Towards a Test-bed for Trading Agents in Electronic Auction Markets

Juan A. Rodríguez-Aguilar, Francisco J. Martín, Pablo Noriega, Pere Garcia and Carles Sierra
Artificial Intelligence Research Institute, IIIA
Spanish Council for Scientific Research, CSIC
08193 Bellaterra, Barcelona, Spain
{*jar,martin,pablo,pere,sierra*}*@iiia.csic.es*
http://www.iiia.csic.es

We present a framework for defining trading scenarios based on fish market auctions. In these scenarios, trading (buyer and seller) heterogeneous (human and software) agents of arbitrary complexity participate in auctions under a collection of standardized market conditions and are evaluated against their actual market performance. We argue that such competitive situations constitute convenient problem domains in which to study issues related with agent architectures in general and agent-based trading strategies in particular.

The proposed framework, conceived and implemented as an extension of FM96.5 (a Java-based version of the *Fishmarket* auction house), constitutes a testbed for trading agents in auction tournament environments, FM97.6.

Finally, we illustrate how to generate tournaments with the aid of our testbed by defining and running a very simple tournament involving a set of rudimentary buyer agents.

Keywords: agent test-beds, multi-agent systems, agent-mediated institutions, auctions, e-commerce

1. Introduction

Auctions are an attractive domain of interest for AI researchers in at least two areas of activity. On the one hand, we observe that the proliferation of on-line auctions in the Internet —such as Auctionline[1], Onsale[2], InterAUCTION[3], eBay[4] and many others— has established auctioning as a main-stream form of electronic commerce. Thus, agent-mediated auctions appear as a convenient mechanism for automated trading, due mainly to the simplicity of their conventions for interaction when multi-party negotiations are involved, but also to the fact that on-line auctions may successfully reduce storage, delivery or clearing house costs in many markets. This popularity has spawned AI research and development in auction servers[31, 23] as well as in trading agents and heuristics[6, 17]. On the other hand, auctions are not only employed in web-based trading, but also as one of the most prevalent coordination mechanisms for agent-mediated resource allocation problems (f.i. energy management[33, 32], climate control[8], flow problems[29]). In this paper we present ideas and tools that are relevant, mainly to the first type of AI activity. We discuss how an agent-mediated electronic auction house can be turned into a test-bed for trading agents.

From the point of view of multi-agent interactions, auction-based trading is deceivingly simple. Trading within an auction house demands from buyers merely to decide on an appropriate price on which to bid, and from sellers, essentially only to choose a moment when to submit their goods. But those decisions—if rational—should profit from whatever information may be available in the market: participating traders, available goods and their expected re-sale value, historical experience on prices and participants' behaviour, etc. However, richness of information is not the only source of complexity in this domain. The actual conditions for deliberation are not only constantly changing and highly uncertain—new goods become available, buyers come and leave, prices keep on changing; no one really knows for sure what utility functions other agents have, nor what profits might be accrued—but on top of all that, deliberations are significantly time-bounded. Bidding times are constrained by the bidding protocol which in the case of DBP[5] auctions—like the traditional fish market[6]—proceeds at frenetic speeds.

[1] http://www.auctionline.com
[2] http://www.onsale.com
[3] http://www.interauction.com
[4] http://www.eBay.com

[5] *Downward bidding protocol*
[6] We will use the expression *fish market* to refer to the actual, real-

ISSN 0921-7126 / $8.00 © , IOS Press

Consequently, if a trading agent intends to behave aptly in this context, the agent's decision-making process may be quite elaborate. It could involve procedural information—when to bid, how to withdraw—, reasoning about individual needs and goals, information and reasoning about supply and demand factors—which may involve other agent's needs and goals—and assessment of its own and rivals' performance expectations—which in turn may require knowledge or reasoning about the external conditions that might affect the auction.

Evidently, many approaches can be taken to deal with this decision-making process. From highly analytical Game-Theoretic ones, to mostly heuristic ones. From very simple reactive traders, to deliberative agents of great plasticity. Moreover, it should be noted that the type of decision-making process involved in auctions is inherent in other common forms of trading and negotiation, and specifically in those that are being identified as likely applications of multi-agent systems[7, 33, 32, 8, 29]. However, it is not really obvious which of the many possible approaches for automatic trading strategies' modelling are better, or under what conditions. We do not intend to present any such evidence in this paper, but instead to sketch a blueprint for the production, assessment and perhaps communication of such evidence. Actually, this paper will focus on the description of a testbed—which permits the definition, activation and evaluation of experimental trading scenarios that we shall refer to as *tournaments*—and will illustrate how it can be used.

As the starting platform for that testbed, we use a Java-based electronic auction house inspired by the traditional fish market, FM96.5 [23]. This provides the framework wherein agent designers can perform *controlled experimentation* in such a way that a multitude of experimental market scenarios of varying degrees of realism and complexity can be specified, activated, and recorded; and trading agents compared, tuned and evaluated.

This exercise will ideally serve to show how one can conveniently devise experimental conditions to test specific features in agent architectures. How, for example, any-time strategies and off-line deliberation may be put to work coherently in a practical way. Or how and when reasoning about other agent's intentions and goals may be profitably turned into a trading advantage. Or how to couple a learning device with a human trader to discover market-dependent heuristics or with a trading agent so as to *watch* it perform the task. Or how to apply data mining techniques to discover patterns of behaviour of rival agents.

We trust this proposal may motivate AI theorists and developers to look into auctions as a challenging problem domain where they can investigate and put their creations through a strenuous test, but we realize that our proposed framework can serve other purposes as well. For instance, these tools may also interest economists who would like to examine issues of *Mechanism Design* under flexible theoretical and experimental conditions ([27]), since our trading scenarios may be seen as pseudo-markets with different degrees of indetermination. Moreover, financial regulatory bodies, and market developers may take advantage of this kind of framework for the design and experimentation with electronic market places, both in terms of those characteristics that new Internet-based trading institutions should have, but also in terms of features and components new market practices may be requiring to facilitate agent-based trading that is practical, reliable and safe.

In Section 2, we outline the essential notions of how an auction house works, how the *Fishmarket* model was implemented to model auctions and how it has been adapted to deal with tournament scenarios. In Section 3 we introduce the concept of tournament descriptor, and in Section 4 we illustrate how to instantiate such tournament descriptor in order to characterize particular tournament scenarios. Finally, Section 5 discusses related work and argues about present and future work.

2. An Auction Tournament Environment

Following [15], the fish market can be described as a place where several *scenes* take place simultaneously, at different locations, but with some causal continuity. The principal scene is the auction itself, in which buyers bid for boxes of fish that are presented by an auctioneer who calls prices in descending order, the *downward bidding protocol*. However, before those boxes of fish may be sold, fishermen have to deliver the fish to the fish market, at the *sellers' admission scene*, and buyers need to register for the market, at the *buyers'*

world, human-based trading institution, and *Fishmarket* to denote the artificial, formal, multi-agent counterpart. Hence, FM96.5 refers to a particular implementation of the *Fishmarket* model of the fish market. Notice that we use the term *institution* in the sense proposed by North [16] as a "... *set of artificial constraints that articulate agent interactions*".

admission scene. Likewise, once a box of fish is sold, the buyer should take it away by passing through a *buyers' settlements scene*, while sellers may collect their payments at the *sellers' settlements scene* once their lot has been sold.

In [23, 15] we present FM96.5, an electronic auction house based on the traditional fish market metaphor. In a highly mimetic way, the workings of FM96.5 also involve the concurrency of several scenes governed by the market intermediaries identified in the *Fishmarket*. Therefore, seller agents register their goods with a seller admitter agent, and can get their earnings (from a seller manager) once the auctioneer has sold these goods in the auction room. Buyers, on the other hand, register with a buyer admitter, and bid for goods which they pay through a credit line that is set up and updated with a seller manager. Buyer and seller agents can trade goods as long as they comply with the *Fishmarket institutional* conventions.

