Information Systems Frontiers 2:1, 65–84, 2000

IBIZA: E-market Infrastructure for Custom-built Information Products

Ashish Arora
The Heinz School of Public Policy and Management, Carnegie Mellon University, Pittsburgh, PA 15213
E-mail: Ashish@andrew.cmu.edu

Gregory Cooper
Center for BioMedical Informatics, University of Pittsburgh, Pittsburgh, PA 15260
E-mail: Gfc@cbmi.upmc.edu

Ramayya Krishnan
The Heinz School of Public Policy and Management and Institute for Complex Engineered Systems, Carnegie Mellon University, Pittsburgh, PA 15213
E-mail: Rk2x@cmu.edu

Rema Padman
The Heinz School of Public Policy and Management, Carnegie Mellon University, Pittsburgh, PA 15213
E-mail: Rpadman@cmu.edu

Abstract. **The merger of electronic commerce, intelligent agent and distributed computing technologies over TCP/IP-based platforms enables the creation of electronic markets in new types of products featuring both human and software agents as actors. One such example is a market in custom-built information products. These are information products that have been constructed to meet specific requirements provided by the consumer. Examples include custom research reports, analysis, and computational objects. How should these markets be designed? What are the market mechanisms that should be used to coordinate the interactions between the actors? What should be the decision strategies employed by the software agents that participate in the market? IBIZA is a computational workbench that enables designers to create and simulate electronic markets in information products. It provides a repository of software agents, bidding strategies, brokering strategies and market mechanisms. Using the repository, designers can instantiate particular designs of electronic markets and conduct experiments to study the impact of design decisions on desired objectives. In this paper, we focus on the key technical and economic issues encountered in the design of IBIZA. We illustrate using examples from our work on designing a software agent-based electronic market for automated model development.**

Key Words. **agents, e-commerce, simulation environments, analysis and design**

1. Introduction

The ubiquity of the TCP/IP platform and rapid developments in intelligent agent and distributed computing technologies has led to the deployment of electronic markets in a variety of domains. The oft quoted examples are e-retailers such as Amazon.com featuring a variety of approaches from fixed price to auctions, business to business marketplaces such as metalsite.net, and information aggregators and suppliers such as forrester.com. However, new and innovative marketplaces are being introduced. Consider the case of flashline.com (see www.flashline.com). Flashline is a marketplace for software

components. Software components in the two dominant component standards—Activex/Com and Java Beans—are available for sale for standardized tasks.

In addition, Flashline also hosts a marketplace for custom software development. Buyers with the need for software components use Flashline to post requests for quotation. Programmers registered with Flashline can search and review the requests and bid to develop the components. As marketplaces for information products such as Flashline evolve, interesting research issues arise with respect to the design and evaluation of these marketplaces. A sample set of questions are as follows.

- What should be the structure of the marketplace? Should it be brokered or not?
- What should be the market mechanism used? Should auctions be used or should the market employ fixed prices?
- What implications does the choice of market mechanism have on quality of the product developed in the marketplace?
- How should the broker choose participants in a market session? What are the implications of the brokers choice on quality of product and on profits for the broker?
- How should a seller decide if it should participate in a market session? What are the implications with respect to its profit?
- What are the impacts of cost structures specific to custom-built information products on market efficiency?

An evolving body of inter-disciplinary work has attempted to answer these questions in a variety of e-commerce contexts. A representative sample of this large literature is given below. Bakos (1997) and Lee and Clark (1996) use economic analysis to study the reduction in transaction costs on electronic marketplaces. Bakos (1998) analyzes the implications of the reduction in buyer search costs for electronic marketplaces. Arunkundrum and Sundararajan (1999) describe issues of firm profitability in electronic secondary markets and Lee (1998) tests hypotheses about whether the realized price is lower in electronic marketplaces using data from AUCNET, a used car auction market. Finally, Chirchu and Kauffman (1999) present case studies of organizations engaging in what is referred to as the IDR (intermediation, disintermediation, reintermediation) cycle Sarkar et al. (1997). Our work on IBIZA is complementary to these works but distinctive in its use of computational modeling as a tool to design, and evaluate through simulation, electronic markets for custom information products. It provides a repository of software agents, bidding strategies, brokering strategies and market mechanisms. Using the repository, designers can instantiate particular designs of electronic markets and conduct experiments to study the impact of design decisions on desired objectives. In this manner, IBIZA provides the infrastructure required to design and evaluate electronic markets.

The rest of the paper is organized as follows. We begin in Section 2 with an illustrative example of an electronic market for automated model development and use it to highlight the requirements of a computational modeling environment for the design of markets for custom built information products. In Section 3, we discuss the design objectives of IBIZA with respect to standards, market mechanisms, intelligence in agents such as brokers and sellers, and technology required to deliver custom information products. In Section 4, we present the IBIZA architecture and the models that were used to analyze and create solutions to meet the design objectives. In Section 5, we discuss the user interface to IBIZA and discuss interaction with IBIZA to create and simulate e-markets. We conclude in Section 6 with a discussion of important implementation decisions and issues that need further work.

2. *Illustrative Scenario*

The context: A researcher wants to develop a predictive model to assist a clinician in treating community acquired pneumonia. She has the MCHD data set that contains information on inpatients discharged from 78 hospitals in 23 states between July 1987 and December 1988. The data set tracks over 250 pieces of clinical information referred to as key clinical findings (KCF). The KCF's include patient demographic characteristics, history, physical examination findings, lab and radiology results collected during up to 3 reviews during the hospitalization. The model should predict mortality of hospitalized patients from their findings at initial presentation with pneumonia. Such predictions would be useful to clinicians since they must decide about where to treat patients with pneumonia. Treating patients at home is less expensive than treating patients at the hospital, and patients with milder cases of the disease are likely to be more comfortable at home. See Cooper et al. (1997) for additional details.

Interaction with an IBIZA e-market: Consider the interactions that the researcher (the buyer) has

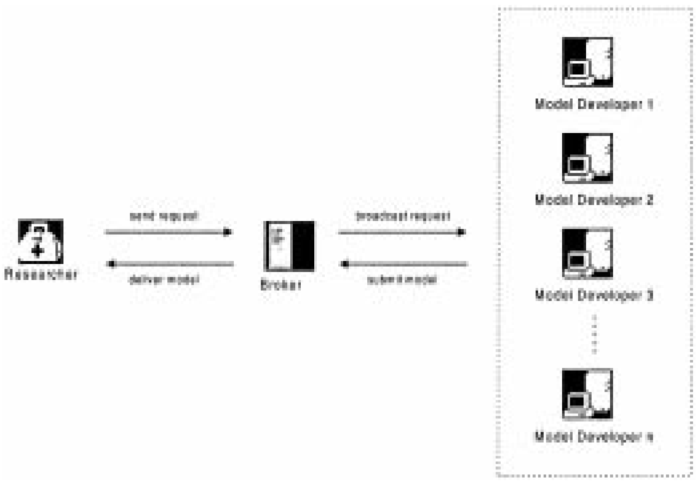


Fig. 1. *An IBIZA e-market for custom model development.*

with the other actors in a particular e-market created using IBIZA (see Fig. 1). The market consists of a broker with sellers that are software agents who have the assets (i.e., machine learning methods) to build the model (see Fig. 1) using data supplied by the buyer.

Data Preparation and request to broker: First, the researcher prepares the custom data set to be used in model creation by selecting 75 out of the 200+ variables that she thinks are relevant to the learning task. She divides the 2,000 records into two parts, a training data set consisting of 1,200 records and an evaluation data set consisting of 800 records to test the quality of the models that will be developed in the market. Using the consumer interface applet, she then poses a request to the broker consisting at the very minimum of the following.

