' can jump over -

variable x will get bounded to the name of the agent in doing so). Notice that besides the scenes described above, the *root* and *output* scenes which represent the initial and final scenes are also represented. Since MASFIT is a federation of auction houses, there are some scenes which are devoted to common activities (federation level), while there are some which are devoted to the activities of a concrete auction house (auction level).

The scenes at the federation level are: *root*, *output*, *buyer admission* and *info-seeking* scenes. There is one single execution of these scenes when the institution is running. The rest of the scenes mediate buyer agents' participation in particular auctions. There is one execution of each of these scenes per auction house connected to the federation, except for the *RCProgramming* scene. In this case, there is one execution of the *RCProgramming* scene per buyer agent admitted within an auction house.

The connection between an auction house and the auction system of a real auction house is managed, from the point of view of the institution, by an agent that enters the institution to play the *auction house boss* role. Once in the *root* scene, the agent has only one path to follow (see Figure 2) which provokes new executions of the following scenes: *auction admission*, *good registering*, *auction results* and *auction room*. Observe that the agent changes its role to play the *buyer admitter role* in the *buyer admission* scene, the *good register* role in the *good registering* scene, and the *auction broker* role in the *auction results* and *auction room* scenes. Furthermore, the agent also goes to the *info-seeking* scene to inform buyer agents that a *new* auction house has been connected to the federation.

Buyer agents must go first to the *buyer admission* scene where the buyer admitter controls their access to the federation. If they are admitted, they can move to the *info-seeking* scene where they can request historical information to the *DBManager*. Moreover, they are also informed about the auction houses connected to the federation. From the *info-seeking* scene, buyer agents can try to enter the different auction houses by moving to the corresponding *auction admission* scenes. In this case, if they are admitted they must coordinate with a newly created RC agent prior to moving to the rest of the scenes of that auction house. Concretely, after being synchronized, the buyer agent goes to the *good registering* and *auction results* scenes, the RC goes to the *auction room* scene, and they go together to a newly created execution of the *RCprogramming* scene. Buyer agents are informed, in the *good registering* scene, as soon as lots are registered in the auction house, and they send the orders to bid to their RCs in the *RCProgramming scenes*. Notice that a buyer agent can receive information from several *good registering* scenes and manage different RCs simultaneously.

4 Implementation

The implementation of the MASFIT system contains three main software components (see Figure 3): the auction system, the institution, and the agents' server. The system also contains a data base, DB MASFIT, which contains information about the auctions and about the buyer agents. Information related to past auctions — sales, quantities, prices, ships, etc— is public, while information about current auctions can be accessed only by authorized agents. Information related to each agent is private to its owner.

4.1 Auction system

The Auction system is the system in charge of controlling the processes occurring within a real auction house. It is composed of several computers connected by a fast Ethernet network. A staff member of the auction house is in charge of registering the lots into the system. The system is also in charge of monitoring the auctioning of the registered lots after receiving an order from the auctioneer. Buyers physically present at the auction can see this information through a big display placed on the market hall. Furthermore, they can submit their bids using a remote control provided by the auction house.

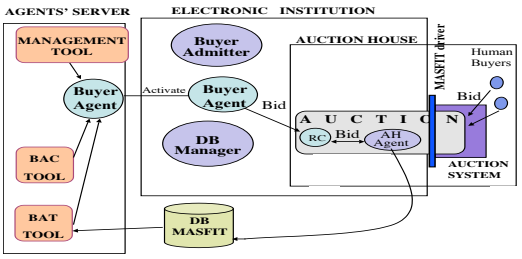

Figure 3. Graphical Scheme of the MASFIT system.

The main new module added to the auction system is the MASFIT Driver which is in charge of the communications with the electronic institution (see Figure 3). It passes the information about the events occurring at the auction house which are relevant for the buyer agents to the institution. Also, it receives information about the buyer agents taking part in the auction and their bids.

4.2 Implementation of the electronic institution

When an institution is executed, agents have their interactions mediated by an infrastructure, called AMELI [5]. The infrastructure facilitates agent participation and communication, within the institution, while enforcing the institutional rules encoded in the specification. At this aim, it loads institution specifications as generated by the ISLANDER editor. AMELIE permits, among many others features, the distributed execution of agents. This is important in our case as the agents must be executed at different places.

As we pointed out in Section 3.1 an electronic institution delegates its services and duties to the staff agents playing internal roles. Hence, it is necessary to completely implement these agents before allowing external agents to enter the institution.