The Fishmarket is an *institution* (cf. North's [16]) that establishes and enforces explicit conventions of three types:

1. *Ontological and Communicational* conventions that determine the types of goods that are exchanged, the pricing and bidding elements and, in general, the content and meaning of those messages that can and may be uttered within an auction house to perform an auction.
2. *Social* conventions that establish the way interactions among participants are to take place. That is, the process through which goods are registered, what is needed for a buyer to be admitted in an auction, how bidding proceeds, or how dues are taken care of.
3. *Individual rules of behaviour*, which establish the duties and rights of participants, and make explicit the obligations and commitments they would incur by participating in an auction.

Following [15], the Fishmarket *social conventions* can be represented in terms of a *Performative Structure*, $\mathcal{PS}_{FM}$ that states the causal and temporal relationship among the Fishmarket *scenes*, as depicted by Fig. 1. Each scene in turn, is characterized by a *Scene Protocol* that indicates how illocutions among participating agents are to be exchanged. For example, Fig. 2 shows the complete Bidding Rounds scene of the Fishmarket (including the bidding round proper, plus preparation and closing), and Table 1 the illocutions used in that diagram[7].

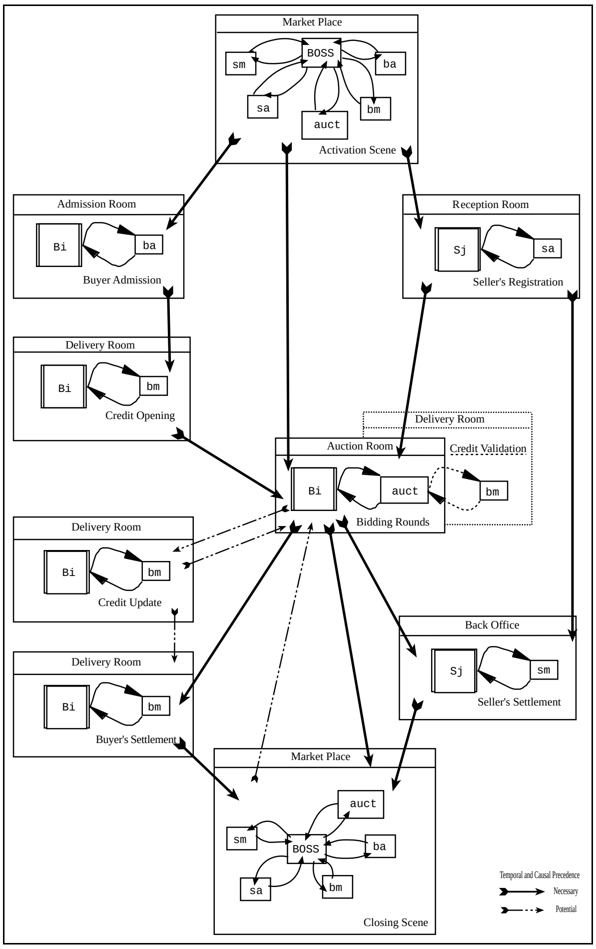

Fig. 1. Fishmarket Superficial Performative Structure

The conditions for generation and interpretation of those illocutions make reference to the "market commitments". Commitments that either have to prevail before an illocution can be uttered by an auction participant, or are affected by a *legitimate* utterance of an illocution by an auction participant.

[7] In Fig. 2, circles represent states (of "commitments"), boxes, other scenes. Arcs are labelled by (abridged) illocutions (see Table 1). Arrows indicate precedence, dotted lines indicate an optional illocutory move.
Note that each illocution in tables 1 and 2 has the form $\iota(\alpha, \Gamma : \varphi; \tau)$, where α is the agent that utters the illocution, Γ the set of agents that receive the illocution, φ the propositional content and τ the time of utterance (in particular, t_{now} represents the moment when the illocution is intended to be uttered). Whenever possible, time is omitted here for simplicity.

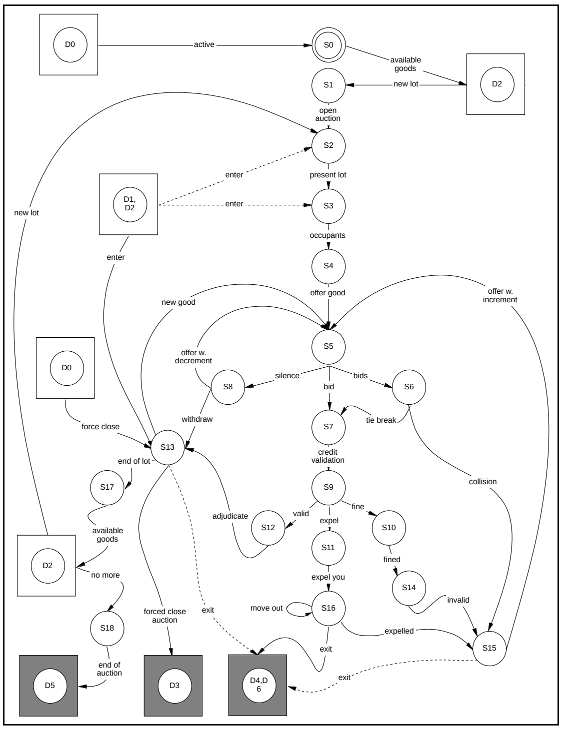

Fig. 2. Fishmarket Bidding Rounds Scene

$$\begin{array}{l}
offer(auct, all : tosell(g, p)) \\
assert(b, auct : bid) \\
request(auct, bm : credit_status(b, p_t)) \\
assert(bm, auct : valid(b)) \\
assert(bm, auct : fined(b, fine)) \\
request(bm, auct : expel(b)) \\
declare(auct, all \cup \{bm\} : sold(g, b, p, t_\omega)) \\
declare(auct, \{b, bm, ba\} : expelled(b)) \\
command(auct, b, outto(DR)) \\
declare(auct, all : expelled(b)) \\
declare(auct, b : fined(b, \kappa)) \\
declare(auct, all : invalidbid) \\
declare(auct, all : collision(g, \bar{b}, p)) \\
declare(auct, all : tiebreak(\bar{b}, b)) \\
declare(auct, all : endoflot) \\
request(b, auct : exitto(DR)) \\
request(s, auct : exitto(BO))
\end{array}$$

Table 1

Illocutions used in a Fishmarket bidding round.

Illocution generation and interpretation conventions are coded into the Fishmarket *individual rules of behaviour*, $\mathcal{BR}_{FM}$, as exemplified in Table 2[8].

Market Commitments in the Fishmarket can be thought as information structures, such as the auction catalogue or the buyer and seller accounts that "interpret" the *terms* that appear in the illocutions.

Ontological and Communicational conventions are formalized as the Fishmarket *dialogical framework*[24, 15], $\mathcal{DF}_{FM}$, which contains those entities that constitute the "auction world" in the Fishmarket; that is, everything that is referred to in a legitimate illocution or rule of behaviour. This can be made explicit in the following way:

Definition 2.1. The *Dialogical Framework* of the Fishmarket is the structure $\mathcal{DF}_{FM} = \langle Agents, Roles_{FM}, \mathcal{SR}_{FM}, Loc_{FM}, L_{FM}, CL_{FM}, ML_{FM}, T\rangle$, where

1. $Agents$ is a set of agent identifiers.
2. $Roles_{FM}$={Boss, Auctioneer, Buyer Admitter, Buyer Manager, Buyer, Seller Admitter, Seller Manager, Seller},
3. $\mathcal{SR}_{FM} = \{\mathbf{authority_{FM}} = \{(\mathbf{Boss}, \mathbf{x}) \mid \mathbf{x} \in \mathbf{Roles_{FM}} - \{\mathbf{Boss}\}\}$, Social relationships in Fishmarket (Boss has authority over all of the participants in an auction).
4. $Loc_{FM} = \{AH, RR, AR, BO, DR, M\}$, the Fishmarket (virtual) locations.
5. L_{FM} the "object language" of Fishmarket. Its signature includes all symbols used to represent the "propositional content" of legitimate illocutions: $\{\{\Delta_{rounds}, credit, bundle, \ldots\}, \ldots$ $\{Admittance, Tosell, \ldots\}, \ldots p_0\}$,
6. CL_{FM} the communication language needed to express legitimate illocutions in Fishmarket (includes $\{$*request, offer, accept, declare* $, \ldots\}$, as part of its signature).
7. ML_{FM}, the metalanguage used to define rules and diagrams (it includes all other languages).
8. T is a model of time (f.i., discrete, acyclic, forward-branching order).

