Simulated Annealing Based Intelligent Bidding Agent for Heterogeneous Environment

Developed by:

Shahbaz Ahmed Khan Ghayyur
(17-CS/MS(SE)/03)

Saeed Ullah
(30-CS/MS(SE)/03)

Supervised by:

Dr. S Tauseef-ur-Rahman

Faculty of Applied Sciences
Department of Computer Science
International Islamic University, Islamabad

2006

بسم الله الرحمن الرحيم

**In the name of ALMIGHTY ALLAH,
The most Beneficent, the most
Merciful.**

Department of Computer Science,
International Islamic University, Islamabad.

12th April, 2006

Final Approval

It is certified that we have read the thesis, titled "**Simulated Annealing Based Intelligent Bidding Agent for Heterogeneous Environment**" submitted by **Shahbaz Ahmed Khan Ghayyur** and **Saeed Ullah** under University Reg. No. **17-CS/MS(SE)/03** and **30-CS/MS(SE)/03** respectively. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad, for the Degree of **MS Software Engineering.**

Committee

External Examiner
Dr. Naveed Ikram
Associate Professor,
Muhammad Ali Jinnah University,
Blue Area Islamabad.

Internal Examiner
Mr. M. Salman
Lecturer
Department of Computer Science,
International Islamic University, Islamabad.

Supervisor
Dr. S Tauseef-UR-Rehman
Head,
Department of Telecommunication Engineering,
FUUAST, Islamabad.

Dedication

Dedicated to Almighty Allah and our Families who have supported us in all aspects throughout our life.

A dissertation submitted to the

Department of Computer Science,

International Islamic University, Islamabad

as a partial fulfillment of the requirements

for the award of the degree of

MS Software Engineering

Declaration

We hereby declare that this research project, neither as a whole nor as a part thereof has been copied out from any source without due credit or reference. It is further declared that we have developed this software entirely on the basis of our personal efforts made under the sincere guidance of our teachers. The thesis is our sole effort and has been developed in accordance with the proposal submitted and approved by faculty of Department of Computer Science under guidance of our supervisor.

Acknowledgements

All praise to the Almighty Allah, the most Merciful, the most Gracious, without whose help and blessings, we would have been entirely unable to complete the project.

Thanks to our Parents who helped us during our most difficult times and it is due to their unexplainable care and love that we are at this position today.

Thanks to our project supervisor Dr. S Tauseef-ur-Rehman, whose sincere efforts helped us to complete my project successfully?

Acknowledgement is also due to our teachers and friends for their help in this project.

Shahbaz Ahmed Khan Ghayyur
17-CS/MSSE/03

Saeed Ullah
30-CS/MSSE/03

Declaration

We hereby declare that this research project, neither as a whole nor as a part thereof has been copied out from any source without due credit or reference. It is further declared that we have developed this software entirely on the basis of our personal efforts made under the sincere guidance of our teachers. The thesis is our sole effort and has been developed in accordance with the proposal submitted and approved by faculty of Department of Computer Science under guidance of our supervisor.

Abstract

The project aims to design and implement an efficient and intelligent bidding agent. As electronic commerce has flourished, the use of agent technology has grown in stature to provide sophisticated and fully automated auction services. This paper deals with the issues of intelligent bidding agent architecture in futuristic integrated electronic commerce systems, via diverse parallel, simultaneous auctions with varying starting-ending times, while incorporating heterogeneous protocols. We propose a modified Belief, Desire, Intention architecture. The proposed mechanism enables optimum gains and efficient learning for concurrent bidding to derive a bidding action plan in highly diverse, fluctuative, fractol and quasi-fractol environment, while taking into account the preferences and demographics of items, bargain leverage, time, supply-demand, auction diversity of interest, and eagerness etc. The agent employs Simulated Annealing to implement its intelligent behavior in not only solitary offline environment but also in a live interactive society, taking it one step further from being a more rule based system. The structure and working of the agent is formulated by classical Architectural Description. This provides a modular description of the semantics about activities performed for optimal service. Explicit specifications on the agent's behavior are also algorithmically formulated as derived form the agent's model.

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

Ever since the known history of mankind the transfer of goods and services under the term of commerce activity is reported. It has ever since been about entities and in the structural form of entities that are valued less by humans, to the people who value them more. Thus creating significance, which in turn gave rise to the term worth. There has always been a rationalization for commerce if it created or gave rise to worth which may be in the form of commodity or service [1].

Bidding is a process which came into existence with the evolution of rudimentary commerce when it was introduced to competition and an imbalance in the mutual supply and demand was created; i.e. more parties striving to gain control of fewer resources due to their potential worth factor [2]. In the modern day world the term bidding still has the same essence i.e. a process in which a party competes with other competitors in order to procure some item at a particular cost. The cost is a reflection of worth and is a price at which the broker/dealer is willing to buy the particular commodity.

As the commerce matured into civilized age, particular specialized institutions were formulated for conductions of bidding to strive for maximum utility form the activity. We have come to know the most promising of such mechanisms as an "Auction" for which the institution of Auction House was introduced. In an Auctioning environment, the process of bidding takes place which is the placement of a bid price presented by a buyer/bidder when he wants to buy a commodity [3]. The bid price is usually just referred to as bid. The bid price stands in disparity to the ask value or the offer, and the divergence between the two is called bid/offer spread [4].

In these auction houses the process of biding can be nut shelled into the broader term of procurement which is defined as "the acquisition of goods or services at the best possible total cost of ownership, in the right quantity, at the right time, in the right place for the direct benefit or use of the governments, corporations, or individuals generally via a contract" [3].

At first the internet technology aided commerce was used to merely offer display and retailing but has quickly engulfed all sorts of business activities. In just a few years the websites have moved form displaying electronic brochures to providing a channel for sales, customer services, and information gathering for small and large enterprises [5]. The streaming of more and more business process on the web paved the way for internet to be turned into a doorway to a virtual business environment [6].

Now a day, e-commerce is growing at an exponential rate. According to one study, the Internet economy grew at a rate of 174.5 percent annually from 1995 to 1998. This growth follows the equation Internet economy. *e*year almost exactly (remember, the natural log, *e*, = 2.718...). A number of analysts forecasted that the Internet economy will exceed US $1 trillion (1012) in 2002 [5]; Forrester Research recently predicted that the worldwide Internet economy has reached US $6.9 trillion in 2004. (The Internet Economy Indicators, Indicators Report, June 1999, available online at [7].

It is generally accepted that e-commerce is the catalyst responsible for creating global competitive markets, changing the business process interface, challenging the organization's structure, culture, and management and business strategies. The literature appears to support the prediction that businesses will need to readily adapt to these changes if they are to remain competitive in what is considered to be a technology driven, global economy. However, from this study a diverse range of enabling and inhibiting factors have emerged, which suggests that the depth of global market change is very much dependent on an industry's regulatory framework and the jurisdictions of each country of operation. Some of these factors are considered to be barriers to global business and impediments to the promulgation of global technologies. Indeed these findings suggest that globalisation exists in concept only, but not in practice [8].

As electronic commerce has flourished, the use of agent technology has grown in stature to provide sophisticated and fully automated auction services. This paper deals with the issues of intelligent bidding agent architecture in futuristic integrated electronic commerce systems, via diverse parallel, simultaneous auctions with varying starting-ending times, while incorporating heterogeneous protocols. We propose a modified Belief, Desire, Intention architecture

1.1 Current Scenario

Intelligent decision making on part of contractors is of critical importance when the result oriented bidding activities involve more players and more rounds of interaction, as is common when the supply web becomes more complicated, and many alternative business deals are possible(BidX, Foogle, Auction Beagle Forums[9]). The process of placing bids becomes much more complicated if the starting and ending time of auctions is different as well along with the different protocols adding the variation in preferences of the selling contractor [10]. To facilitate the contractors many of these auction houses have introduced automated bidders which act on behalf of contractors to take advantage of the huge set of processable information about Auctions available by utilizing their computational power, to get the best deal according to the contractor's preferences. These automated contractors are Agent systems or simply referred to as Agents [10].

The users can always take advantage of the information assembled in the shopping engines like FOOGLE, BIDXS or AUCTION WATCH but the basic limitations imposed by mutually exclusive nature of bidding are still unresolved, as the contractor has to make the final decision and go through the painstaking process of scanning the e-market. Furthermore the contractor is still faced with the towering task of selection of a single bid price which will fetch him the best wining deal. In many cases the customer is trapped in winners curse like phenomenon where they pay more than they should have to secure the win [11].

We have also cross referenced our work with evolutionary programming like GA, which offers another popular approximation technique, we considered this as a possible approach to the problem of training an agent at hand but evolutionary programming and its variants (hybrid approach etc...) were ultimately found wanting. The case is that in a genetic algorithm, several elements of the solution space are looked at simultaneously; these elements are corresponding to individuals which make up a genetic population. Like individuals in a genetic population, the elements of the solution space experience evolution, which occurs through reproduction and continued existence of the fittest. In order for a genetic algorithm to be applied to a problem, elements of the solution space must be programmed in such a way that two elements can reproduce by exchange of some portion of themselves with their associate, just as biological reproduction involves the swapping of bits of Dioxiribo Nucleic Acid. The problem with encoding strategies in heterogeneous multi-protocol auction environment in this way is that it is not obvious how two or more agent strategies[12], by swapping groups of auctioning environment or attributes with each other, could produce a new, valid action plan. The requirements that bidding be conducted by the intentioned preferences provided by the user, and that each agent be bidding and buying only one type of item at a given instant in exactly one auction. Arbitrary swaps of portions of strategy are unlikely to result in valid optimal action plan since the protocol and strategies are different for different environments makes genetic algorithms an unreasonable approach to this problem [14].

1.2 Need of the Hour

Recently, the complexity of logistics involved in mechanized development and in many other areas of commerce has been ever-increasing nearly exponentially. Many processes are being outsourced to outside contractors, making supply chains longer and more convoluted. The increased complexity is often compounded by reduced inventories and accelerated production schedules which demand tight integration of processes across multiple self-interested organizations. At the same time, much commerce is moving on line, where firms can cut costs and improve efficiency by taking advantage of reduced transaction costs, exp edited order cycles, and dynamic pricing available in the network environment.

Current on-line commerce systems characteristically rely on either fixed-price catalogs or simple auctions to set prices, and either industry portals or general search engines to find probable suppliers and customers. Companies over and over again work with pre-qualified suppliers in order to administer risk and complexity, and buyer-supplier relationships depend on factors such as excellence, delivery performance, and flexibility, in addition to cost. In addition, most current e-commerce systems do not have any notion of time (although some can deal with delivery time or lead time), and only the simplest of constraints can be articulated. Exceptions take account of field particular systems such as SABRE used in the travel industry, where one may search for connecting flights. Time and precedence constraints play a elementary role in supply-chain formation and management, since many products are made up of dissimilar parts and require multiple suppliers who have to synchronize their work. [15].

1.3 The Intelligent SA Based Bidding Agent

The term "Agent" has been used to signify many things in recent industry and academic literature, and has been generally degraded by marketing hype. According to Webster's Third New International Dictionary[16], an agent is "one that acts or exerts power... a means or instrument by which a guiding intelligence achieves a result... one that acts for or in the place of another by authority from him." It further says that "An agent is just something that perceives and acts". Bradshaw reviews in detail the various meanings of the term as it has been used in the research community.

The agent and bidding proposed in this thesis enables optimum gains and efficient learning for concurrent bidding to derive a bidding action plan in highly diverse, fluctuative, fractol and quasi-fractol environment, while taking into account the preferences and demographics of items, bargain leverage, time, supply-demand, auction diversity of interest, and eagerness etc. The agent employs Simulated Annealing to implement its intelligent behavior in not only solitary offline environment but also in a live interactive society, taking it one step further from being a more rule based system. The structure and working of the agent is formulated by classical Architectural Description. This provides a modular description of the semantics about activities performed for optimal service. Explicit specifications on the agent's behavior are also algorithmically formulated as derived form the agent's model. Furthermore this thesis also discusses the architecture and implementation of a futuristic electronic marketplace simulator which is used to run various scenarios and agent strategies.

This research effort deals with developing such a bidding agent based on the a retailored Belief-Desire-Intention Architecture with static intentions and by utilizing probabilistic stochastic algorithm of simulated annealing to determine policy based optimal solution in fractal and quazi-fractal distribution environments produced as a result of single or multi- protocol multiple auctions. We chose this particular method because simulated annealing has been known to perform well in areas where the space to be searched is large and not well understood [11] and the problem at hand has to be guided by heuristic since it gradually arrives at better and better solutions.
Since the problem at hand is non-deterministic algorithmic because a large number of solutions exist. This situation is classical for an approximation algorithm which yields naturally towards simulated annealing. It is chosen because it has been proven a success in many difficult optimization problems [12]. Among these problems are the *traveling salesman problem*, *image recognition from noisy data*, *integrated circuit layout*, and *robotic optimal path finding and planning* [12].

References:

[1] Electronic Payment Systems* N. Asokan, Phil Janson, Michael Steiner, Michael Waidner IBM Research Division, Zurich Research LaboratoryCH-8803 Ru"schlikon, Switzerland {aso,pj,sti,wmi} @zurich.ibm.com

[2] eCommerce Trust Study.. Cheskin Research & Studio Archtype/Sapient. 1999. [online]. Available:http://www.cheskin.com/think/studies/ecomtrust.html [viewed July 30, 2001].

[3] http://en.wikipedia.org/wiki/Bidding

[4] David Chaum. Blind Signature for Untraceable Payments, University of California, 1998

[5] http://www.forrester.com

[6] NetBill Security and Transaction Protocol, Carnegie Mellon University Pittsburgh, PA 15213–3890 Benjamin Cox J. D. Tygar Marvin Sirbu thoth+@cmu.edu tygar@cmu.edu sirbu+@cmu.edu

[7] http://www.internetindicators.com/features.html

[8] *The Organisational and Global Environments Relationship: An Investigation of the Key Factors BY Robyn Leslie Watters Diploma Art and Design Graduate Diploma Business Computing*

[9] www.foogle.com www.auctionwatch.com www.auctionbeagle.com

[10] Use of a Colony of Cooperating Agents and MAPLE to Solve the Traveling Salesman Problem Bruno Guerrieri Florida A&M University bruno.guerrieri@famu.edu

[11] Auction Theorey: AGuide to literature. J.Econ. Surv. 13, 3 227-286

[12] Simulated Annealing: Quasi-Fractals and Quasi-Failures. Raymond Lister School of Computing Science Queensland University of Technology GPO Box 2434, Brisbane Q 4001

[13] B. J. Buckham and C. Lambert. Simulated annealing applications. Retrieved September 2003 from the World Wide Web: http://www.me.uvic.ca/~zdong/courses/mech620/SA_App.PDF.

[14] S. Russell and P. Norvig. Arti_cial Intelligence: A Modern Approach. Prentice Hall, Saddle River, New Jersey, 1995.

[15] Agents for electronic commerce By Addison Wesley

[16] www.webster.com

2. Background and Definitions

Since the proposed system deals with the research in multi-agent systems for e-commerce using simulated annealing, following is the description of the concepts used in the proposal.

2.1 Agents

An agent is an autonomous entity with an ontological commitment and an agenda of its own. Each agent possesses the ability to act autonomously; this is an important distinction because a simple act of obedience to a command does not qualify an entity as an agent. Nevertheless in an environment an agent is often acting on a principal's behalf and has a legal duty to act in that person's best interest. An agent may interact or negotiate with its environment and/or with other agents. It may make decisions, such as whether to trust and whether to cooperate with others. [Extracted form article on Wikipedia]

Term agent gives us two orthogonal concepts. The first is the agent's ability for autonomous execution. The second is the agent's ability to perform "domain oriented reasoning."

In Software Engineering terms we can call an agent as a persistent software entity dedicated to a specific purpose. 'Persistent' distinguishes agents from subroutines; agents have their own ideas about how to accomplish tasks, their own agendas. 'Special purpose' distinguishes them from entire multifunction applications; agents are typically much smaller.

In computer science, a software agent is a piece of autonomous or semi-autonomous proactive and reactive, computer software. Many individual communicative software agents may form a multi-agent system(MAS). [Webster Computing Dictionary]

Thus to be considered an agent, a software object must be a self-contained program that is capable of making independent decisions and taking actions to satisfy internal goals based upon its perceived environment. Therefore the characterization of agent processing can be approached from two interrelated directions:

Internal state processing and ontologies for representing knowledge.
Interaction protocols - standards for specifying communication of tasks.

Examples

- User agent - for browsing the World Wide Web
- Mail transfer agent - for serving e-mail
- SNMP agent
- DAML
- Management agents used to manage telecom devices

2.2 Artificial Intelligence

Artificial intelligence (also known as machine intelligence and often abbreviated as AI) is intelligence exhibited by any manufactured (i.e. artificial) system. The term is often applied to general purpose computers and also in the field of scientific investigation into the theory and practical application of AI.

Modern AI research is concerned with producing useful machines to automate human tasks requiring intelligent behavior. It has become an engineering discipline, focused on providing solutions to practical problems. AI methods are often employed in cognitive science research, which explicitly tries to model subsystems of human cognition.

Tush in light of the above discussion we can say that the **Intelligent agents** or **bots** are software elements that work without the assistance of users by making some choices. Choices are based on predicates that developers have identified and built into the software. It is expected that **Intelligent Agents** are a strong focus for AI development, in order to help automate simple tasks.

There are several different classes of intelligent agents in e-commerece

- **Buyer Agents**
- **User Agents (Personal Agents)**
- **Monitoring-and-surveillance (Predictive) Agents**

In such an environment an agent continuously performs the following functions: perception of dynamic conditions in the environment; action to affect conditions in the environment; and reasoning to interpret perceptions, solve problems, draw inferences, and determine actions.

These Intelligent agents are software entities that carry out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and in so doing, employ some knowledge or representation of the user's goals or desires."

The properties of an intelligent agent are [1]

Autonomy: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state

Social ability: agents interact with other agents (and possibly humans) via some kind of agent-communication language

Reactivity: agents perceive their environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the INTERNET, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it

Pro-activeness: agents do not simply act in response to their environment; they are able to exhibit goal-directed behavior by taking the initiative.

The agents can be made more intelligent with the following additional properties:

Rationality: Agents select actions that follow from knowledge and goals.

Adaptivity: Agents are able to modify knowledge and behaviour based on experience.

Collaboration: Agents can plan and execute multi-agent problem solving.

2.3 Stochastic Probabilistic Algorithm

Simulated Annealing [1]

Simulated annealing (SA) is a generic probabilistic meta-algorithm for the global optimization problem, namely locating a good approximation to the global optimum of a given function in a large search space. It was independently invented by S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi in 1983, and by V. Cerny in 1985.

The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one.

It comprises of the following steps

- The basic iteration
- The neighbours of a state
- Transition probabilities
- The annealing schedule
- Convergence to optimum
- Selecting the parameters
- State neighbours
- Transition probabilities
- Annealing schedule

Overview

In the simulated annealing (SA) method, each point s of the search space is compared to a state of some physical system, and the function **E(s)** to be minimized is interpreted as the internal energy of the system in that state. Therefore the goal is to bring the system, from an arbitrary initial state, to a state with the minimum possible energy.

The basic iteration

At each step, the SA heuristic considers some neighbours of the current state s, and probabilistically decides between moving the system to state s' or staying put in state s. The probabilities are chosen so that the system ultimately tends to move to states of lower energy. Typically this step is repeated until the system reaches a state which is good enough for the application, or until a given computation budget has been exhausted.

The neighbours of a state

The neighbours of each state are specified by the user, usually in an application-specific way. For example, in the traveling salesman problem, each state is typically defined as a particular tour (a permutation of the cities to be visited); then one could define two tours to be neighbours if and only if one can be converted to the other by interchanging a pair of adjacent cities.

Transition probabilities

The probability of making the transition to the new state s is a function $\mathbf{P(\delta_E, T)}$ of the energy difference $\delta_E = \mathbf{E(s') - E(s)}$ between the two states, and of a global time-varying parameter T called the temperature.

One essential feature of the SA method is that the transition probability **P** is defined to be nonzero when δ_E is positive, meaning that the system may move to the new state even when it is worse (has a higher energy) than the current one. It is this feature that prevents the method from becoming stuck in a local minimum — a state whose energy is far from being minimum, but is still less than that of any neighbour.

Also, when the temperature tends to zero and δ_E is positive, the probability $\mathbf{P(\delta_E, T)}$ tends to zero. Therefore, for sufficiently small values **T**, the system will increasingly favor moves that go "downhill" (to lower energy values), and avoid those that go "uphill". In particular, when **T** is 0, the procedure reduces to the greedy algorithm — which makes the move if and only if it goes downhill.

Also, an important property of the **P** function is that the probability of accepting a move decreases when (positive) δ_E grows bigger. For any two moves that both have positive δ_E values the P function favours the smaller value (smaller loss).

When δ_E is negative, $P(\delta_E, T) = 1$. However, some implementations of the algorithm do not guarantee this property with the **P** function, but rather they explicitly check whether δ_E is negative, in which case the move is accepted.

Obviously, the effect of the state energies on the system's evolution depends crucially on the temperature. Tentatively speaking, the evolution is sensitive only to coarser energy variations when **T** is large and to finer variations when **T** is small.

Convergence to optimum

It can be shown that, for any given finite problem, the probability that the simulated annealing algorithm terminates with the global optimal solution approaches 1 as the annealing schedule is extended. This theoretical result is, however, not particularly helpful, since the annealing time required to ensure a significant probability of success will usually exceed the time required for a complete search of the solution space.

Pseudo-code

The following pseudo-code implements the simulated annealing heuristic, as described above, starting from state s0 and continuing to a maximum of k_{max} steps or until a state with energy e_{max} or less is found. The call neighbour(s) should generate a randomly chosen neighbour of a given state S; the call random() should return a random value in the range [0, 1). The annealing schedule is defined by the call temp(r), which should yield the temperature to use, given the fraction r of the time budget that has been expended so far.

```
s := s0
e := E(s)
k := 0
while k < kmax and e > emax
  sn := neighbour(s)
  en := E(sn)
  if en < e or random() < P(en - e, temp(k/kmax)) then
    s := sn; e := en
  k := k + 1
return s
```

2.4 Auctions

Auctions have been used as a mechanism to match buyers and sellers for a very long time. For example, in Rome, auctions were used by sellers to market their goods. The first book about auction was written in Britain in the 1600's [2]. The most general auction method, "first-price open price auction", is also called the English auction. Simply stated, an auction is a method of allocating goods that are either scarce or difficult to evaluate based on competition [3]. In a market, a seller wants to sell an item at the highest possible price and the buyers wish to obtain the item at the lowest possible price. An auction helps the seller to identify the buyer who is willing to pay the highest price. Furthermore, the buying price of the item can be determined by the buyers instead of the sellers. This means that sellers can push the burden of pricing to the market. However, sellers may also control prices by choosing an appropriate auction type and by setting a lowest (or "reserve") selling price.

There are various types of auctions. The bidding prices in an auction can be either ascending or descending and they can be either public or private [2]. An auction can also be classified based on the number of items sold (single-item auctions or multi-item auctions). Different auctions have their own characteristics and the most suitable type of auction for selling something depends on many factors. Among them are time to sell the item, the cost of the item and the characteristics of the buyers. With the advent of the Internet, auctions are nowadays widely used for electronic commerce [3]. With online auctions, users could buy/sell items in various regions of the world. Compared to traditional auctions, online auctions bring greater convenience while dramatically decreasing the transaction cost. However, they have some shortcomings that do not exist in traditional auction markets.

Types of Auction

There are many ways to classify auctions. According to the bidding information, auctions can be divided into open-auctions and closed auctions. Based on the variation of prices, we have ascending price auctions and descending price auctions. The difference between single-item auction and multi-item auction is the quantity of items to be sold. Auctions can also be classified into one-sided auctions and double auctions. Figure 1.1 shows the different types of auctions [3]. Auction-based electronic commerce has become very popular in the past few years [4. Numerous auction sites have been set up to carry out different kinds of auctions. We describe several common types of auctions in detail as follows. For simplicity, we assume that the seller also acts as the auctioneer.

2.5 Several classic auctions

English auction

In an English auction, there are one seller and many buyers [5]. The seller sets a reserve price and deadline that are disclosed to buyers and a lowest acceptable price that is known only to the seller and auctioneer. The price is successively raised from the reserve price until only one bidder remains. That bidder wins the item at the final price provided that the final price is not less than the lowest acceptable price and deadline has not been reached. The auction can be run by having the seller announcing prices, the bidders calling out prices, or bids submitted electronically with the best current bid posted.

Dutch auction

A Dutch auction works in the opposite way. First the seller sets a reserved price, a decremental price and a private lowest acceptable price [4]. The auctioneer starts at the reserved price, and then lowers the price continuously.

The first bidder who accepts the current price wins the item provided that it is not less than the lowest acceptable price.

First-price sealed-bid auction (FPSB)

This auction has a deadline and each bidder independently submits a single bid before the deadline without seeing others' bids. The item is sold to the bidder who places the highest bid [30]. Same as the above, the winning price must be equal to or larger than the lowest acceptable price.

Vickrey auction

Vickrey auction is a famous type of auction that is similar to FPSB [5]. The only difference between them is that the bidder who makes the highest bid gets the item at the second-highest bid, or the "second price". The Vickrey auction has a so-called "truth-telling" characteristic. That is, bidders tend to submit bids based on their own value of the item.

Yankee auction

A Yankee auction can be viewed as a generalized type of the English auction because it works in a similar manner but caters for the bidding for multiple items [5]. Basically, the seller allocates items to the buyers according to the descending order of their bid prices until all items are sold out. Bidders with a higher bid will be served first. Each bidder pays what they bid plus the number of items to be bought.

2.6 Single-item auction and multi-item auction

Auctions can also be classified into single-item auctions and multi-item auctions, according to the number of items sold. All auction rules described above only apply to selling one item. Rules for multi-item auctions are more complicated since the quantity requirement of each bidder may be different [3]. The English auction, Dutch auction and sealed-bid auction also support the selling of multi-items as described below. In English and sealed-bid auctions, a bidder is asked to submit k bids, where, to indicate how much he/she is willing to pay for each additional item. Thus b is the amount that the bidder is i willing to pay for one item, is the amount he/she is willing to pay for two items and so on. Given that there are j items to sell in an auction, they will be sold to the buyers with the highest j bids.

In multi-item Dutch auction, the auctioneer announces the price decreasingly and an item is sold to a bidder who agrees to accept the current price. The auction is over when all items are sold. As introduced above, there exist various auctions in the market. However, the key differences among different auctions are related to the following [5]:

Anonymity:

Different information is disclosed during the auction process. For example, sealed-bid auctions are more anonymous than other types of auctions. In sealed-bid auction, only the identity of the final winner is disclosed and all the bids are kept secret. In Dutch auctions, we can know the bids of winners and their identities. In English auctions all bids are public.

Rules for ending an auction:

English auction may end at a predefined closing time. Alternatively, they may also end on the condition that no new bids are submitted within a certain time period. Dutch auctions always end by a new bid or when the price decreases to a predefined price. Sealed-bid auctions have a definite deadline.

