Designing and Implementing Electronic Auctions in a Multiagent System Environment

Davide Roggero[†], Fioravante Patrone[‡], Viviana Mascardi[†]
[†]DISI, Università di Genova,
Via Dodecaneso 35, 16146, Genova, Italy
davide@unige.it,mascardi@disi.unige.it,
[‡] DIPTEM, Università di Genova,
P.le Kennedy - Pad D, 16129, Genova, Italy
patrone@diptem.unige.it

***Abstract*— Agent-Mediated Electronic Commerce is gaining a wide consensus both from the academia and from the industry, since it provides the right abstractions, models and tools to face the challenges that electronic commerce raises. According to C.Sierra, e-commerce can be described as *organization* + *mechanism* + *trust*, where mechanism is concerned with the rules that govern the interaction among agents in such a way that certain properties can be guaranteed.**

This paper describes the design and implementation of a library of customizable agents for simulating auction mechanisms. The purpose of the library is to provide a support to the correct engineering of mechanisms in the e-commerce setting, by providing a flexible tool for the quick prototyping of realistic auctions to the auctions' developers. The auction mechanisms that are included in our library respect the *Revenue Equivalence Theorem*, one of the most important theorems of the formal theory of auctions.

Keywords. Auction Theory, Electronic Auction, Multiagent System

I. Introduction

Information and Communication Technology (ICT) is currently considered as one of the forces that can deeply influence and transform human society. Many people agree on the important role played by ICT in productive growth and international competitiveness, thanks to reduction of transaction costs, support to efficient management, and exchange of a wide amount of information. This happens especially for commerce, radically changing the way enterprises and companies work. For example, large on-line selling enterprises use the Internet strategically to improve service quality, process speed and for cost savings, whereas small enterprises use electronic commerce (e-commerce) primarily to increase their customer base and make themselves known. In order to offer answers suitable to the currently open challenges in the e-commerce area, like business process outsourcing, marketing of agricultural exports and online dispute resolution, new technologies are required. Agent-Mediated Electronic Commerce (AMEC) is the most recent (and one of the most promising) technology born with the purpose of facing the e-commerce challenges.

In his paper "*Agent-Mediated Electronic Commerce*" [1], C.Sierra asserts that e-commerce can be described by the following equation:

$$\textit{e-Commerce} = \textit{organization} + \textit{mechanism} + \textit{trust}$$

In this paper we deal with the second element of the sum: *mechanism*. Mechanism design is concerned with establishing the rules that govern the interaction among agents in such a way that certain properties (such as stability, or equilibrium) can be guaranteed. The definition of the rules of the game determines how the interaction will take place and, based on the assumption of rationality for the agents, tries to achieve a desired behaviour by them, possibly corresponding to dominant strategies.

In order to provide a support to the correct engineering of mechanisms in the e-commerce setting, we have developed a *library of customizable agents for simulating auction mechanisms* with the goal of providing a flexible tool for the quick prototyping of realistic auctions. The design of the auction mechanism exploits the AUML language (`http://www.auml.org/`) for defining the interaction protocols between a bidder and an auctioneer, while the prototyping phase is carried out by exploiting the tools offered by the DCaseLP environment [2], [3]. We have designed and implemented the auction mechanisms that are included in our library, that can be downloaded from the DCaseLP home page, `http://www.disi.unige.it/person/MascardiV/Software/DCaseLP.html`, in such a way that they respect the mathematical theory behind auctions. In fact we have based our work on the results obtained in the Auction Theory area, and in particular on the well-known Revenue Equivalence Theorem (RET, described in [4], [5], [6]). By carrying out many experiments run under different initial conditions, we have experimentally validated that the mechanisms developed as part of our library respect the RET.

The paper is structured in the following way: Section II summarizes the main mathematical results behind auction theory; Section III describes the design of the mechanisms that we provide in our library, while Section IV illustrates their implementation and shows that it respects the RET. Section V concludes.

II. Auction Theory

The typical situation where an auction is suitable to allocate some goods can be described in this way: on one side of the market (the offering side) a monopolist wants to sell some goods; on the other side there are two or more potential buyers. It is implicitly assumed that the monopolist will choose the procedure (or mechanism) to allocate the goods, but this does not necessarily mean that he can extract the entire surplus, because he does not know the buyers' true evaluation of the goods.

