Competitive Scenarios for Heterogeneous Trading Agents

Juan A. Rodríguez-Aguilar Francisco J. Martín Pablo Noriega Pere Garcia
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

Abstract

We present a framework for defining trading scenarios based on fish market auctions. In these scenarios, agents of arbitrary complexity can participate in electronic auctions under a collection of standardized market conditions and be 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.

Our proposal involves a set of conventions for the typification of goods, bidding protocols, availability of goods, buyer endowments and performance evaluation criteria. The proposed framework is implemented as a multi-agent test-bed which is an extension of FM96.5 –a Java-based version of the *Fishmarket* auction house. A simple tournament is used to illustrate these elements.

1 Introduction

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' behavior, 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 Dutch auctions –like the traditional fish market[1]– proceeds at frenetic speeds.

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. However, it is not really obvious which of the many possible approaches for automatic trading strategies 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 multi-agent test-bed –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 test-bed, we use a Java-based electronic auction house inspired by the traditional fish market, FM96.5 [15]. 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

[1] We will use the expression *fish market* to refer to the actual, real-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 [11] as a "... *set of artificial constraints that articulate agent interactions*".

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 behavior 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 ([18]), 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 will introduce the concept of tournament descriptor, and in Section 4, we illustrate how to instantiate such tournament descriptor in order to characterize a particular tournament scenario. Finally, Section 5 discusses related work and argues about present and future work.

2 An Auction Tournament Environment

Following [10], the fish market can be described as a place where several *scenes* run simultaneously, at different places, 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' registration scene*, and buyers need to register for the market, at the *buyers' registration 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 [15, 21, 10] we present 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. Those conventions that affect buyers and sellers have been coded into what we call a *remote control* 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. A remote control gives a permanent identity 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[2].

In order to obtain an auction tournament environment, more functionality has been added to FM96.5 to turn it into a multi-agent test-bed, FM97.6. The resulting test-bed has the following salient characteristics:

- It is *domain-specific* in the sense that it models and simulates an *electronic auction house*.
- It is *realistic*, since it follows the actual conventions of a complex real-world institution, the traditional fish market.
- Being an extension of FM96.5, FM97.6 inherits the mechanism of interaction between buyer agents and the market. This use of the remote controls makes FM97.6 *architecturally–neutral* since no particular agent architecture (or language) is assumed or provided[3].
- FM97.6 allows for very flexible *scenario generation* to enable designers to produce systematic experimentation. FM97.6 allows for the specification, and subsequent activation, of a large variety of market scenarios: from simple artificial scenarios to complex realistic scenarios, from carefully constructed scenarios that highlight certain problems to randomly generated scenarios useful for testing buyer agents' average performance[4].
- Explicit parameter-fixing and participant-registration modes are involved in the scenario generation facility , to allow for the *repeatability* of experiments[5].
- Finally, FM97.6 includes a **Trace Tool** which keeps track of all illocutions and transactions that take place during an auction, hence a whole auction can be audited and re-enacted step-by-step, and the evolving performance of all the agents involved in a tournament can be traced, evaluated, and analyzed.

Figure 1 displays a snapshot of FM97.6 Tournament Definition Panel, the tool utilized by tournament designers to construct tournament scenarios.

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. 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. Thus the purpose of this section will be to

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

[3] Other test-beds such as Tile-world [13], Tæms [12], and Mice [8] have also opted for remaining architecturally neutral, whereas test-beds like Mace [12], Phoenix [3], Archon [12], DVMT [12], or CooperA [12] provide a suite of development facilities for building agents.

[4] Like the one in Section 4 below, for example.

[5] Many DAI test-beds (f.i. Tileworld, Phoenix, DVMT, TÆMS) support repeatability.

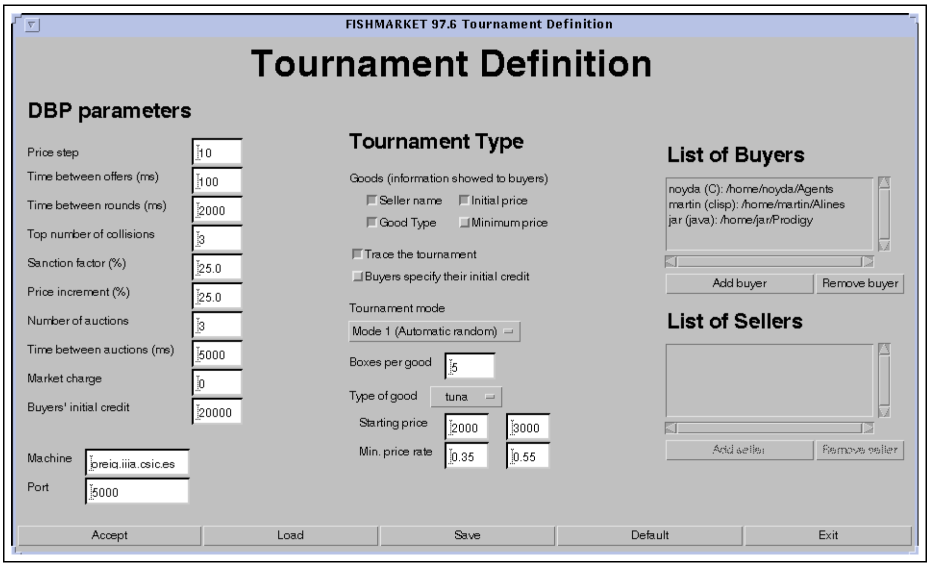

Figure 1: FM97.6 Tournament Definition Panel

sketch the foundations of the formal framework underpinning our current implementation of FM97.6[6].