- The training and evaluation data sets (URLs of these resources)
- Meta data associated with data giving information about the types and value ranges of variables
- The maximum price the user is willing to pay
- The minimum accuracy the user wants the model to provide
- The time window (deadline) available for model development

Broker interaction with Sellers: The broker processes the user request (e.g., partitions the available time window into a time window available to build the model and a time window available to evaluate the model) and executes the market mechanism. In this market, the market mechanism is a simplified type of auction called a **prize scheme**. In

Fig. 2. *An ask object in IBIZA.*

Fig. 3. *The IBIZA bid object.*

this scheme, the broker broadcasts to all the registered seller agents a modified copy of the user request object referred to as an **ask** (Fig. 2).

The object contains all the information in the user request object with the exception of the evaluation data set and the price that the user is willing to pay. In this market (as in all IBIZA markets), the seller agents are software agents. The seller agents independently (using the decision making capability that they are endowed with by the market designer) evaluate the **ask** object and determine if they want to bid taking into account resources required to construct the model, time window available and their likelihood of winning. If they choose to bid, each seller agent has to construct a model using the training data set pointed to in the **ask** object using its machine learning technique. This custom built model is encapsulated as an applet and submitted to the broker as the **bid** (Fig. 3).

The bids (or more specifically, the models submitted with the bid) are evaluated in terms of predictive quality using the evaluation (hold out) data withheld by the broker from the seller agents by a product evaluation service coordinated by the broker. The model that has highest quality is deemed to be the winner in the prize scheme. The seller agent that developed the model wins the prize (this could be set to be the amount the user was willing to pay) and the model is transmitted to the user.

2.1. IBIZA Components

- **Seller agents:** In a market for predictive model development, the producers/sellers are the model developers with learning algorithms that have the capability of generating predictive models based on learning problem specification. Depending on the objective of the market designer, seller agents can be endowed with specific machine learning capabilities, decision making strategies of varying complexity related to processing bids, and can be adaptive (learning over time with respect to bidding strategy). IBIZA provides ''pluggable'' components that can be used to create seller agents with the complexity desired by the market designer.

- **Buyers:** In a market for predictive model development, buyers are agents with the data and the problems that need the model building capabilities of the agents with machine learning techniques. While buyer agents can be software agents, in the example, the buyer agents are human. Buyer agents are required to provide meta data about the data set and supply preference information relating to dimensions such as desired predictive quality of models, time windows and willingness to pay.
- **Broker:** As is the case with seller agents, brokers are software agents in IBIZA e-markets. Market designers can endow brokers with different market mechanisms and decision making capabilities related to executing the market mechanism. Currently, two market mechanisms are supported in IBIZA. They are the generalized auction and the prize scheme. Additional details about these two mechanisms and the way in which they determine the structure of the interactions between the broker and seller agents is described in Section 3.
- **Model evaluation agent:** Model evaluation agents rate the quality of the product submitted by the developers in the market. While the broker could provide this service, IBIZA allows market designers to create specialized agents for this purpose.

2.2. Rationale for the IBIZA architecture

As illustrated in Fig. 1, electronic markets that are created within the IBIZA environment are brokered markets. The rationale for markets and a broker coordinated market architecture can be explained in terms of transaction costs, asset specificity and product complexity (Williamson, 1975).

- While the cost of creating the seller agent asset—the learning algorithm—is high, the marginal cost of producing the information product (i.e., the model) being transacted is comparatively low. Since production costs and communication costs are low, a market-based approach to acquiring these products is indicated (Malone, 1987).
- The processes used to create the information products have low asset specificity. Methods such as learning algorithms are not platform or machine specific. Neither are they function specific in the sense that a learning algorithm could be applied to any learning task despite the fact that some methods may be better suited than others to some tasks. Markets are indicated when the assets required to produce the product have low asset specificity.
- Products can be evaluated using a well specified method. In the case of models, their predictive ability can be accurately determined using the evaluation data. Since a standardized and objective method for evaluating products is available, markets can be used as the coordination mechanism between the buyers and sellers.
- Expressing product requirements can be complex. The availability of market-wide standards for expressing information relevant to product development (e.g., meta data) and standards for encoding the product (e.g., use of an applet with a well understood API) is critical. By enforcing standards and offering matchmaking services, electronic brokerages (Malone, 1987) reduce coordination costs.

3. Design Objectives

The principal objective in IBIZA is to provide market designers with an environment with which they can quickly create and simulate alternative market designs for custom-built information products. The requirements of creating such infrastructure can be summarized into the following categories.

- **Standards**

Standards are required at different layers to ensure that the IBIZA design objectives can be met. We consider standards at three layers.

- **Network architecture:** Given that IBIZA is a distributed system with a decentralized architecture, there is the need to standardize on a network architecture and the appropriate distributed object technology. This is required to ensure the ability to evolve as new features, agents, and market mechanisms are added to the architecture as well as to ensure interoperability between market components. A discussion and comparison of the architectures we considered (e.g., Java distributed object architecture and CORBA) is in Section 6.2.

- **Data encoding and formats:** While network architectures standardize communication protocols (e.g., Java Remote Method Invocation), they do not standardize the format used to encode the messages being exchanged between components. For instance, what should be the format of the meta data and the format of the data set distributed to seller agents that want to bid in a market for predictive model development? The choice of the format has implications for barriers to entry for new seller agents that want to participate in the market. Given that predictive model development was the illustrative application, we chose the UC-Irvine Data Encoding standard. It is employed by a widely used public domain machine learning library distributed by Silicon Graphics called MLC++ (see .http://www.sgi.com/Technology/mlc/source.html.). As IBIZA evolves to support the design of e-markets in other information product application domains, similar standards will be required. Alternatively one could study the impact of having multiple competing standards on the market and study the factors (such as network externalities (Shapiro and Varian, 1998)) that lead to a dominant standard.
- **Product encoding:** The information product developed by the seller agent should be encoded in a format that will be permit it to be evaluated and used by the consumer. This requires standardization of the interface (API) to the product and the technology used to encode it. The latter issue is a function of the assumptions made about the heterogeneity of the operating environments in place with the user as well as at the evaluator. For instance, if the assumption is made that distributed object technology available is the Java object model, products could be encapsulated as applets with an agreed to interface and communicated between agents using object serialization and remote method invocation (Arnold and Gosling, 1996). Alternatively, a similar standardization is possible using the Microsoft Distributed Object Model (Eddon and Eddon, 1999) or CORBA (Seigel, 1996). IBIZA is implemented using the Java distributed object model. Our rationale for this choice is discussed in Section 6.2.

- **Representation of user requirements**

The tradeoffs and the relative importance placed by consumers on attributes such as quality, price and time are important determiners of how seller agents decide to bid in the marketplace. In addition to standardized ways of encoding and communicating this information to the broker and seller agents, alternative means of expressing utility functions need to be supported within IBIZA in order to study the impact of the information provided by the consumers on the efficiency of the market.

- **Market mechanisms**

Market mechanisms play a central role in IBIZA. The choice of the mechanism determines the information processing requirements of the broker as well as that of the seller agents. In IBIZA, two market mechanisms are currently supported, namely a generalized auction procedure and simplified scheme referred to as a prize scheme.

- **Auction:** In the IBIZA auction procedure, a market session begins with the broker notifying all potential participants (it can be all sellers known to the broker or just a subset of them that the broker selected) about the request received from the buyer. Each seller then decides whether to participate in this auction, and generate a bid to get a contract to develop the product. Upon receiving all bids from the sellers, the broker chooses a subset of the bidders (can be one or more participants) that it will award the contract to develop the product. Those that are awarded contracts can then develop the product and submit the finished product to the broker for final evaluation. Upon receiving the finished product, the broker determines which product best meets the buyer's requirement, and delivers that product to the buyer and awards the seller of that product accordingly. Other participants that were awarded contracts but were not selected may receive consolation awards when the product meets or exceeds the specification in their bid. Conversely, they may also be required to pay a penalty fee (when the product does not meet the specification as promised in the bid).