In our case we have developed four types of staff agents that play the internal roles (see Figure 3), namely: *buyer admitter*, *data base manager*, RC, and *auction house* (AH). All these agents have been developed in JAVA. In the case of the *buyer admitter* agent and *data base manager* agent, there is one agent of each type running per institution execution.

Each AH agent manages the connection of an auction system to the institution. It is connected via socket with the MASFIT driver. AH agents enter the institution with the AH role and later on (as explained in Section 3.3), they also play the *auction admitter*, *auction broker* and *good register* roles. There is one AH agent per auction system connected to the institution. Finally, the RC agents play the RC role and participate in an auction on behalf of a buyer agent. There is one RC agent per buyer admitted in an auction house.

Participating agents are distributed among different places, as shown in Figure 3. Each AH agent and all the RC agents participating in the auction house managed by this AH agent, are executed at the corresponding auction house computers, while buyer agents, the *buyer admitter* agent and the *data base manager* agent, are executed

at a service providers place. This distribution is done because the system requires a fast communication between the auction system and the RCs, which is important to guarantee that the RCs receive the offers and submit their bids on time.

4.3 Agents' server

In the current version the Agents' server is the connection point between human users and the MASFIT system. It is developed to help human users to create and control their own agents without requiring them to be experts on agent technology. For this purpose we have developed three tools: Management, Buyer Agent Creative (BAC) and Buyer Agent Trainer (BAT).

The Management tool permits a user to create an agent skeleton. This skeleton contains the necessary code to navigate through the institution and can be customized with buying requests and strategies that are defined using the BAC tool. Complementary, the BAT tool permits users to test their agents in simulation mode in order to tune the parameters of the buying strategies before sending them to compete in real auctions. Finally, we want to mention that users can monitor the bidding of their agents, and that each agent also makes up a report of the auction results that is stored in the DB.

4.3.1 Management tool

The Management tools allow users to create or terminate an agent. Users can check which of their agents are active and participating at the auctions and which are not. They can activate agents to actually buy, to be trained, or deactivate them. Moreover, users can program a delayed activation so that the MASFIT system activates them in due time. Besides, users can check any information related to the agent – buying lists, strategies, etc. When an agent leaves the MASFIT institution, its information state is stored on the DB MASFIT so that a report can be sent to its owner on demand.

4.3.2 Buyer Agent Creative (BAC) tool

The BAC tool helps buyers to customize their agents so that they can buy according to their owner's preferences. The customization is done by means of buying lists, strategies and logistics. They are predefined and depend on several parameters that must be instantiated by users. Each instantiation corresponds to a different request and a different strategy.

A 'buying request' is a list of products to be bought by the agent. Each product is determined by several issues: name, quality, auction house, quantity, and bidding strategy. Some of these issues can be imprecisely defined with ranges, lists or ratios.

When an agent receives the registering of a lot, it checks if the product is required in its buying request. If so, the agent determines a price and sends it to the corresponding RC.

We have developed two kind of parametric families of strategies. The first one follows the tendency of the market and it makes the agents behave very reactive to any change of the prices. The second family predict expected prices based on historical auction data using Case Based Reasoning strategies. It makes the agents be more proactive than with the first strategy. Both of them take into account logistics considerations, like transport costs, in order to calculate the bids in the different auctions, i.e., the farther the auction, the lower the bid.

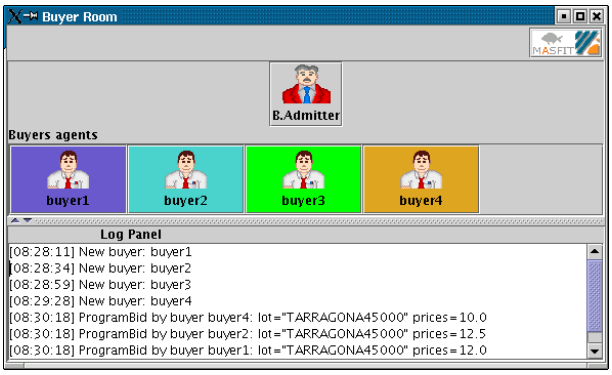

Figure 4. MASFIT institution

4.3.3 Buyer Agent Training (BAT) tool

The BAT tool allows users to test and train buyer agents before sending them to compete in real auctions. This training permits to tune the parameters of the strategies and to check the behavior of the agent within the MASFIT system.

We have implemented two kinds of simulation modes: a) Off line: this mode allows the simulation of past auctions, stored in a DB, to test the agent's bidding mechanisms. It permits the reduction of training time, though agent's reaction time cannot be tested. b) On line: this mode permits to test the reaction speed of the agent with respect to changes in the environment. The only difference between the agent that is being trained and the rest of the buyer agents is that the former bids are ignored by the AH agent.