We shall start by studying the dynamics of the protocol governing the main activity within the *Fishmarket*. 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 (Dutch, English, 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 behaviors of their corresponding bidding protocols. As a particular case, we will concentrate on the downward bidding protocol since it was the one utilized in the first *Fishmarket* tournaments. Thus, we state explicitly the bidding protocol already described in [10, 15, 21] along with its respective parametrization. The description that follows has been encoded in the algorithm in figure 2.

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

Six parameters that control the dynamics of the bidding process are implicit in this protocol definition. Notice, however that other bidding protocols – f.i. English, Yankee, Double auction, etc.– would be characterized by other sets of parameters). 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 7-tuple

$$\mathcal{D}_{DBP} = \langle \Delta_{price}, \Delta_{offers}, \Delta_{rounds}, \Sigma_{coll}, \Pi_{sanction}, \Pi_{rebid} \rangle$$

such that

[6] A detailed proposal is in [10].

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

DBP(B^i_r, g^i_r) = round(B^i_r, g^i_r, p_α, 0)
```

Figure 2: Downward bidding protocol

- $\Delta_{price} \in I\!\!N$ (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* $\mathcal{A}^i_r$ as the 5-tuple

$$\mathcal{A}^i_r = \langle \mathcal{B}^i_r, g^i_r, \mathcal{C}^i_r, d^i_r, \mathcal{I}^i_r \rangle$$

where

- $\mathcal{B}^i_r$ is a non-empty, finite set of buyers' identifiers such that $B^i_r \subseteq \mathcal{B}$, the set of all participating buyers.
- $g^i_r = \langle \iota, \tau, p_\alpha, p_{rsv}, s_j, p_\omega, b_k \rangle$ 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 $b_k \in \mathcal{B}^i_r$ is the buyer of the good. Notice that g^i_r 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.
- $\mathcal{C}^i_r : \mathcal{B}^i_r \longrightarrow I\!\!R$ assigns to each buyer in $\mathcal{B}^i_r$ his endowment during round r of auction i.
- d^i_r stands for an instance of a bidding protocol dynamics descriptor[7].
- $\mathcal{I}^i_r$ is a set of information functions available for the agents during the round. It contains those functions labelling some of the events occurring 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 figure 2, functions for labelling offers, sales, fines, expulsions, collisions, and withdrawals must be provided within this subset.

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 along 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*. This proposal is based on [10] and happens to resemble [9] in the identification of auction parameters.

3.3 Remote Control Protocol

As mentioned above, the *remote control* constitutes the sole means whereby buyer agents interact with the market. The behavior of such device will clearly depend on the bidding protocol type in use.

For the sake of simplicity, we chose finite state machines as the means of representing the model of coordination underlying the structured conversation held between agents and their remote controls. Thus, the *remote control* will

[7] It will always be an instance of a DBP dynamics descriptor for the goods auctioned in the *Fishmarket* tournaments

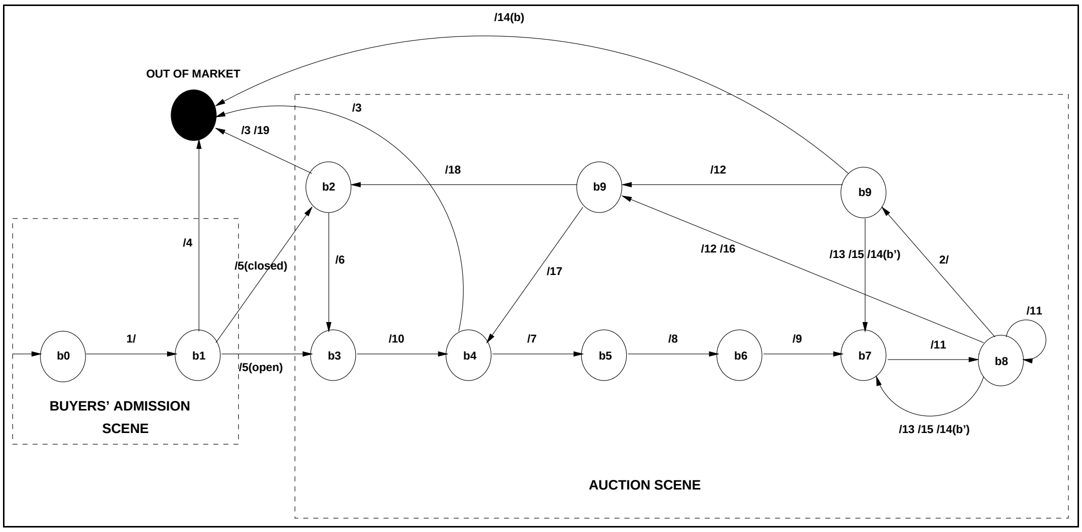

Figure 3: Communication behaviour of buyer agent b ($b' \neq b$) for the DBP

employ a different finite state machine for each auction protocol type.

For instance, figure 3 shows the communication states of a buyer when interacting with his remote control when a good is auctioned following the downward bidding protocol in figure 2. Tables 1 and 2 specify the syntax of the messages labeling the edges of this finite state machine.

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

Table 1: Remote Control Incoming Messages

#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	*closed_market*	

Table 2: Remote Control Outgoing Messages

Notice, however, that this diagram displays the interaction between a buyer agent and his remote control 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 in /11.

In FM97.6, the remote control has been designed to work as a Java process which uses its standard input and standard output to communicate with buyer agents. In adopting such a simple convention, we allow agent programmers to build their agents in any programming language which allows for firstly spawning the remote control as a children process and then plugging to it.

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 9-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 *remote control* indexed by different bidding protocol types (f.i. $P = \{P_{DBP}, P_{English}, \dots\}$).
- $\mathcal{B} = \{b_1, \dots, b_p\}$ is a finite set of identifiers corresponding to all participating buyers.
- $\mathcal{S} = \{s_1, \dots, s_q\}$ is a finite set of identifiers corresponding to all participating sellers.
- $\mathcal{F} = [\mathcal{F}^1, \dots, \mathcal{F}^n]$ is a sequence of n descriptors. Each $\mathcal{F}^i$ specifies the way auction $\mathcal{A}^i$ is dynamically generated.

$\mathcal{T}$						
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	$U[1, 15]$	$U[1200, 2000]$	$U[1500, 3000]$	$U[0.4, 0.5]$