Payment amount:

When an auction is over, the winner must pay for the item. However, the payment amount is not always equal to the winner's bid. In *Discriminative Auctions* (i.e., Yankee auctions), a winner always pays what he/she bids. In *Non Discriminative Auctions*, however, each winner pays the lowest bid among all winners. In Vickrey auction, the winner pays the second highest bid rather than the highest bid.

Restrictions on bid amount:

In all auctions, the seller can specify the bidding parameters. In English auction, the seller typically sets the minimum bidding and the minimum incremental price. In Dutch auctions, the maximum price and the decreasing price are set. A private reserve price can also be specified. Of course, it is confidential to the bidders.

Disclaimer:

These standard definitions have been taken/adopted from different online and offline computing dictionaries and open access resources on web like the computing encyclopedias and Wikipedia which are publically available and permit their use for educational purposes.

References:

[1]. www.wikipedia.org
[2]. [11] http://learntoauction.com/auctionlist.htm
[3]. V. Krishna, "Auction theory", *Academic Press*, San Diego, 2002.
[4]. K. Reynolds, "Going...Going...Gone! ", Agorics, Inc., 1996.
[5]. M. Kumar, S. I. Feldman, "Internet auctions", *in Proceedings of the 3rd USENIX Workshop on Electronic Commerce*, pp. 49-60, 1998.

3. LITERATURE REVIEW

3.1 Introduction

Chapter 3 described the background of the. Although existing online auctions provide many advantages, they also have some shortcomings. This chapter is not intended to give the outline of how different MAS have been developed or are working for agent based auctions but provides a critical analysis of current state of the art systems. A few important factors have been identified from literature and also with discussion and advice from teachers and professionals in industry which are required to make a good intelligent multi agent bidding system. The characteristics which an agent should possess to make it worthy of being fulfilling the research objectives are

- It should work on multiple auctions
- The final bidding plan and its execution should be fully automated which should not need any monitoring
- It should be able to employ artificial intelligence techniques to continuously learn from environment and improve its performance
- It should be able to operate in not only fractal (with peaks on the edges) but also quazi fractal environments (with irregular peaks and valleys)
- IT should be able to evolve to the next intelligence level after gathering data from market environment and knowledge bases
- Since the search space is heuristic. It should be able to provide near optimal results on regular basis
- IT should have successful procurement rates mostly at near optimal prices
- IT should be able to cope with not only local but distributed auction settings.

We are going to examine the proposed on basis of these criterion.

The organization of this chapter is as follows. Section 3.3 briefly describes the basics of agents and their advantages for online auctions. Several famous agent-based online auction systems are also introduced in this section. In section 2.3, we give a critical tabular overview of some bidding strategies proposed by various researchers. Section 2.5 summarizes this chapter.

3.2 E-Business and Online Auctions

Online auctions are an accepted and effective medium for procuring goods and services in both business-to-business and business to consumer electronic commerce [Bapna et al. 2001; He et al. 2003]. Some of the well-known auction houses include **eBay, Amazon.com, Yahoo! Auction, Priceline, UBid** and many others.

These auction houses conduct many different types of auctions, but the more popular ones are English, Dutch, first-price sealed bid and second pricesealed bid (also known as Vickrey). It has however been a limitation that only the human interface can be satisfied with the face to face interfacing or directly via human contact virtual or real.

As a step toward the multiple auctions case, consumers can utilize the services of auction search engines (such as BidXS, AuctionWatch, and AuctionBeagle).

These allow the consumer to monitor multiple concurrent auctions, but they leave the actual bidding decision to the consumer. While this certainly increases the consumer's knowledge of the global marketplace, it does not solve the problem of reducing the amount of time that has to be spent online.

Moreover, deciding what amount to bid for an item requires an intelligent decision where the consumer needs to come up with a strategy to work out the bid value. In many cases, the outcome of this decision making is that the consumer is trapped with the *winner's curse* phenomenon where they pay more than the actual value of the item [Klemperer 1999].

3.3 Agent-based online auction systems

Development of the Internet has spurred a number of attempts to create virtual marketplaces. However several efforts are being made to develop interfaces where artificial agents can interface with not only humans but also among themselves on behalf of humans.

The task becomes more complicated when there are different start and end times and when the auctions employ different protocols. For this reason, some online auctions provide bidding agents (proxy bidders) to assist consumers with these tasks [7].

This section presents some popular and well-known multi agent systems below

3.3.1 eBay and Onsale

Besides a user interface, eBay and Onsale also provide some simple agent-like interfaces [1]. For example, Onsale provides a proxy agent called *BidWatcher*, which is a very simple system and only provides just minimal support for English auctions only. The user provides the bot with a desirable price for an item and the bidding agent tries again and again to procure the item on the same price or alert the user for potential chances to procure. If you want to know more about ebay and onsale and how it works one should refer to [14]

Salient features of birdwatcher analysis are as under.[1], [14]

- It works on a single item at a given time on an auction.
- It is the responsibility of the user to monitor and the bidding and configure the bot accordingly
- It does not have capability of working on multiple auctions at a given instant
- It is also not an intelligent bidder.
- Its intelligentsia is limited
- Its scope of deal is also local and the agent utilizes a direct bidding scheme

3.3.2 AuctionBot

Michigan AuctionBot (http://auction.eecs.umich.edu/) has been in use on the internet for the past few years and has been primarily used for selling old books as may be seen from article found on ACM at http://portal.acm.org/citation.cfm?id=280847. It was developed by university of Michigan [14]. If you want to know more about Auction Bot and how it works one should refer to [14]

Salient features of auction bot analysis are as under.[1][15]

- Auction bot works on multiple auction sites.
- It is the responsibility of the user to monitor and the bidding and configure the bot accordingly
- It has the capability of working on multiple auctions at a given instant
- It is also an intelligent bidder and is opera table in fractal environment however it does not support quazi fractal environment.
- Its intelligentsia is limited and is unavailable and successful procurement rates are below satisfactory when seen on scale of optimal price
- Its scope of deal is also local and the agent utilizes a direct bidding scheme

3.3.3 Kasbah

Kasbah is a Web-based multi-agent system where users can create buying agents and selling agents to trade goods [2]. If you want to know more about KABASH and how it works one should refer to [15]

There are totally three kinds of agents in Kasbah: market agents, selling agents and buying agents. [1][

Salient features of KAbash analysis are as under. [1][[15]

- Kabash does not support multiple types of auctions at a given instant.
- It is the responsibility of the user to monitor and the bidding and configure the bot accordingly
- Its agents does not also does not have the capability of working on multiple auctions at a given instant
- It is also an intelligent bidder but is not operatable in fractal environment .Also it does not support quazi fractal environment.
- Its intelligentsia is limited and is unavailable and successful procurement rates are below satisfactory when seen on scale of optimal price
- Its scope of deal is also local and the agent utilizes a direct bidding scheme

3.3.4 Nomad

EAuctionHouse (EAuctionHouse) utilizes the nomad MAS for making automated bids and facilitate its users [5]. With the Nomad system, mobile agents journey to the eAuctionHouse portal and take part in auctions on the user's behalf. Users can generate agents using Java or can automatically produce agents from Nomad's "template agent library". [15]. If you want to know more about NOMAD and how it works one should refer to [16][17][19]

Salient features of KAbash analysis are as under.[1],[14],[16],[17][19]

- Kabash does not support multiple types of auctions at a given instant.
- It is the responsibility of the user to monitor and the bidding and configure the bot accordingly
- Its agents does not also does not have the capability of working on multiple auctions at a given instant
- It is also an intelligent bidder but is not operatable in fractal environment .Also it does not support quazi fractal environment.
- Its intelligentsia is limited and is unavailable and successful procurement rates are below satisfactory when seen on scale of optimal price
- Its scope of deal is also local and the agent utilizes a direct bidding scheme
- The style Nomad follows is purely reactive

3.3.6 BiddingBot

Different from the above auction sites, BiddingBot is a multi-agent system that supports users in attending, monitoring, and bidding in multiple auctions [14]. If you want to know more about Bidding Bot and how it works one should refer to[14] [15][20]

- Salient features of Bidding Bot analysis are as under.[1][14][15][20

- Bidding Bot supports multiple types of auctions at a given instant and searches for a single item in multiple auctions.

- It is the responsibility of the user to monitor and the bidding and configure the bot accordingly as the last and final decision to procure rests with the user although the bot is more automated that the ones discussed previously

- Its agents does not also does not have the capability of working on multiple auctions at a given instant

- It is also an intelligent bidder but is not operatable in fractal environment .Also it does not support quazi fractal environment.

- Its intelligentsia is limited and is unavailable and successful procurement rates are below satisfactory when seen on scale of optimal price

- Its scope of deal is also local and the agent utilizes a direct bidding scheme

- It uses probabilistic approach which does not serve well in pure quazi fractal environments.

3.3.7 Proxy Bot

If you want to know more about Proxy Bot and how it works one should refer to [15]

Salient features of Proxy Bot analysis are as under.[15].

- Proxybot does not support multiple types of auctions at a given instant.
- It is the responsibility of the user to monitor and the bidding and configure the bot accordingly
- Its agents does not also does not have the capability of working on multiple auctions at a given instant
- It is also an intelligent bidder but is not operatable in fractal environment. Also it does not support quazi fractal environment.
- Its intelligentsia is limited and is unavailable and successful procurement rates are below satisfactory when seen on scale of optimal price
- Its scope of deal is also local and the agent utilizes a direct bidding scheme
- It uses greedy strategy for accomplishments of its goals

3.3.8 ATTac 2000

If you want to know more about ATTac and how it works one should refer to [18][21]

Salient features of KAbash analysis are as under.[18][21]

- ATTac does not support multiple types of auctions at a given instant.
- It is fully automated and user does not have to monitor and the bidding and configure the bot accordingly but sill has the limitation of decision making without human interaction so it does not act on its autonomous agenda
- Its agents does not also does not have the capability of working on multiple auctions at a given instant
- It is also an intelligent bidder but is not operatable in fractal environment .Also it does not support quazi fractal environment.
- Its intelligentsia is limited and is unavailable and successful procurement rates are below satisfactory when seen on scale of optimal price
- Its scope of deal is also local and the agent utilizes a direct bidding scheme

3.3.9 Southampton TAC

If you want to know more about STAC and how it works one should refer to [15][22][21]

Salient features of *Southampton TAC* analysis are as under.[15][22][21]

- *Southampton TAC* does not support multiple types of auctions at a given instant.
- It is the responsibility of the user to monitor and the bidding and configure the bot accordingly
- Its agents does not also does not have the capability of working on multiple auctions at a given instant
- It is also an intelligent bidder but is not operatable in fractal environment .Also it does not support quazi fractal environment.
- Its intelligentsia is limited and is unavailable and successful procurement rates are below satisfactory when seen on scale of optimal price
- Its scope of deal is also local and the agent utilizes a direct bidding scheme
- Southampton TAC is easy to automate and control since it follows a rule based strategy

3.3.10 Aster

If you want to know more about Aster and how it works one should refer to [15][21]

Salient features of Aster analysis are as under.[15][21]

- Aster does not support multiple types of auctions at a given instant.
- It is the responsibility of the user to monitor and the bidding and configure the bot accordingly
- Its agents does not also does not have the capability of working on multiple auctions at a given instant
- It is also an intelligent bidder but is not operatable in fractal environment .Also it does not support quazi fractal environment.
- Its intelligentsia is limited and is unavailable and successful procurement rates are below satisfactory when seen on scale of optimal price
- Its scope of deal is also local and the agent utilizes a direct bidding scheme

To address these shortcomings, we believe it is necessary to develop an autonomous agent that can participate in multiple heterogeneous auctions, that is empowered with trading capabilities and that can make purchases autonomously.

Our e-commerce model extends and builds on the e-commerce structures presented in (Galant, 2000), (Chmiel, 2004a) and (Paprzycki, 2004). Basically, our environment acts as a distributed marketplace that hosts e-sellers and allows e-buyers to visit them and purchase products. Buyers have the option to negotiate with the sellers, to bid for products and to choose the seller from which to make a purchase. Conversely, sellers may be approached" instantly" by multiple buyers and consequently, through auction-type mechanisms, have an option to choose the buyer.The implementation of the proposed solution consists of two broad pieces of software namely and electronic marketplace and the bidding agent.

A tabular analysis of above is given on the page 26

Name of Agent System	Works on multiple Auction Types	Final Bidding Plan and Monitoring	Intelligent Bidding	Operate able in Fractal Environment	Operate able in Quasi - Fractal Environment	Intelligence Evolution 0→ 3 level	Near Optimal Results	Successful procurement rates	Scope of Deal	Agent Architecture
Auction Bot	Yes	User	Yes	Yes	No	No	---	Below Satisfactory	Local	Direct Bidding
Kasbah	No	User	No	No	No	No	Not Always	Below Satisfactory	Local	Direct Bidding
Nomad	No	User	Yes	Yes	No	No	Below Average	Below Satisfactory	Local	Purely Reactive
MAGMA	No	Automated	Yes	Yes	No	No	Mostly	Below Satisfactory	Global	Historic Data Based
Bidding Bot	No	User	Yes	Yes	No	No	Average	Below Satisfactory	Local	Probabilistic
Proxy Bot	No	User	Yes	Yes	No	No	Not Always	Below Satisfactory	Local	Greedy
ATTac 2000	No	Automated	Yes	Yes	No	No	Mostly	Below Satisfactory	Local	Greedy Reactive
Southampton TAC	No	Automated	Yes	Yes	No	No	Mostly	Below Satisfactory	Local	Rule Based
Aster	No	User	Yes	Yes	No	No	Not Always	Below Satisfactory	Local	Greedy

Evaluation of various state of the art systems on basis of Research Objectives

References:

[1] R. H. Guttman, A. G. Moukas and P. Maes, "Agent-mediated electronic commerce: a survey", *Knowledge Engineering Review*, Vol. 13, No. 2, pp. 143-152, June 1998.

[2] J. Baumann, "Mobile agents: control algorithms", *Springer Press*, 2000.

[3] W. Brenner, R. Zarnekow and H. Wittig, "Intelligent software agents: foundations and applications", *Springer Press*, 1998.

[4] P. R. Wurman, M. P. Wellman, W. E. Walsh, "The Michigan Internet AuctionBot: a configurable auction server for human and software agents", *in Proceedings of the Second International Conference on Autonomous Agents* (*Agents-98*), pp. 301-308, May 1998.

[5] T. Sandholm, Q. Huai, "Nomad: mobile agent system for an Internet-based auction house", *IEEE Internet Computing*, pp. 80-86, March/April 2000.

[6] M. Tsvetovatyy, M. Gini, B. Mobasher, Z. Wieckowski, "MAGMA: an agent-based virtual market for electronic commerce", *Journal of Applied Artificial Intelligence*, Vol. 11, No. 6, pp. 501-523, September 1997.

[7] Y. A. Tung, R. D. Gopal and A. B. Whinston, "Multiple Online Auctions", *Computer Journal*, Vol. 36, No. 2, pp. 100-102, February 2003.

[8] P. Anthony, W. Hall, V. D. Dang, N. R. Jennings, "Autonomous Agents For Participating In Multiple Online Auctions", *in Proceedings of the IJCAI Workshop on E-Business and the Intelligent Web*, pp. 54-64, July 2001.

[9] P. Anthony, N. R. Jennings, "Evolving Bidding Strategies for Multiple Auctions", *in Proceedings of 15th European Conference on AI (ECAI-2002)*, pp. 178-182, 2002.

[10] R. E. Schapire, P. Stone, D. McAllester, M. L. Littman and J. A. Csirik, "Modeling Auction Price Uncertainty Using Boosting-based Conditional Density Estimation", *in Proceedings of the 19th International Conference in Machine Learning*, pp. 546-553, 2002.

[11] www.forrester.com

[12] Multiagent systems: a modern approach to distributed artificial intelligence By Gerhard Weiss Published by MIT Press, 2000 ISBN 0262731312, 9780262731317

[13] *http://en.wikipedia.org/wiki/Auction*

[14] The Michigan Internet AuctionBot: a configurable auction server for human and software agents by Peter R. Wurman, Michael P. Wellman, William E. Walsh

[15] The Economics of E-commerce By Nir Vulkan

[16] *www.ecommerce-journal.com/tags/j_nomad_system*

[17] www.enteruk.co.uk

[18] ATTac-2000: An Adaptive Autonomous Bidding Agent Journal of Articial IntelligenceResearch 15 2001 189-206

[19] T. Sandholm, Q. Huai, "Nomad: mobile agent system for an Internet-based auction house", IEEE Internet Computing, pp. 80-86, March/April 2000.

[20] T. Ito, N. Fukuta, T. Shintani, and K. Sycara, "BiddingBot: A multiagent support system for cooperative bidding in multiple auctions", in Proceedings of 4th Internet Conference on Multi-Agent Systems, pp. 182-189, 2000.
[21] W. Brenner, R. Zarnekow and H. Wittig, "Intelligent software agents: foundations and applications", Springer Press, 1998.
[22] *Southampton TAC*: An adaptive autonomous trading agent. Resource URI: http://www4.wiwiss.fu-berlin.de/dblp/resource/record/journals/toit/HeJ03
[21] Developing a bidding agent for multiple heterogeneous auctions ACM Transactions on Internet Technology (TOIT)Volume 3 , Issue 3 (August 2003)

4. Problem Domain

There have been several attempts to design sophisticated and efficient bidding strategies for agents participating in online auctions. Some of them are discussed here in detail

Faratin et al. [1998] is broadly similar to the mechanism defined in this article. However, there are several important differences between one-to-one negotiations and multiple auctions. Chief amongst these, are the type of the tactics that are considered relevant and the aspects of the domain that need to be reflected in these tactics. An extension to Faratin's model is given by **Matos, Sierra, and Jennings [2001]** who analyzed the evolution of the negotiation strategies using Genetic Algorithms, and determined which of them are appropriate in which situations.

The aim of this work was to perform an evaluation of the range of negotiation strategies by analyzing their relative success, and how these strategies evolve over time to become a fitter population. This approach is somewhat similar to our work, but the main difference is in the domain that we are dealing with (multiple auctions versus bilateral negotiations using simulated annealing).

BiddingBot is a multi-agent system that supports users in attending, monitoring and bidding in multiple auctions through a process called co-operative bidding [Ito et al. 2000]. This approach demonstrates how agents can cooperate and work together to do the bidding process in multiple auctions. It consists of one leader and several bidder agents, where the leader agent acts as the coordinator and the facilitator of the whole bidding process. Bidding is done by exchanging messages between the user, the leader agent and the bidder agents.

However, the main problem with this approach is that the agents do not actually make the bidding decision. This decision is left to the user. Thus, the agents do not have full autonomy and the decision-making process is slow since the agent needs to interact with the user from time to time.

The **trading agent competition (TAC) [19]** provided a platform for agent designers to develop autonomous agents that can compete with one another in multiple simultaneous auctions for complimentary and substitutable goods.

The key feature of TAC is that it required autonomous bidding agents to buy and sell multiple interrelated goods in auctions of different types **[Greenwald and Stone 2001]**. Each participating agent is a simulated travel agent with the goal of assembling a number of travel packages for its eight clients.

Each client is characterized by a random set of preferences for the possible arrival and departure dates, hotel rooms and entertainment tickets. The objective of a TAC agent is to maximize the total satisfaction of its customers (i.e., the sum of the customer's utilities). The competition attracted a number of alternative agent designs (e.g., **ATTac-2000 [Stone et al. 2001]**, RoxyBot **[Boyan and Greenwald 2001]**, Aster **[Greenwald and**

Stone 2001] and SouthamptonTAC [He and Jennings 2003]). Although there are clearly some similarities with our scenario, there are also a number of important differences. In particular, we concentrate on the bidding strategies to obtain a single item rather than worrying about the complementary goods that need to be bundled with the desired item.

Moreover our algorithm proposes a coordination mechanism to be used in an environment where all the auctions terminate simultaneously, and a learning method to tackle auctions that terminate at different times.

Byde [2001] also considers this environment, but utilizes stochastic dynamic programming to derive formal methods for optimal algorithm specification that can be used by an agent when participating in simultaneous auctions for a single private-value good. Both of these works are designed specifically for purchasing items in the multiple English auctions and their algorithms are not applicable in a heterogeneous protocol context. **Byde et al. [2001]** presented another decision theoretic framework that an autonomous agent can use to bid effectively across multiple auctions with various protocols (namely, English, Dutch, first price sealed bid and Vickrey auctions).

In order to come up with the best bid value that guarantees the delivery of the item, an agent must always speculate about future events. To do this, [**Byde**] presented an approximation function that provides an estimate of the expected utility of participating in the set of future auctions. The decision making algorithm presented by [**BIDE**] works in this way; it selects all the Dutch, English, and sealed bid auctions that the agent wishes to consider. It then tests the union of all three sets of bids to determine the utility of this course of action (using the approximation function).

This process would be repeated for all the possible combinations of Dutch, English and sealed bid auctions and returns the auction set with the highest expected utility. This auction set contains the list of all the auctions that the agent should bid in at that particular point in time. This approach can be employed to purchase single or multiple items in online auctions.

However, at this time, the evaluation of the algorithm's operational effectiveness has not been reported and so we cannot determine whether it will outperform our heuristic methods.

No Known system up to date concerns with the issues of specific Agent Architecture, market payment protocols and development of bidding strategies via Simulated Annealing and concerning the Quasi-Fractol landscape produced as a result of heterogeneous market protocols. Furthermore there is no known system which addresses the simultaneous use of large number of resource, dynamic resource requirements, complex communication structure and stringent performance requirements in e-commerce multi-agent systems.

To address these shortcomings, we believe it is necessary to develop an autonomous agent that can participate in multiple heterogeneous auctions, that is empowered with trading capabilities and that can make purchases autonomously.

4.1 Objectives in order of Priority

The objectives of the projects are

- The model and develop a test bed for simulating multiple auctions using heterogeneous auction and payment protocols.
- To develop an e-commerce intelligent bidding agent to facilitate user to automated participate in dozens of heterogeneous auctions at a given time and produce the best deal with in given constraints
- To implement an intelligent bidding strategy for the user agent using artificial intelligence and probabilistic stochastic algorithm of simulated annealing.
- To eliminate any chances of the agent's strategy getting struck at local maxima in a factoral and quazi factoral landscape distribution produced as a result of auction environment.
- To introduce into the e-commerce test bed the theory and practice of payment and contracting protocols to make the simulated market environment more realistic and fair to trade in.
- The introduction of intelligent level 0, 1, 2, and 3 agents in the simulated environment.
- To introduce the concept of look-ahead and no look ahead in virtual auctions.
- To introduce a model and framework for enabling trust among trading parties and eliminate the concept of cheating and information abuse.

4.2 Domain of this Research Venture

As it is obvious from the Title of the research project and the above discussion, that it comprises of the broad fields in computer science including

- Auction Theory (Heterogeneous Auction protocols and payment protocols)
- Distributed Artificial Intelligence
- Agent Oriented Systems
- Probabilistic Stochastic optimization algorithms
- Optimal solution in Fractal and Quasi-Fractal distributions

4.3 Project Scope

E-commerce is growing at an exponential rate. According to one study, the Internet economy grew at a rate of 174.5 percent annually from 1995 to 1998. This growth follows the equation Internet economy *every* year almost exactly. A number of analysts forecasted that the internet economy will exceed US $1 trillion (1012) in 2002; Forrester Research (http://www.forrester.com) recently predicted that the worldwide Internet economy has reached US $6.9 trillion in 2004. (The Internet Economy Indicators,

Indicators Report, June 1999, available online at http://www.internetindicators. com/features.html.)

The most popular way of conducting business in the modern day e-commerce environment is by Auctioning. (http://en.wikipedia.org/wiki/Auction) It is the process of buying and selling, by offering them up for bid, taking bids, and then selling the item to the successful bidder.

Microsoft discovered that 70 per cent of its purchases were for relatively small items which took up something of the order of 3 per cent of its purchase volume. The company discovered that a large amount of employee time was spent on the procurement process and hence invested $1.1m on a system known as MSMarket. When a Microsoft employee wishes to buy some item such as stationery they log into MSMarket, the system identifies them from their login identity and consults its database to discern what rules should be applied to purchases from that employee. The employee informs the system that they require some stationery and a screen of items and prices negotiated with a supplier are displayed. The employee purchases what is required and the order is sent over the Internet; an e-mail is then sent to their manager to inform them of this and a tracking number generated which can be used to query the supplier if the item has not been delivered by a certain time. The use of MSMarket has increased exponentially since it was deployed and it now handles more than $3 billion of orders. According to one study, the Internet economy grew at a rate of 174.5 percent annually from 1995 to 1998. This growth follows the equation Internet economy. *every* year almost exactly[11].

Since the research will produce a bidding agent to compete in auctions on users behalf along with a set of strategies that are very diverse and can be implemented either standalone or in collaboration in any such future automated bidding system to act in any heterogeneous e-commerce environment.

So it is obvious that the research concerning improvement and facilitating of worlds 25% and growing commodity trading is obviously very beneficial and cost effective.

4.4 Proposed Solution

Literature in both books and research articles from contributors defines various types of auctions. This section gives a brief account of the various proposed auction types

4.1 Auctions By Value
There are three qualitatively different auction settings depending on how an agent's value of the item is formed. [12]

2.1.1 Private value Auctions [12]
In case of a PVA (private value auction) the actual worth of the commodity depends on your own preferences. For example auctioning off the cake that the winner bidder will eat. The key is that the winning bidder will not resell the item or get utility from showing it off to others. [12]

2.1.2Common value Auctions [12]
In case of CVA (common value auctions) the value of the commodity depends entirely on other persons view of the value of an item.

Correlated value Auctions [12]
In COVA(correlated value auctions) the value of the commodity depends partly on own preferences and partly on others' values for it.

2.3 Auctions by Protocol
Wikipedia the worlds largest online resource of information divides the auctions protocols into two main categories namely Primary and secondary type of auctions having four and seven types of sub auctions each. These may again be subdivided into two types as seen in table 2.1

Primary Auctions	**Secondary Auctions**
The English Auction	All-pay auction
The Dutch Auction	Buyout auction
The First-Price, Sealed-Bid Auction	Combinatorial auction Lloyd's syndicate auction
The Vickrey Auction	No-reserve auction (NR)
	Reserve auction
	Silent auction
	Top-Up Auction
	Walrasian auction

Table 2.1 *Major types of auctions and their classifications (More can be read about them at http://en.wikipedia.org/wiki/Auction) [13]*

For purpose of this research we are only considering Primary Auctions. The agent working on user's behalf should monitor and collect information from the ongoing auctions, make decisions on behalf of the consumer and endeavor to guarantee the delivery of the item. The agent must ensure that it never bids above the private valuation (the maximum amount that the consumer is willing to pay) and it tries to get the item in a manner that is consistent with the consumer's preferences.

To this end, this project is on our work in developing such a bidding agent. The agent has a range of strategies that it can employ depending on the user's aims and the environment in which the agent finds itself. The strategies themselves are heuristic in nature because the multiple heterogeneous auction environment is very complex, dynamic and unpredictable, making it impossible to find an optimal strategy that can be used in practical contexts. Moreover, the effectiveness of the strategies is heavily influenced by the nature of the environment [Anthony et al. 2001]. For this reason we decided to have different strategies for different circumstances. As the range of potential strategies is huge, we decided to use a Simulated Annealing to search for effective strategies for each of the various environments that we identified on basis of [Anthony and Jennings 2002].