- **Prize:** The prize mechanism is a simplified version of the auction procedure and works like a contest. All participants are required to develop the product based on a given specification, and the best submitted product is awarded the prize. In this case, the broker plays a role of contest organizer that advertises the contest to all potential participants, and those interested in participating would develop a product and submit it before a deadline set by the broker. After the deadline, the broker determines the best product and awards the winner with the prize (part of the price that the buyer pays to the broker).
- **Broker decision making strategy:** As noted above, the market mechanism determines information processing requirements of the broker. In the auction procedure, the broker selects participants (seller agents) for a market session. Several alternatives strategies may be chosen by the market designer who would like to investigate the impact of broker decision making strategies on market efficiency. For example, a simple approach would be to allow every seller registered with the broker to participate in a market session for a user request. However, this could become resource intensive when there are a large number of potential participants in the market resulting in large increases in the broker's coordination cost. An alternative approach is to enable the broker to select a subset of all registered model developers to participate in the market session. However, the choice of sellers has implications for product quality. Mapping product quality into indicators of market performance is contingent on the objective function of the broker. For example, the market designer could design and evaluate alternative market structures in IBIZA—one with a profit maximizing broker and the other with a welfare maximizing broker. Details of a broker decision making strategy in IBIZA are given in Section 4.2.1.
- **Model developer (seller) strategies:** As with the broker, the seller agent has to process communication from the broker to determine if it wants to participate in a market session. Alternatives range from simple to complex. An example of a simple strategy would be to participate in every market session regardless of whether the model developer's capability matches with the desired product requirement. However, this approach could become problematic both to the broker as well as the model developer. As the number of model developers increases, the broker will have to process a large number of bids that have a low probability of yielding high quality products. Similarly, given limited resources, the model developer should participate only in market sessions in which it has a high probability of winning. Several factors have an impact on model developer strategy. An important one is the time available for bidding and model building. Since any time taken to assess if a bid should be made is time taken away from building the product required to be part of the bid, the available time window set by the broker is an important determinant of seller strategy with implications for quality as well. The market designer can design and test the impact of the relationships between these factors by endowing seller agents with differing participation strategies. We present details of a seller agent participation strategy in Section 4.2.2.
- **Product evaluation:** As discussed earlier, product evaluation is an important pre-condition to creating e-markets in IBIZA. While the dimensions along which product evaluation is to be done is an important consideration, the agency used to conduct the evaluation is equally important. This latter choice has an impact on how seller agents choose to allocate the time window within which they are required to bid (i.e., build a model) and have their bid evaluated. If a market for evaluation services were available, seller agents could independently determine how to allocate the time available to them. Alternatively, the broker could decide the time available to bid (referred to as the build window) and the time available to evaluate the bid (referred to as the eval window). Market designers can test the impact of a product evaluation mechanism and the means used to set bid, build, and evaluation windows on factors such as profitability of seller agents and market efficiency.

4. IBIZA Architecture and Components

Responding to the issues discussed in the previous sections, IBIZA was designed using a object-oriented analysis and design techniques (Rumbaugh et al., 1991; Fowler and Scott, 1998; Gamma et al., 1995). While space limitations prevent a detailed discussion of these models, in this section we present a series of time sequence diagrams to illustrate the interactions between the principal components in IBIZA. We begin with a brief introduction to the software components in IBIZA (see architecture Fig. 4) and their mapping to the market actors discussed in Section 2.

- Client: The Client is the component that serves as the starting point of a market session. Through the Client module, user requests that initiate market sessions are created. The Client module communicates with the Data Repository and the Broker. In a simulation, the client encodes the preferences and the willingness to pay of the consumer.
- Data Repository: The Data Repository provides a storage facility for the data sets that are to be used to create the product (i.e., the model) desired by the consumer. Through the client module, the data set to be submitted to the Broker can be customized (e.g., recall the customization in the illustrative scenario of the MHCD data set). During a market session, the Broker can retrieve the data set from the data repository.
- Broker: The Broker component coordinates the market and initiates and terminates market sessions. Upon receiving a request from the Client, the Broker initiates a market session and notifies the Model Developers (the seller agents). As discussed in the previous section, the Broker executes the market mechanism, processes bids, has them evaluated by the Model Evaluators and identifies the winning bid to conclude the transaction.
- Model Developer: The Model Developer is the seller agent and offers model development services. Each Model Developer in the market employs a unique combination of learning algorithm and computing resources (software and hardware used to develop models) that sets its product apart from the products of other Model Developers. Each Model Developer can employ its own unique strategy to determine whether it should participate in a market session.

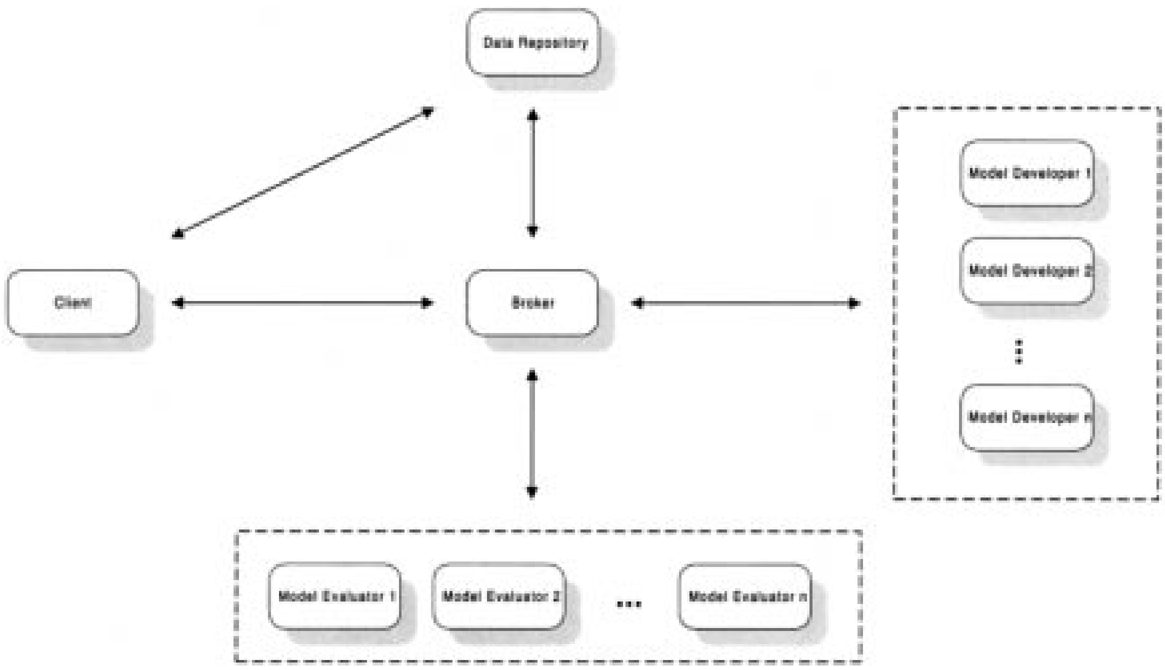


Fig. 4. Ibiza architecture diagram.

- Model Evaluator: The Model Evaluator is a component that acts as secondary service provider to the Broker.

4.1. Interaction models

The interaction between IBIZA components can be divided into two phases—data set preparation and market session. We present time sequence models that define the structure and sequence of interactions in IBIZA. These time sequence models provided the specification used to implement IBIZA.

Dataset preparation

- The Client requests a listing of the dataset available at the Data Repository. The Data Repository returns a list of dataset along with the description of each dataset.
- The Client selects a dataset to be used and requests the metadata of that dataset from the Data Repository. The Data Repository returns the metadata that describes the characteristics of the dataset.
- The Client uses information in the metadata to determine which variables of the dataset are to be used in the creation of the model. The Client also determines the size of the custom dataset to be created by the Data Repository.
- The Client requests a custom dataset creation to the Data Repository. Upon receiving the request, the Data Repository generates the custom dataset, stores it and returns back a string that uniquely identifies the custom dataset. The Client uses the string in its request to the Broker or whoever else that will be using the custom dataset for model development (Fig. 5).

Market session

The Market Session is initiated by the request to the Broker.