There are many different auction mechanisms that can be classified according to their features [7], [8]. The first distinction can be made between *open* and *sealed-bid* auctions. In the *open* auction mechanisms, the seller announces prices or the bidders call out the prices themselves, thus it is possible for each agent to observe the opponents' moves. The most common type of auction in this class is the *ascending* (or *English*) auction, the well-known procedure typical of artwork auctions, where the price is successively raised until no one bids anymore and the last bidder wins the object at the last price offered. Another diffused type, the *descending* or *Dutch* auction works in exactly the opposite way: the auctioneer starts at a high price and then lowers it continuously (notice that this kind of auction essentially belongs to the "sealed bid" type). The first bidder that accepts the current price wins the object at that price. The *sealed-bid* auction mechanisms are characterized by the fact that offers are only known to the respective bidders (as the name suggest, offers are submitted in sealed envelopes). In the *first-price* sealed-bid auction each bidder independently submits a single bid without knowing the others' bid, and the objects is sold to the bidder who made the best offer. First-price auction are especially used in government contract. Another widely used and analyzed auction in this class is the *second-price* sealed-bid auction, that works exactly as the first-price one except that the winner pays the second highest bid. This auction is sometimes called Vickrey auction after William Vickrey, who wrote the seminal paper on auctions [9].

An auction mechanism is said to be *efficient* if and only if the offered object is always given to the buyer with the highest valuation for it. The four basic mechanisms just described (English, Dutch, first price sealed-bid, and second-price sealed-bid) are all efficient (assuming that bidders are rational and their bids are in equilibrium). Assuming that an auction mechanism is efficient, an interesting problem is to establish which procedure can guarantee the maximum revenue to the seller. Economic theory provided some fundamental and surprising results on the equivalence (at equilibrium) of the expected revenues of various auction mechanisms. Vickrey provided the earliest conceptualization and results in [9], which was, together with [4], a major factor in his 1996 Nobel prize. Myerson [5] and Riley and Samuelson [6] showed that Vickrey's results apply very generally.

The Revenue equivalence theorem is stated in the following way:

(**Revenue equivalence theorem**) Assume that

- there are N risk-neutral potential buyers (i.e., they are indifferent between, for example, playing a lottery which gives 0 euro with probability $1/2$ and 1000 euro with probability $1/2$, and gaining 500 euros for sure);
- the independent private-value model applies (i.e. each bidder: 1) has a private evaluation of the object, unknown to the other bidders; 2) believes that the other bidders' evaluation of the object can be described by a probability distribution that is identical for all the bidders; 3) believes that there is statistical independence between the individual evaluation);
- the buyers are symmetric (i.e. they cannot be distinguished one from the other).

Then all the efficient auction mechanisms guarantee to the seller the same expected revenue, and each bidder makes the same expected payment as a function of his valuation.

This theorem implicitly defines a wide class of equivalence of auction mechanisms (in terms of expected revenue) and both the first-price and the second-price sealed auction belong to this class. This could be surprising: in the first-price sealed auction, the winner pays the price that he called while in the second-price one the winner pays a price equal to the highest bid made by the other players. The fact is that the players' best strategy in the second-price auction is to bid their true valuation while in the first-price auction the bidders face a trade-off between lowering the offer (thus obtaining a better payoff in case of success) and getting higher probability of success (but paying more for the object). It is optimal for a bidder in a first-price auction to bid his valuation minus a discount: the revenue equivalence theorem states that this discount compensates exactly (in expected value) the reduction of payment caused by the second-price mechanism.

III. Analysis and design of the auction mechanisms

Considering that the Dutch auction mechanisms is completely equivalent under any value model to the first-price sealed-bid auction, we have implemented the remaining three standard mechanisms described in Section II: English, first-price sealed-bid and second-price sealed-bid mechanism. Since the English mechanism is the most complex (and interesting) one among the three, in this section we concentrate on it, by analyzing the communication protocol that governs the interaction between auctioneer and bidders, and by describing the design of the agents' behavior. The details on the sealed-bid mechanisms can be found in [10].

Each auction mechanisms require two types of agent at least:

1) The *Auctioneer* agent that puts items on sales, receives offers, distributes information on what is going on and decides the auction winner

2) The *Bidder* agents that try to buy the items on sale by evaluating newly acquired information and sending offers

We implemented an English auction mechanism for a single indivisible object. Our analysis of this auction led us to the definition of the interaction protocol in Figure 1. This protocol is described using AUML that extends UML with agent roles, multithreaded lifelines, extended message semantics, parameterized nested protocols, and protocol templates.