We chose this particular method because SA has been known to perform well in areas where the space to be searched is large and not well understood [Mitchell 1996]. Having evolved the strategies, the agent adopts the one that is most appropriate to its prevailing context.

The agent has a range of strategies that it can employ depending on the user's aims and the environment in which the agent finds itself, the agent adopts the one that is most appropriate to its prevailing context.

The functional diagram of the proposed agent based e-commerce system is as under

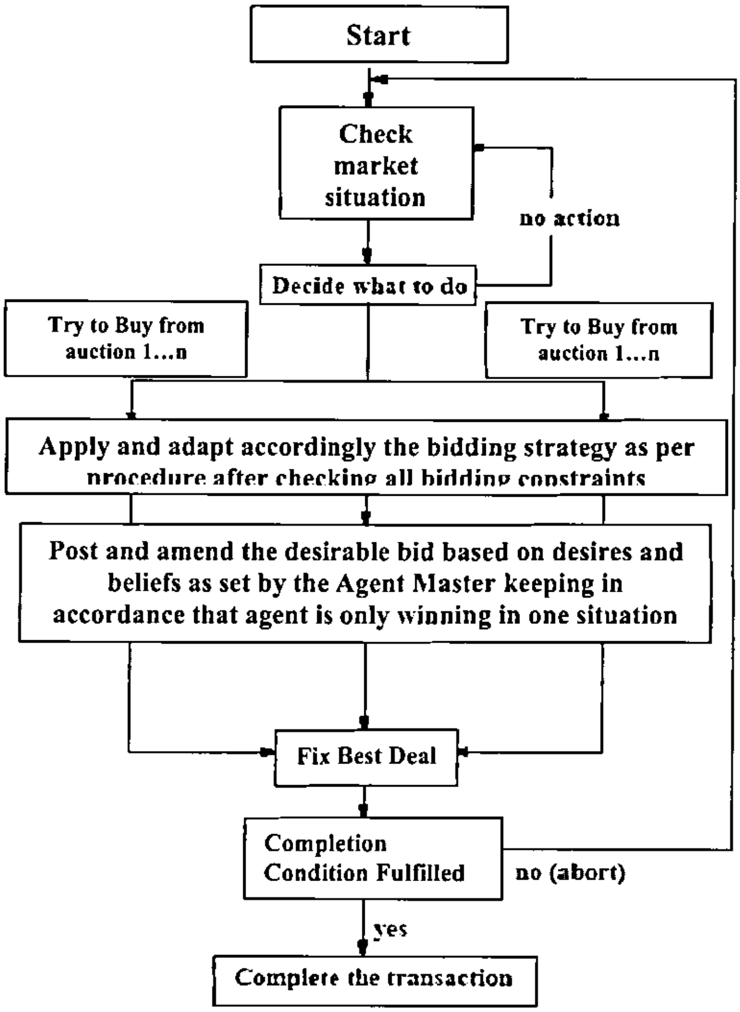


Figure 4.1 Fuction Diagram of SabibAgent

- At the start of each auction, the electronic marketplace generates a group of random bidders to simulate other auction participants. These participants operate in a single auction and have the intention of buying the target item and possessing certain behavior. There are however experiments where multiple auction bidders compete for commodities. They maintain information about the item they wish to purchase, their private valuation of the item, the starting bid value and their bid increment. These values are generated randomly from a standard probability distribution. Their bidding behavior is determined based on the type of auction that they are participating in. In an English auction, they initiate bidding at their starting bid value; when making a counter offer, they add their bid increment to the current offer, and they stop bidding when they acquire the item or when their private valuation is reached. In a Dutch auction, they wait until the offer value is equal to or less than their private valuation before making an offer. Finally, in a Vickrey auction, they bid at their private valuation. Tradeoffs have to be made on basis of the intelligent strategy of the Agent.

- We will identify more than six most influential intensions on behalf of both user and the heterogeneous marketplace, and make tradeoffs among them after making the possible selection patterns form the customers view point. These intensions include time tactics, speculation, remaining auction, desire for bargain, price limit, eagerness to procure, availability etc. The agent will automatically do the tradeoffs and derive a strategy on runtime based on the pattern application scheme.

- The evaluation function measures how well the particular solution performs against the others. Designing the evaluation function is one of the key facets of Simulated Annealing. Individual success rate in obtaining the item is considered as the evaluation function. Success rate is calculated by running the simulations 500 times and than calculate the average success rate.

- A suitable initial temperature T_0 is one that results in an average increase of acceptance probability P_0 of about 0.7. In other words, there is a 70% chance that a change which increases the objective function will be accepted.

- In SA algorithm the final temperature is determined by fixing
    - The number of temperature values to be used.
    - The total number of solutions to be generated.

- Alternatively, the search can be halted when it ceases to make progress. Lack of progress can be defined in a number of ways:

- No improvement (i.e. no new best solution) being found in an entire Markov chain at one temperature, combined with

- The acceptance ratio falling below a given value.

- The process is very lengthy and will be performed as follows.
    - After very comprehensive "all to all" pattern matching simulations run to identify the resultant successful scheme for the given pattern. Make results on the basis of agent's success and best solution factors. After the run of all agent strategies 500 times and get the resultant data. The best strategy form the above steps will be selected for single preference. This best strategy is not for all scenarios and the above set of actions will be performed again for each scenario. Make patterns for user and market

preferences based on the combination of preferences and perform the above defined steps for all to all combinations.

- Thus at the end of these activities we will have an ecommerce system with
    - Well defined, definite Domain
    - Categorization of components comprising the Domain
    - Representative samples which will integrate for full functionality
    - Readily available feature of reusability, modifiability and enhancement by means of Black Box implementation of Object Oriented principles
- Concluding, this proposal advances the current state of the art research activities in the following ways.
- First, we are going to develop a high level decision-making framework for an agent to bid across multiple concurrent auctions of varying protocols with varying start and end times.
- Secondly, this framework is heuristic in nature and uses tactics and strategies to vary the agent's behavior so as to ensure a good fit with the user's auction objectives.
- Thirdly, we intend to evolve a strategy that is effective in our multiple auctions context. This strategy consists of multiple evolved sub-behaviors that are appropriate in different environmental settings and with different user objectives. This strategy can be termed the *intelligent bidding strategy.*
- Fourthly we security and trustworthiness mechanisms imposed are not implemented in any market simulation as to our knowledge; these mechanisms not only make the simulation results very close to reality but are also very strong candidate for being the strategies to eradicate any sort of unfairness in any auction based electronic system.

Prior to our work, no solution satisfied this wide variety of such Hybrid Heterogeneous requirements. The resultant system is very heavy in development but is very resource friendly in runtime environment since a simple switch will be required for the strategy selection.

4.5 Output of the Proposed Project

The output of the proposed projects is

- An e-market place as a simulation test bed for simulating the multi agent e-commerce system.
- A new model and framework for ecommerce systems using probabilistic stochastic algorithm of simulated annealing.
- Research and the implementation of the Simulated Annealing algorithm for production of local maxima problem free optimal solutions.
- An intelligent, very light to run bidding agent for facilitation of user in e-commerce environment.
- Our proposed infrastructure will be suitable for simultaneous use of large number of resource, dynamic resource requirements, complex communication structure and stringent performance requirements.

References:

[1].FARATIN, P., SIERRA, C., AND JENNINGS, N. R. 1998. Negotiation decision functions for autonomous agents. Int. J. Robot. Autonom. Syst. 24, 3-4, 159–182. IEEE Internet Comput. 52–60.

[2].MATOS,N., SIERRA, C., AND JENNINGS,N. R. 2001. Successful negotiation strategies: An evolutionary approach. In Proceedings of the 3rd International Conference on Multi Agent Systems (Paris, France). IEEE Computer Society Press, Los Alamitos, Calif., 182–189.

[3].BOYAN, J. AND GREENWALD, A. 2001. Bid determination in simultaneous actions: An agent architecture. In Proceedings of the 3rd ACM conference on Electronic Commerce. ACM, New York, 210–212.

[4].BOYAN, J. AND GREENWALD, A. 2001. Bid determination in simultaneous actions: An agent architecture. In Proceedings of the 3rd ACM conference on Electronic Commerce. ACM, New York, 210–212.

[5].SANDHOLM, T. W. 1999. Distributed rational decision making. In Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, G. Weiss, Ed. The MIT Press, Cambridge, Mass., USA, 201–258. STONE, P., LITTMAN, M. L., SINGH, S., AND KEARNS, M. 2001. Attac-2000

[6].DUMAS, M.,GOVERNATORI,G.,HOFSTEDE, A. T., AND RUSSELL,N. 2002. An architecture for assembling agents that participate in alternative heterogeneous auctions. In Proceedings of theWorkshop on Research Issues in Data Engineering. IEEE Computer Society Press, Los Alamitos, Calif., 75–83.

[7].http://www.forrester.com

[8].http://en.wikipedia.org/wiki/Auction

[9].ANTHONY, P., DANG, V. D.,HALL, W., AND JENNINGS, N. R. 2001. Autonomous agents for participating in multiple online auctions. In Proceedings of the IJCAI Workshop on E-Business and Intelligent Web. 54–64.

5. Software Architecture

The chief objective is to develop a framework for intelligent agents for effective use in simultaneous and parallel multi-protocol auctions scheme. The objective is to develop a relevant philosophical agent framework that has the capability to resolve the bidding problem with respect to a definite scenario and bring into play the gained knowledge to surface the bidding strategy mechanisms while gaining maturity for its present and future ventures in universal marketplace by way of consultation of its intelligent strategies and Decision Aid Repositories.

As the number of online customers and businesses are greater than ever, bid management is becoming radically more complex, and analyzing the relevant data form the electronic marketplace can be an overwhelming never ending work. It is apparent now that the use of intelligent agent technology mitigates the risks coupled with meeting current and future information management needs to formulate automated bidding.

As the number of online customers and businesses are greater than ever, bid management is becoming radically more complex, and analyzing the relevant data form the electronic marketplace can be an overwhelming never ending work. It is apparent now that the use of intelligent agent technology mitigates the risks coupled with meeting current and future information management needs to formulate automated bidding.

We have opted intelligent agent technology because it allows the customer to more resourcefully track objectives of a target buying, administration and utilization of resources, unfolding market proceedings and internal/external state of affairs are a good reliable source of perceived inputs to the dynamic bidding plan generation procedure. The use of Intelligent Agent technology empowers the stake holders with enhanced market exploitation vis-à-vis analyzing masses of data.

Since an agent is an autonomous entity with an ontological commitment and an agenda of its own [1]. Every agent possesses the ability to act autonomously. In the e-commerce environment an agent is often acting on a principal's behalf and has a legal duty to act in that person's best interest. An agent may interact or negotiate with its broker and/or with other agents. It may make decisions, such as whether to trust and whether to cooperate with others. They are capable of making independent decisions and taking actions to satisfy internal goals based upon their perceived environment.

Our agent implementation has a stronger notion of autonomy than traditional systems in addition to a reactive, proactive or social behavior as affected on the concerned scenario. If the states of the Scenarios/ Environments can be characterized as a set $S_c = \{S_{c1}, S_{c2} \ldots S_{cn}\}$ where Sc_i is the scenario. At any given instant of time the agent can be faced with only one element of the set of scenarios then the action of our agent can be one element of the set of predefined actions $A_c = \{A_{c1}, A_{c2}, A_{c3}, A_{c4} \ldots, A_{cn}\}$. By application of automate theory it can be represented in the functional form of

A_{cx} : Sc* → A_n which maps environment states encountered into appropriate action. We are assuming that the set of environments is limited, predictable and deterministic.

On the agent architectures like *reactive agent architecture, layered agent architecture, belief desire intention architecture* and logic based architectures, we were unable to make a match with our requirements due to highly versatile and hybrid environment constraints. The BDI architecture seems philosophically closest to the scenario in demand. Since the BDI Architecture has its ancestry in the philosophical ritual of understanding *practical reasoning*, the process of deciding, moment by moment, which action to perform in the furtherance of our goals [2], we found it most convenient if it was molded to fulfill our requirements, but since the changes needed were drastic, it resulted into a whole new style. We suggest that [2] should be consulted for anyone who wants further insight on BDI Architecture.

The tailored BDI agent has a set of plans, which defines sequences of actions and steps available to achieve a certain goal or react to a specific situation. The agent reacts to events, which are generated by modifications to its beliefs, additions of new goals, or messages arriving from the environment or from another agent. An event may trigger one or more plans. The agent commits to execute one of them, that is, that plan becomes the intention. Plans are executed one step at a time. A step can query or change the beliefs, performs actions on the external world, and submits new goals. The operations performed by a step may generate new events that, in turn, may start new plans. A plan succeeds when all its steps have been completed; it fails when certain conditions are not met.

We have personalized the traditional BDI to the E-BDIArchitecture with *static intention centric focal point*, while the Desires and Beliefs are persistently updated according to the real-time input data. Since a lot of effort is spent on the development of Architecture Description Languages (ADLs) as can be seen from Rapide[3], Darwin [4], Aseop [5], Unicon [6], Wright [7], Acme [8] and Faulkner [9]. The theoretical aspects of the philosophical Intentions, Beliefs and Desires in E-BDIArchitecture along with their Architectural Description are given below.

5.1. Intentions

These are options laid down by the user, and are unswervingly responsible for formulation of the outcome in the ongoing process according to user's requirements. Given that Intentions are equivalent to owner's guidelines, thus they not only impel the deliberation process but are utterly accountable for mean-ends reasoning by serving as means of legalization for Desires and Beliefs. The Intentions are answerable for the SABIBAgent's current focus.

Intention → Cumulative Weight Load (CWL)
CWM → AIW
| AIW connective CWL
AIW → Intention | 0.0|0.1|...|1.0
Intention → RT | RA | BD | EP | LP | SD
RT → Predicate

RA → Predicate
BD → Predicate
EP → Predicate
LP → Predicate
SD → Predicate
Predicate → Function (Predicate)
| Function
Function →Functions of ECOMMBDI
Connective → $\wedge$ | $\vee$ | $\Rightarrow$

Legend
CWL → Cumulative Weight Load (Intentions)
AIW → Atomic Intention Weight
RT → Remaining Time
RA → Remaining Auctions
BD → Bargain Desire
EP → Eagerness to Procure
LP → Limit Price
SD → Supply Demand

5.2. Desires

These are the set of options generated during the progression of agent pre bid training. They comprise of the set former solutions by parties for the current problem being on hand, which in this case will be the values of preceding successful bids for procurement of the same item sought for by SABIBAgent.

$D_{set} \subseteq Option_{set}$
Desire $\in D_{set}$
D_{set} → (Desire)*
Desire → Atomic Desire | $\neg$ Atomic Desire
Atomin Desire → Procurement Price
Procurement Price → FPV_1 | FPV_2 | …| FPV_n

Legend
D_{set} → Set of Desires
FPV → Floating Point Value
* → Zero or more repetitions

5.3. Beliefs

The Desires if validated by matchmaking with the Intensions become Beliefs. For example if the Desire was to buy an item for 20$, and the Intention was to buy it for 40$ or less then a Belief is established that the item can be bought and the bidding will instigate on this Belief. Similarly the optimized Desire set gives augment to the Beliefs set which are consistent with the Intentions.

$B_{set} \subseteq Desire_{set}$

Belief $\in B_{set}$
$B_{set} \rightarrow (Belief)^* \mid (Belief\ connective\ (B_{set}))^*$
Belief $\rightarrow$ Atomic Belief
$\mid \neg$ Atomic Belief
Connective $\rightarrow \wedge \mid \vee \mid \Rightarrow$

Legend
B_{set} ➔ Set of Beliefs
* ➔ Zero or more repetitions

In short SABIBAgent is a large scale work flow engine based on a modified philosophy of BDI (Belief Desire Intention) Architecture called e-BDIArchitecture, with static Intentions. The agent communicates with global e-marketplace simulator and other agents by means of a built-in asynchronous *message passing scheme* (MPS). The agent exploits all potential resources available at hand to work on user's behalf and continuously reworks the solutions as problem, parameters, constraints or execution environment changes.

The intentions still have the central status in the retailored system and have the following properties and role in the mean-ends reasoning process

5.4. Intentions and their effect on mean ends-reasoning in e-BDI Framework

• **Non Varying Intentions**

Intentions for a bidding cycle are constant and have precedence weights associated with them provided by the owner. Only the owner can withdraw the agent if no procurement has been made and the whole bidding cycle has to be restarted if Intentions or Intention weights are changed.

• **Intentions impel means-ends reasoning.**

If an Intention has been made to buy an item from the market place, then the agent will attempt to achieve the Intention, which involves, amongst other things, deciding *how* to achieve it, for example, by entering an auction and bidding for the desired item. Moreover, if one particular course of action fails to achieve an Intention, then It will typically attempt other generated action plans. Thus if it fails to gain an item in one auction, it will try another auctions selling the same commodity.

• **Intentions constrain future deliberation.**

If Intention is to buy a PC, then it will not entertain options that are incompatible with this Intention. Only those Intentions are entertained in the SABIBAgent which are mutually exclusive and the probability of achieving both simultaneously is no infinitesimal. For example bidding for an item at the lowest price ever, with desperation factor of a 100%.

•Intentions persist.

The SABIBAgent if intelligent will not typically give up on its Intentions without good reason—they will persist, typically until either it believe it has successfully achieved them, if it believes they cannot be achieved or are unrealistic, or else because the purpose for the Intention is no longer present.

• Intentions manipulate Beliefs upon which future realistic reasoning is based.

If the agent adopts the Intention to buy an item, then it can plan for the future on the assumption that it *will* bid for that item and acquire it. For if it intends to procure some item while simultaneously believing that it will never be able to procure one, then it is being irrational.

• Re-evaluation of Beliefs and Desires

Beliefs and Desires are re-evaluated if they are not according to the required criteria during the first training phase of the agent. They also determine if the Intentions are realistic or not. If the set of Beliefs is an empty set then the Intentions are not realistic because no such Desires could be gathered or the Desires are inconsistent with the Intentions.

The Beliefs and Desires may be reevaluated in the initial or middle part once the bidding cycle commences if the critical e-market parameters like availability, supply demand etc change drastically or if the other agent's bidding strategies or prices start varying drastically.

5.5. The working of Agent

The agent is created by the user and is initialized by a set of Intentions. The agent is trained on basis of real-time market data repository. It then negotiates with the heterogeneous market simulator [10] and requests for general market auctions set repository. It reads the repository for selection on committing to the suitable auctions. Suitable auctions are the one which present the in demand item according to our Intentions. These suitable auctions are then and an option set is generated and again filters to act on only the best auctions available. These refined options are the foundation stone of Desires built right in the next step. Desire sets again go though a filtering process and are then translated into Beliefs by matchmaking with the Market Data Repository. As a result of this matchmaking only the valid Beliefs are filtered to be part of the mean-ends reasoning process. If no valid Belief(s) are filtered though for effective mean-ends reasoning to take place the process is again repeated form rebuilding Desires onward until a valid Belief set is established which is consistent with our Intentions since newer auctions and results are updated in the data clusters of repository. The agent then commences bidding on basis of this Belief set initial most optimal value and if unsuccessful, the next optimal Desire and Belief is established for next bidding to commence.

IAP&PM creates and awakens the EBAE by subscribing scenario specific information and EBAE then starts perusing its procurement goal in a particular auction synchronizing its actions with other EBAE's. Each EBAE collaborate in one to many relationships or roles with others of its same kind. They are aware of each other via IAP&PM in SABIBAgent system. These roles are defined as DuchEBSE, EnglishEBSE, VickeryEBSE, 1st Price SealedBidEBSE etc. The agent Architecture is scalable and other roles can be easily incorporated by managing them in the Problem Solving Knowledge Base.

The action strategy provides traceable information on the overall current task progress. The intentions which remain unchanged during the course of action (i.e. static Intentions) are the cornerstone of action strategy progress. They are the answer to the what, the why, the how, the when, the where, the who's of the bidding problem. Furthermore the action module also decides whether to carry on, quit or suspend the bidding according to the intelligent action plan generated by the intelligentsia process.

A standalone working scenario of the Agent's means-end reasoning plan in ADL form can be represented as

Goal (Item, Market, Auction Set, Protocol) /* Universal Goal*/

Achieve (Item (Cumulative Weight Load))

Run (IWCF, DCF, ASF, AFF, (OGF, OFF, DGF, DFF, BGF, BBMRF, IVF)*)$^+$

Run ($EBAE_1$, ….. , $EBAE_n$) $^+$

Failed (IVF = = 0) && Abort

Succeed ((IVF != 0) && Bid Successful

Legend
IWCF ➔ Intention Weight Calculation Function
DCF ➔ Data Clustering Function
ACF ➔ Auction Selection Function
AFF ➔ Auction Filtering Function
OGF ➔ Option Generation Function
DGF ➔ Desire Set Generation Function
DRF ➔ Desire Set Revision Function
BGF ➔ Belief Generation Function
BBMRF ➔Belief Based means-end Reasoning Function
IVF ➔ Intention Validation Function
The mid term goals can be defined as
Sub-Goal → Validate (Belief)
| Achieve (Belief)
| Abort (Belief)
| Load (Belief)
| Abort (Session) | Abort

As the environment variables change the re-planning includes how the Belief/Desire are to be effected accordingly thus generating the new scenario centric optimized solution. SABIBAgent continuously monitors the plan and forces re-planning if and only if real-time changes in environment variables are drastic, forcing the agent to take a bolder approach.

The service descriptions provided to and by the agent in the MAS (Multi agent simulation) are descripted by the following specification.

<Train for Auction Participation>
[AuctionMembership= = no]
    → FetchIntentions( )
SetCumulativeLoad(Intentions)
FetchMarketData()
    WHERE
(Auction ε {Total Auctions} && time > Auction.End && Desired_Item)

SelectAuctions( )
    WHERE
(Auction ε {Total Auctions} && time < Auction.End && Desired_Item)

FilterAuctions( )
    WHERE
(Auction ε { Selected Auctions } && time > Auction.End && Desired_Item)
&&($P_i^w(v) = (\sum_{P>v} P_i^c(P) + P_i^c(v)/2) \geq 50\%$) || ($P_i^w(v) = (1/\sum n_i/x) \geq 50\%$)

GenerateDesireSet( )
    FOR
(($\Sigma_{i=1....n} W_j * I_j$)- $W_{DP} < W_{DP}$) &&
(($\Sigma_{i=1....n} W_j * I_j$)- $W_{BD} \leq W_{BD}$)
    DesireSetRevision( )
    WHERE
($Desire_i$.value > Private Value) && ($Desire_i$.count > Threshold)

GenerateBelief( )
    FOR
(($\Sigma_{i=1....n} W_j * I_j$)- $W_{DP} < W_{DP}$) &&
(($\Sigma_{i=1....n} W_j * I_j$)- $W_{BD} \leq W_{BD}$)

Means-endsReasoning( )
    SimulatedAnnealing( )
    ValidateIntention( )
IF
    Belief ! ≥ Intention

```
<Join Auction>
    → time
Set Connection = Connection ϕ
Submit_Request(Auction_ID, Members_ID, Items)
WHERE
Current.time < Auction.StartTime && Desired_Item


<Get Market Membership>
[Join Auction (Auction_ID,    Members_ID, Items)]
→       AuctionMembership = = yes && AuctionID = = AID, MembershipID = = MID
        IF
Auctioneer_Accept(MID) = = ok

<Menas-ends Reasonig>
        [AuctionMembership= = yes]
    →    Means-endsReasoning( )
{
 CentralizedLerning(EAData)
Decentralized Learning(DAData, VAData)
        ReinforcementLearning(SD)
QLearning(RT, RA, BD, EP, LP)
}

<Initiate Bidding>
        [AuctionMembership = = yes]
Submit_Bid(Auction,MID,BeliefValue)&& FilterdAuctions.Auctioninfo &&
Auction.Auct.ID = = Auction

<accept Acknowledgement of offer>
[Initiate Bidding (Auction,MID,BeliefValue)]
    → !Bid_ID = = bid_ID
        IF
AcutioneerAgent [ Bid_Adjust (Self, AuctionID,midbidID)

<Adjust Bid After Failure>
        CentralizedLerning(EAData)
Decentralized Learning(DAData, VAData)
        ReinforcementLearning(SD)
        Q-Learning(RT, RA, BD, EP, LP)
[Submit_Bid(Auction,MID,Belief.Valu-e)]
Belief + = Belief
    → Submit_Bid(Auction,MID,Belief.Value)
        IF
Bid_Failed (Self, AuctionID, mid, bidID)
```

WHERE
AID = = Acution &&Bid_ID = = bidID && MID = =mid

<Pay acknowledged offer price>
[Submit_Bid(Auction,MID,Belief.Valu-e)]
→ Pay (Bid_ID, Payment)
IF
Bid_Accepted(mid, AID, bidID)= =yes WHERE
AID = = Auction && Bid_ID = = bidID && MID= = mid
<Quit From Auction>
→ Quit_Auction (AcutionID)
Membership = = no
IF
Bid_Failed (Self, AuctionID, mid, bidID)
WHERE
AID = = Acution &&Bid_ID = = bidID && MID = =mid
IF
Bid_Success (Self, AuctionID, mid, bidID)
WHERE
AID = = Acution &&Bid_ID = = bidID && MID = =mid

The degree of boldness or cautiousness [11] is heavily dependent on the active environment of the marketplace simulator as it is in any e-commerce auction house like yahoo auction [12], eBay[13], Amazon[14], Priceline[15], UBid[16] and many others. The implemented system starts off as a cautious agent and evolves towards a bolder approach. If the critical market factors like supply-demand, competent agent density etc. are highly fluctuative. The boldness factory has a threshold of 50% since above this value the agent gets stuck in reevaluating its strategical plan and does little effective work as was the case shown by experimental data sets.

References:

[1] Distributed Artificial Intelligence MIT press.

[2] M.E. Bratman, D.J. Israel, and M.E. Pollack. Plans and resource-bounded practical reasoning. Computational Intelligence, 4:349-355, 1988.

[3] D. C. Luckham, J. J. Kenney, L. M. Augustin, J. Vera, D. Bryan, and W. Mann. Specification and Analysis of System Architecture Using Rapide. *IEEE Trans. on Software Engineering*, pp. 336-355, April 1995.

[4] J. Magee and J. Kramer. Dynamic Structure in Software Architectures. In *Proc. of the 4th Symposium on the Foundations of Software Engineering* (FSE4), pp. 3-14, San Francisco, CA, October 1996.

[5] D. Garlan, R. Allen, and J. Ockerbloom. Exploiting Style in Architectural Design Environments. In *Proceedings of SIGSOFT'94: Foundations of Software Engineering*, pages 175–188, New Orleans, Louisiana , USA, December 1994.

[6] M. Shaw, R. DeLine, D. V. Klein, T. L. Ross, D. M. Young, and G. Zelesnik. Abstractions for Software Architecture and Tools to Support Them. *IEEE Trans. On Software Engineering*, pp. 314-335, Apr. 1995.

[7] R. Allen and G. Garlan. *Formal Connectors*. Technical Report, CMU-CS-94-115, Carnegie Mellon University, March 1994.