This agent-mediated abstract institution, $\mathcal{FM} = \langle \mathcal{DF}_{FM}, \mathcal{PS}_{FM}, \mathcal{BR}_{FM}\rangle$ has FM96.5 as its computational realization. In it, all those conventions that affect buyers and sellers have been coded into what we call an *institutor* which constitutes the sole and exclusive means through which a trader agent—be it a software agent or a human trader—interacts with the market institution. An institutor gives a permanent identity

[8]Fishmarket rules involve terms such as: SG and UG (the lists of sold and unsold goods), $Pend_{auct}$ (the pending tasks of the auctioneer), $bundle(b)$ (all the purchases of buyer b), Δ_{rounds} (waiting period between two bidding rounds), etc. They may also include standard programming language constructs such as $WAIT$ and $REST$ that refer to the intended computational realizations.

Rule 1. $(\mathbf{adjudicate_{auct}})$
IF $assert(bm, auct : valid(b), t)$
$THEN$ $credit_{t_{now}}(b) := credit_t(b) - p_t(g)$
$buyer(g) := b$
$bundle(b) := bundle(b) \cup \{g\}$
$t_\omega(g) := t$
$SG := APPEND(SG; g)$
$UG := REST(UG)$
$Pend_{auct} := UG)$
$declare(auct, all :$
$sold(g, buyer(g), p_\omega(g), t_\omega(g)); t_{now})$

Rule 2. $(\mathbf{newgood_{auct}})$
IF $(declare(auct, all :$
$sold(g, buyer(g), p_\omega(g), t_\omega(g)); t))$
AND $(UG \neq \emptyset)$
$THEN$ $g := FIRST(UG)$
$p(g) := p_0(g)$
$WAIT(t_{now} \geq t + \Delta_{rounds})$
$offer(auct, all : tosell(g, p(g)); t_{now})$

Rule 3. $(\mathbf{newgood'_{auct}})$
IF $declare(auct, all :$
$sold(g, buyer(g), p_\omega(g), t_\omega(g)); t)$
AND $UG = \emptyset$
$THEN$ $request(auct, sa : moregoods; t_{now})$

Table 2
Adjudication Rules in Fishmarket

to the trader and enforces an *interaction protocol* that establishes what illocutions can be uttered by whom and when —and consequently what their language and content, sequencing and effects may be[9]. In FM96.5 we decided to implement institutors as a particular instantiation of the so-called market interagents introduced by Martin et al. in [12].

In order to obtain an auction tournament environment, more functionality has been added to FM96.5 to turn it into a testbed, FM97.6. This must be regarded as a *domain-specific* environment that models and simulates an electronic auction house. Nonetheless, notice that the resulting multi-agent testbed is *realistic* since it has been grown out of a complex real-world application, FM96.5.

Being an extension of FM96.5, FM97.6 inherits the mechanism of interaction between buyer agents and the market. This ensures that our testbed shows a crisp distinction between agents and the simulated world. Furthermore, the use of institutors or market interagents permits to consider FM97.6 as an *architecturally neutral* environment since no particular agent architecture (or language) is assumed or provided. Analogously, other test-beds such as Tile-world[20], Tæms[18], and Mice[13] have also opted for remaining architecturally neutral, whereas test-beds like Mace[18], Phoenix[4], DVMT[18], Archon[18], or CooperA[18] provide a suite of development facilities for building agents.

As to the systematization of our experiments, the complete parametrizability of FM97.6 allows for the generation of different market scenarios. This capability of *scenario generation* appears as a fundamental feature of any multi-agent testbed if it intends to guarantee the *repeatability* of the experiments to be conducted. Concretely, the customizability of FM97.6 allows for the specification, and subsequent activation, of a large variety of market scenarios: from simple toy scenarios to complex real-world scenarios, from carefully constructed scenarios that highlight certain problems to randomly generated scenarios useful for testing buyer agents' average performance[10]. Fig. 3 displays a snapshot of FM97.6 Tournament Definition Panel, the tool utilised by tournament designers to construct tournament scenarios. Observe that most DAI[11] test-beds (f.i. Tileworld, Phoenix, DVMT, TÆMS) also support repeatability.

Finally, there is the matter of evaluating a buyer agent's performance. FM97.6 keeps track of all illocutions taking place during an auction, so that a whole auction can be audited step-by-step, and the evolving performance of all the agents involved in a tournament can be traced, calculated, and analyzed. Fig. 4 displays a snapshot of FM97.6 Trace Tool, the component of the testbed which allows to trace the behaviour of all agents within the market, and follow the progress of the participants in tournaments.

[9] In [23] we used the term *nomadic agent interface*; in [15, Chpt.10] the notion of *institutor* is defined and discussed.

[10] These are the kind of scenarios that we actually generated for our first tournament as we explain in section 4.

[11] DAI stands for *Distributed Artificial Intelligence.*

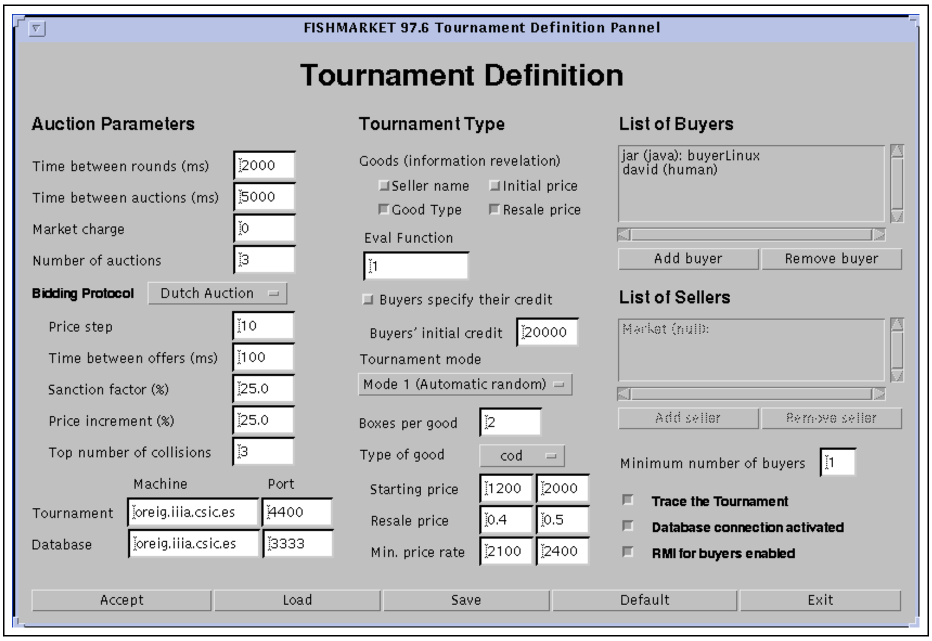

Fig. 3. FM97.6 Tournament Definition Panel

Summarizing, the resulting environment, FM97.6, thus constitutes a multi-agent testbed where a very rich variety of experimental conditions can be explored systematically and repeatedly, and analyzed and reported with lucid detail if needed.

3. Defining Standard Market Conditions

A trading scenario will involve a collection of explicit conventions that characterize an artificial market. Such conventions define the bidding conditions (timing restrictions, increment/decrement steps, available information, etc.), the way goods are identified and brought into the market, the resources buyers may have available, and the conventions under which buyers and sellers are going to be evaluated. This proposal combines the ideas presented in [22] and [15] and shares some commonalities with [14, 31] in the identification of auction parameters. In this section we discuss these underlying ideas from a formal point of view and introduce some of the elements needed to make a precise instantiation of actual tournament scenarios in section 4.