In the registration phase, p Bidder agents ask to be registered in the Auction by sending a message with a communicative act `request`, the Auctioneer can accept the request (sending back a message of `confirm` to each accepted agent) or deny the request (with a `refuse` communicative act).

Once the registration time is over, the Auctioneer sends an `inform` message to the n registered agents specifying its reservation price, this warns the Bidder about the minimal acceptable offer. Then the Auctioneer sends another `inform` communicative act to start the offering phase.

In the offering phase, the Bidder agents send `propose` messages that contain offers: every time a Bidder x offers a bid that is better than the highest received bid, the Auctioneer sends an `inform` message back to x to notify that is winning the object Then the Auctioneer has three possibilities:

1) to broadcast to all n participants what is the new highest offer
2) to broadcast to all n participants that there is an extension to the original auction span
3) to declare the end of the offering phase

All these possibilities are communicated by `inform` messages and each of them causes different behaviors of the Bidder agents: the first two messages leave to the Bidders the chance to make new offers (shown in the Figure 1 by the loop back arrows), while the last message moves the communication protocol to the next phase, the object attribution.

The object attribution phase of an English auction mechanisms with continuous bidding is simple because the evaluation of the best bid is completely done in the offering phase, so the winner agent is already determined once that phase is finished. Thus, the Auctioneer broadcasts an `inform` message with the name of the winner, then wait for a `confirm` message.

The next step in our analysis was to describe the behavior of Auctioneers and Bidders of each auction mechanism, and we decided that Pascal pseudo-code was the right tool for this activity. Since, for space constraints, we cannot include the pseudo-code that we have defined for both the auctioneer and the bidders, we only include - as an example - a portion of the code defining the core activity of the bidder's offering stage. This is the activity of the Bidder after receiving the information about the current bid from the Auctioneer. The bidder 1) updates its value model according to it; 2) evaluates its new offer, according to the (updated) value model; and 3) offers a new bid, if its new offer is better than the current one.

```
if (receive('inform','present-bid(Bid)',auctioneer) and not
I-win)
then
        Present-bid:=Bid;
        update-value-model(Bid,[],Value-model);
        New-bid:=eval-offer(Present-bid,Value-model,
                     Bidder-number,End-auction);
        if(better(New-bid,Present-Bid)
        then
               send('propose','offer(New-bid)',auctioneer);
        endif
...
```

IV. Implementation of the Library of Auction Mechanisms

Each auction mechanism in our library is constituted by the implementation of the code of the auctioneer, and the code for the bidders. The files that contain the auctioneer code are named 'AUCT_*type*.pl' while the files with the bidders code are named 'G*X*_*type*.pl', where *type* refers to the auction mechanism and *X* is an integer. The agents are implemented in tuProlog in the DCaseLP environment based on the JADE platform (`http://jade.tilab.com/`). The code of the agents can be downloaded from the DCaseLP home page (`http://www.disi.unige.it/person/MascardiV/Software/DCaseLP.html`), together with the packages that constitute DCaseLP, and the Master thesis by D. Roggero [10] (in English) that describes the application.

The characteristics of the auctioneer that can be customized by the user are:

Registration time. It is the duration of the registration phase in minutes.

Acceptance of registration. The user can customize the predicate that define the rules by which an agent can be accepted as a bidder. These rules can be private of the auctioneer or depend on an external reputation system.

Auction time. It is the duration of the offering phase in minutes.

Alarm time. Only in the English auctions. It is the interval of time at the end of the offering phase, during which any new offer will trigger the extension of the auction time.

Extension time. Only in the English auctions. It is the interval of time that the auctioneer adds to the auction time if any offer has arrived during the alarm time.

Wait time. It is the interval of time that the auctioneer waits for a message of confirmation from the winning bidder.

Reservation price. The reservation price is the lowest bid accepted by the auctioneer to sell the object.

Bid comparison. The auctioneer must choose if a new bid is better than another. The user can customize this feature to reflect the preferences of the auctioneer over offers.

Attribution of the object. In case the auction ends with two or more bidders owning the best offer, the auctioneer must decide who is the real winner using a lottery whose definition can be customized.