- After a custom dataset has been created, the Client prepares a request by asking the user many parameters that describe user's requirements (e.g., maximum price user is willing to pay, deadline, minimum quality level, etc.). The Client then submits the request to the Broker.
- Upon receiving the Client's request, the Broker prepares a Market Session by retrieving the custom dataset needed for model development from the Data Repository using the identification string (the return value from the Data Repository to the Client after custom dataset creation in the dataset preparation sequence).
- The Broker starts the market session using one of the market mechanisms (i.e., auction or prize). The sequence of interaction from here until the point where the Model Developer delivers their products depends on which market mechanism is used. Both market mechanisms feature the use of an Ask Object which encapsulates the communication from the broker to the model developers (seller agents). The object diagram illustrates the structure of the Ask Object:
- The Ask Object is a container class used to package all the parameters necessary for the execution of a model construction task. The Model Developers utilize the information within the Ask Object to determine the feasibility of taking on the task and the attributes within the Ask Object constitute the determinants in this decision. The attributes of the Ask Object are a design parameter controlled by the market designer. The extent of the information revealed by the Broker to the seller agents can have a significant impact on

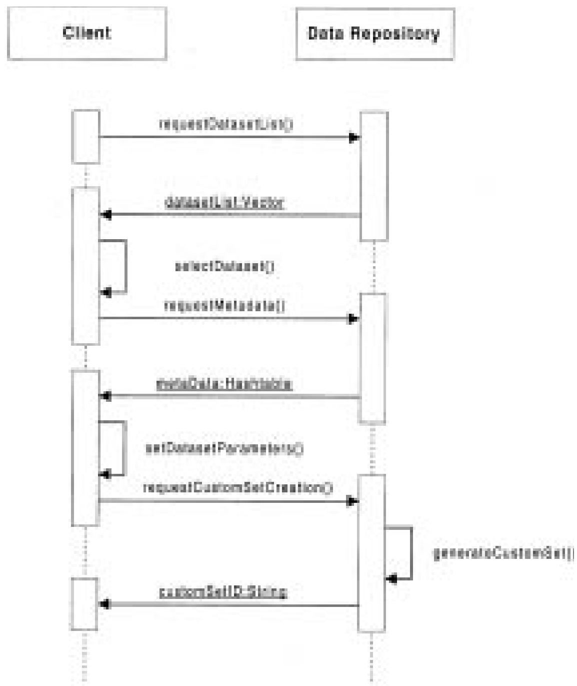


Fig. 5. *Dataset preparation sequence diagram.*

factors such as market efficiency and profitability of the seller agents. For example, each object can assess its probability of winning in a Market Session as 1/*N*, where *N* is the number of market participants.

- When the Broker receives a model from the Model Developer, it will then send the model and the evaluation data set to the Model Evaluator.
- The Model Evaluator tests the predictive accuracy of the model on the evaluation data set and reports the results back to the Broker.
- Upon receiving evaluation results for all submitted models, the Broker ranks the models and determines using the user specified preference function the model that best meets requirements.

The best model is delivered to the Client, and the participating Model Developers are notified about the result of the Market Session (Fig. 6).

Differences between the two market mechanisms
The auction mechanism includes the prize mechanism as a special case. The sequence of interactions is presented first for the auction mechanism. The special case of prize is then presented. The time sequence diagrams illustrating the difference is in Fig. 7.

- The Broker determines the potential participants of the market, and sends request for bids to these participants. Details of one method available to market designers in IBIZA is presented in the following section.

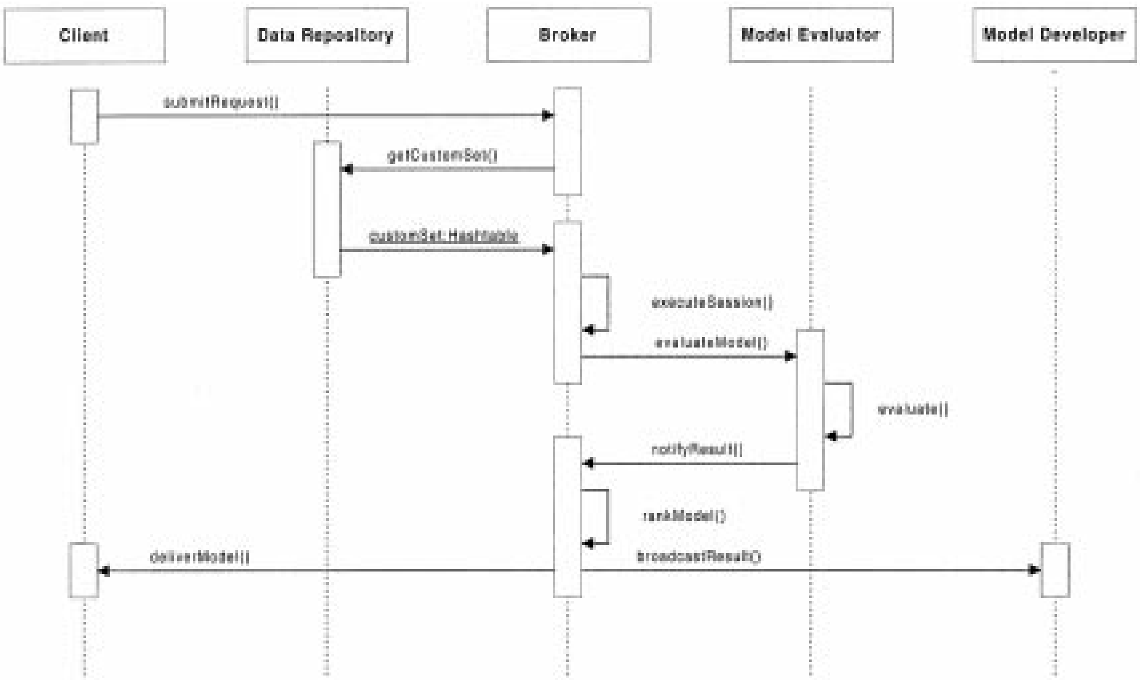


Fig. 6. *Market Session sequence diagram.*

- Upon receiving the request for bid, the Model Developer uses its strategy to determine whether it is worthwhile to participate in the market session. If the decision is not to participate, the Model Developer drops the request and does not participate in the market session. If it chooses to participate, the Model Developer prepares a bid and submits it to the Broker before the deadline for submitting bids has been reached. The Bid Object is shown below.
- The Bid Object, like the Ask Object, is a container class that stores information relevant to the Broker's decision as to which Model Developers to award contracts. If awarded a contract, the Model Developer then builds the model using its own learning algorithm. When the model is done, it is delivered to the Broker for evaluation. The sequence from this point on is the same as the last four steps in the Market Session interaction sequence above (Fig. 6).

Prize

The prize mechanism is simpler than the auction scheme and consists of the following steps.

- The Broker determines the potential participants of the market, and sends announcement about a ''contest'' to meet user's request to those participants.
- Upon receiving the announcement, the Model Developer decides using its strategy whether to participate in the ''contest'' or not.
- If Model Developer decided to enter the ''contest,'' it proceeds with building the model.
- After the model is finished, it is submitted to the Broker for evaluation.
- The sequence from this point on is the same as the last four steps in the Market Session interaction sequence above.