[8] D. Garlan, R. Monroe, and D. Wile. ACME: An Architecture Description Interchange Language, In *Proc. of CASCON 97*, Toronto, pp. 169-183 Jan. 1997.

[9] P. C. Clements. A Survey of Architecture Description Languages. In *Proc. of the Eighth International Workshop on Software Specification and Design*, Paderborn, Germany, March 1996.

[10] http://www.forrester.com

[11] D.Kinny and M.Georgeff. Commitment and effectiveness of situated agents. In proceedings of Twelfth International joint conference on Artificial Intelligence (IJCIA-91), pages 82-88, Sydney, Australia, 1991.

[12] www.yahoo.com

[13] www.ebay.com

[14] www.amazon.com

[15] www.ubid.com

[16] www.priceline.com

6. Design

6.1 The SABIBAgent Framework

Like any other Intelligent Agent SABIBAgent is Agent that is situated in some environment and, that is capable of autonomous actions in this environment in order to meet its design objectives [1].

The agent Architecture we used to develop our intelligent agent for competition in the e-commerce bidding agent society is a revamp of Belief Desire Intension Architecture retailored from the scratch to fit our requirement goals. The consequential Architecture is named as e-COMMBDI and is drastically different form the original approach. It employs user provided static Intentions (Deliberations) with attached priority, and the Beliefs and Desires are naturally inspired and judged by analysis on basis of Intensions(means end reasoning) for accomplishment of the goal.

Since the BDI Architecture has its ancestry in the philosophical ritual of understanding *practical reasoning*—the process of deciding, moment by moment, which action to perform in the furtherance of our goals [6] we found it most convenient if it was molded to fulfill our requirements, but since the changes needed were drastic it resulted into a whole new style. We suggest that [6] should be consulted for anyone who wants further insight on BDI Architecture.

We have personalized the E-COMMBDI Architecture with static Intention Centric focal point, while the Desires and Beliefs are persistently updated according to the real-time input data.

The conventional Agent based software solutions (Give names) and search engines like BidX[2], Auction Watch[3] and Auction Beagle[4] etc. Do not even have the architectural potential to maneuver for an optimal deal in a Distributed Heterogeneous Multi-Auction-Protocol Environment, are if implemented prove too much error prone and cumbersome when managing unique items even with above defined planning process. Thus we felt necessary to implement the full SABIBAgent framework along with the efficient bid planning process. The proposed agent Architecture separates policy form mechanism effectively thus making decision aid results that provide dynamic planning and execution monitoring capabilities.

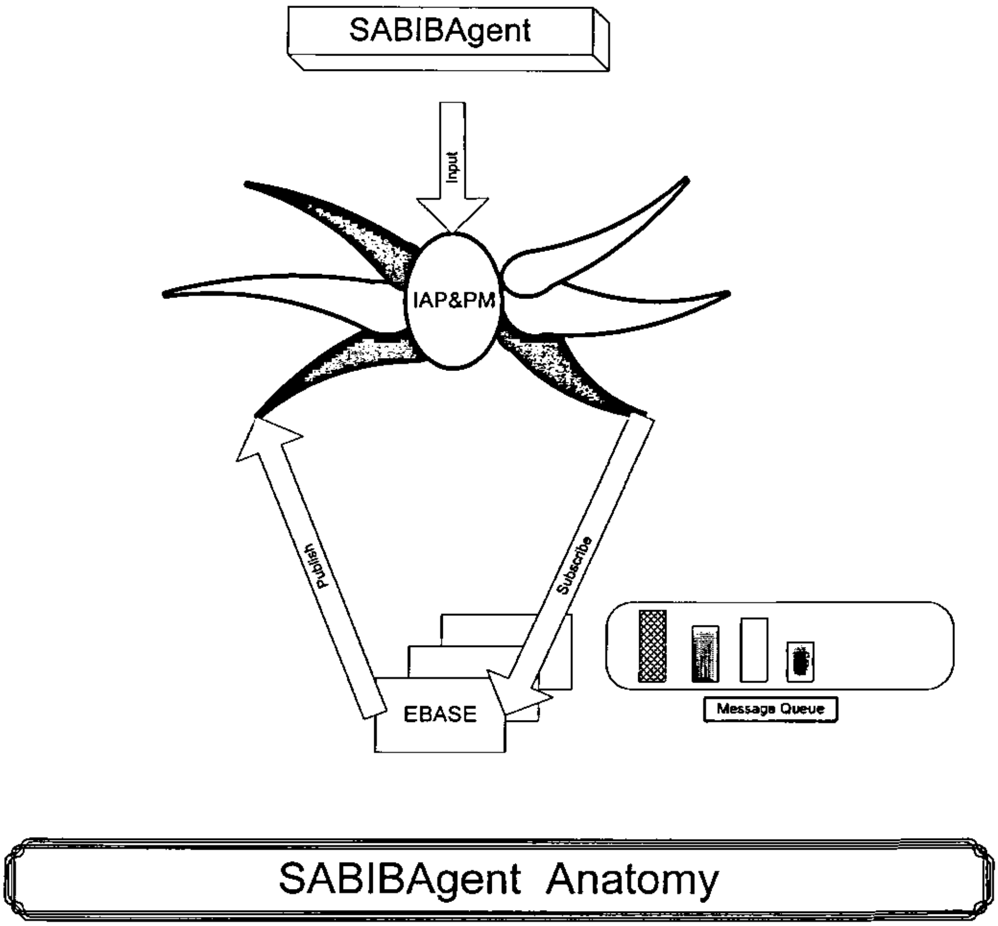
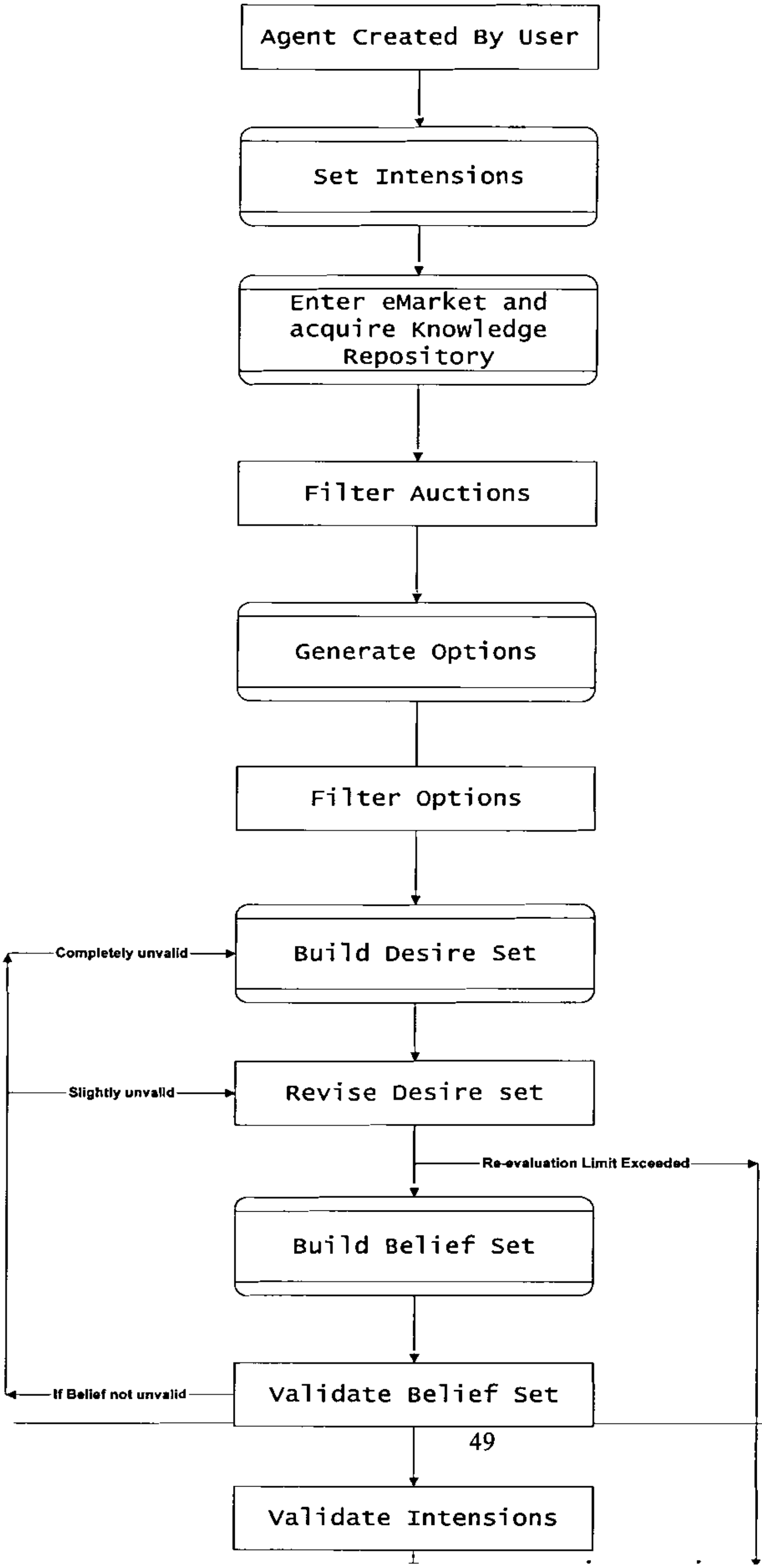


Figure 6.1 SABIBAgent Anatomy

In short SABIBAgent is a large scale work flow engine based on a virtually self tailored BDI (Belief Desire Intention) Architecture called e-COMMBDI Architecture, with static Intentions. The agent communicates with global e-marketplace simulator and other agents by means of a built-in asynchronous Message Passing Scheme (MPS). The agent exploits all potential resources available at hand to work on user's behalf and continuously reworks the solutions as problem, parameters, constraints or execution environment changes.

6.2 Working cycle of SABIBAgent

The agent is created by the user and is initialized by a set of Intentions. The agent is trained on basis of Real-time Market data repository. It then enters the heterogeneous market simulator [5] and requests for general market auctions set repository. It reads the repository for selection on committing to the suitable auctions. Suitable Auctions are the one which present the in demand item according to our Intentions. These suitable auctions are then and an option set is generated and again filters to act on only the best auctions available. These refined options are the foundation stone of Desires built right in the next step. Desire sets again go though a filtering process and are then translated into Beliefs by matchmaking with the Market Data Repository. As a result of this matchmaking only the valid Beliefs are filtered to be part of the mean-ends reasoning process. If no valid Belief or not enough valid Beliefs are filtered though for effective mean-ends reasoning to take place the process is again repeated form rebuilding Desires onward until a valid Belief set is established which is consistent with our Intentions since newer auctions and results are updated in the data clusters of repository. The agent then commences bidding on basis of this Belief set initial most optimal value and if unsuccessful, the next optimal Desire and Belief is established for next bidding to commence.



Process Flow Diagram of SABIBAgent

Degree of Boldness and Cautiousness

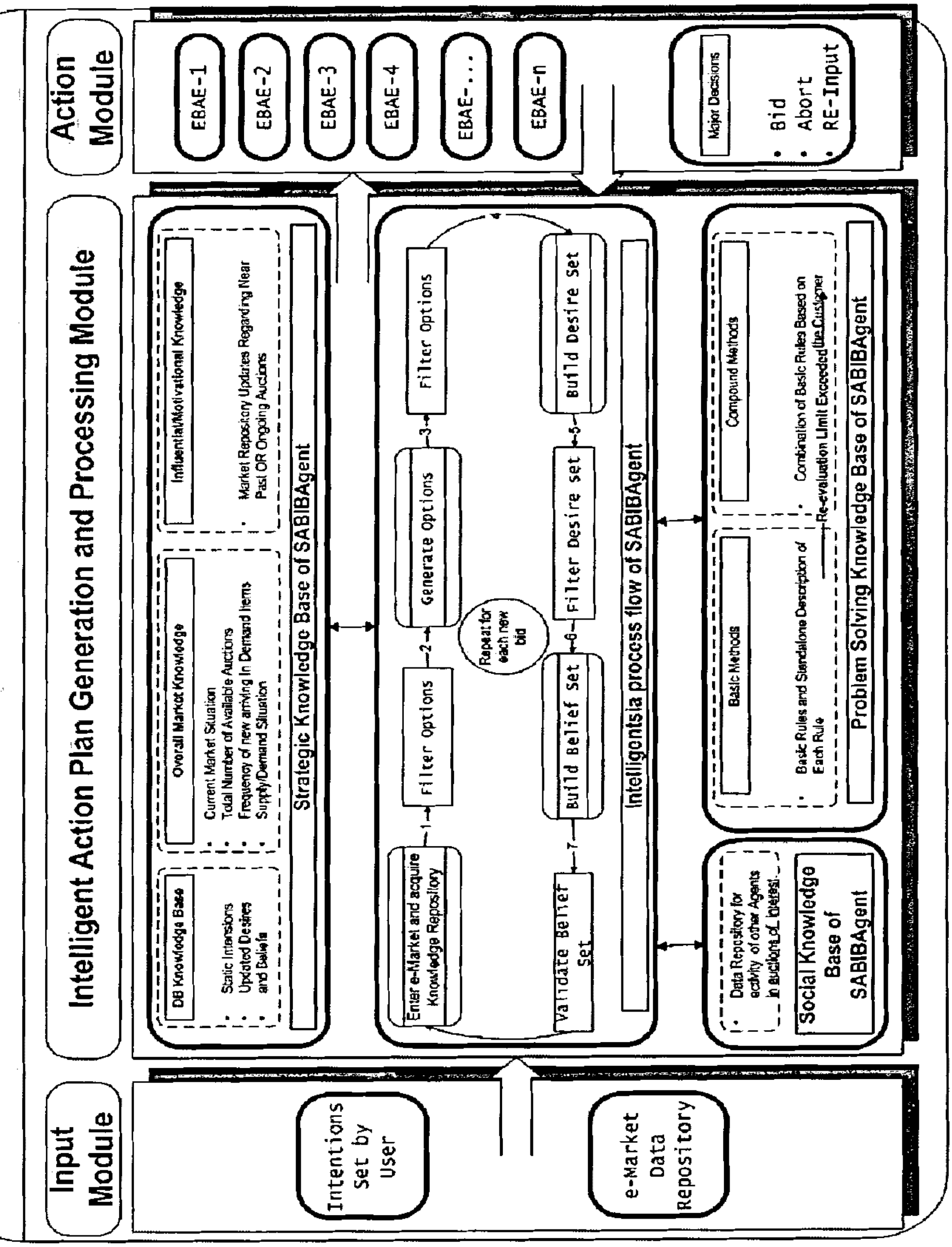
Since the environment in the market is fairly consistent according to our studies on various markets of private value auctions the agent starts off as a cautious agent but on acquiring and reviewing the market repository if it finds the critical market factors highly fluctuating it increased its level of Boldness, but it never goes above 50% because above this value the agent would get stuck in reconsidering its Desires and Beliefs and would do little real effective work towards its goal. The Boldness of the Agent in the tailored framework is limited to changing and re-evaluating its Desires and Beliefs. For more on bold and cautious agents please read [7]

6.3 Concrete Architecture of SABIBAgent a Detailed Look

The SABIBAgent comprises of three main components

I. Input Module
    a. Prioratized Intention Fetching sub module
    b. Data Repository Fetching sub module
II. Intelligent Action Plan and Processing Module
    a. Strategic Knowledge Base
        i. Desire / Belief Knowledge Base
        ii. Overall Market Knowledge
        iii. Motivational Knowledge
    b. Intelligent Processing
        i. Intension Weight Calculation Function
        ii. Data Clustering Function
        iii. Auction Selection Function
        iv. Auction Filtering Function
        v. Option Generation Function
        vi. Option Filtering Function
        vii. Desire set Generation Function
        viii. Desire set Revision Function
        ix. Belief Generation Function
        x. Belief Based means-ends reasoning Function
        xi. Intention Validation Function
    c. Social Knowledge Base
        i. EBAE Intercommunication
        ii. Agent Monitoring Function
    d. Problem Solving Knowledge Base
        i. Simple Methods
        ii. Compound Methods
III. Action Module
    a. Major Decision module
    b. EBAE 's

Detailed Architectural Diagram of SABIBAgent



The input module is responsible for introduction of agent in the market place. It is an engine responsible for populating the agent's data repository with not only the intension set and priority scheme delivered by the owner but also with the relevant item market data. Along with all this the input module also provides the *Intelligent Action Plan and Processing Module* with the set on ongoing auction attributes e.g. (start time/ finish time/ type of Auction/ open /closed/private/ public etc.) and the desired item attributes in these auctions. The input modules can also do filtering of auctions, thus targeting only those auctions with the agent's desired object presented for procurement.

The intelligent Action plan module serves for three crucial tasks

1. It is the communication channel for the Environment Based Action Engines participating in multiple individual auctions for procurement of a single item. All their intra engine communication originates or terminates here.
2. It serves as repository for any information needed by that Agent itself for policy making or Environment Based Action Engine (EBSA) to participate in distributed decision making process.
3. Most important function of the Intelligent Action Plan and Processing Module (IAP&PM) is that it also serves as the intelligentsia or generating unit of the SABIBAgent. It comprises of Basic Methods and Compound Methods as part of problem solving knowledge base, aided in its task by Strategic Knowledge Base and Social Knowledge Base.

The main functional components responsible for the intelligentsia of Agents employing the E-COMMBDI Architecture are

- **Intension Weight Calculation Function**

  This function is responsible for formulation of priority based Intention hierarchy formation which is considered when the intelligent bidding strategy of SABIBAgent is formulated. For example giving a higher weight to the desperateness factor results in a more aggressive bidding strategy. While giving more time to bid evolves into a more mild approach towards procurement, thus increasing the probability for a better and cheaper procurement.

- **Data Clustering Function**

  This function is responsible for keeping the data gathered from both internal (EBAE) and external (current and previous e-Market and other active agent monitoring etc.) sources. This organization obviously makes the job of scheduling tasks much easier because tons of relevant data is now in well organized clusters.

- **Auction Selection Function**
  This function simply selects a set of Auction from the ongoing Auction pool on basis of availability of Desired item or items. Furthermore it keeps track of new auctions and frequently updates the Filtered Auction set with new auctions of interest.

- **Auction Filtering Function**
  This function filters the Auction set which was produced as a result of Auction Selection Function. It screens out the target Auctions where the probability of success OR effective time utilization is maximum.

- **Option Generation Function**
  This Function module is responsible for building a list of options which are potential candidates of being the Desires of the SABIBAgent. This process is carried out by scanning the data repository and making a list of all committed transaction attributes about the item of concern.

- **Desire set Generation Function**
  The list of options is validated against the Intentions and only those options and Intentions which are mutually inclusive are adopted as Desires of SABIBAgent.

- **Desire set Revision Function**
  This Desire set though trimmed down can still be very large and diverse, thus if required, it is further optimized by the Revision Function.
- **Belief Generation Function**
  This function now generates the Beliefs on which the bidding plan commences. This Belief set along with the prioritized Intentions take us gradually to the means-ends-reasoning process associated with the SABIBAgent.

- **Belief Based means-ends reasoning Function**
  This function takes perceptual input and the current set of Beliefs constructs the logic to proceed with the bidding plan.

- **Intention Validation Function**
  The means ends reasoning plan generated and the Intentions are validated against each other to make sure the final goal is realistic and has a good enough probability of being achievable.

All these functions are part of the problem solving knowledge base which combines simple methods to formulate compound methods. The conceptual breakup of the problem solving knowledge base is presented in figure.

schema incorporated in it to act in a scenario according to a specific e-market protocol for that particular unique auction.

EBSE's exchange information with each other through publish /Subscribe transactions via intelligent action plan and processing Module so that they can synchronize their actions according to the global action plan.

IAP&PM creates and awakens the EBAE by subscribing scenario specific information and EBAE then starts perusing its procurement goal in a particular auction synchronizing its actions with other EBAE's. Each EBAE collaborate in one to many relationships or roles with others of its same kind. They are aware of each other via IAP&PM in SABIBAgent system. These roles are defined as DuchEBSE, EnglishEBSE, VickeryEBSE, 1st Price SealedBidEBSE, 2nd price Sealed BidEBSE, etc. The agent Architecture is scalable and other roles can be easily incorporated by managing them in the Problem Solving Knowledge Base.

The inter-EBSE relationship is in a way that EBSE's are aware of other EBASE's and synchronize their actions to work collaboratively for achievigna single universal

distributed goal. Any EBSE at any given instant may be participating in its deployed environment in a predefined role and other invoked EBSE's will be simultaneously operating in their respective environments with single role capacity.

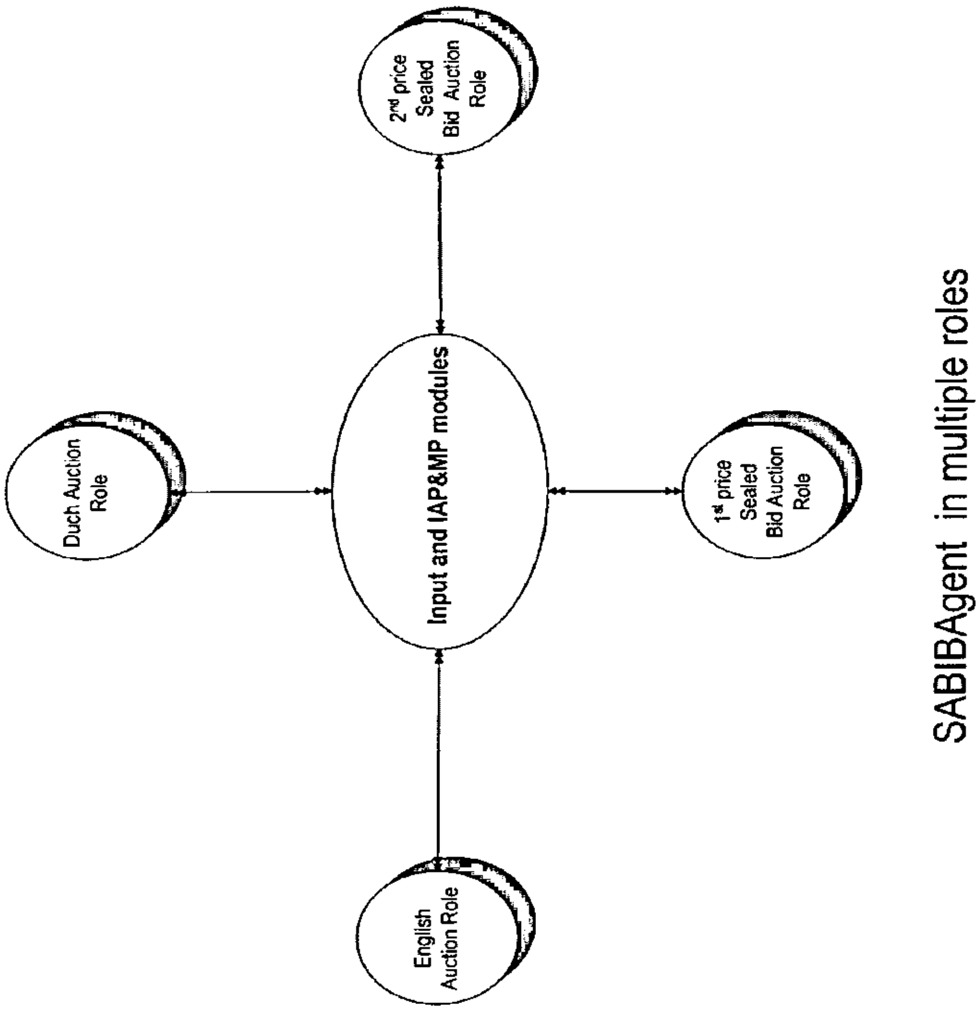


Figure 6.5 SABIBAgent Roles

This scheme conceptually distributes a bidding plan throughout the heterogeneous environment. Each EBASE contains its portion of the overall bidding strategy and collaborates via the IAP&PM to achieve the goal. The action strategy is composed of multiple sub-Strategies which group together in approaching a single objective. These sub-strategies when implemented in a sequence produce a work flow of the agent intelligentsia.

The action strategy provides traceable information on the overall current task progress. The intentions which remain unchanged during the course of action (i.e. static Intentions) are the cornerstone of action strategy progress. They are the answer to the what, the why, the how, the when, the where, the who's of the bidding problem.

The IAP&PM constituents define and control the plan execution throughout the lifecycle of the SABIBAgent. While there may be some minor classes the heart and soul of the IAP&PM are the following modules

- Strategic Knowledge Base

- Social Knowledge Base
- Problem Solving Knowledge Base

As it is obvious from the strategic knowledge base SABIBAgent provides a mechanism for tracking and reporting the observed non-allocated and allocated bid results for Auctions(In progress or completed). This feedback capability is provided by underlying architecture and makes the process of dynamic planning at next run and allocation of Desires/Beliefs possible. This too is how the continuous refinement of a plan can grow from being merely a speculation to actual optimal solution.

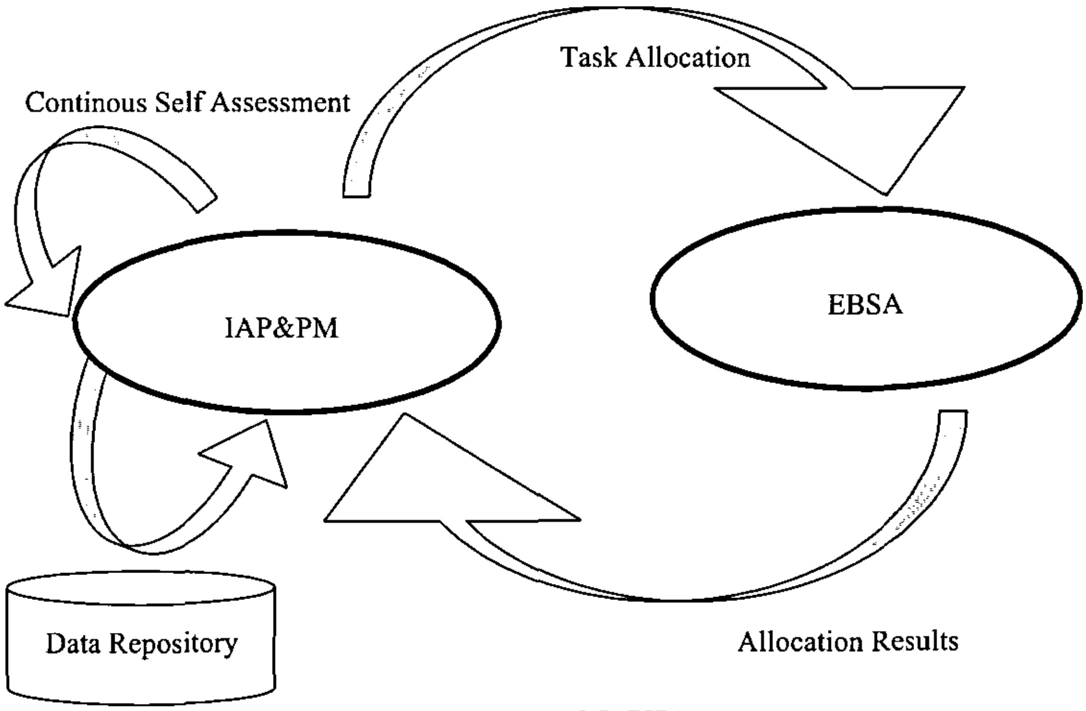


Figure 6.6 Learning Process of SABIBAgent

As the environment variables change the re-planning includes how the Belief/Desire are to be effected accordingly thus generating the new scenario centric optimized solution. SABIBAgent continuously monitors the plan and forces re-planning iff real-time changes in environment variables are drastic, forcing the Agent to take a bolder approach.