As far as the bidders are concerned, the characteristics of the bidders that can be customized by the user are:

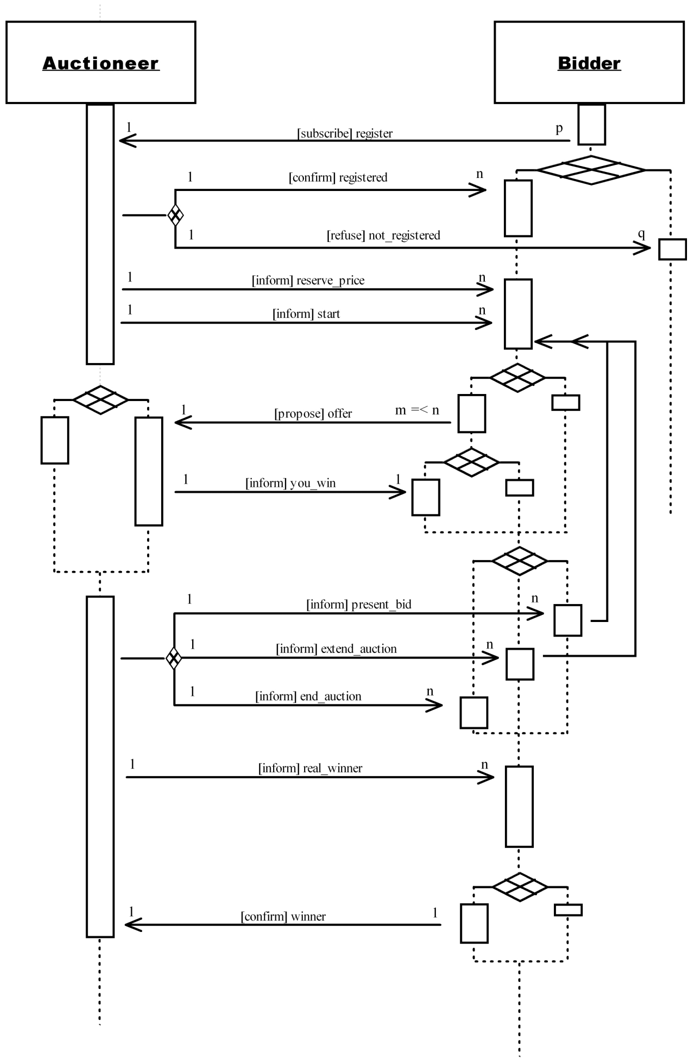


Fig. 1. English auction mechanism with continuous bidding

Value Model. The value model of a bidder determines its object's monetary worth . The default value model implemented in our bidders is the private one: the bidder asserts a static value for the object. The user can customize this feature, defining a predicate that calculates the object's worth for the bidder using both private and public information. Notice that the software works independently of the assumption of private value, so that other agents' bids can be informative.

Strategy. The strategy of a bidder determines the value and the time of its offers. It depends mainly on the value model and on other bidder's behavior, but other aspects can be considered, like time and information from sources external to the auction.

In order to test our implementation, we ran all the mechanisms of our library with the same parameters to show that they satisfy the RET discussed in Section II.

In each auction, the name of the auctioneer agent is of the form '`auct_mech`' where $mech$ is the type of auction mechanism while the names of the bidder agents are of the form '`gX_mech`' where X is a integer in the interval $[1, 4]$ and $mech$ is the type of auction mechanism. The complete address will be `name@Vento:1099/JADE` since all the agents are deployed on a single computer called '`Vento`'. In the text output, each agent's output can be recognized by its address at the beginning of the line.

In the following, we only discuss the outcomes of our experiments with the English auction; a complete account of the implementation of the sealed-bid mechanisms can be found in [10]. Game theory suggests that, in an English auction with private values, the best strategy for any bidder is to remain in the competition, making small raising, until the price reaches his evaluation of the object, then drop out of the auction: in this way the winner will get the object at a price just a little higher than the second-highest private value. In our implementation, each bidder uses this strategy (implemented in Prolog):

```
eval_offer(New_bid) :-
        object_value(Value), present_bid(Present_bid),
        Present_bid < Value, New_bid is Present_bid + 1,!.
```