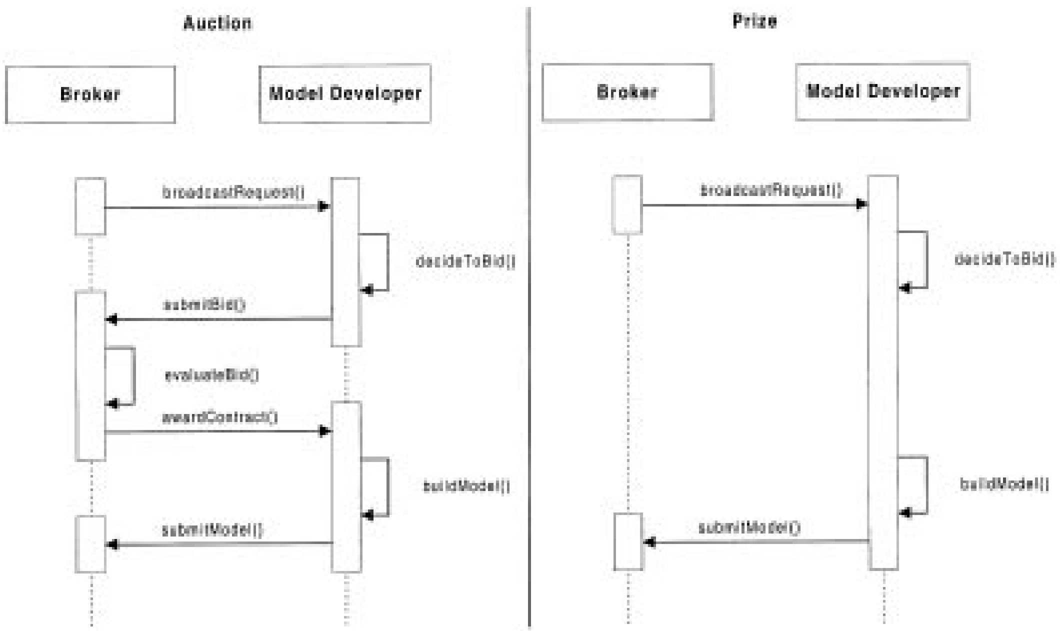


Fig. 7. *Auction vs. prize sequence diagrams.*

4.2. Decision strategies of Broker and seller agents

4.2.1. The Broker's decision problem. As discussed in Section 3, a key step in the execution of an auction procedure is the decision by the Broker to award contracts in response to bids. Strategies to solve this decision problem range from simple strategies that would invite all bidders to more complex strategies that take benefit and costs into account. In IBIZA, market designers should be offered a set of alternatives including the alternative of implementing their own decision rules. In the current implementation of IBIZA, two decision procedures are offered. We provide a sketch of one of the decision models available to a market designer in IBIZA.

Each bid from a Model Developer specifies a promised quality and a desired price for the service. Let Q^* denote the promised quality, and X^*, the compensation required for the construction of that model. Each bid is then a 2-tuple (Q^*,X^*). Given N such bids, the Broker's task is to select a number $K \leq N$ of model developers to award contracts. Note that the actual quality of the model produced by a Model Developer is not known to either the developer or the broker. Instead, there is some distribution of quality conditional on the nature of the task, i.e., on the size and characteristics of the dataset and possibly also of the time and resources devoted by the Model Developer to the task. There are several alternative objective functions that could be used. One that is used in the model we present maximizes expected quality. This in turn becomes a metric that will be used to monitor the performance of the Broker and the market.

$$\text{MAX}(E(Q(K)) - \textbf{\textit{Sum of Costs}})$$

where $Q(K)$ denotes the highest quality attained among the K models submitted. Technically, $Q(K)$ is largest Order Statistic in the actual quality of models. The ''Sum of Costs'' includes the required compensation and any overhead costs incurred by the Broker.

Distribution of $Q(K)$

There does not exist any general expression for $Q(K)$ since the probability distributions of quality of the various Model Developers differ. However, one can use the theory of order statistics of identically distributed variables as an approximation. The largest order statistic from any sample of identically distributed random variables can have only three possible limiting probability density functions, if such limiting distributions do exist (David, 1981). The most widely used of these is the so called extreme value distribution.[1]

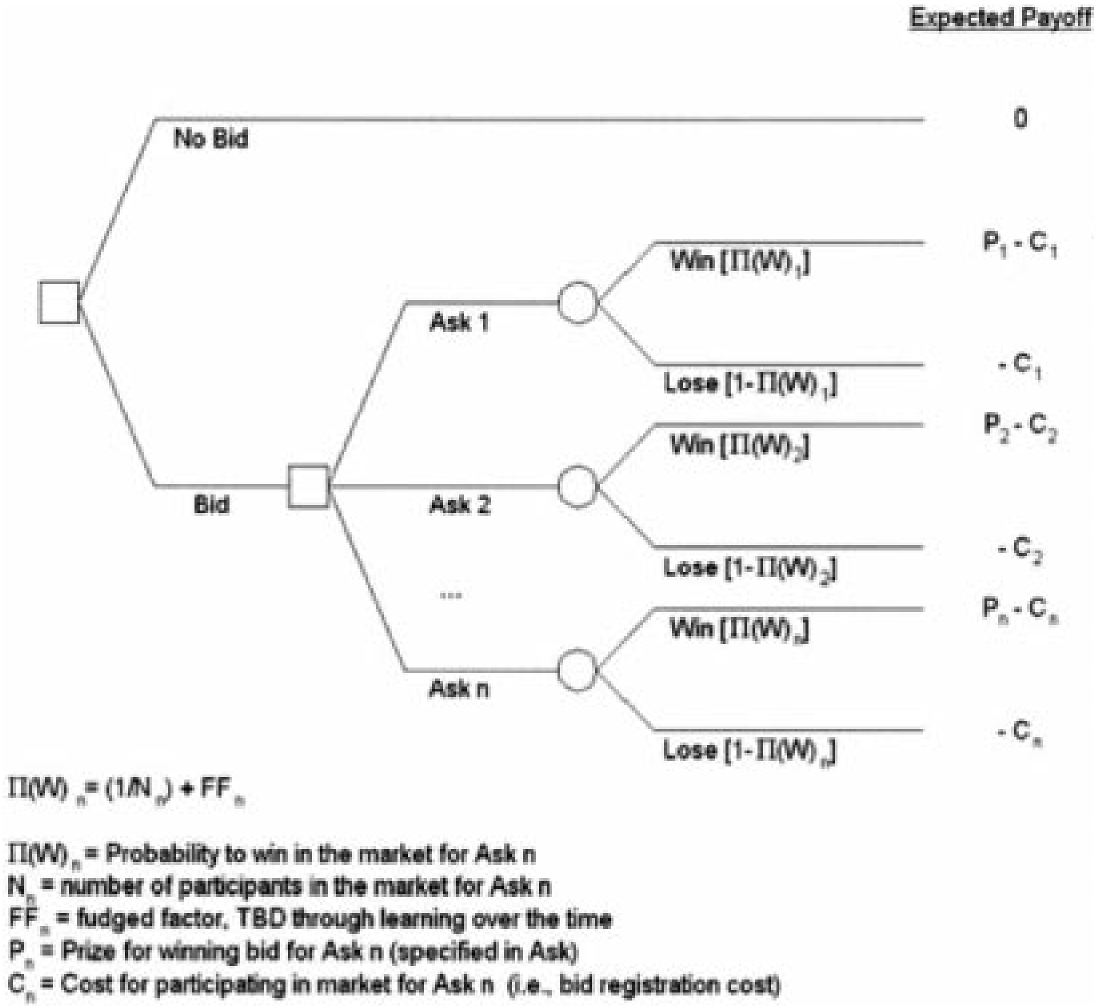
The extreme value distribution has the attractive property that its largest order statistic is also has an extreme value distribution. Thus, if Q_i, the quality of an individual Model Developer has an extreme value distribution with the c.d.f. given by $\exp^{-(e-(x-\eta)/\beta)}$, with $E(Q_i) = \eta + \beta\gamma$ and $\text{var}(Q_i) = (\beta^2)(\Pi^2)/6$, then $Q(K)$has an extreme value distribution as well with a c.d.f. given by $\exp^{-(\exp-(x-\eta-Ln(K))/\beta)}$ with $E(Q(K)) = \eta + \beta \cdot \gamma + \beta \log\ K$, where γ is Euler's constant. Note that β indicates the uncertainty associated with the distribution of Q. This value can be more accurately determined using historical information. A reasonable value of β needs to be provided by the market designer.