6.3.1 Working of Agent in its Environment

The SABIBAgent when implemented and simulated on the custom made global multi auction-protocol e-Market simulator was a smashing accomplishment, and proved to be a natural fit since it was personalized to meet the requirements on demand.

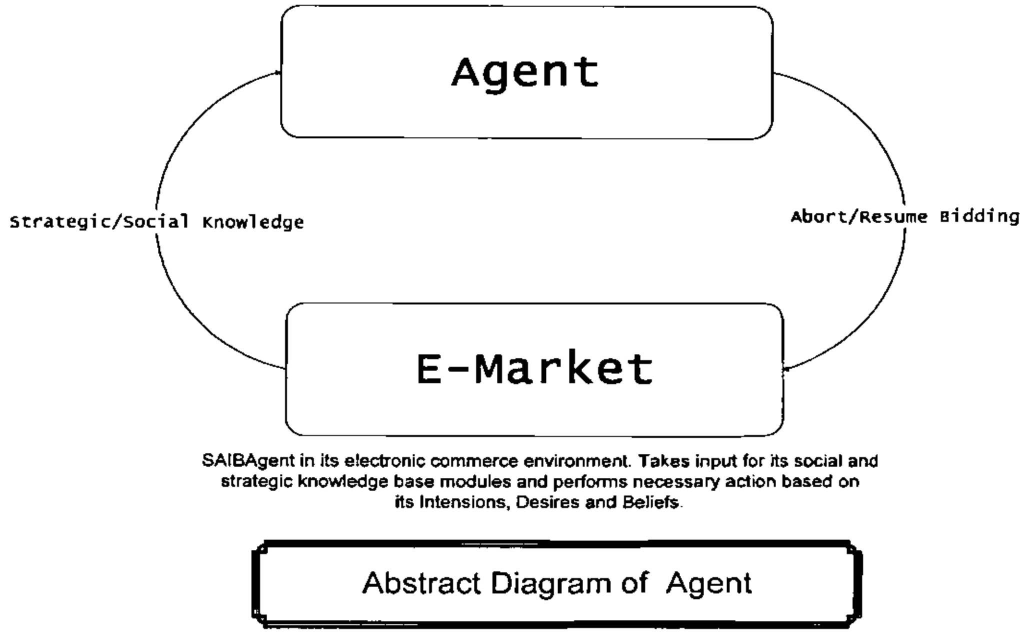
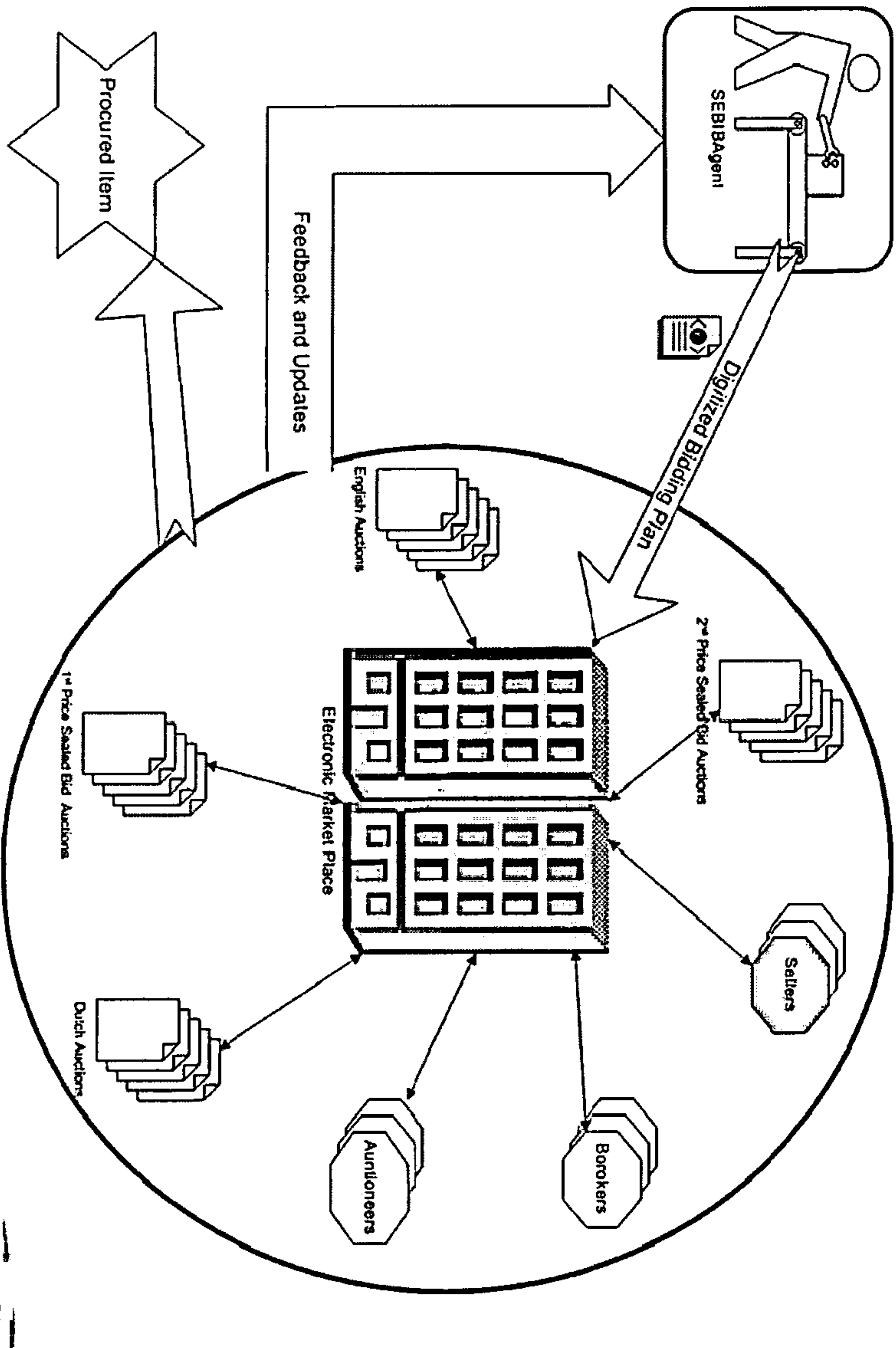


Figure 6.7 Abstract Diagram of SABIBAgent

The rational disjointing of policy form the mechanism proved to be real neat in making implementation adaptations and this effectual design strategy resulted in a Simulated Annealing based Intelligent Bidding Agent for Heterogeneous Marketplace with high scalability, automatic Adaptability to changing scenarios, and a very high degree of modifiability, Upgradeability, Interoperability, Reusability, Distributed Artificial Intelligence Management, Maintainability and Salability features for future enhancement involving multi-agent negotiation and contracting in the near future.

6.3.2 Overview of SABIBAgent with its Whole Working Environment

The system is a entirely automated e-commerce suit, designed while keeping future requirements into consideration. It comprises of a Global Electronic Market place with thousands of auctions running heterogeneous protocols at a given instant. The customers, sellers, brokers etc are represented as autonomous agents coded to behave unselfishly. The market place implementation makes it sure that once an agent commits a bid it cannot back out from its commitment. For user facilitation they have the option to select the payment protocol scheme of his choice. Similar facilities are also available to the seller or Auctioneer Agent. It too makes it certain that the item is presented where it has the maximum chance of fetching the maximum value. The handshaking process for Auctioneer agent and Market place for item placement can be shown with the help of following diagram.



System overview of Simulated SABIBAgent and its Heterogeneous working Environment

As is obvious that not only the buyer but the customer agent is also dependent on time for optimal result generation and an earlier start time and late finish time suits both the parties making it a win-win situation. The SABIBAgent Architectural Framework ensures that it is simultaneously bidding in multiple auctions but can only win one. The owner of the agent (actual customer) never has to motor the activities and is saved form the painstaking work of selection and management of bidding in thousands of Auction houses.

6.3.2.1 Personalized Developmental Items

The agent Architecture is highly moldable with the utmost ease for any new scheme or scenario. The user just has to create an EBASE template and relevant set of simple and compound rules for the making the agent personalized for own use in e-commerce environment, since they are the only engines that can enhance or modify system behaviour. This is precisely why this Architecture is so much attractive as a developmental framework. The developer need not to concern himself with the development of underlying management and means-ends reasoning infrastructure.

6.3.2.2 Human Interaction with the SABIBAgent

The human machine interaction in this model is limited to the Intention gathering and prioritization. This is made fairly simple by the excellent GUI which provides a set of options to choose from thus making the task so simple that even a rookie would be able to perform it without much effort.

6.3.2.3 Benefits

The benefits of our proposed SABIBAgent are significant from the above discussion. It personalizes a whole new Architectural style which is radically different form the traditional BDI Architecture. Any planning application or real world example would match perfectly to the e-COMMBDI, and the underlying Architecture automatically provides features like discovery, task generation, scheduling and allotment etc.

It is ideal for use in situations where a solitary agent has to execute in a multi role capacity synchronizing all its actions. The presented Framework promotes reuse, makes maintenance negligible, up gradation or modification in behaviour just needs to deal with the EBAE engines only. Furthermore it has a exceedingly comprehensive, specialized planning structure as part of its intelligentsia functions.

6.3.2.4 Weaknesses

The logic for EBAE engine is very complex and thus has a precipitous learning arch which further intricate by the fact that the exceedingly quasi fractal environment makes convergence towards optimum very difficult. Furthermore the system if implemented on the World Wide Web will not be suitable for low bandwidth environments. This proposed Intelligent Agent framework is for use in multifaceted situations and may have too much operating cost for simple planning problems, as its powers will not be fully realized, but nevertheless it can be implied if no other means to solve simple problems are available.

6.4 Design of Marketplace Simulator

A *market* is a forum for commerce in a particular commodity or business area. There could be markets devoted to banking, publishing and printing, construction, transportation, industrial equipment, etc. Each market includes a set of domain-specific services and facilities. The primary functions of the market are to act as a matchmaker between customers and suppliers, to define terms of discourse among participating agents, and to collect and publish statistics to support agent decision processes. In addition, each market encapsulates a set of sessions that represent the active transactions within the market.

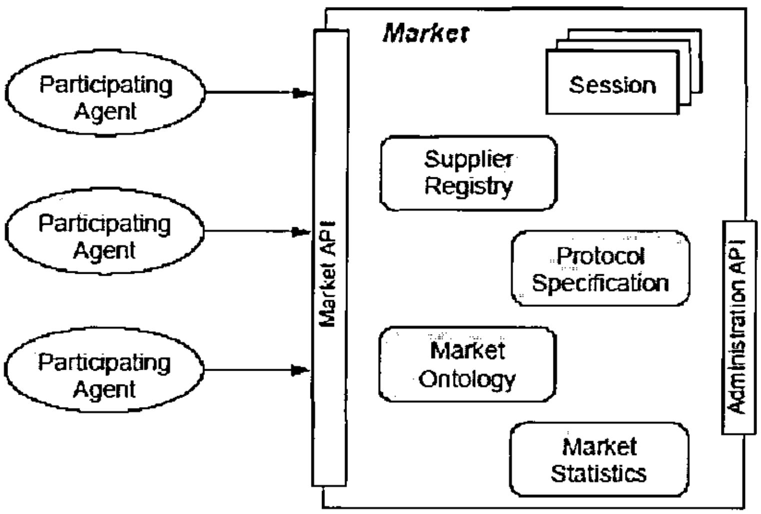


Figure 6.9 Marketplace Simulator Architecture

A Marketplace is a virtual simulation that supports commerce in a particular business area, and encapsulates a set of sessions representing the active transactions in the Market. Our simulated electronic marketplace consists of a number of auctions that run concurrently. There are four types of auctions running in the environment: English, Dutch, First Price and Vickrey. The English, First Price and Vickrey auctions have a finite start time and duration generated randomly from a standard probability distribution, the Dutch auction has a start time but no pre-determined end time.

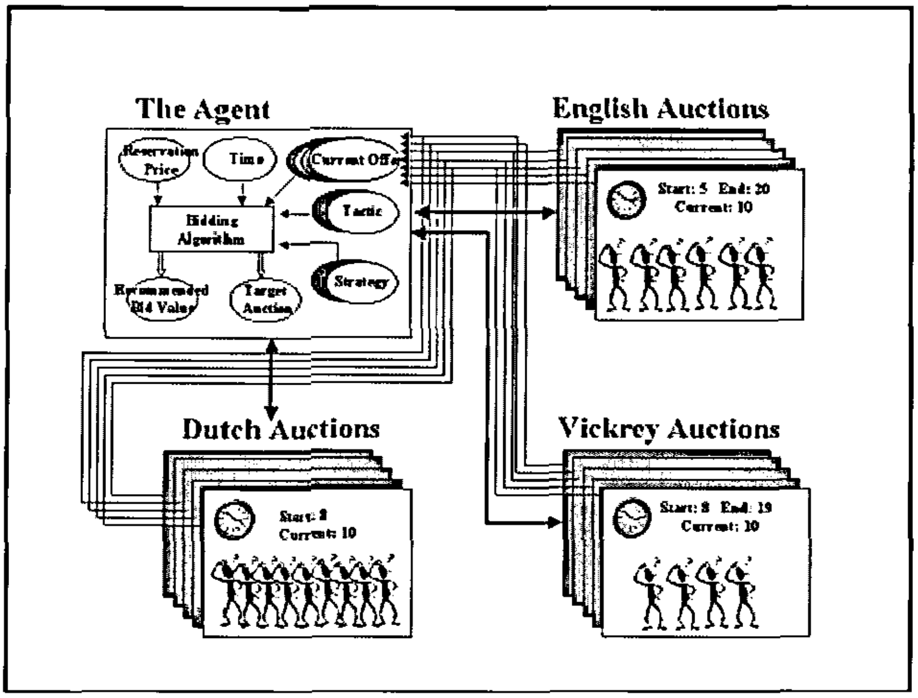


Figure 6.10 Marketplace Architecture

The start time Marketplace Simulator Algorithm and the end time varies from one auction to another. At the start of each auction (irrespective of the type), a group of random bidders are generated to simulate other auction participants. These participants operate in a single auction and have the intention of buying the target item and possessing certain behaviour. They maintain information about the item they wish to purchase, their private valuation of the item (reservation price), the starting bid value and their bid increment. These values are generated randomly from a standard probability distribution. Their bidding behaviour is determined based on the type of auction that they are participating in. In an English auction, they start bidding at starting bid value; when making a counter offer, they add their bid increment to the current offer (provided the total is less than the reservation price), and they stop bidding when they acquire the item or when their reservation price is reached. In a Dutch auction, they wait until the offer value is equal to their reservation price before making an offer. Finally, in case of sealed bid auctions, they bid at their reservation price. These strategies are based on the dominant strategies of the respective one-shot single auctions [Sandholm, 1999].

The auction starts with a predefined starting value; a small value for an English auction and a high value for a Dutch auction. There is obviously no start values for sealed bid auctions. Offers and counter offers are accepted from bidders who are picked randomly from the group of bidders in that particular auction. These processes are repeated until the reservation price is reached or until the end time for that auction is reached. The winner in an English auction is the bidder with the highest bid value at the end of the auction. In a Dutch auction, when no offer is received from the bidders, the value is reduced (based on a fixed decrement value) and the whole process is repeated again. The item is sold when a bidder agrees to buy the item at the offer price. If there is more than one bidder who is interested at the same price, the item will be sold to the bidder who offered to buy the item first. There may be cases where there is no offer from the bidders at all throughout the auction. In this situation, the auction terminates when the decremented offer reaches the reservation price. Bidders in sealed bid auctions submit their bid values before the end of the auction. Bids are opened at the end of the auction and the winner is the one who offered the highest price. If there is a tie, the winner is the bidder who submits the earliest bid. The marketplace is flexible and can be configured to take up any number of auctions and any value of discrete time. We assume that all the auctions in the marketplace are auctioning the item that the consumers are interested in. Our bidder agent is allowed to bid in any of the auctions at any time when the marketplace is active. The objective of the bidder agent is to participate across the multiple auctions, bid in the auctions and deliver the item to its consumer in a manner that is consistent with their preferences. The bidder agent is given a deadline by when it needs to obtain the item. The bidder agent utilizes the available information to make its bidding decision; this includes the consumer's reservation price, the time it has left to acquire the item, the current offer of each individual auction, and its set of tactics and strategies. The reservation price is derived from the item's closing price distribution, observed from past auctions. The tactics and strategies are the main constituents that drive the agent's behaviour in making the bidding decision. The output of the bidding decision is the auction the agent should bid in and the recommended bid value that it should bid in that auction. If the agent does not purchase the item by its deadline, it returns to the consumer for further instructions. Apart from the bidding agent, our proposed system comprises of the following agents

6.4.1 Interface Agent

Interface agent will be responsible for collecting and collating relevant information from the user to initiate a workflow process, presenting the returned results and explanations to the user and optionally requesting the user for additional information.

6.4.2 Scheduling Agent

The scheduling agent is the agent doing the real workflow enactment. It will execute all the tasks and sub-processes according to the control and data-flow given in the definition.

6.4.3 Task Agent

Task agents act as wrappers of the actual applications. A typical task agent knows the meta-model of the task that it is associated with and the procedures for executing the task or accessing the data repository. It also communicates with scheduling agents to report the current situation of the task (e.g., committed, failed, executing, etc.).

6.4.4 Facilitator Agent

It acts as a facilitator for agents in the system. It collects advertisements of the agents in terms of their capabilities and facilitates agents to find each other to satisfy their needs. Facilitator agents should know each other's address and query each other to answer requests of the agents. Therefore in our system, facilitator agents advertise themselves to other facilitator agents.

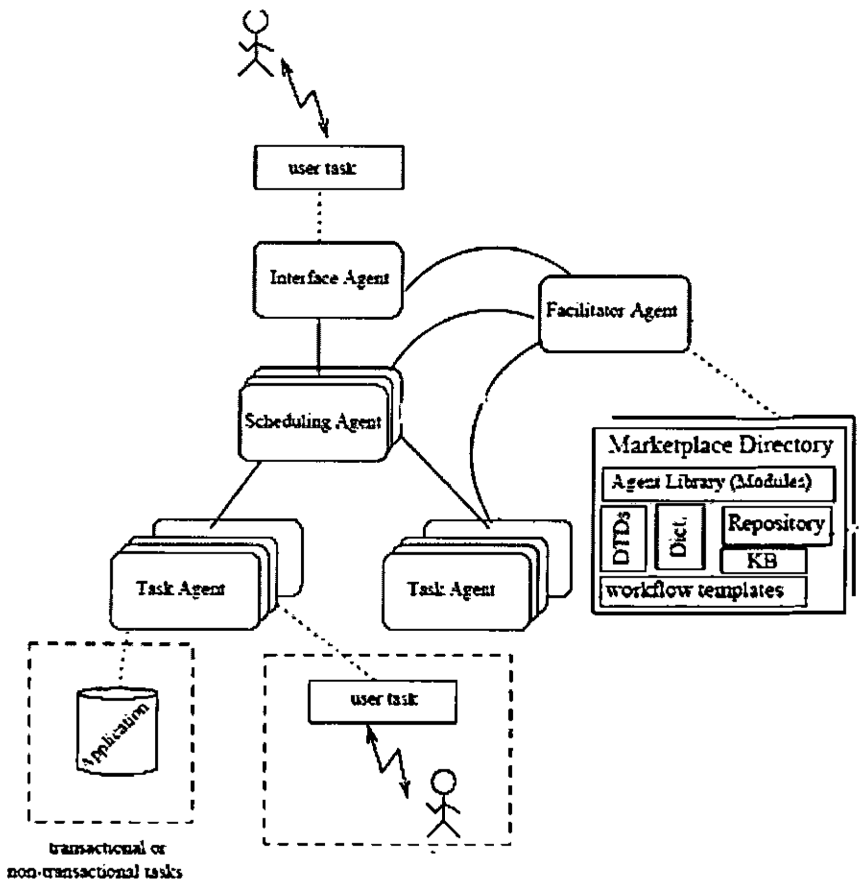


Figure 6.11 Agents Marketplace Interaction[Galant, 2000]

Our e-commerce model extends and builds on the e-commerce structures presented in (Galant, 2000), (Chmiel, 2004a) and (Paprzycki, 2004). Basically, our environment acts as a distributed marketplace that hosts e-sellers and allows e-buyers to visit them and purchase products. Buyers have the option to negotiate with the sellers, to bid for products and to choose the seller from which to make a purchase. Conversely, sellers may be approached" instantly" by multiple buyers and consequently, through auction-type mechanisms, have an option to choose the buyer.

- For the successful approach to this research we need an electronic market place for simulation and result generation in which agents bid in tandem with our developed agents on behalf of the owners and have different bidding strategies. We intend a new smart strategy to outperform the current strategies using

References:

[1] M.Wooldridge and N.R.Jennings. Intelligent agents: Theory and practice. The knowledge Engineering Review, 02:115-152,1995.

[2] www.bidx.com

[3] www.auctionwatch.com

[4] www.auctionbeagle.com

[5] A Framework for e-market By Shahbaz A.K.Ghayyur, Saeed Ullah, Dr.S.Tauseef-UR-Rehman (Paper in process)

[6] M.E. Bratman, D.J. Israel, and M.E. Pollack. Plans and resource-bounded practical reasoning. Computational Intelligence, 4:349-355, 1988.

[7] D.Kinny and M.Georgeff. Commitment and effectiveness of situated agents. In proceedings of Twelfth International joint conference on Artificial Intelligence (IJCIA-91), pages 82-88, Sydney, Australia, 1991.

```
Input :         Join Market and Request Data
Output:               Repository Data

Procedure: START

                Set Connection =  Connection ϕ
/* Criterian → Fetch Acuction Data where the desired items were sold*/

                FOR EACH Auction_i ε {Total Auctions}
                        FOR EACH Item_i ε {Auction_i}
                                Submit_Request(Auction_ID, Members_ID, Winnig_Bid)
                                WHERE
                                (Auction ε {Total Auctions} && time > Auction.End &&
                                Desired_Item)
                END FOR EACH
                END FOR EACH


/* For new and Upcoming Auctions data selling the item of interest*/

                after interval i
                FOR EACH Auction_i ε {Total Auctions}
                WHERE (timestamp_Auction > timestamp_Local_Repository)
                FOR EACH Item ε {Auction_i}
                        Submit_Request(Auction_ID, Members_ID, Winnig_Bid)
                END FOR EACH
                END FOR EACH
```

7.1.3 Auction Selection Function

This function simply selects a set of Auction from the ongoing Auction pool on basis of availability of Desired item or items. Furthermore it keeps track of new auctions and frequently updates the Filtered Auction set with new auctions of interest.

```
Input :         Join Market and Request Data
Output:               Selected Auctions

Procedure: START

                Set Connection =  Connection ϕ
/* Criterian → Fetch Acuction Data where the desired items were sold*/

                FOR EACH (Auction_i ε {Total Auctions} && Item_i ε {Auction_i})
```

Submit_Request(Auction_ID, Members_ID,
Minimum_Bid)

WHERE
(Auction ϵ {Total Auctions} && time < Auction.End &&
Desired_Item)

END FOR EACH

7.1.4 Auction Filtering Function

This function filters the Auction set which was produced as a result of Auction Selection Function. It screens out the target Auctions where the probability of success OR effective time utilization is maximum.

Input: {Auction Set}
Output: {Optimized Auction Set}
Processing: START

FOR EACH Auction$_i$ ϵ {Selected Dutch Auctions || Selected English Auctions}

Submit_Request(Auction_ID, Members_ID, Winnig_Bid)
WHERE
(Auction ϵ { Selected Auctions } && time > Auction.End && Desired_Item) &&
$P_i^w(v) = (\sum_{P>v} P_i^c(P) + P_i^c(v)/2) \geq 50\%$

/* where $P_i^w(v) = \sum_{P>v} P_i^c(P) + P_i^c(v)/2$ is the winning probability of English and Dutch Auctions as indicated in [Reference] */

END FOR EACH

FOR EACH Auction$_i$ ϵ {Selected Vickery Auctions}

Submit_Request(Auction_ID, Members_ID, Winnig_Bid)
WHERE
(Auction ϵ { Selected Auctions } && time > Auction.End && Desired_Item) && $P_i^w(v) = (1/\sum n_i / x) \geq 50\%$

/* where $P_i^w(v) = (1/\sum n_i / x)$ is the winning probability of Vickery Auctions as indicated in [Reference] */

END FOR EACH

7.1.5 Desire Generation Function

The list of options is validated against the Intentions and only those options and Intentions which are mutually inclusive are adopted as Desires of SABIBAgent.

Input: {Optimized Auction List}
Output: {Desire Set}

Procedure: START

Load the Cumulative Weight Structure

IF ($\Sigma_{i=1....n}$ W_j * I_j)- $W_{BD} \leq W_{BD}$

/* if the weight of Bargain tactic W_{BD} is higher or euqual in magnitude to the Desprateness tactic select the Early Start Late Finish (ESLF) and Late Start Late Finish (LSLF)Auctions*/

FOR i = 1...n
FOR EACH $Auction_i \in$ {*ESLFAcutions* ||*LSLFAuctions*}
FOR EACH $Item_i \in \{ESLFAcution_i \,||LSLFAuction_i\}$
Set $Desire_i$ = Price of $Item_i$

/* Get procurement price and place it in the Desire Set/*

END FIR EACH
END FOR EACH
END FOR
END IF

IF ($\Sigma_{i=1....n}$ W_j * I_j)- $W_{DP} < W_{DP}$

/* if the weight of Desprateness to procure tactic W_{DP} is higher or euqual in magnitude to the Desprateness tactic select the Early Start Early Finish (ESEF) and Late Start Early Finish (LSEF)Auctions*/

FOR i = 1...n
FOR EACH $Auction_i \in$ {*ESEFAcutions* ||*LSEFAuctions*}
FOR EACH $Item_i \in \{ESEFAcution_i \,||LSEFAuction_i\}$
Set $Desire_i$ = Price of $Item_i$
/* Get procurement price and place it in the Desire Set/*

END FIR EACH
END FOR EACH

END FOR
END IF

IF $D_{set} = = \phi$
Fetch Market Data( )
END IF

7.1.6 Desire Filtering Function

This Desire set though trimmed down can still be very large and diverse, thus if required, it is further optimized by the Revision Function.