The bidder makes the evaluation of a new offer each time the auctioneer inform all the participants that the present winning price is changed. With this strategy, every bidder (except the one who made the last winning bid) makes the same offer as soon as they get the message: being in an English auction with continuous bidding, the auctioneer will accept the first arrived offer as the temporary winning bid (in fact, it bested the old one by 1 euro[1]) and discard all the subsequent identical offer made by other bidders.
The messages on lines 318, 319, 320, 321 of Figure 2 are `inform` messages that contain the present temporary best offer. As soon as they get this message, all the bidders (except `g1` who was the present winner) send `propose` messages (lines 322, 323, 324) containing new identical offers calculated with the `eval_offer` predicate seen before. The auctioneer gets the first offer (line 322), sees that it is better than the last winning bid and take it as the new winning bid (at line 325): when the auctioneer examines the other offers, it finds that they are *equal* to the present winning bid, so it discards them.
In Figure 3, we can see that, at the end of the auction, agent `g4_eng_c` wins with an offer of 301. We can also note that the auction time was expired before the real conclusion of the competition and this has triggered the extension mechanism that permits to establish the final price of the object; in fact, at the end of the auction time the winner was agent `g3_eng_c` with a bid of 300.
From the theorical point of view, if an English auction has no limits of time, the best selling price will emerge for sure, but a more realistic approach suggests to limit the duration of the auction, like we did. This can create consequences: for example, if the private value of at least two bidder is much bigger than the reservation price, the extended time could expire before the competition is over, thus denying the individuation of the best offer and not attributing the object to the bidder with the highest private value. This fact is inevitable but we realized that, in this implementation, the order in which the bidders register to the auction influences the order in which they bid, thus giving advantage to a bidder that registered earlier than another: this leads to an unfair attribution of the object. Hence, we decided to implement a round-based version of the English auction that could change this unfair behavior. The English auction with rounds behaves like the continuous one, apart form the attribution stage, where a fair approach to determine the winner is adopted (more details can be found in [10]).

We run all four auction mechanisms implemented under common conditions to verify the RET. Examining all the auction run, we can notice that every one of them terminated with agent `g4_eng_r` as winner, thus demonstrating to be efficient auctions. The two sealed bid mechanisms individuated an auctioneer's revenue of 300, while for the two English mechanisms the revenue was of 301: this difference is caused by the *discrete bidding* strategy that our bidders use. In fact, if the strategy in the English auctions had been to raise the last winning price by 0.1, then the difference between the revenues would have been not 1 but 0.1; if the strategy had been to raise the price by 0.01, then difference would have been 0.01; and so on. Thus, we can say that our implementation verifies the RET.

[1]We assume that the granularity of the bid is 1 euro.

V. Conclusions and future work

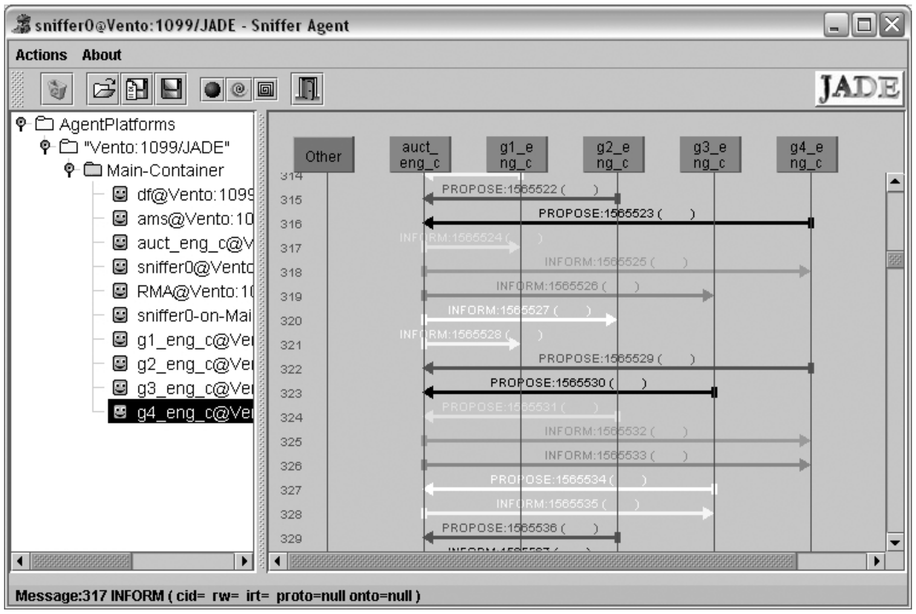
In this paper we have described the work done to develop a library of agents for simulating auction mechanisms. We have analyzed and implemented four different mechanisms:

- the first-price sealed-bid auction mechanism,
- the second-price sealed-bid auction mechanism,
- the open English auction mechanism with continuous bidding,
- the open English auction mechanism with rounds.