We shall start by studying the dynamics of the protocol governing the main activity within *Fishmarket*, that is, a part of the performative Structure $\mathcal{PS}_{FM}$. Next, we define the notions of *Auction round*, *Auction*, and *Tournament descriptor*. Finally, we close this section defining the framework wherein buyers and sellers may be evaluated.

3.1. Bidding Protocol

When auctioning a good, one could choose among a wide range of bidding protocols (DBP,UBP[12], etc.). Each of these protocols can be characterized by a set of parameters that we refer to as *bidding protocol dynamics descriptors*, so that different instantiations of such descriptors lead to different behaviours of their corresponding bidding protocols. As a particular case, we will concentrate on the downward bidding protocol (DBP) since it was the one utilized in the *Fishmarket* tournaments. Thus, we state explicitly the bidding protocol of Fig. 2 (as described in [15, 23][13]) along with its respective parametrization. The description that follows has been encoded in the algorithm in Fig. 5.

[Step 1] The auctioneer chooses a good out of a lot of goods that is sorted according to the order in

[12] DBP stands for *downward-bidding protocol* ('or 'Dutch") and UBP stands for *upward-bidding protocol* (or "English")

[13] http://www.iiia.csic.es/Projects/fishmarket

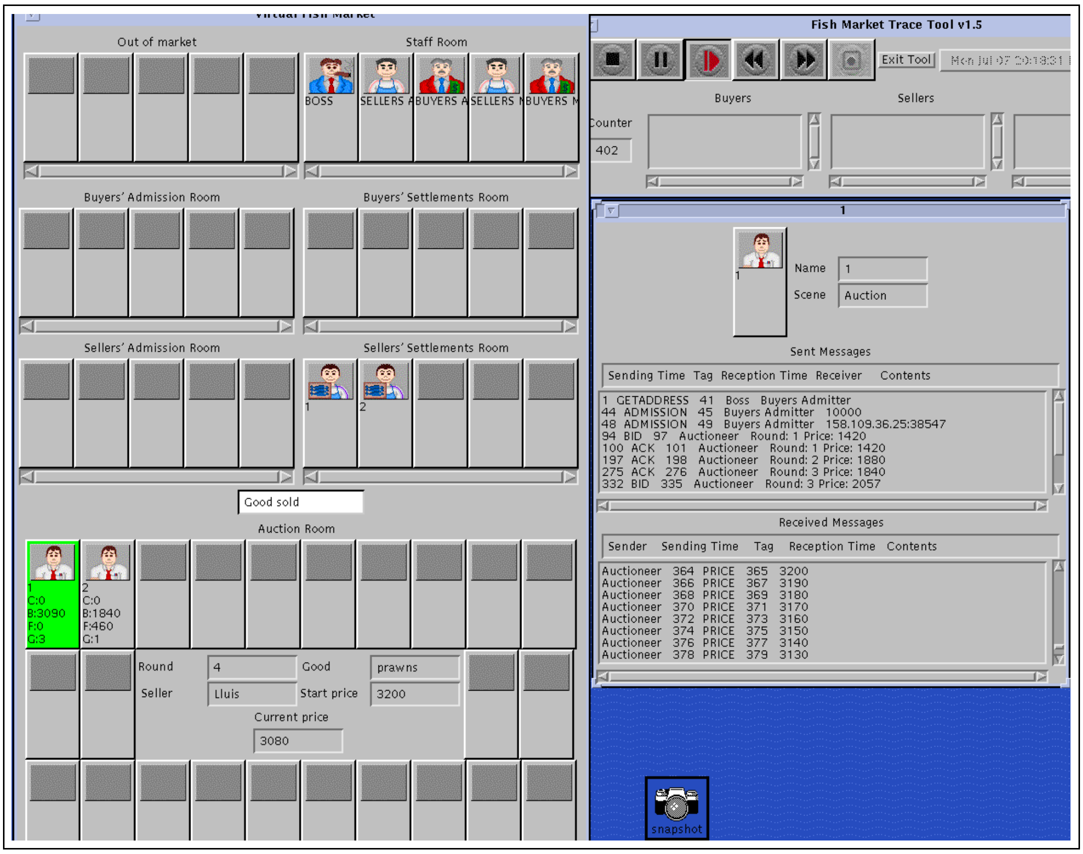

Fig. 4. FM97.6 Trace Tool

which sellers deliver their goods to the sellers' admitter.

[Step 2] With a chosen good g, the auctioneer opens a *bidding round* by quoting offers downward from the good's starting price, (p_α) previously fixed by the sellers' admitter, as long as these price quotations are above a *reserve price* (p_{rsv}) previously defined by the seller.

[Step 3] For each price called by the auctioneer, several situations might arise during the open round:

Multiple bids Several buyers submit their bids at the current price. In this case, a collision comes about, the good is not sold to any buyer, and the auctioneer restarts the round at a higher price. Nevertheless, the auctioneer tracks whether a given number of successive collisions (Σ_{coll}) is reached, in order to avoid an infinite collision loop. This loop is broken by randomly selecting one buyer out of the set of colliding bidders.

One bid Only one buyer submits a bid at the current price. The good is sold to this buyer whenever his credit can support his bid. Whenever there is an unsupported bid the round is restarted by the auctioneer at a higher price, the unsuccessful bidder is punished with a fine, and he is expelled out from the auction room unless such fine is paid off.

No bids No buyer submits a bid at the current price. If the reserve price has not been reached yet, the auctioneer quotes a new price which is obtained by decreasing the current price according to the price step. If the reserve price is reached, the auctioneer declares the good *withdrawn* and closes the round.

[Step 4] The first three steps repeat until there are no more goods left.

```
Function round (B_r^i, g_r^i, p, coll, D_DBP) =
let Function check_credit(b_i) =
  if C_r^i(b_i) ≥ p then
    update_credit(b_i, p);
    sold(g_i^r, b_i, p);
  else if C_r^i(b_i) ≥ p * Π_sanction then
         update_credit(b_i, p * Π_sanction);
         round(B_r^i, g_r^i, p * (1 + Π_rebid), 0, D_DBP);
       else
         round(B_r^i − {b_i}, g_r^i, p * (1 + Π_rebid), 0, D_DBP);
in
  offer(g_r^i, p);
  wait(Δ_offers);
  let B = {b_i | bid(b_i) = true, b_i ∈ B_r^i} in
    case
      ||B|| = 0 : if p = p_ω then withdraw(g_r^i);
                  else round(B_r^i, g_r^i, p − Δ_price, 0, D_DBP);
      B = {b_i} : check_credit (b_i);
      ||B|| > 1 : if coll < Σ_coll then
                    round(B_r^i, g_r^i, p * (1 + Π_rebid), coll + 1, D_DBP);
                  else check_credit(random_select(B));
    end case
  end
end

DBP(B_r^i, g_r^i) = round(B_r^i, g_r^i, p_α, 0)
```

Fig. 5. Downward bidding protocol

Six parameters that control the dynamics of the bidding process are implicit in this protocol definition[14]. We shall enumerate them now, and require that they become instantiated by the tournament designer as part of a tournament definition.

Definition 3.1 (DBP Dynamics Descriptor). We define a Downward Bidding Protocol Dynamics Descriptor $\mathcal{D}_{DBP}$ as the 6-tuple $\langle \Delta_{price}, \Delta_{offers}, \Delta_{rounds}, \Sigma_{coll}, \Pi_{sanction}, \Pi_{rebid} \rangle$ such that

- $\Delta_{price} \in I\!N$ (price step). Decrement of price between two consecutive quotations uttered by the auctioneer.
- $\Delta_{offers} \in I\!N$ (time between offers). Delay between consecutive price quotations.
- $\Delta_{rounds} \in I\!N$ (time between rounds). Delay between consecutive rounds belonging to the same auction.
- $\Sigma_{coll} \in I\!N$ (maximum number of successive collisions). This parameter prevents the algorithm from entering an infinite loop as explained above.
- $\Pi_{sanction} \in I\!R$ (sanction factor). This coefficient is utilized by the buyers' manager to calculate the amount of the fine to be imposed on buyers submitting unsupported bids.
- $\Pi_{rebid} \in I\!R$ (price increment). This value determines how the new offer is calculated by the auctioneer from the current offer when either a collision, or an unsupported bid occur.