5 Example, MASFIT at work

The purpose of this section is to illustrate how the MASFIT system works. For this purpose we have monitorized a system execution in which there are two auction houses connected to the federation, identified as *Tarragona* and *Vilanova* [3].

Figure 4 shows which agents have been admitted in the institution, namely: *buyer1*, *buyer2*, *buyer3* and *buyer4*. In the lower part of the window some of the events occurring within the federation are shown. Concretely, we can observe that $buyer1$, $buyer2$, and $buyer4$ have sent an order to their respective RCs to bid for lot $TARRAGONA45000$, at 12, 12.5 and 10 euros respectively.

Complementary, Figure 5 shows a snapshot of the participation of agents in *Tarragona*'s auction house. It shows which RCs are participating at the auction room scene and information about the auction rounds. There are three RCs participating in the auction room, namely: *buyer1RCTarragona* (on behalf of $buyer1$), *buyer2RCTarragona* (on behalf of $buyer2$), and *buyer4RCTarragona* (on behalf of $buyer4$). Observe that $buyer3$ is not participating in this auction house. The messages exchanged during the execution of the auction room – corresponding to the auction rounds– are shown in the log panel. In this example, they correspond to a round in which the lot $TARRAGONA45000$ is auctioned. We can observe that the auctioner sends lower offers for the product until the $buyer2RCTarragona$ agent submits a bid at 12.40 euros. Note that this is the first offer lower or equal than the programmed bid (12.50 euros).

Finally, in Figure 6 we can see some information about $buyer2$'s participation in the system. The figure shows, in the upper part, the

[3] In fact, these are the two auction houses that participated in the pilot development funded by the European Union

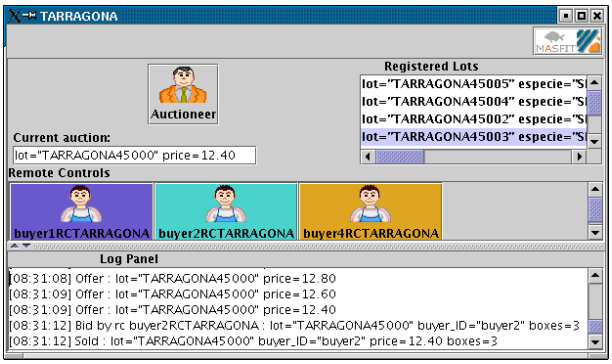

Figure 5. Auction at MASFIT

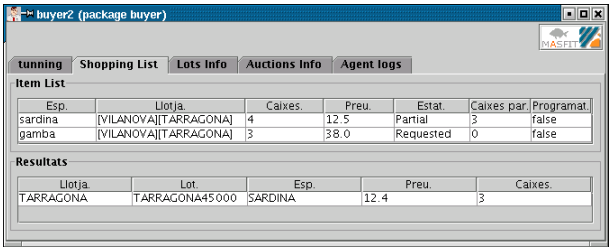

Figure 6. Agent at MASFIT

products it is interested in (Item List), while the lower part shows the purchases done by the agent (Resultats). For each row on the Item list we can see the fish species (Esp.), the auction houses where the product can be bought (Llotja), the price to buy the product (Preu), the number of boxes to buy (Caixes), if the total amount of boxes of the product has been bought (Estat), the number of bought boxes (Caixes p.) and whether there is any pending order to a RC related to that item (Programat). For instance, the first row in the Item list on this figure means that the buyer is interested in buying four boxes of "sardina" (sardine) at 12.5 euros at either the auction house of Vilanova or Tarragona. The order is partially complete, as the buyer has already bought three boxes, and there are no pending orders sent to a RC related to that item. On the lower part, the purchases of the agent are shown in detail. Concretely, we can see that the three boxes of sardine have been bought in Tarragona at 12.40 euros corresponding to the lot Tarragona45000.

6 Conclusions and Related Work

We have presented a multi agent system that allows intelligent autonomous agents to participate remotely in simultaneous auctions on behalf of their owners. Using this system, buyers can participate in multiple auctions without intermediaries while better fulfilling their buying preferences. Besides, the market becomes more competitive and therefore, more profitable for both buyers and sellers.