For each auction mechanism, the interaction between auctioneer and bidder has been analyzed and an Interaction Protocol has been produced. In the design phase, the internal behavior of each type of agent has been studied and their customizable features have been highlighted. Each agent's behavior has been written down in a pseudo-Pascal listing. Finally, each agent has been implemented with tuProlog in the DCaseLP environment, thus achieving the goal of providing customizable tools for simulating auction mechanisms. For example, by modifying the reservation price of the English auctioneer and the value model of the related bidders, it is possible to simulate English multi-dimensional auctions. Moreover, DCaseLP and JADE supply many tools for analyzing message exchange and debugging agent behaviors, thus helping the user in the analysis of the bidders' strategy.

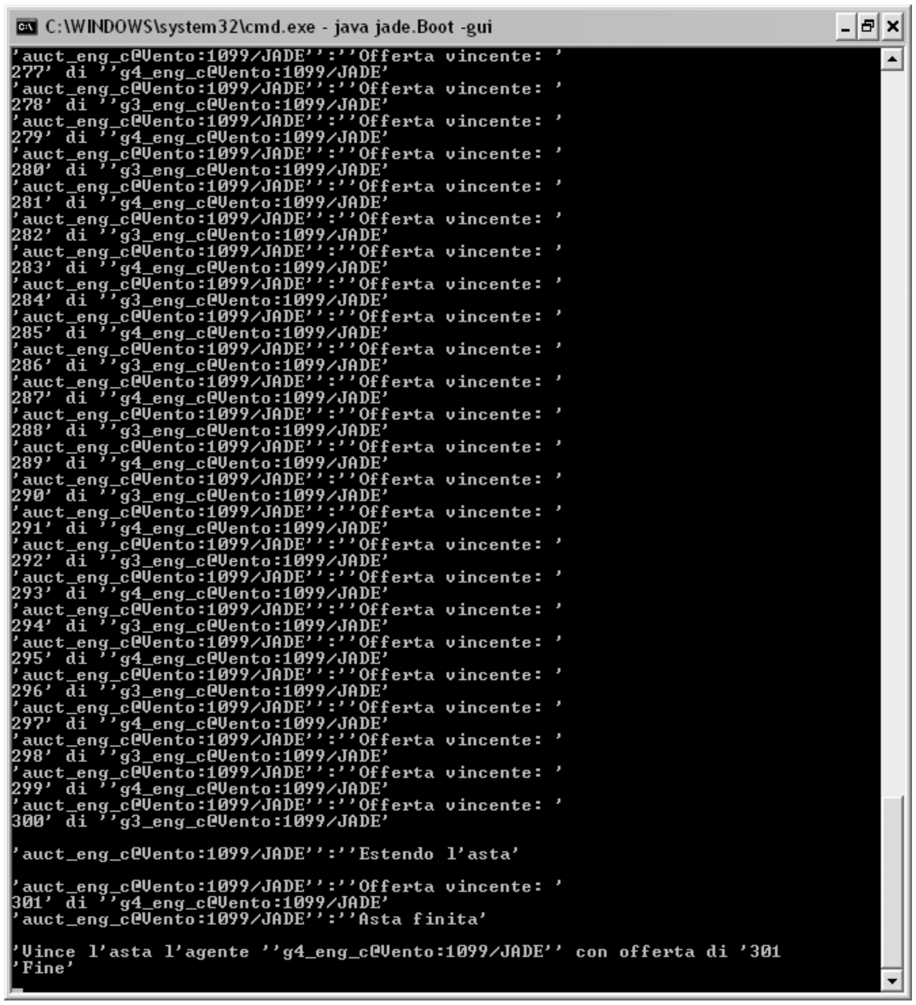
We have ran all the implemented mechanism using risk-neutral bidders with independent private value taken from a uniform distribution. Under these hypothesis, Game Theory demonstrated that there exist an optimal bidder's strategy for each of the implemented mechanism: we programmed our test bidders with these strategies and we verified that all the simulated auctions gave the same revenue to the auctioneer and the same payoff to the bidders. The fact that RET is satisfied (up to some error clearly due to discretization) can be seen as a check for the correctness of the implementation.