Sum of costs

The sum of costs term in the objective function is the total cost of having K Model Developers participate in a Market Session. The cost of each Model Developer participating can be expressed as the sum of the actual compensation paid to the Model Developer, and any overhead cost incurred by the Broker. This overhead cost can be in the form of an out-of-pocket cost payable to Model Developers in return for participating in the auction.

Thus, $\text{Cost} = X_i + O$ where $X_i = X_i{}^* + \theta(Q_i{}^* - Q_i)$ and O is the overhead cost. The symbol, θ, denotes the price-quality tradeoff as expressed by the Client and Q is the minimal quality level acceptable to the Client. Expanding this expression, we get $C_i = X_i{}^* + \theta Q_i{}^* - \theta Q_i + O$.

The sum of costs is therefore: $\Sigma_i C = \Sigma_i X_i{}^* + \theta\Sigma_i Q_i{}^* - \theta\Sigma_i Q_i + K^* O$, so that the expected total cost is $E(\Sigma_i X_i{}^* + \theta\Sigma_i Q_i{}^* - \theta\Sigma_i Q_i + K^* O) = \Sigma_i X_i{}^* + \theta\Sigma_i Q_i{}^* + K^* O - \theta\Sigma_i E(Q_i)$. Note that $\Sigma_i X_i{}^*$ and $\Sigma_i Q_i{}^*$ can be easily computed from the bids submitted by the Model Developers, but the broker needs to be supplied with estimates for $E(Q_i)$. For simplicity, we assume that all the Model Developers are identical so that $E(Q_i)$ is equal to μ and $E(\text{total cost}) = K(X^* + \theta Q^* + O + \theta\mu)$.[2]



Calculation of *K*, the number of contracts to be awarded

Rewriting the objective function as:

$$\mathrm{MAX_K}(\eta + \beta.\gamma + \beta \mathrm{logK} - K(X^* + \theta Q^* + O + \theta\mu)).$$

The first order condition for an interior maximum is given by

$$\beta/K = (X^* + \theta Q^* + O + \theta\mu),$$

so that $K = \beta/(X^* + \theta Q^* + O + \theta\mu)$.
The only unknowns in that expression are β and μ, which can be approximated using historical information. Market designers need to set values for β and $E(Q)$. Under more general assumptions, one cannot get an analytic solution. However, the general approach outlined here, namely comparing the additional gain in the expected quality of the largest order statistic to the additional expected cost can be implemented to find the optimal value of *K*.

Discussion

While the model presented encapsulates a particular decision strategy, empirical testing is needed to determine if the computational effort (time) to solve this decision problem (once every Market Session) is justified. Would a simpler strategy that took less effort and time lead to the same or higher expected quality of the products built in the market. An example of such a simple strategy is one of inviting every registered model developer to submit a bid. This is just the sort of question that IBIZA simulations can answer to help market designers with the design of brokers.

4.3. The seller agents decision problem

Recall that each seller agent in IBIZA is a developer of a customized information product. In our illustrative example, the model developer is the seller agent. Each Model Developer is endowed by the market designer with a learning algorithm (its asset that is used to create models on demand), and a participation strategy that is used by the agent to evaluate an Ask Object to determine if it will submit a bid in a market session. As shown in Fig. 7, both the auction and the prize mechanism require agents to make a decision about whether they will participate (either by submitting a bid that indicates a willingness to participate as in the auction mechanism or by submitting a bid that contains the model as in the prize mechanism). Further, these decisions need to be made under tight time constraints since any time required to determine if the agent should participate is time taken away from model building and evaluation. As with the Broker's decision problem, there are several alternative strategies ranging from simple to complex that are possible. The Model Developer could decide to bid in every market session under the assumption that the marginal cost of submitting a bid is lower than the cost of determining whether it should bid or not. A more complex strategy is one that makes a decision based on a model of expected profit from a bid. A simple decision tree-based model that is implemented in IBIZA is shown below. The model assumes that the agent has a collection of Ask Objects (e.g., those that need a model built within a specified time period) that need to be evaluated. Associated with each ask object is probability of winning and an expected profit. Clearly, probability of winning depends on the competition and an analytic model to assess this probability is extremely complex. A naive approach for an agent would be to assume that if there are N registered developers (something made known by the broker to all participants), every agent has an equal chance of winning ($1/N$). An adaptive approach could be used to modify this probability by a factor (see FF in the model) based on performance in the market so that the agent develops a model of which Ask Objects demand capabilities best suited to its asset (i.e., learning method). For instance, an agent with a neural network method might be best suited to building models for data sets with large number of categorical variables. Variations of this decision tree-based approach could take the time available for model building into account. In this case, the decision would be to determine both the set of asks to respond to as well as the order in which models required by the ask need to be developed. Each of these alternative strategies can be made available as components on IBIZA to be used by a market designer to determine the factors that indicate the strategy that would be the best to adopt.

As of this writing, three participation strategies are provided in IBIZA. The first strategy is to always participate in a market session regardless of the consequences. The second is to use the simple decision tree when many participation opportunities are presented, and the third is to use the result of regression analysis done on the historical participation results. These strategies are designed as modules that can be "plugged in" to the Model Developer as needed.

Learning algorithms

In our prototypical application, the participation strategy and the learning algorithm are the assets of the model developer. Five algorithms were selected based on their availability and simplicity in implementation for use in prototypical application. The algorithms selected were ID3 (decision tree algorithm), Naive Bayes (probabilistic algorithm), K-Nearest Neighbor (instance-based algorithm), Random (randomly guess the answer), and Constant (always select the value that showed up the most historically) (Mitchell, 1997). From an economic point of view, the agents employing these methods are horizontally differentiated. For example, K-Nearest Neighbor method is efficient in constructing models. But models constructed using the method take a long time to evaluate. The predictive quality of the models are in general quite good. In contrast, the Random and Constant algorithms have very fast construction and evaluation time, but the predictive quality of their models is not as good. This diversity permits market designers to test the relationship between participation strategies of seller agents and knowledge of the capabilities of their assets.

5. Interacting with IBIZA

IBIZA is a distributed system implemented using Java making extensive use of Java RMI (remote method invocation). It runs on Windows NT and Solaris