Several agent-based virtual market places have been developed in the last years: Kasbah [2], ZEUS [3], AuctionBot [10], MAGNET [6] or eMediator [14]. AuctionBot [10] and MAGNET [6] allow a software trader to directly connect to an auction server, while MASFIT obliges every buyer to have his participation mediated by a special type of facilitator provided by the auction house. Moreover, MASFIT is prepared to support agents created with different tools than the ones provided by the Agents' server —as emediator, AuctionBot and ZEUS—, and it also allows agent designers to include their own customized strategies into the agents, as Zeus and Kasbah.

However, the MASFIT system is a real, flexible and robust agent-based auction house where software and human buyers can actually trade. Its biggest advantage is that it is devised as an open agent-mediated electronic institution, having its organizational structure inspired in traditional fish auction houses. The MASFIT system naturally extends Fishmarket [8, 13] in two directions: (i) is distributed to permit several auctions to run simultaneously, which allows for the addition of new auctions on the fly without modifying the rest of the system, and (ii) the specification of the institution permits buyer agents to participate at the traditional auctions following their specific protocol and rules. Moreover, it integrates the virtual and the real auction houses, so that agents can compete against human buyers present at the auctions. Note that none of the previous mentioned markets can assume both characteristics at a time.

The system has been tested using two auction houses at a time. The performance of two buying strategies – fixed price and simple mimetic of the market– has been assessed across many user preferences, through the simulation environment. Some users have checked whether the bids submitted by the buyer agents are reasonable and whether the outcome of the buying process fits into their expectations. Furthermore, this permits to test buyer agents performance competing with human buyers. The system is currently entering commercial exploitation.

Acknowledgments

This work has been developed under the MASFIT project (IST-2000-28221 EUTIST-AMI) in collaboration with AUTEC, "Automatismes Electrònics i Control SL". We want to acknowledge the support provided by the UDT, J. L. Arcos and J.A. Rodrìguez, from the IIIA.

REFERENCES

[1] A.Byde and N. Jennings C. Preist, 'Decision procedures for multiple auctions', in *Proceedings of the AAMAS'02*, pp. 613–620, (2002).
[2] A.Chavez and P.Maes, 'Kasbah: An agent marketplace for buying and selling goods', in *Proceedings of the PAAM'96*, pp. 75–90, (1996).
[3] J.C. Collins and L.C. Lee, 'Building electronic marketplaces with the zeus agent tool-kit', in *Agent Mediated Electronic Commerce*, LNAI 1571, 1–24, Springer-Verlag, (1998).
[4] M. Esteva, *Electronic Institutions: from specification to development*, number 19 in IIIA Phd Monograph, 2003.
[5] Marc Esteva, Bruno Rosell, Juan A. Rodríguez-Aguilar, and Josep Ll. Arcos, 'Ameli: An agent-based middleware for electronic institutions', in *Proceedings of the AAMAS'04*, (2004).
[6] J.Collins, B. Youngdahl, S.Jamison, B.Mobasher, and M.Gini, 'A market architecture for multi-agent contracting', in *Proceedings of the AGENTS'98*, pp. 285–292, (1998).
[7] M.Esteva, D.de la Cruz, and C.Sierra, 'Islander: an electronic institutions editor', in *Proceedings of the AAMAS'02*, (2002).
[8] P. Noriega, *Agent-Mediated Auctions: The Fishmarket Metaphor*, number 8 in IIIA Phd Monograph, 1997.
[9] P. Noriega and C. Sierra, 'Auctions and multiagent systems', in *Intelligent Information Agents*, ed., Klusch M., pp. 153 –175. Springer-Verlag, (1999).
[10] P.R.Wurman, M.P.Wellman, and W.E.Walsh, 'The michigan internet auctionbot: A configurable auction server for human and software agents', in *Proceedings of the AGENTS'98*, pp. 301–308, (1998).
[11] J.A. Rodriguez, P. Noriega, C. Sierra, and J. Padget, 'Fm96.5. a java-based electronic auction house', in *Proceedings of the PAAM'97*, pp. 207–224, (1997).
[12] J. A. Rodríguez-Aguilar, *On the Design and Construction of Agent-mediated Electronic Institutions*, number 14 in IIIA Phd Monograph, 2001.
[13] J. A. Rodríguez-Aguilar, F. J. Martín, P. Noriega, P. Garcia, and C. Sierra, 'Towards a test-bed for trading agents in electronic auction markets', *AI Communications*, **11**(1), 5–19, (1998).
[14] Tuomas Sandholm, 'emediator: A next generation electronic commerce server', *Computational Intelligence*, **18**(4), 656–676, (2002). Special Issue on Agent Technology for Electronic Commerce.