As far as the related work is concerned, today there are many commercial and research applications for implementing real electronic auctions, or simply for simulating them. The Trading Agent Competition (`http://www.sics.se/tac/`), for example, is carried out every year, in order to promote and encourage high quality research into the trading agent problem, while the well-know electronic commerce portals, eBay (`http://www.ebay.com/`) and Amazon (`http://www.amazon.com/`), demonstrate the commercial applicability of the research on agent-mediated auctions.



Message:317 INFORM (cid= rw= irt= proto=null onto=null)

Fig. 2. English auction with continuous bidding: offering phase



```
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
277' di ''g4_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
278' di ''g3_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
279' di ''g4_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
280' di ''g3_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
281' di ''g4_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
282' di ''g3_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
283' di ''g4_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
284' di ''g3_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
285' di ''g4_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
286' di ''g3_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
287' di ''g4_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
288' di ''g3_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
289' di ''g4_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
290' di ''g3_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
291' di ''g4_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
292' di ''g3_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
293' di ''g4_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
294' di ''g3_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
295' di ''g4_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
296' di ''g3_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
297' di ''g4_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
298' di ''g3_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
299' di ''g4_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
300' di ''g3_eng_c@Vento:1099/JADE'

'auct_eng_c@Vento:1099/JADE'':''Estendo l'asta'

'auct_eng_c@Vento:1099/JADE'':''Offerta vincente: '
301' di ''g4_eng_c@Vento:1099/JADE'
'auct_eng_c@Vento:1099/JADE'':''Asta finita'

'Vince l'asta l'agente ''g4_eng_c@Vento:1099/JADE'' con offerta di '301
'Fine'
```

Fig. 3. English auction with continuous bidding: shell output.

Despite the wide range of available applications, we decided to implement and distribute our own, in order to implement (as part of our future work) some extensions to the basic auction mechanism that may benefit from the reasoning capabilities provided by our tuProlog agents. In particular, we would be interested in:

- analyzing and implementing other less common but interesting auction mechanisms, like double auctions and all-pay auctions. The last kind of auctions is quite common, often at a non formalised level: just considering lobbying activities, or competition for a given (potential) boy/girl friend...
- building a society of agents, with "advertising" agents that contain information (like starting and ending time, type of object on sale, type of auction mechanism) on the auctions that are going to be held and "searcher" agents that look for interesting auction using user's preferences and informs the bidder.
- implementing a reputation system, where reliable "notarial" agents calculates the reputation of the subscribers using other agents' opinions and past behaviors and making it public to the agent community.

References

[1] C. Sierra, "Agent-mediated electronic commerce," *Autonomous Agents and Multi-Agent Sytems*, vol. 9, pp. 285–301, 2004.

[2] E. Astesiano, M. Martelli, V. Mascardi, and G. Reggio, "From Requirement Specification to Prototype Execution: a Combination of a Multiview Use-Case Driven Method and Agent-Oriented Techniques," in *Proc. of SEKE'03*, 2003, pp. 578–585.

[3] I. Gungui and V. Mascardi, "Integrating tuProlog into DCaseLP to engineer heterogeneous agent systems," in *Proc. of CILC 2004.* Available at http://www.disi.unige.it/person/MascardiV/Download/CILC04a.pdf.gz.

[4] W. Vickrey, "Auction and bidding games," in *Recent advances in Game Theory*. Princeton University Conference, 1962, pp. 15–27.

[5] R. Myerson, "Optimal auction design," *Mathematics of Operations Research*, vol. 6, pp. 58–73, 1981.

[6] J. Riley and W. Samuelson, "Optimal auctions," *American economic review*, vol. 71, pp. 381–92, 1981.

[7] P. Klemperer, *Auctions: Theory and practice*. Princeton University Press, 2004.

[8] V. Krishna, *Auction Theory*. Academic Press, 2002.

[9] W. Vickrey, "Counterspeculation, auctions and competitive sealed tenders," *Journal of Finance*, vol. 16, pp. 8–37, 1961.

[10] D. Roggero, "Aste elettroniche in ambiente multi-agente," Master's thesis, DISI, University of Genoa, Italy, 2005.