Input: {Desire Set}
Output: {Optimized Desire Set}

Procedure: START

/* Remove the desires with highest and lowest values if Desire Set is too large or the distribution in not even. Also Remove the Desire if its value is above the private value*/

```
FOR EACH Desire_i ∈ {DesireSet}
IF Desire_i.value > Private Value
THEN
        {DesireSet} = DesireSet - Desire_i
END IF
END FOR EACH


IF Desire_i.count > Threshold
THEN
        FOR EACH Desire_i ∈ {DesireSet}
                Find Desire_i where Desire_i − Desire_{i-1} < Difference
                between two consecutive elements in case of even
                distribution
                {DesireSet} = DesireSet - Desire_i
        IF Desire.count ≤ Threshold
                EXIT
        END IF
        END FOR EACH
ENF IF
```

/* Threshhold is the number of bid attempts we can make to procure the desired item within the allocated time frame, and is highly dependent on the variables like the number and type of auctions and their starting and ending times */

7.1.7 Belief Generation Function

This function now generates the Beliefs on which the bidding plan commences. This Belief set along with the prioritized Intentions take us gradually to the means-ends-reasoning process associated with the SABIBAgent.

Input: {Optimized Desire Set}
Output: {Belief Set}

Procedure: START

FOR EACH Auction$_i \in \{EnslishAuctions\} || \{DuchAuctions\}$
FOR EACH $Desire_i \in \{OptimizedDesireSet\}$
IF $P_i^w(Desire_i) = (\sum_{P>v} P_i^c(P) + P_i^c(Desire_i)/2) \leq 50\%$
THEN
{DesireSet} = DesireSet - Desire$_i$
END IF
END FOR EACH
END FOR EACH

FOR EACH Auction$_i \in \{SealedAuctions\}$
FOR EACH $Desire_i \in \{OptimizedDesireSet\}$
IF $P_i^w(Desire_i) = (1/ \sum_{Desire_{i_{1...n}}} n_i / x) \leq 50\%$
THEN
{DesireSet} = DesireSet - Desire$_i$
END IF
END FOR EACH
END FOR EACH
{Belief Set} = {Optimized Desire Set}

7.1.8 Means-ends Reasoning Function

This function takes perceptual input and the current set of Beliefs constructs the logic to proceed with the bidding plan.

Input: {Belief Set, Optimized Auction Set}
Output: {Means-ends Reasoning Plan}

Procedure: START

For English Auctions 1...n in Selected Auction Set DO
CurrentAcution = $EnglishAuction_i$
IF
$Auction_i$ time < $Auction_{i+1}$ time AND
Auctioni current pice < $Auction_{i+1}$ current price
EnglishAuctionListj = Current Auction
J=J+1
END IF

For English Auctions 1...n in Selected Auction Set DO
CurrentAcution = $EnglishAuction_i$
IF
$Auction_i (\sum_{P>v} P_i^c(P) + P_i^c(v)/2) > Auction_{i+1} (\sum_{P>v} P_i^c(P) + P_i^c(v)/2)$
EnglishAuctionListj = Current Auction
J=J+1
END IF

FOR Duch Auctinos 1...n in Selected Auction Set DO
CurrentAuction = $DuchAuction_i$
IF
$Auction_i$ time < $Auction_{i+1}$ time AND
$Auction_i$ current pice < $Auction_{i+1}$ current price
$DuchAuctionList_j$ = Current Auction
J=J+1
END IF

For Duch Auctions 1...n in Selected Auction Set DO
CurrentAcution = $DuchAuction_i$
IF
$Auction_i (\sum_{P>v} P_i^c(P) + P_i^c(v)/2) > Auction_{i+1} (\sum_{P>v} P_i^c(P) + P_i^c(v)/2)$
DutchAuctionListj = Current Auction
J=J+1
END IF
FOR SealedBid Auctions 1...n in Selected Auction Set DO
CurrentAuction = SealedBidAuctioni
IF
$Auction_i$.endingtime < $Auction_{i+1}$.endingtime AND
$Auction_i$ current pice < $Auction_{i+1}$ current price

SealedBidAuction$_J$ = Current Auction
J=J+1
END IF

FOR SealedBid Auctions 1...n in Selected Auction Set DO
CurrentAuction = SealedBidAuctioni
IF
Auction$_i$.$(1/\sum n_i/x)$ >Auction$_{i+1}$.$(1/\sum n_i/x)$

SealedBidAuction$_J$ = Current Auction
J=J+1
END IF

/* Now the Auctions are separated by protocol and are in a precedence hierarchy wr.t time price and probability of winning*/

Start Bidding in Auctions from lowest hierarchy in Non Sealed Bid Auctions

/* The Dutch Auctions have highest precedence followed by English and Sealed Bid Auctions */

Get Private Value of other agents form current and previous Auctions
/* The Private Value of Auctions can be easily obtained form English Auctions by noticing their last bids in an auction. i.e. where they drop off

Leave auction where an agent is present whose private value > own Private Value

IF not in a winning position in lower hierarchy auction AND Bid not pending in a sealed Bid Auction
Allowed to bid in the upper hierarchy English and Dutch Auctions

IF Auction remaining time in Sealed Bid Auctions < 1/99 of total time AND not in a winning position in any of the Auctions AND not bidding in any Sealed Auctions
Allowed to bid in the Selaed Bid Auction of highest priority

/* This ensures that we can scan and bid all possible auctions of interest and still be able to come out with only a single or desired number of items in case of a success*/

7.1.9 Assigning Beliefs to Bids

Input: {Belief Set}
Output: {Bid}

Procedure: START

FOR EACH $Belief_i \in \{BelefSet\}$ with ASSIGN flag = = N

IF
(Belief.ASSIGN.count < Belief.total) && (Belief.slelect.cycle ≤ max.slelectcyle)
THEN
Set Belief.ASSIGN = Y
END IF
END FOR EACH

7.1.10 Intention Validation Function

The means ends reasoning plan generated and the Intentions are validated against each other to make sure the final goal is realistic and has a good enough probability of being achievable.

7.2 Development of the Marketplace Simulator

Our simulated electronic marketplace consists of a number of auctions that run concurrently. There are four types of auctions running in the environment: English, Dutch, First Price and Vickrey. The following algorithm depicts the abstract working scenario of the marketplace simulator with multi-protocol variant-time heterogeneous auction environment.

Number of Auctions in the simulation: W
/ W depends upon the auction complexity */*
Initial Price: P_i
Sequence of Auctions: A_1, A_2, A_3 ,---, A_n

Initial Configuration:
Randomly create initial sellers populations;
/ for purpose of simplicity, we suppose that seller=auctioneer */*
Start Auction A_i *from the possible auction Sessions: English, Dutch, Sealed Bid and send the message to all bidders containing the auction preferences and a time-out;*

While not (Stopping Criterion) do
For each auction, generate a number of internal random bidders from a standard probability distribution;
Select the bidding strategy for the generated auction and initiate auction;
Get the Bids from all the bidders
Coordinate the generated auction until it concludes;
Get payment for N best bids and notify the unsuccessful bids
Generate Social Knowledge Statistics;
If number of current auctions is less than number of total auctions then create new auctions where new auctions=total auction – current auctions;
Set Random closing time for newly generated auctions
End while;

Following is the abstract code representation for the marketplace simulator giving only description in brief of the implemented functions.

```
public class MarketPlace implements Runnable {
String name;
Thread t;

//variable for holding the Size of Auction for choosing auction complexity
int bidSize=0;

//name of the bidders for generating statistics
public ArrayList auctionList;

//array for holding the statistics of all bid winners and bids
public ArrayList bidWinners;

// Is there any active auctions
boolean auctionAvailable=true;


public MarketPlace(BidEditorPanel editor,String threadName, long price, double
bidSize) {
}

public void setBidSize(double bs)
{

}

public void shutDown()
{


}

synchronized public void addWinner(bidWinner bidWinner) {

}

synchronized public boolean addAuction(BidAuctionGroup ba) {

}

synchronized public boolean removeBid(String name) {
```

```
}

synchronized public boolean removeBids() {

}
}


public class BidsGenerator implements Runnable {

Thread t;
String name;
//Composition for Marketplace
MarketPlace m_market;

public BidsGenerator(BidEditorPanel m_editor,MarketPlace market, String threadName,
long price) {

}


//Randomly create initial sellers populations;
public void addStartBids() {

}


public void shutDown()
{

}

//Generate new auctions
public void run() {

}
```

7.3 Working Strategy of SA Agent with e-Marketplace Simulator

As the environment variables change the re-planning includes how the Belief/Desire are to be effected accordingly thus generating the new scenario centric optimized solution. SABIBAgent continuously monitors the plan and forces re-planning if and only if real-time changes in environment variables are drastic, forcing the agent to take a bolder approach.

The service descriptions provided to and by the agent in the MAS (Multi agent simulation) are descripted by the following specification.

<Train for Auction Participation>
[AuctionMembership= = no]
→ FetchIntentions( )
SetCumulativeLoad(Intentions)
FetchMarketData()
WHERE
(Auction ϵ {Total Auctions} && time > Auction.End && Desired_Item)

SelectAuctions( )
WHERE
(Auction ϵ {Total Auctions} && time < Auction.End && Desired_Item)

FilterAuctions( )
WHERE
(Auction ϵ { Selected Auctions } && time > Auction.End && Desired_Item)
$\&\&(P_i^w(v) = (\sum_{P>v} P_i^c(P) + P_i^c(v)/2) \geq 50\% \;||\; (P_i^w(v) = (1/\sum n_i/x) \geq 50\%)$

GenerateDesireSet( )
FOR
$((\Sigma_{i=1....n} W_j * I_j)- W_{DP} < W_{DP})$ &&
$((\Sigma_{i=1....n} W_j * I_j)- W_{BD} \leq W_{BD})$
DesireSetRevision( )
WHERE
($Desire_i$.value > Private Value) && ($Desire_i$.count > Threshold)

GenerateBelief( )
FOR
$((\Sigma_{i=1....n} W_j * I_j)- W_{DP} < W_{DP})$ &&
$((\Sigma_{i=1....n} W_j * I_j)- W_{BD} \leq W_{BD})$

Means-endsReasoning( )
SimulatedAnnealing( )
ValidateIntention( )
IF
Belief !≥ Intention

<Join Auction>
→ time
Set Connection = Connection ϕ
Submit_Request(Auction_ID, Members_ID, Items)
WHERE

Current.time < Auction.StartTime && Desired_Item

<Get Market Membership>
[Join Auction (Auction_ID, Members_ID, Items)]
→ AuctionMembership = = yes && AuctionID = = AID, MembershipID = = MID
IF
Auctioneer_Accept(MID) = = ok

<Menas-ends Reasonig>
[AuctionMembership= = yes]
→ Means-endsReasoning( )
{
CentralizedLerning(EAData)
Decentralized Learning(DAData, VAData)
ReinforcementLearning(SD)
QLearning(RT, RA, BD, EP, LP)
}

<Initiate Bidding>
[AuctionMembership = = yes]
Submit_Bid(Auction,MID,BeliefValue)&& FilterdAuctions.Auctioninfo &&
Auction.Auct.ID = = Auction

<accept Acknowledgement of offer>
[Initiate Bidding (Auction,MID,BeliefValue)]
→ !Bid_ID = = bid_ID
IF
AcutioneerAgent [Bid_Adjust (Self, AuctionID,midbidID)

<Adjust Bid After Failure>
CentralizedLerning(EAData)
Decentralized Learning(DAData, VAData)
ReinforcementLearning(SD)
Q-Learning(RT, RA, BD, EP, LP)
[Submit_Bid(Auction,MID,Belief.Valu-e)]
Belief + = Belief
→ Submit_Bid(Auction,MID,Belief.Value)
IF
Bid_Failed (Self, AuctionID, mid, bidID)
WHERE
AID = = Acution &&Bid_ID = = bidID && MID = =mid

<Pay acknowledged offer price>
[Submit_Bid(Auction,MID,Belief.Valu-e)]
→ Pay (Bid_ID, Payment)

```
        IF
        Bid_Accepted(mid, AID, bidID)= =yes        WHERE
        AID = = Auction && Bid_ID = = bidID        && MID= = mid
<Quit From Auction>
    → Quit_Auction (AcutionID)
        Membership = = no
        IF
        Bid_Failed (Self, AuctionID, mid, bidID)
        WHERE
        AID = = Acution &&Bid_ID = = bidID && MID = =mid
        IF
        Bid_Success (Self, AuctionID, mid, bidID)
        WHERE
        AID = = Acution &&Bid_ID = = bidID && MID = =mid
```

The degree of boldness or cautiousness is heavily dependent on the active environment of the marketplace simulator as it is in any e-commerce auction house like yahoo auction, eBay, Amazon, Priceline, UBid and many others. The implemented system starts off as a cautious agent and evolves towards a bolder approach. If the critical market factors like supply-demand, competent agent density etc. are highly fluctuative. The boldness factory has a threshold of 50% since above this value the agent gets stuck in reevaluating its strategical plan and does little effective work as was the case shown by experimental data sets.

8. Testing

8.1 Objective

Software testing is the process of devising a set of inputs to a given piece of software that will cause the software to exercise some portion of its code. The developer of the software can then check that the results produced by the software are in accord with his or her expectations. The following test cases were designed taking into consideration the nature of project at hand. This document describes the testing procedures which have been adopted to test the functionality of the system.

8.2 Testing Strategy

Testing is the process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item. The purpose of the Testing Strategy is to define the overall context for the entire testing process. We have chosen Boundary value analysis as the testing strategy to check that the outputs of the system, given certain inputs, conform to the functional specification of the software.

8.3 Scope

This section contains a description of the testing to be performed by the project team to confirm the proper functioning of the software components of the system. It describes the scope and basis for software testing, the initial review of documentation to support software testing, and the review of the system source code . Further testing of the system software is addressed in the following sections:

1. Section 1, for specific tests of the Marketplace Simulator and
2. Section 2, for testing the intelligentsia of the bidding agent as proposed in this research project.

8.4 Features to be tested

The test strategy consists of a test designed to check that the system is working properly and a large set of unit tests. Its primary goal is to verify the design and the implementation of the system. The software features that have been tested include:

8.4.1 Module Testing: To perform the verification of the smallest unit of software design, the module. Using the detailed design specification as a guide, important control paths are tested to uncover errors within the boundary of the module.

8.4.2 System Testing: To ensure that the system as a whole satisfies input/output specifications and that interfaces between modules/programs/subsystems are correct. Emphasis was placed on usability, performance, and Correctness capabilities.

8.5 Pass/Fail Criteria

It describes the exit criteria that a component or system must satisfy in order to be accepted by a user, customer, or other authorized entity. The Pass/Fail Criteria is dependant on the type of test case. Each test case depending on its state can be pass, fail or incomplete.

8.5.1 Module Testing

This Section describes the pass/fail criteria for different modules tested in unit testing phase.

Pass criteria

- All the specified test cases have been run. If time runs out the test cases with highest priority must have been run.
- No open Critical defects

Fail criteria

- Not all test cases have been executed.
- Open critical defects

8.5.2 Acceptance testing

This Section describes the pass/fail criteria for the system tested in acceptance testing phase with the user.

Pass criteria

- All the specified test cases have been run. If time runs out the test cases with highest priority must have been run.
- No open Critical defects
- The most important functionalities work properly

Fail criteria

- Not all test cases have been executed
- Open critical defects

8.6 Marketplace Simulator Testing

8.6.1 Test Case 1: Auction Creation

Purpose:	Test that auctions are created according to the configuration set by the user
Pre-req:	Marketplace simulator is active. All other configuration information is valid.
Test Data:	User preferences Auction protocols Randomly generated items and sellers
Steps:	1. visit input dialog 2. enter required information 3. click to inaugurate a new session

	4. verify system with specification
Expected Results	Start Successfully by creating Auctions according to set user preferences.
Actual Results	Detailed Boundary value analysis was performed and marketplace simulator initiates properly with the desired configuration and main configuration dialogues popped up after the splash screen. Only one session was created per ID i.e. auction complexity as set by user, limit on auctions was observed properly, user preferences for bidding plan generation were stored in order, task type matching was ok and boundary of expected price violation resulted in warning.

8.6.2 Test Case 2: English Auction Simulation

Purpose:	Test that English auctions are simulated according to pre-defined specification and constraints.
Pre-req:	Marketplace simulator is active. Auctions are created according to test case 1 All other configuration information is valid.
Test Data:	User preferences Auction protocols An auctioneer with desired items is active in an English Auction Randomly generated items and sellers
Steps:	1. visit input dialog 2. enter required information 3. click to inaugurate a new session 4. verify system with specification 5. If English auction is initiated with acceptable price and is being incremented with hard coded English protocol logic 6. Whether an English auction concluded with desired attributes 7. After deadline the market repository attributes are respectively updated w.r.t. auction
Expected Results	English Auctions are created and they exhibit behavior in accordance with English Auction Protocol. i.e. progressive open bidding.
Actual Results	As the correct initiation of system was verified by test case one the Client.log file which keeps the session information was parsed and checked for English auctions after a session expired. It was found that English

	Auctions were created according to their creation density function and neither exceeded the minimum value not the maximum value permitted by the function for a single auction size specified i.e. small, medium, large w.r.t auction size and complexity specified. It was also confirmed that not a single auction violated the English protocol for business conductance.

8.6.3 Test Case 3: Dutch Auction Simulation

Purpose:	Test that Dutch auctions are simulated according to pre-defined specification and constraints
Pre-req:	Marketplace simulator is active. Auctions are created according to test case 1 All other configuration information is valid.
Test Data:	User preferences Auction protocols An auctioneer with desired items is active in an Dutch Auction Randomly generated items and sellers
Steps:	1. visit input dialog 2. enter required information 3. click to inaugurate a new session 4. verify system with specification 5. If Dutch auction is initiated with acceptable price and is being decremented with hard coded Dutch protocol logic 6. Whether an Dutch auction concluded with desired attributes 7. After deadline the market repository attributes are respectively updated w.r.t. auction
Expected Results	English Auctions are created and they exhibit behavior in accordance with Dutch Auction Protocol. i.e open decrementing offer.
Actual Results	As the correct initiation of system was verified by test case one the Client.log file which keeps the session information was parsed and checked *for English auctions after a session expired.* It was found that English Auctions were created according to their creation density function and neither exceeded the minimum value not the maximum value permitted by the function for a single auction size specified i.e. small, medium, large w.r.t auction size and complexity specified. It was also confirmed that not a single auction violated the Dutch protocol for business conductance.

8.6.4 Test Case 4: Sealed-Bid Auction Simulation

Purpose:	Test that Sealed Bid Auctions are simulated according to pre-defined specification and constraints
Pre-req:	Marketplace simulator is active. Auctions are created according to test case 1 All other configuration information is valid.
Test Data:	User preferences Auction protocols An auctioneer with desired items is active in an Sealed Bid Auction Randomly generated items and sellers
Expected Results	English Auctions are created and they exhibit behavior in accordance with Sealed-bid Auction Protocol. i.e sealed bidding.
Actual Results	1. visit input dialog 2. enter required information 3. click to inaugurate a new session 4. verify system with specification 5. If Sealed Bid Auction is initiated with acceptable price and is being conducted as advertised 6. Whether an Sealed Bid Auction concluded with desired attributes 7. After deadline the market repository attributes are respectively updated w.r.t. auction
Actual Results	As the correct initiation of system was verified by test case one the Client.log file which keeps the session information was parsed and checked for English auctions after a session expired. It was found that English Auctions were created according to their creation density function and neither exceeded the minimum value not the maximum value permitted by the function for a single auction size specified i.e. small, medium, large w.r.t auction size and complexity specified. It was also confirmed that not a single auction violated the Sealed-Bid protocol for business conductance.

8.6.5 Test Case 5: Market Negotiation Protocol

Purpose:	Test the correctness of agent/market communication
Pre-req:	Marketplace simulator is active. Auctions are created according to test case 1 All other configuration information is valid. SA Agent is initialized
Test Data:	User preferences Auction protocols

	Randomly generated items and sellers SA Agent preferences
Steps:	1. visit input dialog 2. enter required information 3. click to inaugurate a new session 4. verify system with specification 5. verify agent properties and interaction mechanism
Expected Results	Proper publish subscribe relationships are established and log for communication and negotiation kept.
Actual Results	The log for keeping message queues was checked and some errors were observed in keeping error or message losses(since only successful acknowledgements were stored) which was then fixed, and the test was re run which confirmed that the problem had been fixed and all the messages of concern generated by either parties were being properly stored.

8.7 Testing the Bidding Agent

8.7.1 Test Case 6: Strategy for Multi-Protocol Bidding

Purpose:	Test for agent intelligent action plan for individual auction protocol
Pre-req:	Marketplace simulator is active. Auctions are created according to test case 1 All other configuration information is valid. SA Agent is initialized
Test Data:	User preferences Auction protocols Randomly generated items and sellers All auction protocols are up and running SA Agent preferences
Steps:	1. visit input dialog 2. enter required information 3. click to inaugurate a new session 4. verify system with specification 5. verify agent properties and interaction mechanism 6. verify intelligent bidding strategy for individual auction protocol
Expected Results	Brings out the best global deal available for individual auction protocol for realistic values in.
Actual	The marketplace simulator initiated properly with the desired configuration

Results	and only one type of auction was left activated be un checking all the other auction types from auction activation dialog box. Boundary value analysis was performed by setting optimal user intentions. The bidding process was initiated and after successful procurement it was confirmed by checking the log file Client.log that the solution achieved was nearly always optimal, i.e. the winning price was always lowest for a particular item type in a given session. Whenever out of bound or unrealistic values were given a warning was generated after parsing the market historical repository and if pressed further resulted in unsuccessful procurement. During this process an anomaly was notified that the agent was scanning all types of auctions in the history file thus wasting resources. The anomaly was removed by placing extra checks in coding on auction type parsing in the auction filtering function. The system was re run and the log file confirmed that only the selected auction types or type were processed during parsing.

8.7.2 Test Case 7: Strategy for Multi-Protocol Bidding

Purpose:	Test for agent intelligent action plan for multi-protocol auctions
Pre-req:	Marketplace simulator is active. Auctions are created according to test case 1 All other configuration information is valid. SA Agent is initialized
Test Data:	User preferences Auction protocols Randomly generated items and sellers All auction protocols are up and running SA Agent preferences
Steps:	1. visit input dialog 2. enter required information 3. click to inaugurate a new session 4. verify system with specification 5. verify agent properties and interaction mechanism 6. verify intelligent bidding strategy for multi-auction protocols
Expected Results	Brings out the best global deal available for multi protocol auctions for realistic values in.
Actual Results	The marketplace simulator initiated properly with the desired configuration and all types of auction were left activated be checking all the auction types from auction activation dialog box. Boundary value analysis was performed by setting optimal user intentions. The bidding process was initiated and

	after successful procurement it was confirmed by checking the log file Client.log that the solution achieved was nearly always optimal, i.e. the winning price was always lowest for a particular item type in a given session. Furthermore it was also confirmed that the agent was winning random auctions and there was no set pattern which could indicated the dominance of a particular strategy on agent behavior thus enabling it to be more successful and better results for previous sessions. Whenever out of bound or unrealistic values were given a warning was generated after parsing the market historical repository and if pressed further resulted in unsuccessful procurement.

8.7.3 Test Case 8: Strategy for Successful Procurement

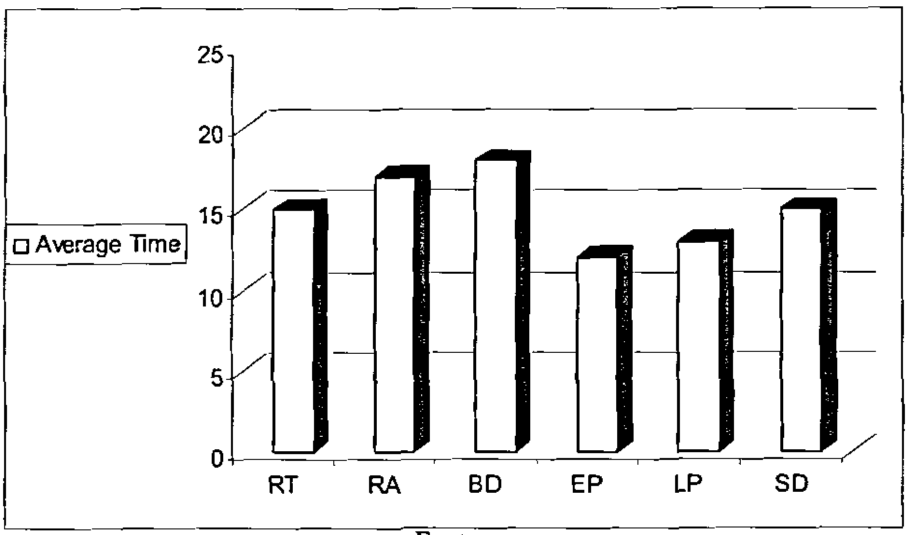
Purpose:	Test for agent intelligent action plan for procurement in multi-protocol auctions
Pre-req:	Marketplace simulator is active. Auctions are created according to test case 1 All other configuration information is valid. SA Agent is initialized
Test Data:	User preferences Auction protocols Randomly generated items and sellers All auction protocols are up and running SA Agent preferences
Steps:	1. visit input dialog 2. enter required information 3. click to inaugurate a new session 4. verify system with specification 5. verify agent properties and interaction mechanism 6. verify item procurement/failure via specified mechanism
Expected Results	Final bidding plan generation and monitoring is fully automated till the procurement process is completed. Successful optimal procurement in Quasi -Fractal Environment (Peaks dispersed though out the landscape). Capability to evolve its intelligence by continuous learning of not only environment but also social behavior patterns of other agents. (0→ 3 level). Must always be able to achieve near optimal results. Very high successful procurement rates. Deal made must be globally optimal. Agent Architecture should be Probabilistic Stochastic Rule Based Reactive
Actual Results	The log file after successful agent procurement session confirmed that it observed the constraints in intelligent action plan i.e it was present in at most twelve auctions at a given instant and procured only a single item by timely

	coordinating its bidding mechanisms. It was also monitoring other agent behavior patterns after nearly half of its total time spent in the market thus confirming its status of intelligentsia improvement form level zero when it started to level 3. It was also managing its eagerness to procure factor nicely since its procurement rate was nearly 100% for realistic input data and price was also optimal. Also it was checking every new auction and adding updating its potential bidding list by validating bidding constraints. The diversity in winning behavior after a number of sessions confirmed that the deal was global and the strategy generation was totally dynamic and situation specific as was our research objective.