Note that the identified parameters impose significant constraints on the trading environment. For instance, Δ_{offers} and Δ_{rounds} affect the agents' time-boundedness, and consequently the degree of situatedness viable for bidding strategies.

3.2. Auctions

Auction rounds aim at identifying and characterizing the ontological elements involved in each bidding round.

Definition 3.2 (Auction Round). For a given round r of auction i we define the *auction round* $\mathcal{A}_r^i$ as the 5-tuple

$$\mathcal{A}_r^i = \langle \mathcal{B}_r^i, g_r^i, \mathcal{C}_r^i, d_r^i, \mathcal{I}_r^i \rangle$$

where

- $\mathcal{B}_r^i$ is a non-empty, finite set of buyers' identifiers such that $B_r^i \subseteq \mathcal{B}$, the set of all participating buyers.
- $g_r^i = \langle \iota, \tau, p_\alpha, p_{rsv}, s_j, p_\omega, b_k \rangle$ is a good where ι stands for the good identifier, τ stands for the type of good, $p_\alpha \in I\!N$ stands for the starting price, $p_{rsv} \in I\!N$ stands for the reserve price, $s_j \in \mathcal{S}$—the set of all participating sellers—is the seller of the good, $p_\omega \in I\!N$ stands for the sale price, and $b_k \in \mathcal{B}_r^i$ is the buyer of the good. Notice that g_r^i is precisely the good to be auctioned during round r of auction i, and that p_ω and b_k might take on empty values when the round is over, denoting that the good has been withdrawn.
- $\mathcal{C}_r^i : \mathcal{B}_r^i \longrightarrow I\!R$ assigns to each buyer in $\mathcal{B}_r^i$ his available credit during round r of auction i.
- d_r^i stands for an instance of a bidding protocol dynamics descriptor[15].
- $\mathcal{I}_r^i$ is a set of information functions available for the agents during the round. It contains those functions labelling some of the events occurring

[14] Other bidding protocols—f.i. UBP, Yankee, Double auction, etc.—would be characterized by other sets of parameters)

[15] In the *Fishmarket* tournaments It will always be an instance of the DBP dynamics descriptor.

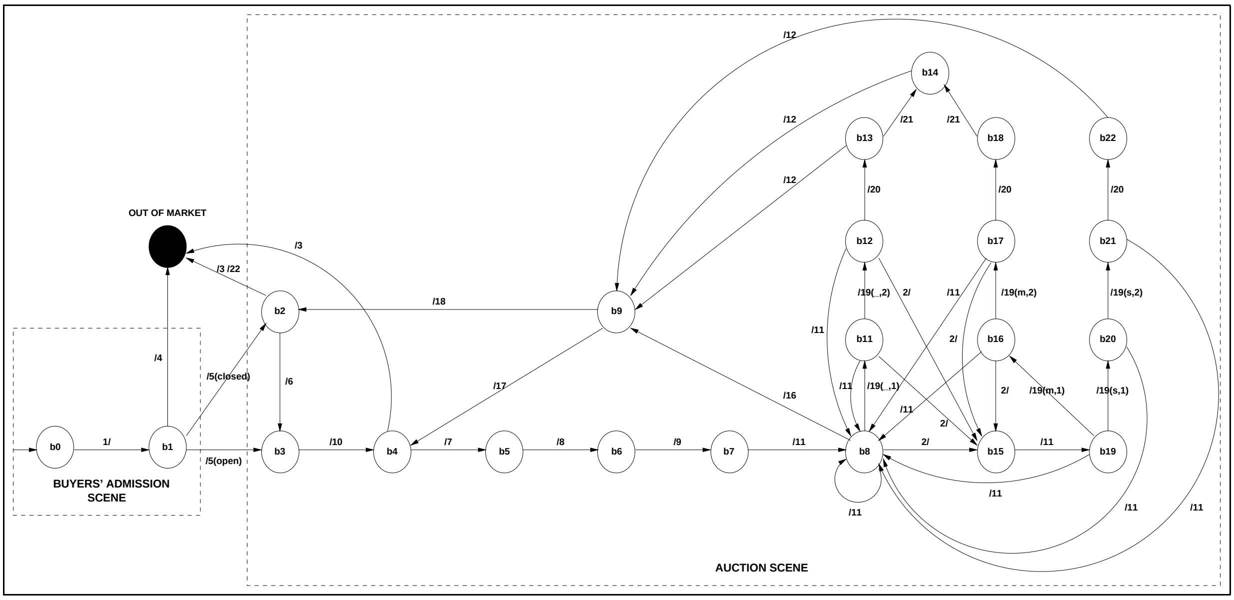

Fig. 6. Communication behaviour of buyer agent b ($b' \neq b$) for the UBP

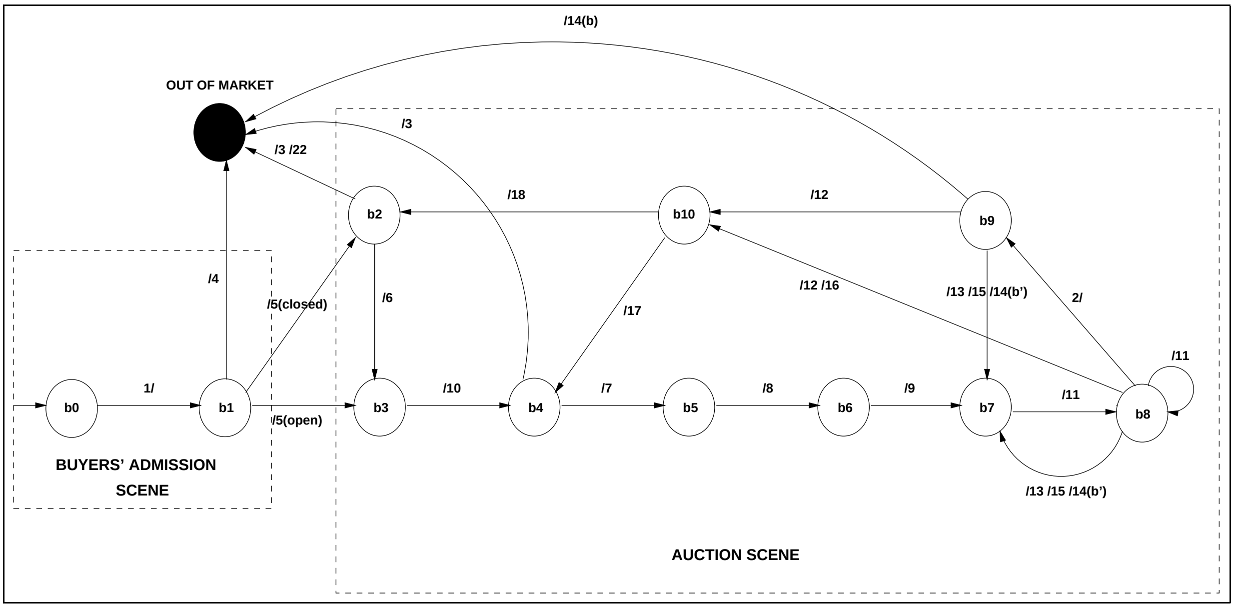

Fig. 7. Communication behaviour of buyer agent b ($b' \neq b$) for the DBP

during the round. Thus, the contents of this set will depend on the bidding protocol governing each round. For instance, following the description of the downward bidding protocol in Fig. 5, functions for labelling offers, sales, fines, expulsions, collisions, and withdrawals must be provided within this subset. For example, the auction catalogue could be included as an element of this set.

FM97.6 lets the tournament designer decide on the degree of transparency to be attached to *auction rounds*. In other words, the designer will have to decide what information about *auction rounds* is to be conveyed to the contenders, whether these should be informed about the participating buyers, and the subset of the set of information functions to be transmitted.

Finally, a notion of *Auction* arises naturally from the definition above.