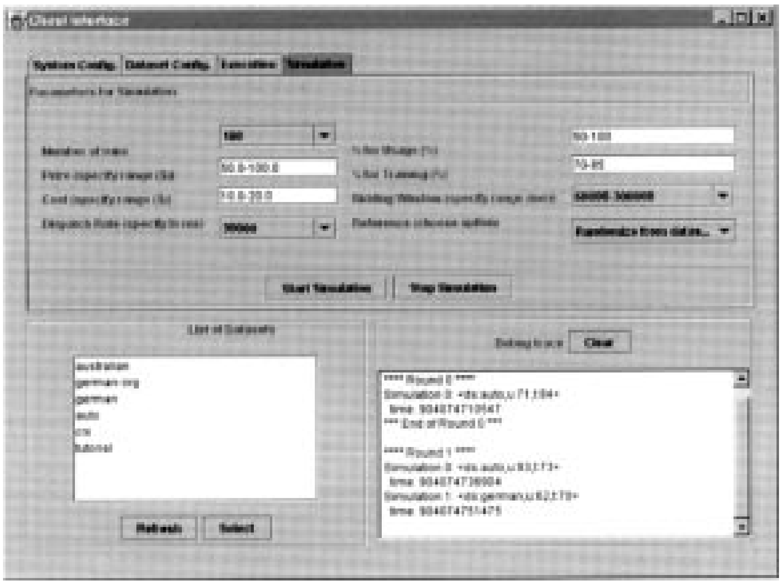


Fig. 8. *Screen shot from the Client GUI.*

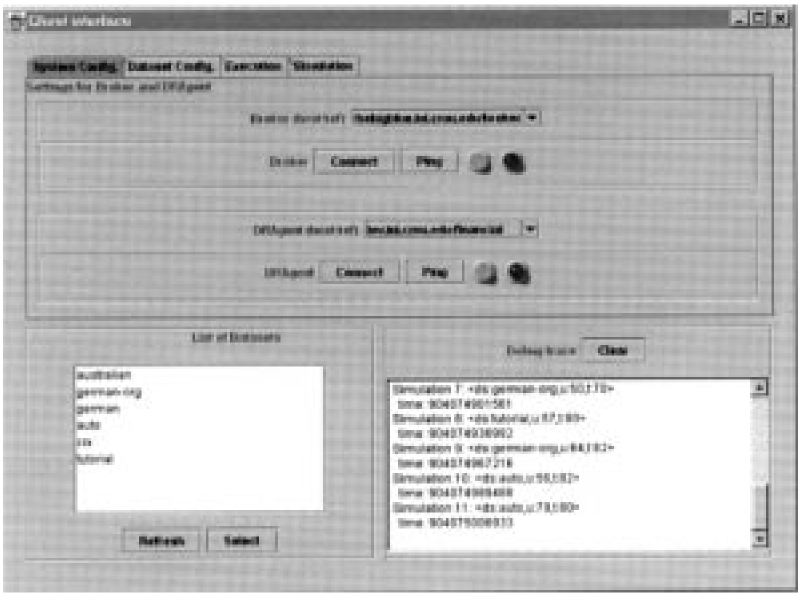


Fig. 9. *Screen shot from Client GUI's connection panel.*

Platforms. Interaction with IBIZA is through three graphical user interface panels implemented using the Java AWT (abstract windowing toolkit) and JFC (Java Foundation Classes called Swing). The user interfaces are presented and discussed in this section.

Intuitive client interface

While much of our discussion of IBIZA has focused on its use as a simulation environment for electronic markets with software agents, IBIZA can also host simulations where some of the agents are human. To facilitate interaction with human agents, the Client module in IBIZA features two interfaces: an interface that permits the user to interact with a Broker and the Data Repository to create custom data sets and user requests (Fig. 8), and an interface to establish and monitor connections to the Broker and Data Repository (Fig. 9).

Monitoring the broker
The interface to the Broker is designed to be used by the market designer and it sets parameters such as the market mechanism to be used, the Model Developer agents and the Model Evaluator agents that should be included in a simulation. The interface also allows the designer to view, archive and export detailed data and traces collected during the simulation (in a tab delimited format) to facilitate further analysis.

6. Discussion

We conclude with a discussion of several issues that were first mentioned in Section 2 based on the lessons learned from implementation and experimentation with IBIZA. The issues we will address fall into two categories: standards and infrastructure.

6.1. Standards

Standards are an important issue in an environment such as IBIZA. Standard data representation formats are required to communicate to the seller agents in the market. Since the agents produce an information product in the form of a mobile applet, a standard API is required both to test the product as well to enable any future use of the product. Since the objective of IBIZA is to enable the study of different broker and seller agent strategies in a market, standard API's are required in IBIZA to make it extendible. We designed standards based on a careful requirements analysis. Space limitation prevent a detailed discussion of all standards. In the following we present our choice of a particular data representation format for use in IBIZA. The interested reader is referred to a detailed technical report on the design of IBIZA (Setiawan and Teo, 1999).

Dataset representation format
Since the prototypical application was in a market for machine learning services, our task was to choose a format for data submitted by a user to the Broker and subsequently to seller agents in the market. The two obvious approaches were to use an existing format or to define a completely new format. With the first approach, time can be saved since the format is available and all that is needed is to provide translation services to those agents that cannot process the format. This approach makes sense when there is a dominant format (Table 1).

Table 1. *Using available format vs. defining new format comparison*

	Using available format	Defining new format
Time consumption	Low	High
Public Dataset Conversion	May be necessary	Required

After studying several public data repositories, one format stood out: the data format used by University of California, Irvine, a public Data Repository that has been widely used by many in the machine learning academic community. UC-Irvine's format is also very straightforward; each dataset consists of two parts: a metadata file and a data file. The metadata file contains information regarding the characteristics of the dataset (i.e., how to interpret the data), such as the possible values of target functions, number of independent variables, type of independent variables, etc. An example of a metadata file is given in Fig. 10 The first line declares the possible prediction outcome. The second line consists of the names of the independent variables. The following lines consist of the possible values that each of the independent variables can take (with keyword ''continuous'' if the independent variable takes continuous value).

The data file contains the actual data itself. This file can be further split into two parts, for training and evaluation. An example data file is given in Fig. 11.

```
accept, reject
hair color, car, age, own house
hair color: black, brown, blonde, grey.
car : cheap, average, expensive.
age: continuous.
own house: yes, no.
```

Fig. 10. *Example of UC-Irvine metadata contents.*

```
black, cheap, 20, yes, reject.
black, cheap, 18, no, reject.
grey, average, 23, no, reject.
blonde, cheap, 31, yes, reject.
grey, expensive, 25, yes, accept.
```

Fig. 11. *Example of UC-Irvine data file content.*

6.2. Infrastructure

Model shipping and delivery

As we have discussed, the product transacted in the market is a model. How should this model be created, submitted for evaluation and eventually delivered to the buyer? How should the model be communicated from one player to another. Clearly, these choices have implications in relation to barriers to entry into the IBIZA market as well as to switching costs for agents choosing to leave the IBIZA-based market to participate in another market for information services. We compared four alternatives (Table 2) and chose the Java distributed object environment as the technology infrastructure for model shipment and delivery.

6.3. Scalability of Broker implementation

IBIZA is a broker-centric architecture. The Broker manages all session objects and is potentially a point of congestion. Consider the state of each session managed by the Broker.

- Number of Model Developer in auction.
- Details on the user's request.
- The market scheme employed.
- Auction scheme used.
- State of each Model Developer with respect to each session.
- The models returned from the Model Developer.
- Results of the evaluation of their models.
- Connection to Model Developers.
- Connection to Model Evaluators.
- Which Model Developer won the auction.

The memory requirements for the Broker scale up very quickly with many concurrent sessions, as can be seen from the amount of information that the Broker has to keep track of in each session. We did study the possibility of serializing this state information to disk. While this has been applied effectively in other memory management schemes like demand paging (Stallings, 1997), it is difficult to implement it in the Broker without additional study of the spatial and temporal characteristics of the auctions executing in our electronic market. Not considering these implications could lead to poor caching performance, and might even degrade into severe thrashing.

In the current implementation of IBIZA, memory is allocated by specifying a substantial maximum heap size within the Java Virtual Machine (JVM) executing the Broker and as the memory requirement increases with longer simulation runs or more Model Developers, this amount is correspondingly increased. However, this is a temporary arrangement and detailed study of the characteristics of auctions is required before a scalable memory management solution is devised.

6.4. Timing allocation across sessions

User requests in IBIZA specify the time by which the user expects to receive the product. This is referred to as the *time* window. This time window has to be divided up to account for the subsequent interactions between the Broker, the Model Developers and the Model Evaluators. The time window is divided into the *bid* window (bids have to returned within this amount of time), the *build* window (predictive models

Table 2. *Distributed communication technology comparison*

	TCP/IP sockets	Remote procedure calls	Java remote method invocation	CORBA IIOP
Platform independence	Yes	No	Yes	Yes
Language independence	Yes	No	No	Yes
Implementation complexity	High	Low	Low	Low
Supports object serialization	No	No	Yes	Depends on implementation
3rd party package necessary	No	No	No	Yes
Remote object execution	No	No	Yes	Yes

have to be built and submitted to the Broker within this amount of time) and lastly, the *evaluation* window (models have be evaluated within this given time). Currently, the broker has to make the decision about how the time window is to be divided. Alternative time allocation schemes are possible. For example, a simple scheme that can be implemented in the prize mechanism involves allocating the full time window to the build window, and setting the evaluation window to zero. Note that in the prize mechanism, there is no bid window since all seller agents that want to participate can do so. The onus of ensuring that enough time remains for the model to be evaluated rests with the Model Developers in this case.