9. Experimental Evaluation

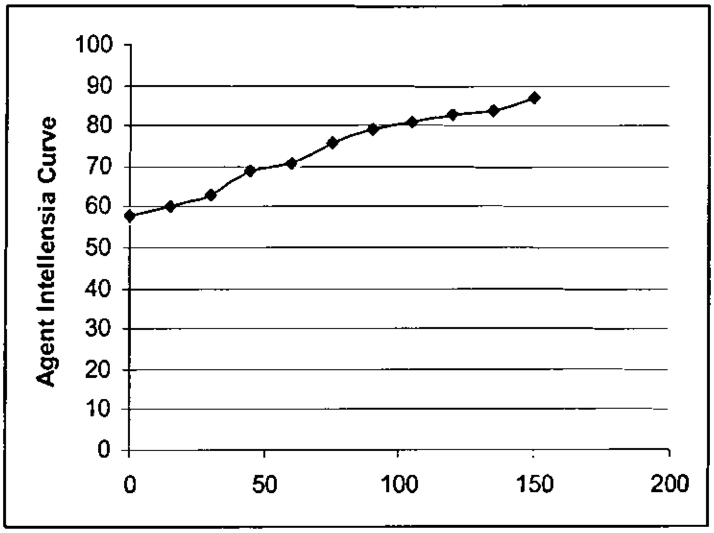
We conducted numerous sets of experiments to verify our system performance and preformed comparison of the generated results with a variety of Simulated Annealing parameter settings. In these experiments, we observed a significant margin of improvement after incorporating SA in matchmaking of the user expectations to a number of scenarios. It is observed that the improvement linearly increases with the number of system runs. The simulated annealing based procedure results in a rule based best case option weight set for implementing the bidding action plan, providing us with a rule based strategy for gaining the best deal within the specified constraints.
Our experiments consist of 150 runs of the scenario based strategies in the marketplace simulator. The parallel/simultaneous multi-protocol auctions with maximum duration of 22 hours, with 5-12 randomly generated agents in an auction at a given instant. The most popular auction protocols are selected which include English, Dutch, First Price Sealed Bid and Second Price Sealed Bid. Our agent participates in this multi auction environment on the factors of Remaining Time (RT), Remaining Auctions (RA), Desire to Bargain (DB), Eagerness to Procure (EP), Limit Price (LP) and Market Supply-Demand Situation (SD). In order to test the robustness of the above model, we considered the following dimensions:

- Agent Success Rate[1]
- Average Time Taken for each strategy[2]
- Agent Strategy Optimization[3]



Factors

Figure 9.1 Average Time



Number of Auctions →

Figure 9.2 Agent Intelligentsia curve

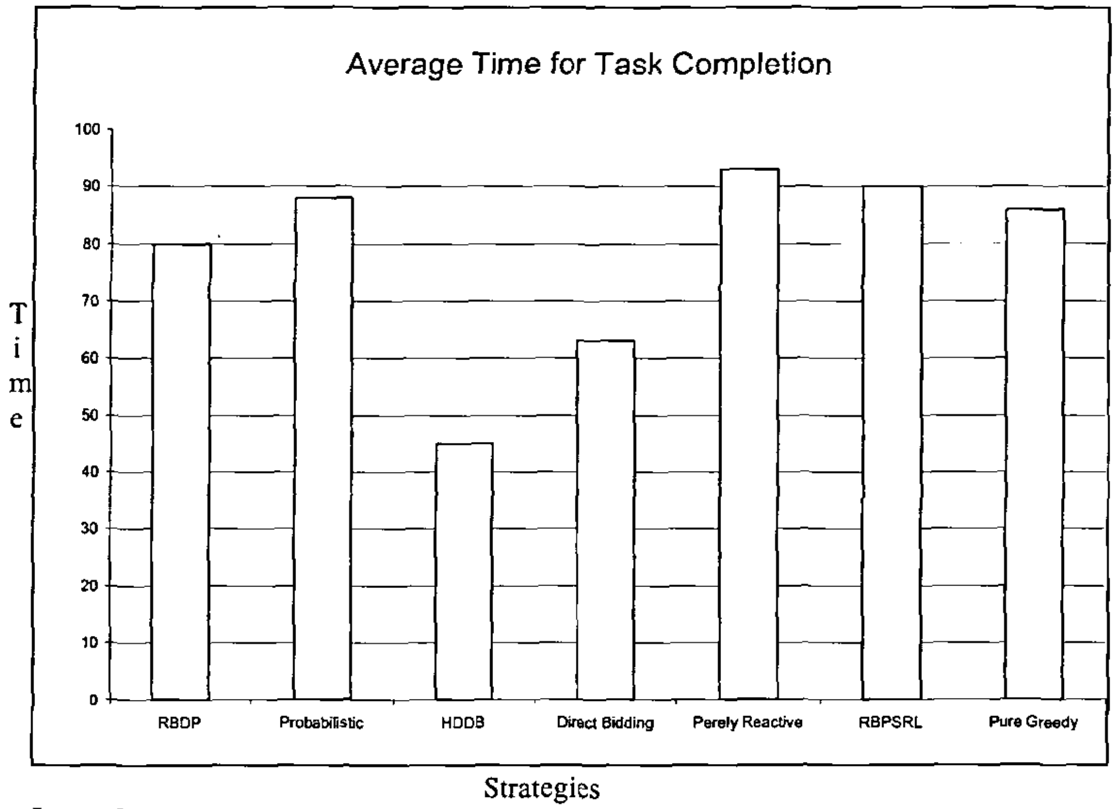
For measurement of effectiveness and strategically standing of the proposed bidding agent a trading game was conducted via implementing the current popular agent bidding strategies namely

- Rule Based Agent using dynamic Programming
- Probability based bidding Agents
- Agents bidding on basis of Historic Data
- Direct Bidding Agents
- Purely Reactive Agents
- Probabilistic Stochastic Rule Based Reactive Agents enforcing Reinforcement Learning

The above mentioned strategies on basis of the following measurement factors

- Average time taken for task completion
- Total number of Auctions scanned
- Average winning prices
- Total number of Auctions lost
- Simultaneous bidding in number of Auctions

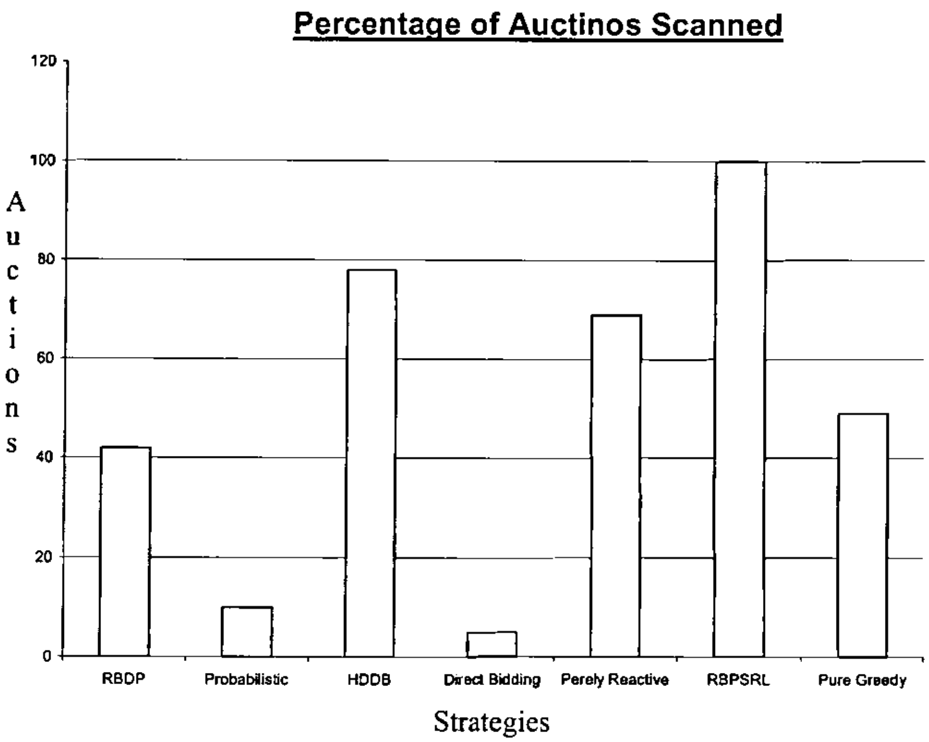
The series of runs were conducted for procurement of same item in same allocated time and the compiled results can be viewed as follows



Legend

RBDP	Rule Based Dynamic Programming
Probabilistic	Probabilistic Bidding
HDDB	Distoric Data based Direct Bidding
Purely Reactive	Purely Reactive Strategy
RBPSRL	Rule Based Probalistic Stocastic Reactive Strategy

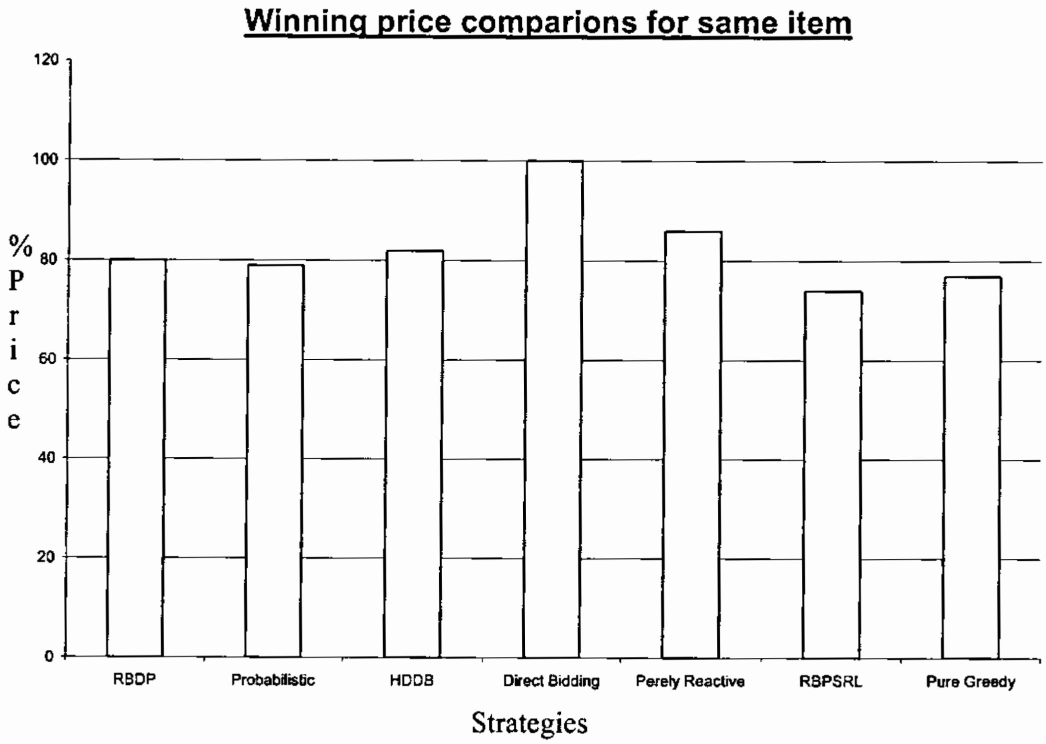
Figure 9.3 Average Time for task completion



Legend

RBDP	Rule Based Dynamic Programming
Probabilistic	Probabilistic Bidding
HDDB	Distoric Data based Direct Bidding
Purely Reactive	Purely Reactive Strategy
RBPSRL	Rule Based Probalistic Stocastic Reactive Strategy

Figure 9.4 Percentage of Auctions Scanned

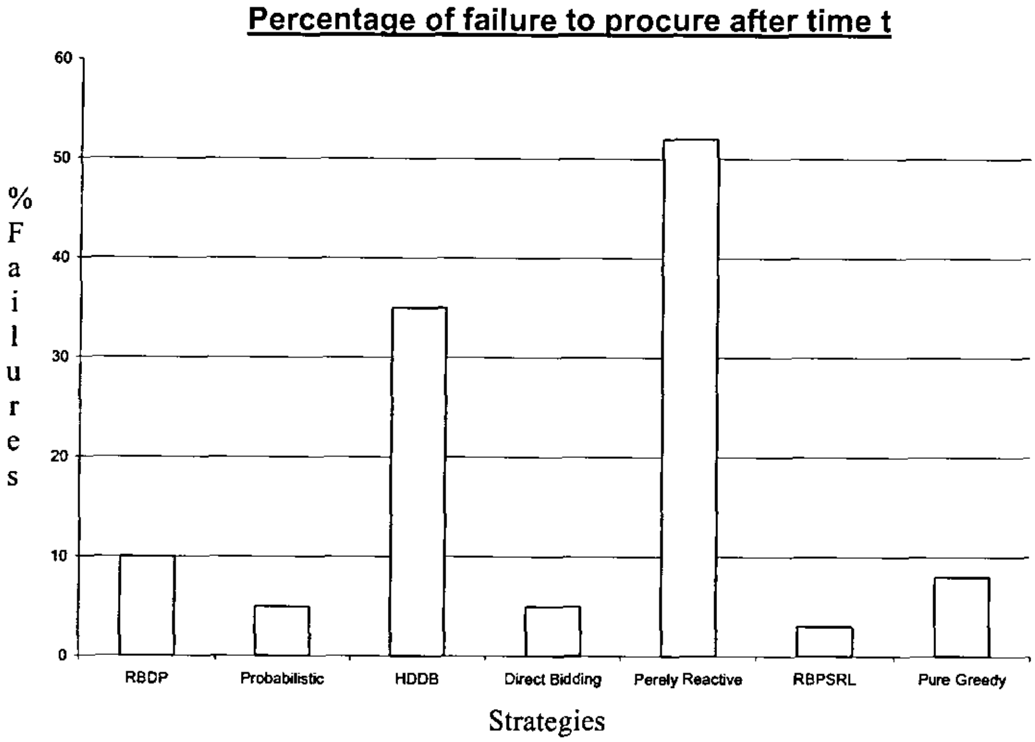


Legend

RBDP	Rule Based Dynamic Programming
Probabilistic	Probabilistic Bidding
HDDB	Distoric Data based Direct Bidding
Purely Reactive	Purely Reactive Strategy
RBPSRL	Rule Based Probalistic Stocastic Reactive Strategy

Note ➔ Price is displayed in form of percentage of limit price which is same for all the agents implemented

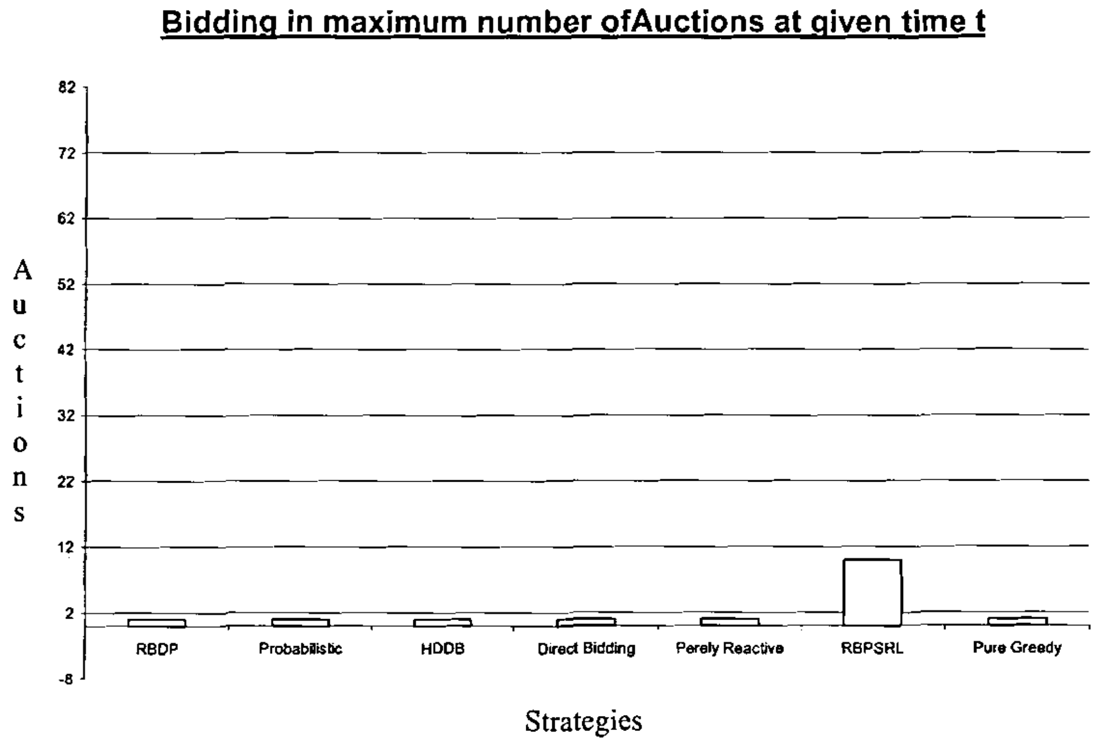
Figure 9.5 Wining price comparisons of same times



Legned

RBDP	Rule Based Dynamic Programming
Probabilistic	Probabilistic Bidding
HDDB	Distoric Data based Direct Bidding
Purely Reactive	Purely Reactive Strategy
RBPSRL	Rule Based Probalistic Stocastic Reactive Strategy

Figure 9.6 Percentage of failure to procure after time t



Legend

RBDP	Rule Based Dynamic Programming
Probabilistic	Probabilistic Bidding
HDDB	Distoric Data based Direct Bidding
Purely Reactive	Purely Reactive Strategy
RBPSRL	Rule Based Probalistic Stocastic Reactive Strategy

Figure 9.7 Simultaneous bidding in number of Auctions

As it can be seen form the above implementation that the proposed strategy lacks in terms of time consumption, and average winning departments but the design strategy is rewarded in terms of fetching best(lowest) price for a particular item in a set of auctions, thus achieving the implementation goal. This is made possible because the agent scans all possible auctions and exhausts every possibility of gaining an advantage in terms of lower bargain.

10. Conclusion

In this thesis we presented the Architectural Framework of a bidding agent, competent enough to toil in a heterogeneous, multi-protocol, multi auction environment, with user driven focus. The proposed agent framework enables the service demanding party to bid in parallel, simultaneous or combinational auctions employing the means-ends reasoning based action plan, to buy a single or multiple items. The agent is capable of competing and outsmarting potential competitors while working integratedly with other agent systems in a marketplace simulator. The agent framework employs Simulated Annealing as a method of inventing rules for optimal performance and then employees the machine learning techniques of Reinforcement Learning and Q- Learning after starting its interaction with other agents and environment. By employing these techniques the agent grows form a level zero agent to a level two agent and implies stimulus based intelligentsia to outperform other such systems. The composition and operational aspects of the designed agent is formulated by orthodox Architectural Description Language proposed for multi agent systems. The architectural description language in this paper provides a modular account of the semantics of the behavior proposed for optimal service. This also enables unambiguous stipulation on the agent's conduct. Hence, the designated environment can be clearly acknowledged for developers on both patron and market stake holder's sides. The same pattern can also be used by developers of service requesters so that the application can be effortlessly integrated without too much stipulate of methodological support.

Future Work

This thesis was not projected to converse the ceremonial characterization of intelligent agents or e-commerce/ m-commerce / i-commerce systems. Neither does it examine or gives synopsis about a variety of approaches for implementing agent oriented systems for electronic environments, and how these approaches counterpart up to various programming paradigms like Object Oriented Programming and Aspect Oriented Programming or Process Oriented Programming etc. not does it present formal definitions and equations regarding any of the machine learning techniques since they are not of primary concern and any one who wants to be more familiar with the technologies and techniques can consult the referenced articles for any further insight into the classical literature.

Since we are in the early stages of implanting the learning behavior of a bidding agent via Machine Learning, there is a need to investigate Machine Learning Reinforcement Learning and Q-Learning techniques further for gaining better results. The issue of look-ahead in auctions, enforcing trust, preventing selfish behavior, avoiding dishonest auctioneers via certificate generation and avoidance of repudiation are concerns of future investigation. We intend to test our proposed framework on various marketplace architectures and play bidding games with agents empowered by other architectures and techniques to investigate the effectiveness of our approach.

Intelligent SA Agent for Simultaneous Multi-protocol Auctions

Shahbaz A.K.Ghayyur[1], *Saeed Ullah*[1], *S.T. Rehman*[2]
[1]Department of Computer Science, International Islamic University
[2]Department of Telecommunication Engineering Federal Urdu University of Arts, Science and Technology
email : shahbazghayyur@gmail.com , saeedullah@gmail.com, tauseef@fuuastisb.edu.pk

Abstract

As electronic commerce has flourished, the use of agent technology has grown in stature to provide sophisticated and fully automated auction services. This paper deals with the issues of intelligent bidding agent architecture in futuristic integrated electronic commerce systems, via diverse parallel, simultaneous auctions with varying starting-ending times, while incorporating heterogeneous protocols. We propose a modified Belief, Desire, Intention architecture. The proposed mechanism enables optimum gains and efficient learning for concurrent bidding to derive a bidding action plan in highly diverse, fluctuative, fractol and quasi-fractol environment, while taking into account the preferences and demographics of items, bargain leverage, time, supply-demand, auction diversity of interest, and eagerness etc. The agent employs Simulated Annealing to implement its intelligent behavior in not only solitary offline environment but also in a live interactive society, taking it one step further from being a more rule based system. The structure and working of the agent is formulated by classical Architectural Description. This provides a modular description of the semantics about activities performed for optimal service. Explicit specifications on the agent's behavior are also algorithmically formulated as derived form the agent's model.

Keywords: Multi Agent System Framework, Distributed Artificial Intelligence, Architectural Description Language, Simulated Annealing

Introduction

Ever since the known history of mankind the transfer of goods and services under the term of commerce activity is reported. It has ever since been about entities and in the structural form of entities that are valued less by humans, to the people who value them more. Thus creating significance, which in turn gave rise to the term worth. There has always been a rationalization for commerce if it created or gave rise to worth which may be in the form of commodity or service [1].

Bidding is a process which came into existence with the evolution of rudimentary commerce when it was introduced to competition and an imbalance in the mutual supply and demand was created; i.e. more parties striving to gain control of fewer resources due to their potential worth factor [2]. In the modem day world the term bidding still has the same essence i.e. a process in which a party competes with other competitors in order to procure some item at a particular cost. The cost is a reflection of worth and is a price at which the broker/dealer is willing to buy the particular commodity.

As the commerce matured into civilized age, particular specialized institutions were formulated for conductions of bidding to strive for maximum utility form the activity. We have come to know the most promising of such mechanisms as an "Auction" for which the institution of Auction House was introduced. In an Auctioning environment, the process of bidding takes place which is the placement of a bid price presented by a buyer/bidder when he wants to buy a commodity [3]. The bid price is usually just referred to as bid. The bid price stands in disparity to the ask value or the offer, and the divergence between the two is called bid/offer spread [4].

In these auction houses the process of biding can be nut shelled into the broader term of procurement which is defined as "the acquisition of goods or services at the best possible total cost of ownership, in the right quantity, at the right time, in the right place for the direct benefit or use of the governments, corporations, or individuals generally via a contract."[3]

At first the internet technology aided commerce was used to merely offer display and retailing but has quickly engulfed all sorts of business activities. In just a few years the websites have moved form displaying electronic brochures to providing a channel for sales, customer services, and information gathering for small and large enterprises [5]. The streaming of more and more business process on the web paved the way for internet to be turned into a doorway to a virtual business environment [6].

Now a day, *e-commerce is growing at an* exponential rate—literally. According to one study, the Internet economy grew at a rate of 174.5 percent annually from 1995 to 1998. This growth follows the equation Internet economy . *e*year almost exactly (remember, the natural log, *e*, = 2.718...). A number of analysts forecasted that the Internet economy will exceed US $1 trillion (1012) in 2002 [5]; Forrester Research recently predicted that the worldwide Internet economy has reached US $6.9 trillion in 2004. (The Internet Economy Indicators, Indicators Report, June 1999, available online at [7]

This paper first gives a brief overview of the intelligentsia process of the agent which involves isolated offline learning by means of probabilistic stochastic algorithm of Simulated Annealing producing a rule based optimal bidding action plan. Then the online learning behaviour of agent is briefly elaborated by means of machine learning techniques of interactive learning, reinforcement learning and Q-learning making the agent grow in stature from a mere level zero agent to a level two agent. The next section proposes an abstract model of the bidding agent able to bid in heterogeneous protocol, parallel, combinational, simultaneous auction environment with variable start and end times for procurement of single or multiple auctions. This policy for procurement can be termed as the *automated clever bidding policy,* and in the later part the Architectural Description Language is used to define the architecture of agent and interaction with its working environment. Our work satisfies the wide variety of hybrid heterogeneous requirements and the resultant system suits to the problem accordingly.

This research effort deals with developing such a bidding agent based on the a retailored Belief-Desire-Intention Architecture with static intentions and by utilizing probabilistic stochastic algorithm of simulated annealing to determine policy based optimal solution in fractal and quazi-fractal distribution *environments produced as a result of single or* multi- protocol multiple auctions. We chose this particular method because simulated annealing has been known to perform well in areas where the space to be searched is large and not well understood [11] and the problem at hand has to be guided by heuristic since it gradually arrives at better and better solutions.
Since the problem at hand is non-deterministic *algorithmic because a large number of solutions* exist. This situation is classical for an approximation algorithm which yields naturally towards simulated annealing. It is chosen because it has been proven a success in many difficult optimization problems [12]. Among these problems are the *traveling salesman problem*, *image recognition from noisy data*, *integrated circuit layout*, and *robotic optimal path finding and planning* [12].

Related Work

Intelligent decision making on part of contractors is of critical importance when the result oriented bidding activities involve more players and more rounds of interaction, as is common when the supply web becomes more complicated, and many alternative business deals are possible(BidX, Foogle, Auction Beagle Forums[8]). The process of placing bids becomes much more complicated if the starting and ending time of auctions is different as well along with the different protocols adding the variation in preferences of the selling contractor [9]. To facilitate the contractors many of these auction houses have introduced automated bidders which act on behalf of contractors to take advantage of the huge set of processable information about Auctions available by utilizing their computational power, to get the best deal according to the contractor's preferences. These automated contractors are Agent systems or simply referred to as Agents [9].

The users can always take advantage of the information assembled in the shopping engines like FOOGLE, BIDXS or AUCTION WATCH but the basic limitations imposed by mutually exclusive nature of bidding are still unresolved, as the contractor has to make the final decision and go through the painstaking process of scanning the e-market. Furthermore the contractor is still faced with the towering task of selection of a single bid price which will fetch him the best wining deal. In many cases the *customer is trapped in winners curse like* phenomenon where they pay more than they should have to secure the win [10].

We have also cross referenced our work with evolutionary programming like GA, which offers another popular approximation technique, we considered this as a possible approach to the problem of training an agent at hand but evolutionary programming and its variants (hybrid approach etc...) were ultimately found wanting. The case is that in a genetic algorithm, several elements of the solution space are looked at simultaneously; these elements are corresponding to individuals which make up a genetic population. Like individuals in a genetic population, the elements of the solution space experience evolution, which occurs through reproduction and continued existence of the fittest. In order for a genetic algorithm to be applied to a problem, elements of the solution space must be programmed in such a way that two elements can reproduce by exchange of some portion of themselves with their associate, just as biological reproduction involves the swapping of bits of Dioxiribo Nucleic Acid. The problem with encoding strategies in heterogeneous multi-protocol auction environment in this way is that it is not obvious how two or more agent strategies, by swapping groups of auctioning environment or attributes with each other, could produce a new, valid action plan. The requirements that bidding be conducted by the intentioned preferences provided by the user, and that each agent be bidding and buying only one type of item at a given instant in exactly one auction. Arbitrary swaps of portions of strategy are unlikely to result in valid optimal action plan since the protocol and strategies are different for different environments makes genetic algorithms an unreasonable approach to this problem [13].

Abstract view of SABIBAgent

An agent is an autonomous entity with an ontological commitment and an agenda of its own [16]. Every agent possesses the ability to act autonomously. In the e-commerce environment an agent is often acting on a principal's behalf and has a legal duty to act in that person's best interest. An agent may interact or negotiate with its broker and/or with other agents. It may make decisions, such as whether to trust and whether to cooperate with others. They are capable of making independent decisions and taking actions to satisfy internal goals based upon their perceived environment.