Definition 3.3 (Auction). We define an auction $\mathcal{A}^i$ as a sequence of *Auction rounds*

$$\mathcal{A}^i = [\mathcal{A}^i_1, \ldots, \mathcal{A}^i_{r_i}]$$

To summarize, firstly we have identified all the essential elements characterizing bidding rounds: the participating buyers and their credits, the sellers and their goods, those features typifying the bidding protocol, and the most relevant information produced during the round that allows the participating agents to know the current state of the bidding process. Secondly, we have introduced the notion of auction in terms of our view of *Auction rounds*.

3.3. Market Interagents

When developing our testbed, a major question arose: how to handle the interdependencies among the agents situated in a market setting? On the one hand, there is the matter of coordinating the activities of the several market intermediaries composing the market institution so as to guarantee the proper workings of the institution itself. On the other hand, there is the matter of coordinating the interplay between trading (buyer and seller) heterogeneous (human and software) agents and the market institution.

In general, it is widely accepted that when several computational entities interact by exchanging messages, a higher level of interaction concerning the conventions shared during the exchange should be addressed [30, 9, 2, 19, 15]. Making such conventions explicit allows the management of the interdependencies among agents' activities.

In [15] the abstract role of an *institutor* is discussed and in [12] a computational analogue is developed. In [12] we introduce an *interagent* as an autonomous software agent which intermediates the communication and coordination between an agent and the agent society wherein this is situated. An interagent allows interdependencies between agents' communicative acts, expressed as performatives of a high-level agent communication language, to be ordered by means of *conversation protocols* which represent the conventions adopted by agents when interacting through the exchange of messages. We model and implement conversation protocols as a special type of pushdown automaton because unlike finite state machines, pushdown automata allow to store and subsequently retrieve the context of an ongoing conversation. These conversation protocols can be easily defined in a declarative way for each interagent.

The functionality provided by an interagent will highly depend on the role played by the agent interacting with it. Thus we will distinguish two distinct roles for agents making use of interagents:

- *the user* of an interagent will regard it as the sole and exclusive means through which he can interact with the agent society thanks to the set of communication and coordination services provided by the interagent, but previously defined by the owner
- *the owner* of an interagent is provided with a wide range of facilities to either load or program—before and during the user's run-time—into the interagent the communication and coordination services that the user is allowed to employ

Needless to say that an agent can possibly play both roles at the same time.

Thus, interagents have been incorporated into our testbed in order to handle the intra-market coordination problems as well as the interplay between trading agents and the market institution. Notice, though, that we draw a distinction between the so-called *internal market interagents* and the *external market interagents* based on two criteria: ownership and usage. Fig. 8 depicts the two types of interagents included into the *Fishmarket*.

Whereas internal market interagents are both owned and used by those agents functioning as market intermediaries within the market institution, external market interagents are owned by the institution too but

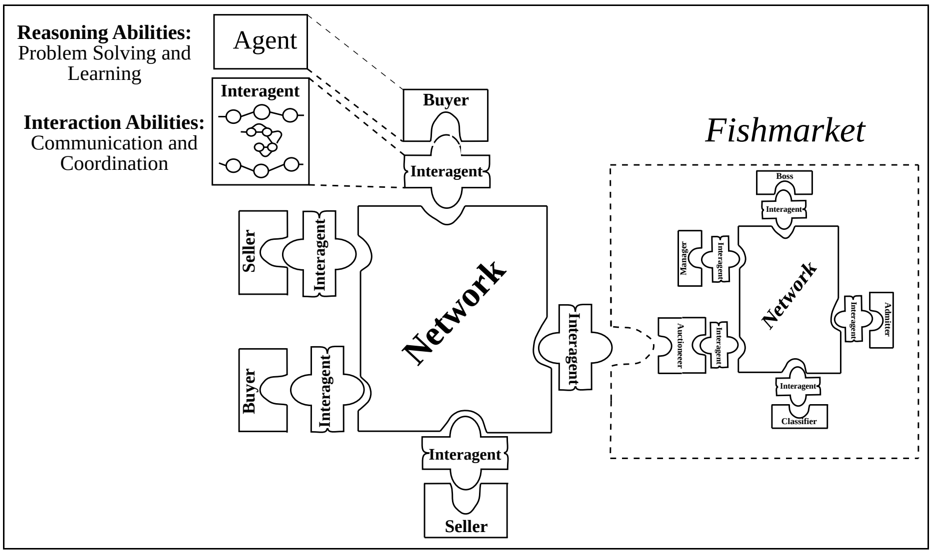

Fig. 8. *Fishmarket*: A multi-agent system using interagents

used by trading agents to interact with the market.

For instance, Fig. 7 depicts the communication states of a buyer when interacting with his interagent when a good is auctioned following the downward bidding protocol in Fig. 5, while Fig. 6 depicts the analogous finite state control for the UBP. Tables 3 and 4 specify the syntax of the messages labelling the edges of both finite state controls[16]. These messages correspond to the propositional content of the illocutions in Table 1.

#Message	Predicate	Parameters
1	*admission*	*buyerlogin password*
2	*bid*	[*price*]
3	*exit*	

Table 3
Market Interagent Incoming Messages

Notice, however, that both diagrams display the interaction between a buyer agent and his interagent from the agent's view. Therefore, message numbers followed by / stand for messages sent by a buyer agent, while message numbers preceded by / stand for messages received by a buyer agent. For instance, 2/ means that the buyer submits a bid at the price called by the auctioneer within /11.

In FM97.6, external market interagents have been designed to work as Java processes which use its standard input and standard output to communicate with buyer agents (the users) and (TCP-)stream sockets

#Message	Predicate	Parameters
4	*deny*	$deny\_code$
5	*accept*	$open\|closed\ auction\_number$
6	$open\_auction$	$auction\_number$
7	$open\_round$	$round\_number$
8	*good*	$good\_id\ good\_type$ $starting\_price\ resale\_price$
9	*buyers*	$\{buyerlogin\}^*$
10	*goods*	$\{good\_id\ good\_type$ $starting\_price\ resale\_price\}^*$
11	*offer*	$good\_id$ *price*
12	*sold*	$good\_id$ *buyerlogin price*
13	*sanction*	*buyerlogin fine*
14	*expulsion*	*buyerlogin*
15	*collision*	*price*
16	$withdrawn$	$good\_id\ price$
17	end_round	$round\_number$
18	$end\_auction$	$auction\_number$
19	$going$	$\{single\|multiple\} + \{1, 2\}$
20	$gone$	
21	tie_break	*buyerlogin*
22	$closed\_market$	

Table 4
Market Interagent Outgoing Messages

[16] For the sake of simplicity, these examples restrict to show the *finite state controls* of the conversation protocols used for encoding the coordination patterns underlying the dialogues held between buyer agents (the users) and the interagents attached to them by the institution (the owner).

to communicate with the institution (the owner). In adopting such a simple convention, we allow agent programmers to build their agents in any programming language that allows for firstly spawning the interagent as a child process and then plugging to it.

Finally, we would like to emphasize that two major benefits derive from the usage of interagents within our testbed (as summarized in [12]):

- they permit agents to reason about both communication and coordination at a higher level of abstraction, and
- they provide a complete set of facilities that allows agent engineers to concentrate on the design of their agents' internal and social behaviour

3.4. Tournament Descriptor

By bundling together all the elements introduced so far, we can formulate descriptions of tournament scenarios.

Definition 3.4 (Tournament Descriptor). We define a Tournament Descriptor $\mathcal{T}$ as the 8-tuple

$$\mathcal{T} = \langle n, \Delta_{auctions}, \mathcal{D}, \mathcal{P}, \mathcal{B}, \mathcal{S}, \mathcal{F}, E \rangle$$

such that:

- n is the number of auctions to take place during a tournament.
- $\Delta_{auctions}$ is the time between consecutive auctions.
- $\mathcal{D}$ is a finite set of bidding protocols' dynamics descriptors.
- $\mathcal{P}$ is a finite family of communication protocols that a buyer agent must employ to interact with its *interagent* indexed by different bidding protocol types (f.i. $P = \{P_{DBP}, P_{UBP}, \ldots\}$).
- $\mathcal{B} = \{b_1, \ldots, b_p\}$ is a finite set of identifiers corresponding to all participating buyers.
- $\mathcal{S} = \{s_1, \ldots, s_q\}$ is a finite set of identifiers corresponding to all participating sellers.
- $\mathcal{F} = [\mathcal{F}^1, \ldots, \mathcal{F}^n]$ is a sequence of n descriptors. Each $\mathcal{F}^i$ specifies the way auction $\mathcal{A}^i$ is dynamically generated.
- $E = \langle E_b, E_s \rangle$ is a pair of winner evaluation function that permit to calculate respectively the score of buyers and sellers.