		tunafish	$U[1, 15]$	$U[800, 1500]$	$U[1200, 2500]$	$U[0.3, 0.45]$
		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	$\langle E_b, E_s \rangle = \langle \beta^r_a [\frac{4}{r^2}(\pi(r - \pi))], \varnothing\rangle$					

Table 3: UPC Tournament Descriptor

- $E = \langle E_b, E_s\rangle$ 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 $\mathcal{A}_i$ along 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.
- $\mathcal{A} = [\mathcal{A}^1, \dots, \mathcal{A}^n]$ is a finite sequence of *Auctions.*

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

At this point, we would like to emphasize that throughout this section we have pursued to obtain a formalization which can be easily generalized to serve also as the means of describing tournaments in an auction house where goods are auctioned under the rules of different bidding protocols, which we envision as the natural evolution of our current system.

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 [21].

We opted for a simple scenario characterized by the tournament descriptor in table 3. 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 $\mathcal{A}^i$ ($\mathcal{C}_1^i(b) = 50000\ \forall b \in \mathcal{B}_1^i$) 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 3. 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 [21].

- 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_a^r), 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 evalutation functions) favoured voratious 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 [4]) devised variants to these tournament 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[8].

5 Related and Future Work

Several attempts have been made by researchers in electronic commerce concerning the proposal of electronic marketplace architectures [2, 17]. Such efforts share the common goal of building electronic markets where both buying and selling agents can trade on behalf of their users. Nonetheless up to our knowledge no remarkable proposal has been made in order to to provide 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 test-bed environment that can be used to test and tune trading agents, FM97.6, that happens to be built on an actual agent-mediated auction house, FM96.5, and can thus be taken as a training or tuning test-bed for FM96.5 based auctions. But because of the agent-architecture neutrality of our test-bed, and because of the primal nature of the bidding mechanism, such training and tuning can be applied in other agent-mediated trading forms. The AuctionBot initiative [22, 20] has lately moved in a similar direction, and now provides an API to build trading agents that can participate in an AuctionBot-designed auction.

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*[6] that encourages both AI researchers and robotics researchers to make their systems play soccer; or the *AAAI Mobile Robot Competition*[7] 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 [16]. One can hardly argue that any of these agent competitions involve features that are directly relevant for agent-mediated trading. Along these lines, 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 [19], or in the use of artificial intelligence to study economic markets [14]. 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] and tools and techniques for deploying and testing trading-agent shells, strategies and actual agents. Secondly, FM96.5 and FM97.6 will be made to evolve towards other (even more flexible) agent-mediated institutions and test-beds. In particular, we expect to release in the near future an agent-mediated auction house where goods can be traded under the rules of any auction protocol (not only Fishmarket variants). Later on, we shall release an agent-mediated market place where other forms of price-fixing mechanisms (double auction, discounting, open negotiation) can take place. Both directions pursue to develop practical and trust-worthy forms of agent-mediated trading, but pretend also to help us advance in the understanding of the fascinating reality of situated reasoning.

6 Acknowledgments

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] 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.

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

[3] 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.

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

[5] P. Garcia, E. Giménez-Funes, 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)(submitted paper)*, 1998.

[6] 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.

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

[8] 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.

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

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

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

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

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

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

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

[20] 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.

[21] The FishMarket Project. http://www.iiia.csic.es/-Projects/fishmarket.

[22] The Michigan AuctionBot Project. http://auction.eecs.umich.edu.