Inefficiencies in the market can result from Model Developers submitting their models within the build window but given the nature of their models, exceeding the allocated evaluation window. If prior knowledge of the behavior of these models had been considered and factored into the timing allocation module, Model Developers that produce models inappropriate to the given timing requirements can be excluded from certain sessions. As such, state information on these ''undesirable'' Model Developers need not be unnecessarily stored, and overall memory needs of the Broker can be streamlined.

Additionally, intelligent timing allocation schemes can also notify users of potentially infeasible timing requirements. Given historical information of operational characteristics of participating Model Developers, the Broker can notify the user immediately if the specified time window is unreasonable, or if the existing pool of Model Developers cannot meet those requirements. Thus, the user does not need to wait for the expiration of the time window before discovering the fact. The Broker's memory space is also not wasted on auctions that do not produce results.

6.5. Scheduling the model evaluator

Model Evaluators perform the role of evaluating of models submitted by the Model Developers. To ensure that the evaluation process scales well with an increase in the rate of model submission, the simulation framework has been developed such that additional Model Evaluators can be registered with the Broker as and when the need arises. The Broker will then distribute the model evaluation tasks across these Model Evaluators.

A round-robin scheduling method has been employed to distribute Model Evaluation tasks across all Model Evaluators registered with the Broker. Although it has the property of simplicity, this method does not take into consideration the nature of the models being evaluated. For instance, the model construction process for an agent featuring the K-Nearest Neighbor technique is very simple and hardly uses any model building time. It basically consists of storing the training dataset as the model and returning the model to the Broker. When this model is submitted to the Model Evaluator for evaluation, each evaluation tuple is compared against the model's repository. The model then returns the most likely outcome as the result. Considering that the average training dataset contains hundreds, if not thousands of tuples, and that each tuple itself contains about eight independent variables, the computational overhead of evaluating each tuple is substantial and evaluation time for such models is many times greater than that of the other models. As such, when a simple scheme such as round robin scheduling is used to distribute the evaluation load, the Model Evaluator becomes congested resulting in bids from other agents that are in the queue not being evaluated within their allocated evaluation time window. Solutions to this problem require more sophisticated load balancing strategies such as sender-initiated or receiver-initiated (Stallings, 1997) methods to ensure better response times from the model evaluators.

Beyond the issue of load balancing methods was the insight we gained into specific services to be provided in the market. Before conducting simulations, we did not think of model evaluation services as a constrained resource. The simulations also helped us recognize that a centralized Broker determined time allocation scheme may be inefficient when models vary considerably in the amount of time required to build and evaluate them. We believe that given the complexity of the interactions that happen in electronic markets, their design should facilitated through careful experimentation such as is supported in IBIZA. We are still in the preliminary phase of conducting experiments and hope to learn and improve the computational infrastructure required to facilitate the design of new and innovative e-markets.

Acknowledgments

This work was funded in part by NSF CISE/IIS/KDI 9873005. We also gratefully acknowledge the work on

Rudy Setiawan and Kevin Teo who implemented IBIZA as part of their INI thesis project.

Notes

1. This probability density function pdf is also the limiting distribution for well known distributions such as the exponential, logistic and the normal.
2. This assumption is not necessary and can be relaxed at the cost of more notation and algebra.

References

Arnold K, Gosling J. *The Java Programming Language*. Addison Wesley Publishing Company, 1996.

Bakos Y. Towards friction-free markets: The emerging role of electronic marketplaces on the internet. *Communications of the ACM* 1998;41(8):35–42.

Bakos Y. Reducing buyer search costs: Implications for electronic marketplaces. *Management Science* 1997;43(12).

Sundararajan A, Arunkundram R. An economic analysis of electronic secondary markets: Installed base, technology, durability and firm profitability. *Decision Support Systems* 1999.

Chircu AM, Kauffman R. Strategies for internet middlemen in the intermediation/disintermediation/reintermediation Cycle. *Electronic Markets* 1999;9(2).

Cooper GF, Aliferis CF, Ambrosino R, Aronis J, Buchanan BG, Caruana R, Fine MJ, Glymour C, Gordon G, Hanusa BH, Janosky J, Meek C, Mitchell T, Richardson T, Spirtes P. An evaluation of machine-learning methods for predicting pneumonia mortality. *Artificial Intelligence in Medicine* 1997;9:107–138.

David HA. *Order Statistics*. New York: John Wiley and Sons, 1981.

Downey T, Meyer J. *The Java Virtual Machine*. McGraw Hill, 1996.

Eddon G, Eddon H. *Inside Distributed Com*. Microsoft Press, 1999.

Fowler M, Scott K. *UML Distilled*. Reading, MA: Addison-Wesley, 1998.

Gamma E, Helm R, Johnson R, Vlissides J. *Design Patterns: Elements of Reusable Object-oriented Software*. Reading, MA: Addison-Wesley, 1995.

Lee HG. Do electronic marketplaces lower the price of goods? *Communications of the ACM* 1998;41(1):73–80.

Lee HG, Clark T. Impact of electronic market places on transaction cost and market structure. *International Journal of Electronic Commerce* 1996;1(1):127–149.

Malone T, Yates J, Benjamin R. Electronic markets and electronic hierarchies: effects of information technology on market structure and corporate strategies. *Communications of the ACM* 1987;30(6):484–497.

Mitchell T. *Machine Learning*. NY: McGraw-Hill, 1997.

Rumbaugh J, Blaha M, Premerlani W, Eddy F, and Lorenson W. *Object-Oriented Modeling and Design*. NJ: Prentice Hall, 1991.

Sarkar M, Butler B, Steinfiled C. Intermediaries and cybermediaries: A continuing role for mediating players in the electronic marketplace. *Journal of Computer-Mediated Communication* 1(3).

Setiawan R, Teo K. *IBIZA: A Simulation Environment for Designing Electronic Markets, MSIN Thesis*:. Information Networking Institute, Carnegie Mellon University.

Shapiro C, Varian HR. *Information Rules: A Strategic Guide the Network Economy*. Boston: Harvard Business School Press, 1998. http://www.inforules.com

Siegel J. *CORBA Fundamentals and Programming*. John Wiley and Sons, 1996.

Stallings W. *Operating Systems: Internals and Design Principles*. Prentice Hall, 1997.

Williamson OE. *Markets and Hierarchies*. New York: Free Press, 1975.

Ashish Arora is Associate Professor of Economics and Public Policy at the Heinz School of Public Policy and Management. Professor Arora's research centers around the areas of economics of technological change, management of technology, intellectual property rights, and technology licensing. He has in the past worked on questions of the productivity of university research, and the growth and development of biotechnology and the chemical industry. He is currently studying software industry, and specifically, the development of the Indian software industry and its links to the United States. An enduring research interest is in understanding the rise and functioning of markets for technology and their consequences.

Greg Cooper is Associate Professor of Medicine at the University of Pittsburgh Medical Center. His primary research interests involve the application of decision theory, probability theory, and artificial intelligence to medical informatics research tasks, including causal discovery from observational patient data, computer-aided medical diagnosis, computer-aided clinical outcome prediction, computer-assisted development of clinical care guidelines, machine learning of medical expert systems, and computer-assisted retrieval of medical information. Additional information is available at http://www.cbmi.upmc.edu/ ~ gfc/

Ramayya Krishnan is the W.W. Cooper and Ruth F. Cooper Professor of Information Systems at the Heinz School of Public Policy and Management and the Institute for Complex Engineered Systems at Carnegie Mellon University. His current research studies the technology, business and policy aspects of electronic commerce. Additional information about his research

projects are at http://dnet.heinz.cmu.edu/dcsrg/index.html. His work is funded by the National Science Foundation and DARPA.

Rema Padman is Associate Professor of Operations and Information management. Professor Padman's area of research interest is deterministic optimization including optimization-based decision support systems, and applications of network flow modeling in domains such as project management and scheduling. Her current research involves the use of novel computing approaches, new algorithms and search techniques such as parallel computing, neural networks and tabu search in solving problems in the above domains.