First of all, notice that the tournament designer will include a non-empty $\mathcal{D}_{DBP}$ in $\mathcal{D}$, for the *Fishmarket* tournaments, and that the designer will have to specify also the time between consecutive auctions. Observe as well that the sets $\mathcal{D}, \mathcal{P}, \mathcal{B}$, and $\mathcal{S}$ are the domains taken by the set of descriptors $\mathcal{F}$ in order to dynamically generate the contents of each auction $\mathcal{A}_i$ during the tournament, for instance, the set of buyers participating in round r of auction i must be a subset of the domain $\mathcal{B}$. Note also that any given auction $\mathcal{A}^i$ will not be fully instantiated till all their bidding rounds $\mathcal{A}^i_r$ are over, because although some elements in $\mathcal{A}^i_r$ are known before this round starts, the rest are produced during the round. On the other hand, notice that different sets of descriptors determine different tournament modes. In FM97.6, tournament designers can choose among some standard modes whose main features are:

Automatic The lots of goods are automatically generated based on functions of arbitrary complexity provided by the tournament designer in $\mathcal{F}$, and so no sellers are involved in these tournaments.

Random The lots of goods are randomly generated based on uniform distributions given in $\mathcal{F}$ provided by the tournament designer, and thus no sellers are involved in these tournaments either.

One auction Once all participating sellers have submitted their goods, the same auction is repeated over and over with the same lot of goods till the number of auctions set by the tournament designer is reached.

Fishmarket The mode closest to the workings of the fish market. The tournament designer simply specifies the starting and closing times. During that period of time buyers and sellers can enter, submit goods, bid for goods, and leave at will.

Observe that the degree of complexity of the scenarios that trading agents will face results from the combination of the chosen tournament mode, the amount and complexity of the information supplied within $\mathcal{F}$, and the transparency attached to each *auction round*.

3.5. Tournament Evaluation Framework

Finally, the following definition provides the framework that the tournament designer is to use when tracing, evaluating, and analyzing tournament scenarios.

Definition 3.5 (Tournament Evaluation Framework). We define a Tournament Evaluation Framework $\mathcal{E}$ as the pair $\langle \mathcal{T}, \mathcal{A} \rangle$ such that:

- $\mathcal{T}$ is a Tournament Descriptor.
- $\mathcal{A} = [\mathcal{A}^1, \ldots, \mathcal{A}^n]$ is a finite sequence of *Auctions*.

n	21					
$\Delta_{auctions}$	*5000 msec*					
$\mathcal{D}$	$\{d_{DBP}\} = \{\langle 10ptas, 1000msec, 3000msec, 3, 0.25, 0.25\rangle\}$					
$\mathcal{P}$	$\{P_{DBP}\}$					
$\mathcal{B}$	*warakaman akira fishbroker tindalos dolphin f2422080 panipeixos josnat satan xanquete e0934125*					
$\mathcal{S}$	$\varnothing$					
$\mathcal{F}^i(i = 1 \dots n)$	r_i	cardinal of goods				
	$\mathcal{B}^i_r$	$\mathcal{B}$				
	$goods$	τ	*#Boxes*	p_α	p_{rsl}	p_{rsv}
		cod	$U[1,15]$	$U[1200,2000]$	$U[1500,3000]$	$U[0.4,0.5]$
		tuna fish	$U[1,15]$	$U[800,1500]$	$U[1200,2500]$	$U[0.3,0.45]$
		prawns	$U[1,15]$	$U[4000,5000]$	$U[4500,9000]$	$U[0.35,0.45]$
		halibut	$U[1,15]$	$U[1000,2000]$	$U[1500,3500]$	$U[0.4,0.6]$
		haddock	$U[1,15]$	$U[2000,3000]$	$U[2200,4000]$	$U[0.35,0.55]$
	$\mathcal{C}^i_r$	$\mathcal{C}^i_1(b) = 50000\ \forall b \in \mathcal{B}^i_1, \mathcal{C}^i_{k+1}(b) = \mathcal{C}^i_k(b) - expenses^i_k(b)$				
	d^i_r	d_{DBP}				
	$\mathcal{I}^i_r$	fine, expulsion, sanction, sale, offer, collision, withdrawal				
E	$\langle E_b, E_s \rangle = \langle \beta^r_a \left[\frac{4}{r^2}(\pi(r-\pi))\right], \varnothing\rangle$					

Table 5
UPC Tournament Descriptor

The sequence of *Auctions*, $\mathcal{A}$, must be regarded as the tournament history, i.e., the complete instantiation of the auctions composing the tournament. Moving to the implementation level, we find that such history of tournaments is kept by FM97.6 in a database.

Next section aims at introducing a rather straightforward example that intends to illustrate how to generate a market scenario, and, at the same time, to report on the results of the first *Fishmarket* tournament conducted at the Technical University of Catalonia(UPC).

4. A Toy Fish Market Tournament: The UPC Tournament

This section presents the definition of the first *Fishmarket* tournament which involved a group of final year students at the UPC. For the sake of brevity, we only describe the main features of the tournament scenario. For more detailed information, we address the reader to the tournament web page[17].

We opted for a simple scenario characterized by the tournament descriptor in Table 5. There are some comments to be made on the resulting scenario:

- Buyer agents were identified by a unique login and password delivered to their owners after registering. Then, once admitted into the auction room, all buyer agents were endowed with the same credit at the beginning of each auction $\mathcal{A}^i$ ($\mathcal{C}^i_1(b) = 50000\,\forall b \in \mathcal{B}^i_1$) of the tournament $\mathcal{T}$.
- Because the tournament mode was set to *random*, the number of fish boxes for each type of fish (τ) were randomly generated for each auction $\mathcal{A}^i$, and the starting price (p_α), resale price (p_{rsl}), and reserve price (p_{rsv}) of each one of these fish boxes were also randomly generated according to the uniform distributions in Table 5. All those distributions except those referring to the reserve prices were known by the contenders.
- Skeleton programs for buyer agents were provided in Java, C, Prolog, and Common Lisp.
- The chosen evaluation function (E_b) calculates the performance for each buyer at round number r of auction number a based on the accumulated benefits (β^r_a), the accumulated number of purchases (π), and the number of rounds (r) where that buyer is active.

In spite of the rudimentary character of this experience, two considerations are worth reporting:

- The experimental conditions defined (mainly starting prices, available endowments, and evaluation functions) favoured voracious strategies (buy as much as possible, as soon as possible).
- The setting of time-delays (like Δ_{offers}, Δ_{rounds} and $\Delta_{auctions}$) acted against deliberative agents.

De Toro (in [5]) devised variants to these tourna-

[17] http://www.iiia.csic.es/Projects/fishmarket/tournament97.html

ment conditions and showed that deliberative agent performance, relative to simple reactive heuristics, improved with scarcity of resources and experience, as long as time delays between rounds and between auctions were kept above a threshold[18].

5. Related and Future Work

Several attempts have been made by researchers in electronic commerce concerning the proposal of electronic marketplace architectures [3, 23, 26, 31]. Such efforts share the common goal of building electronic markets where both buying and selling agents can trade on behalf of their users. Nonetheless there is the intricate matter of providing agent developers (and agent users) with some support to help them face the arduous task of designing, building, and tuning their trading agents, before letting them loose in wildly competitive scenarios. We have attempted to contribute in that direction. We have developed a testbed that can be used to test and tune trading agents, FM97.6, that happens to be built as an extension of an actual agent-mediated auction house, FM96.5. Our test-bed shares many commonalities with the AuctionBot initiative [31][19]. AuctionBot is a highly versatile online auction server that permits the generation of a wide range of auction environments wherein both human and software agents (they provide an API to help build trading agents) can participate. It has already proven its usefulness as a research platform hosting large-scale experiments to study computational market mechanisms and agent strategies.

The lack of agent-mediated trading test-beds is paradoxical in light of the popularity of agent competitions and the inherently competitive nature of trading. Recall for instance *Robocup*[10] that encourages both AI researchers and robotics researchers to make their systems play soccer; or the *AAAI Mobile Robot Competition*[11] where autonomous mobile robots try to show their skills in office navigation and in cleaning up the tennis court; and even automated theorem proving systems are pitched against each other in [25], although one can hardly argue that any of these agent competitions involve features that are directly relevant for agent-mediated trading. However, our proposal is closer to the *Double auction* tournaments held by the Santa Fe Institute[1] where the contendants competed for developing optimized trading strategies. Though similar enough, our approach has a wider scope. We are interested not only in testing agent strategies and building trading agents [28], or in the use of artificial intelligence to study economic markets [21]. We are also interested in the study of market conditions and market conventions, thus our emphasis on the flexibility of the specification framework, and the generality of the underlying definitions.

Our future work shall proceed in two complementary directions. Firstly, trading agents. We envision as an immediate future task the deployment of more complex buyer agent models such as those already introduced in [5–7] and tools and techniques for deploying and testing trading-agent shells, strategies and actual agents. Secondly, FM97.6 will be made to evolve to host other (even more flexible) agent-mediated institutions. In particular, we expect to release in the near future an agent-mediated auction house where goods can be traded under the rules of several bidding protocols. Later on, we shall concentrate on agent-mediated marketplaces where other forms of price-fixing mechanisms (double auction, discounting, open negotiation) can take place.

[18] Using a more standard relative-performance common-value evaluation function.

[19] http://auction.eecs.umich.edu

Acknowledgements

This work has been partially supported by the Spanish CICYT project SMASH, TIC96-1038-C04001, and the Mexican CONACYT grant [69068-7245]. J. A. Rodríguez-Aguilar and F. J. Martín enjoy the CIRIT doctoral scholarships FI-PG/96-8.490 and FI-DT/96-8.472 respectively.

References

[1] M. Andrews and R. Prager. *Genetic Programming for the Acquisition of Double Auction Market Strategies*, pages 355–368. The MIT Press, 1994.

[2] M. Barbuceanu and M. S. Fox. Cool: A language for describing coordination in multi-agent systems. In *Proceedings of the First International Conference in Multi-Agent Systems* (*ICMAS-95*), pages 17–24. AAAI Press, June 1995.

[3] A. Chavez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. In *First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology* (*PAAM'96*), pages 75–90, 1996.

[4] P. Cohen, M. Greenberg, D. Hart, and A. Howe. Trial by fire: Understanding the design requirements for agents in complex environments. *AI Mag.*, 10(3):33–48, 1989.

[5] M. C. de Toro. A hybrid buyer agent architecture for the fish-market tournaments. Master's thesis, Universitat Autonoma de Barcelona, 1997.

[6] P. Garcia, E. Giménez, L. Godo, and J. A. Rodríguez-Aguilar. Possibilistic-based design of bidding strategies in electronic auctions. In *The 13th biennial European Conference on Artificial Intelligence* (*ECAI-98*), 1998.

[7] E. Giménez, L. Godo, J. A. Rodríguez-Aguilar, and P. Garcia. Designing bidding strategies for trading agents in electronic auctions. In *Proceedings of the Third International Conference on Multi-Agent Systems* (*ICMAS-98*), 1998.

[8] B. A. Huberman and S. Clearwater. A multi-agent system for controlling builging environments. In *Proceedings of the First International Conference on Multi-Agent Systems* (*ICMAS-95*), pages 171–176. AAAI Press, June 1995.

[9] N. R. Jennings. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 8(3):223–250, 1995.

[10] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. Robocup: The robot world cup initiative. In *First International Conference on Autonomous Agents*, 1997.

[11] D. Kortenkamp, I. Nourbakhsh, and D. Hinkle. The 1996 AAAI Mobile Robot Competition and Exhibition. *AI Mag.*, 18(1):25–32, 1997.

[12] F. J. Martín, E. Plaza, J. A. Rodríguez-Aguilar, and J. Sabater. Jim a java interagent for multi-agent systems. In *Proceedings of the AAAI-98 Workshop on Software Tools for Developing Agents*, 1998.

[13] T. Montgomery, J. Lee, D. Musliner, E. Durfee, D. Dartmouth, and Y. So. Mice users guide. Technical report, Dept. of Electrical Engineering and Computer Science, Univ. of Michigan. Technical Report, CSE-TR-64-90, 1992.

[14] T. Mullen and M. Wellman. Market-based negotiation for digital library services. In *Second USENIX Workshop on Electronic Commerce, Oakland, CA*, 1996.

[15] P. Noriega. *Agent-Mediated Auctions: The Fishmarket Metaphor*. PhD thesis, Universitat Autonoma de Barcelona, 1997. Also to appear in IIIA mongraphy series.

[16] D. North. *Institutions, Institutional Change and Economics Performance*. Cambridge U.P., 1990.

[17] C. S. Noyda Matos and N. R. Jennings.

[18] G. M. P. O'Hare and N. R. Jennings. *Foundations of Distributed Artificial Intelligence*. John Wiley & Sons, Inc, 1996.

[19] H. V. D. Parunak. Visualizing agent conversations: Using enhanced dooley graph for agent design and analysis. In *Proceedings of the Second International Conference on Multi-Agent Systems*, 1996.

[20] M. Pollack and M. Ringuette. Introducing the tileworld: Experimentally evaluating agent architectures. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 183–189, 1990.

[21] V. Rajan and J. R. Slagle. The use of artificially intelligent agents with bounded rationality in the study of economic markets. In *AAAI-96*, 1996.

[22] J. A. Rodríguez-Aguilar, F. J. Martín, P. Noriega, P. Garcia, and C. Sierra. Competitive scenarios for heterogeneous trading agents. In *Proceedings of the Second International Conference on Autonomous Agents* (*AGENTS'98*), 1998.

[23] J. A. Rodríguez-Aguilar, P. Noriega, C. Sierra, and J. Padget. Fm96.5 a java-based electronic auction house. In *Second International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology*(*PAAM'97*), 1997.

[24] C. Sierra, N. R. Jennings, P. Noriega, and S. Parson. A framework for argumentation-based negotiation. In *Proceedings of the 4th International Workshop on Agent Theories, Architectures and Languages*, 1997.

[25] C. B. Suttner and G. Sutcliffe. *ATP System Competition*, volume 1104 of *Lecture Notes in Artificial Intelligence*, pages 146–160. Springer Verlag, 1996.

[26] M. Tsvetovatyy and M. Gini. Toward a virtual marketplace: Architectures and strategies. In *First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology* (*PAAM'96*), pages 597–613, 1996.

[27] H. R. Varian. Economic mechanism design for computerized agents. In *First USENIX Workshop on Electronic Commerce*, 1995.

[28] J. M. Vidal and E. H. Durfee. Building agent models in economic societies of agents. In *Workshop on Agent Modelling* (*AAAI-96*), 1996.

[29] M. P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, (1):1–23, 1993.

[30] T. Winograd and F. Flores. *Understanding Computers and Cognition*. Addison Wesley, 1988.

[31] P. R. Wurman, , M. P. Wellman, and W. E. Walsh. The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents. In *Second International Conference on Autonomous Agents* (*AGENTS'98*), 1998.

[32] F. Ygge and H. Akkermans. Power load management as a computational market. In *Proceedings of the Second International Conference on Multi-Agent Systems* (*ICMAS-96*), 1996.

[33] F. Ygge and H. Akkermans. Making a case for multi-agent systems. In M. Boman and W. V. de Velde, editors, *Advances in Case-Based Reasoning*, number 1237 in Lecture Notes in Artificial Intelligence, pages 156–176. Springer-Verlag, 1997.