Our agent implementation has a stronger notion of autonomy than traditional systems in addition to a reactive, proactive or social behavior as affected on the concerned scenario. If the states of the Scenarios/ Environments can be characterized as a set $S_c = \{S_{c1}, S_{c2} \ldots S_{cn}\}$ where Sc_i is the scenario. At any given instant of time the agent can be faced with only one element of the set of scenarios then the action of our agent can be one element of the set of predefined actions $A_c = \{A_{c1}, A_{c2}, A_{c3}, A_{c4} \ldots, A_{cn}\}$. By application of automate theory it can be represented in the functional form of $A_{cx} : Sc^* \rightarrow A_n$ which maps environment states encountered into appropriate action. We are assuming that the set of environments is limited, predictable and deterministic.

On the agent architectures like *reactive agent architecture, layered agent architecture, belief desire intention architecture* and logic based architectures, we were unable to make a match with our requirements due to highly versatile and hybrid environment constraints. The BDI architecture seems philosophically closest to the scenario in demand. Since the BDI Architecture has its ancestry in the philosophical ritual of understanding *practical reasoning*, the process of deciding, moment by moment, which action to perform in the furtherance of our goals [17], we found it most convenient if it was molded to fulfill our requirements, but since the changes needed were drastic, it resulted into a whole new style. We suggest that [17] should be consulted for anyone who wants further insight on BDI Architecture.

The tailored BDI agent has a set of plans, which defines sequences of actions and steps available to achieve a certain goal or react to a specific situation. The agent reacts to events, which are generated by modifications to its beliefs, additions of new goals, or messages arriving from the environment or from another agent. An event may trigger one or more plans. The agent commits to execute one of them, that is, that plan becomes the intention. Plans are executed one step at a time. A step can query or change the beliefs, performs actions on the external world, and submits new goals. The operations performed by a step may generate new events that, in turn, may start new plans. A plan succeeds when all its steps have been completed; it fails when certain conditions are not met.

We have personalized the traditional BDI to the E-BDIArchitecture with *static intention centric focal point*, while the Desires and Beliefs are persistently updated according to the real-time input data. Since a lot of effort is spent on the development of Architecture Description Languages (ADLs) as can be seen from Rapide[18], Darwin [19], Aseop [20], Unicon [21], Wright [22], Acme [23] and Faulkner [24]. The theoretical aspects of the philosophical Intentions, Beliefs and Desires in E-BDIArchitecture along with their Architectural Description are given below.

Intentions
These are options laid down by the user, and are unswervingly responsible for formulation of the outcome in the ongoing process according to user's requirements. Given that Intentions are equivalent to owner's guidelines, thus they not only impel the deliberation process but are utterly accountable for mean-ends reasoning by serving as means of legalization for Desires and Beliefs. The Intentions are answerable for the SABIBAgent's current focus.

Intention → Cumulative Weight Load (CWL)
CWM → AIW
| AIW connective CWL
AIW → Intention | 0.0|0.1|...|1.0
Intention → RT | RA | BD | EP | LP | SD
RT → Predicate
RA → Predicate
BD → Predicate
EP → Predicate
LP → Predicate
SD → Predicate
Predicate → Function (Predicate)
| Function
Function → Functions of ECOMMBDI
Connective → $\wedge$ | $\vee$ | $\Rightarrow$

Legend
CWL → Cumulative Weight Load (Intentions)
AIW → Atomic Intention Weight
RT → Remaining Time
RA → Remaining Auctions
BD → Bargain Desire
EP → Eagerness to Procure
LP → Limit Price
SD → Supply Demand

Desires
These are the set of options generated during the progression of agent pre bid training. They comprise of the set former solutions by parties for the current problem being on hand, which in this case will be the values of preceding successful bids for procurement of the same item sought for by SABIBAgent.

$D_{set} \subseteq Option_{set}$
Desire $\in D_{set}$
D_{set} → (Desire)*
Desire → Atomic Desire | $\neg$ Atomic Desire
Atomin Desire → Procurement Price
Procurement Price → FPV_1 | FPV_2 | ...| FPV_n

Legend
D_{set} → Set of Desires
FPV → Floating Point Value
* → Zero or more repetitions

Beliefs
The Desires if validated by matchmaking with the Intensions become Beliefs. For example if the Desire was to buy an item for 20$, and the Intention was to buy it for 40$ or less then a Belief is established that the item can be bought and the bidding will instigate on this Belief. Similarly the optimized Desire set gives augment to the Beliefs set which are consistent with the Intentions.

$B_{set} \subseteq Desire_{set}$
Belief $\in B_{set}$
B_{set} → (Belief)* | (Belief connective (B_{set}))*
Belief → Atomic Belief
| $\neg$ Atomic Belief
Connective → $\wedge$ | $\vee$ | $\Rightarrow$

Legend
B_{set} → Set of Beliefs
* → Zero or more repetitions

In short SABIBAgent is a large scale work flow engine based on a modified philosophy of BDI (Belief Desire Intention) Architecture called e-BDIArchitecture, with static Intentions. The agent communicates with global e-marketplace simulator and other agents by means of a built-in asynchronous *message passing scheme* (MPS). The agent exploits all potential resources available at hand to work on user's behalf and continuously reworks the solutions as problem, parameters, constraints or execution environment changes.

The intentions still have the central status in the retailored system and have the following properties and role in the mean-ends reasoning process

Intentions and their effect on mean ends-reasoning in e-BDI Framework

• *Non Varying Intentions*
Intentions for a bidding cycle are constant and have precedence weights associated with them provided by the owner. Only the owner can withdraw the agent if no procurement has been made and the whole bidding cycle has to be restarted if Intentions or Intention weights are changed.

• *Intentions impel means-ends reasoning.*
If an Intention has been made to buy an item from the market place, then the agent will attempt to achieve the Intention, which involves, amongst other things, deciding *how* to achieve it, for example, by entering an auction and bidding for the desired item. Moreover, if one particular course of action fails to achieve an Intention, then It will typically attempt other generated action plans. Thus if it fails to gain an item in one auction, it will try another auctions selling the same commodity.

• *Intentions constrain future deliberation.*
If Intention is to buy a PC, then it will not entertain options that are incompatible with this Intention. Only those Intentions are entertained in the SABIBAgent which are mutually exclusive and the probability of achieving both simultaneously is no infinitesimal. For example bidding for an item at the lowest price ever, with desperation factor of a 100%.

• *Intentions persist.*
The SABIBAgent if intelligent will not typically give up on its Intentions without good reason—they will persist, typically until either it believe it has successfully achieved them, if it believes they cannot be achieved or are unrealistic, or else because the purpose for the Intention is no longer present.

• *Intentions manipulate Beliefs upon which future realistic reasoning is based.*
If the agent adopts the Intention to buy an item, then it can plan for the future on the assumption that it *will* bid for that item and acquire it. For if it intends to procure some item while simultaneously believing that it will never be able to procure one, then it is being irrational.

Re-evaluation of Beliefs and Desires
Beliefs and Desires are re-evaluated if they are not according to the required criteria during the first training phase of the agent. They also determine if the Intentions are realistic or not. If the set of Beliefs is an empty set then the Intentions are not realistic because no such Desires could be gathered or the Desires are inconsistent with the Intentions.

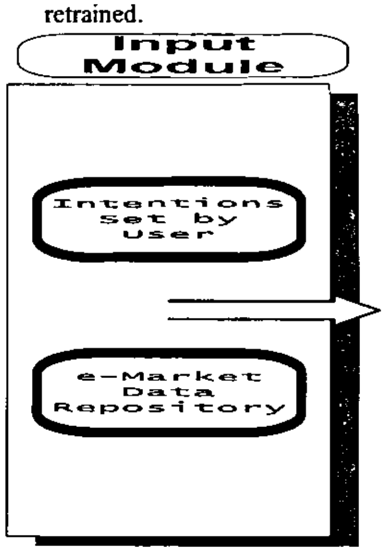
The Beliefs and Desires may be reevaluated in the initial or middle part once the bidding cycle commences if the critical e-market parameters like availability, supply demand etc change drastically or if the other agent's bidding strategies or prices start varying drastically.

SABIBAgent A Detailed Look

The Agent is composed of three main components
1) The input module
2) The Intelligent Action Plan and Processing module
3) The action module

The input module

This module is further subdivided into two sub modules
1. The prioritized intention fetching module which gets the intention set form the user
2. The data repository fetching sub-module which is responsible for providing the data on which agent training is evaluated and the agent is retrained.



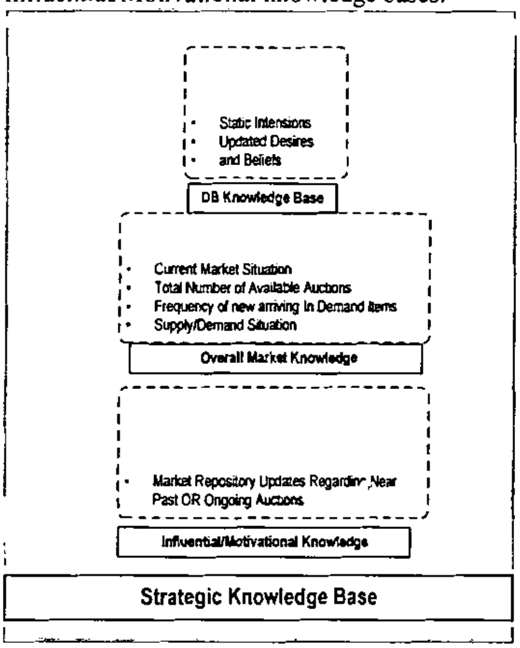
This module is a collection of classes that are responsible with not only the computer human interaction bur also provides the agent with critical data for intelligent decision making. This information is then transferred for use to the Intelligent Action Plan and Processing Module. The functionality of this module can also be enhanced such that it may provide with filtered auctions with some advantageous attributes. This feature can come in real handy if we are using the agent for fixed attribute items.

The Intelligent Action Plan and Processing module

This is the core of the agent and is subdivided into the following functional units

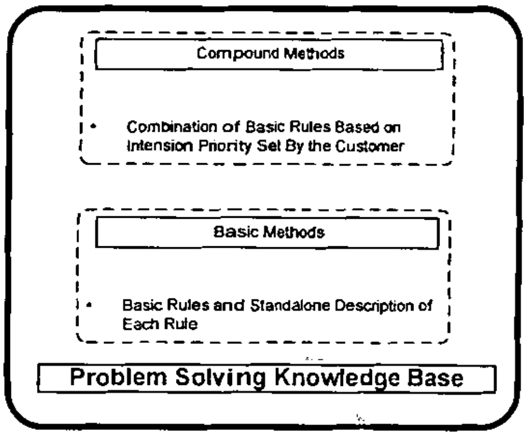
1. Strategic Knowledge Base
2. Problem Solving Knowledge Base
3. Social Knowledge Base
4. Intelligentsia Flow Engine

The knowledge babes provide the means of tracking and reporting, completed and ongoing activity in auctions of interest. It consists of Desire/Belief, Market and Influential/Motivational knowledge bases.



This capability is makes the process of dynamic planning and re-planning as easy as one two three and plays critical role in the allocation of desires and beliefs. The social knowledge base not only provides the means of interaction for Environment Based Action Engines (EBAE's) described ahead but also provide the way of monitoring competitive agents and taking into account the supply demand situations, thus implementing the capability for an agent to grow in stature from being a mere level zero agents initially to a level two agent as the time progresses and agent becomes smarter. [16] should be consulted for further insight into level of agents.

The problem solving knowledge base comprises of basic and compound methods implementation which are used by the intelligentsia process flow engine to formulate strategies.

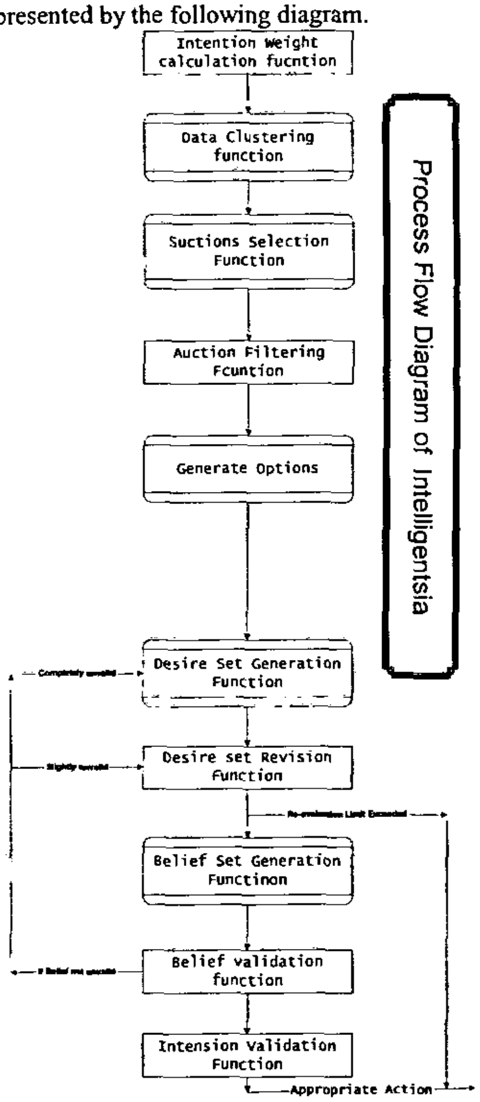


The intelligentsia process flow engine works by manipulation of all the knowledge bases for formulation of smart action plan with the help of following 10 functions implemented in problem solving knowledge base, namely

a. Intension Weight Calculation Function
b. Data Clustering Function
c. Auction Selection Function
d. Auction Filtering Function
e. Option Generation Function
f. Desire set Generation Function
g. Desire set Revision Function
h. Belief Generation Function
i. Belief Based Means-ends Reasoning Function
j. Intention Validation Function

These functions are responsible for the formulation of user provided priority based intention hierarchy, keeping gathered data in organized form for fast and efficient utilization, selection of auctions on interest form the global auction set and keeping track of new events in auctions, filtration of auctions with the highest priority of winning and conflict resolution, building a list of options which are potential candidates of being desires, the desire set production and revision, belief generation and validation and intention validation functions. All these functions are implemented as part of the problem solving knowledge base but are used by this module to evolve the means end reasoning and action plan.

The working of the Intelligentsia process can be represented by the following diagram.

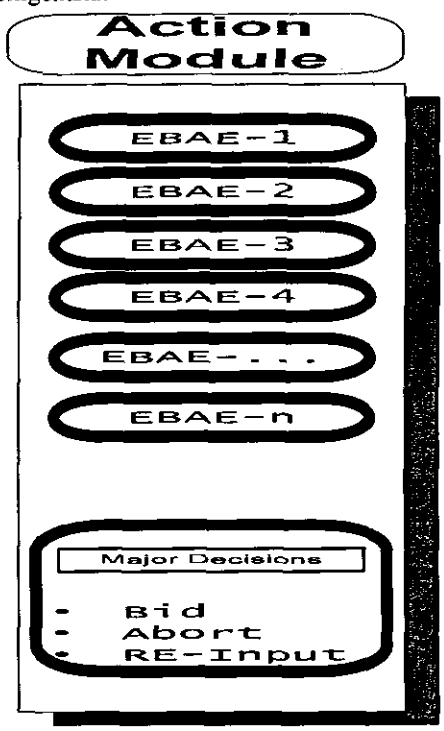


The Action Module

The action module comprises of numerous Environment Based Action Engines (EBSE's). Each EBSE is created and associated with a particular running auction and has a template schema is hard coded to act in a scenario according to a specific e-market protocol for that particular unique auction.

EBSE's exchange information with each other through publish /Subscribe transactions via intelligent action plan and processing Module so that they can synchronize their actions according to the global action plan.

This scheme conceptually distributes a bidding plan throughout the heterogeneous environment. Each EBASE contains its portion of the overall bidding strategy and collaborates via the IAP&PM to achieve the goal. The action strategy is composed of multiple sub-strategies which group together in approaching a single objective. These sub-strategies when implemented in a sequence produce a work flow of the agent intelligentsia.



The working of Agent

The agent is created by the user and is initialized by a set of Intentions. The agent is trained on basis of real-time market data repository. It then negotiates with the heterogeneous market simulator [5] and requests for general market auctions set repository. It reads the repository for selection on committing to the suitable auctions. Suitable auctions are the one which present the in demand item according to our *Intentions*. These suitable auctions are then and an option set is generated and again filters to act on only the best auctions available. These refined options are the foundation stone of Desires built right in the next step. Desire sets again go though a filtering process and are then translated into Beliefs by matchmaking with the Market Data Repository. As a result of this matchmaking only the valid Beliefs are filtered to be part of the mean-ends reasoning process. If no valid Belief(s) are filtered though for effective mean-ends reasoning to take place the process is again repeated form rebuilding Desires onward until a valid Belief set is established which is consistent with our Intentions since newer auctions and results are updated in the data clusters of repository. The agent then commences bidding on basis of this Belief set initial most optimal value and if unsuccessful, the next optimal Desire and Belief is established for next bidding to commence.

IAP&PM creates and awakens the EBAE by subscribing scenario specific information and EBAE then starts perusing its procurement goal in a particular auction synchronizing its actions with other EBAE's. Each EBAE collaborate in one to many relationships or roles with others of its same kind. They are aware of each other via IAP&PM in SABIBAgent system. These roles are defined as DuchEBSE, EnglishEBSE, VickeryEBSE, 1st Price SealedBidEBSE etc. The agent Architecture is scalable and other roles can be easily incorporated by managing them in the Problem Solving Knowledge Base.

The action strategy provides traceable information on the overall current task progress. The intentions which remain unchanged during the course of action (i.e. static Intentions) are the cornerstone of action strategy progress. They are the answer to the what, the why, the how, the when, the where, the who's of the bidding problem. *Furthermore the action module also* decides whether to carry on, quit or suspend the bidding according to the intelligent action plan generated by the intelligentsia process.

A standalone working scenario of the Agent's means-end reasoning plan in ADL form can be represented as

Goal (Item, Market, Auction Set, Protocol) /* Universal Goal*/

Achieve *(Item (Cumulative Weight Load))*

Run (IWCF, DCF, ASF, AFF, (OGF, OFF, DGF, DFF, BGF, BBMRF, IVF)*)$^+$

Run (EBAE$_1$, ….. , EBAE$_n$) $^+$

Failed (IVF = = 0) && Abort

Succeed ((*IVF != 0) && Bid Successful*

Legend
IWCF ➔ Intention Weight Calculation Function
DCF ➔ Data Clustering Function
ACF ➔ Auction Selection Function
AFF ➔ Auction Filtering Function
OGF ➔ Option Generation Function
DGF ➔ Desire Set Generation Function
DRF ➔ Desire Set Revision Function
BGF ➔ Belief Generation Function
BBMRF ➔Belief Based means-end Reasoning Function
IVF ➔ Intention Validation Function

The mid term goals can be defined as

Sub-Goal → Validate (Belief)
| Achieve (Belief)
| Abort (Belief)
| Load (Belief)
| Abort (Session) | Abort

As the environment variables change the re-planning includes how the Belief/Desire are to be effected accordingly thus generating the new scenario centric optimized solution. SABIBAgent continuously monitors the plan and forces re-planning if and only if real-time changes in environment variables are drastic, forcing the agent to take a bolder approach.

The service descriptions provided to and by the agent in the MAS (Multi agent simulation) are descripted by the following specification.

<Train for Auction Participation>
[AuctionMembership= = no]
→ FetchIntentions( )
SetCumulativeLoad(Intentions)
FetchMarketData()
WHERE
(Auction ε {Total Auctions} && time > Auction.End && Desired_Item)

SelectAuctions( )
WHERE
(Auction ε {Total Auctions} && time < Auction.End && Desired_Item)

FilterAuctions( )
WHERE
(Auction ε { Selected Auctions } && time > Auction.End && Desired_Item)
&&(

$$P_i^w(v) = (\sum_{P>v} P_i^c(P) + P_i^c(v)/2) \geq 50\%$$

|| $(P_i^w(v) = (1/\sum n_i / x) \geq 50\%)$

GenerateDesireSet( )
FOR
$((\Sigma_{i=1....n} W_j * I_j)- W_{DP} < W_{DP})$ &&
$((\Sigma_{i=1....n} W_j * I_j)- W_{BD} \leq W_{BD})$
DesireSetRevision( )
WHERE
($Desire_i$.value > Private Value) &&
($Desire_i$.count > Threshold)

GenerateBelief( )
FOR
$((\Sigma_{i=1....n} W_j * I_j)- W_{DP} < W_{DP})$ &&
$((\Sigma_{i=1....n} W_j * I_j)- W_{BD} \leq W_{BD})$

Means-endsReasoning( )
SimulatedAnnealing( )
ValidateIntention( )
IF
Belief !≥ Intention

<Join Auction>
→ time
Set Connection = Connection ϕ
Submit_Request(Auction_ID, Members_ID, Items)
WHERE
Current.time < Auction.StartTime && Desired_Item

<Get Market Membership>
[Join Auction (Auction_ID, Members_ID, Items)]
→ *AuctionMembership = = yes &&*
AuctionID = = AID, MembershipID = = MID
IF
Auctioneer_Accept(MID) = = ok

<Menas-ends Reasonig>
[AuctionMembership= = yes]
→ Means-endsReasoning( )
{
CentralizedLerning(EAData)
Decentralized Learning(DAData, VAData)
ReinforcementLearning(SD)
QLearning(RT, RA, BD, EP, LP)
}

<Initiate Bidding>
[AuctionMembership = = yes]
Submit_Bid(Auction,MID,BeliefValue) && FilterdAuctions.Auctioninfo && Auction.Auct.ID = = Auction

<accept Acknowledgement of offer>
[Initiate Bidding (Auction,MID,BeliefValue)]
→ !Bid_ID = = bid_ID
IF
AcutioneerAgent [Bid_Adjust (Self, AuctionID,midbidID)

<Adjust Bid After Failure>
CentralizedLerning(EAData)
Decentralized Learning(DAData, VAData)
ReinforcementLearning(SD)
Q-Learning(RT, RA, BD, EP, LP)
[Submit_Bid(Auction,MID,Belief.Valu-e)]
Belief + = Belief
→ Submit_Bid(Auction,MID,Belief.Value)
IF
Bid_Failed (Self, AuctionID, mid, bidID)
WHERE

AID = = Acution &&Bid_ID = = bidID
&& MID = =mid

<Pay acknowledged offer price>
[Submit_Bid(Auction,MID,Belief.Valu-e)]
→ Pay (Bid_ID, Payment)
IF
Bid_Accepted(mid, AID, bidID)= =yes
WHERE
AID = = Auction && Bid_ID = = bidID
&& MID= = mid

<Quit From Auction>
→ Quit_Auction (AcutionID)
Membership = = no
IF
Bid_Failed (Self, AuctionID, mid, bidID)
WHERE
AID = = Acution &&Bid_ID = = bidID
&& MID = =mid
IF
Bid_Success (Self, AuctionID, mid, bidID)
WHERE
AID = = Acution &&Bid_ID = = bidID
&& MID = =mid

The degree of boldness or cautiousness [25] is heavily dependent on the active environment of the marketplace simulator as it is in any e-commerce auction house like yahoo auction [26], eBay[27], Amazon[28], Priceline[29], UBid[30] and many others. The implemented system starts off as a cautious agent and evolves towards a bolder approach. If the critical market factors like supply-demand, competent agent density etc. are highly fluctuative. The boldness factor has a threshold of 50% since above this value the agent gets stuck in reevaluating its strategical plan and does little effective work as was the case shown by experimental data sets.

For more insight on Reinforcement learning, and Q-Learning [31] and [32] should be consulted.

Conclusion

In this paper we presented the Architectural Framework of a bidding agent, competent enough to toil in a heterogeneous, multi-protocol, multi auction environment, with user driven focus. The proposed agent framework enables the service demanding party to bid in parallel, simultaneous or combinational auctions employing the means-ends reasoning based action plan, to buy a single or multiple items. The agent is capable of competing and outsmarting potential competitors while working integratedly with other agent systems in a marketplace simulator. The agent framework employs Simulated Annealing as a method of inventing rules for optimal performance and then employees the machine learning techniques of Reinforcement Learning and Q- Learning after starting its interaction with other agents and environment. By employing these techniques the agent grows form a level zero agent to a level two agent and implies stimulus based intelligentsia to outperform other such systems. The composition and operational aspects of the designed agent is formulated by orthodox Architectural Description Language proposed for multi agent systems. The architectural description language in this paper provides a modular account of the semantics of the behavior proposed for optimal service. This also enables unambiguous stipulation on the agent's conduct. Hence, the designated environment can be clearly acknowledged for developers on both patron and market stake holder's sides. The same pattern can also be used by developers of service requesters so that the application can be effortlessly integrated without too much stipulate of methodological support.

Future Work

This paper is not projected to converse the ceremonial characterization of intelligent agents or e-commerce/ m-commerce / i-commerce systems. Neither does it examine or gives synopsis about a variety of approaches for implementing agent oriented systems for electronic environments, and how these approaches counterpart up to various programming paradigms like Object Oriented Programming and Aspect Oriented Programming or Process Oriented Programming etc. not does it present formal definitions and equations regarding any of the machine learning techniques since they are not of primary concern and any one who wants to be more familiar with the technologies and techniques can consult the referenced articles for any further insight into the classical literature[31].

Since we are in the early stages of implanting the learning behavior of a bidding agent via Machine Learning, there is a need to investigate Machine

Learning Reinforcement Learning and Q-Learning techniques further for gaining better results. The issue of look-ahead in auctions, enforcing trust, preventing selfish behavior, avoiding dishonest auctioneers via certificate generation and avoidance of repudiation are concerns of future investigation. We intend to test our proposed framework on various marketplace architectures and play bidding games with agents empowered by other architectures and techniques to investigate the effectiveness of our approach.

References:

[1] Electronic Payment Systems* N. Asokan, Phil Janson, Michael Steiner, Michael Waidner IBM Research Division, Zurich Research LaboratoryCH-8803 Ru"schlikon, Switzerland {aso,pj,sti,wmi} @zurich.ibm.com

[2] eCommerce Trust Study.. Cheskin Research & Studio Archtype/Sapient. 1999. [online]. Available:http://www.cheskin.com/think/studies/ecomtrust.html [viewed July 30, 2001].

[3] http://en.wikipedia.org/wiki/Bidding

[4] David Chaum. Blind Signature for Untraceable Payments, University of California, 1998

[5] http://www.forrester.com

[6] NetBill Security and Transaction Protocol, Carnegie Mellon University Pittsburgh, PA 15213-3890 Benjamin Cox J. D. Tygar Marvin Sirbu thoth+@cmu.edu tygar@cmu.edu sirbu+@cmu.edu

[7] http://www.internetindicators.com/features.html

[8] www.foogle.com
www.auctionwatch.com
www.auctionbeagle.com

[9] Use of a Colony of Cooperating Agents and MAPLE to Solve the Traveling Salesman Problem Bruno · Guerrieri Florida A&M University bruno.guerrieri@famu.edu

[10] Auction Theorey: AGuide to literature. J.Econ. Surv. 13, 3 227-286

[11] Simulated Annealing: Quasi-Fractals and Quasi-Failures. Raymond Lister School of Computing Science Queensland University of Technology GPO Box 2434, Brisbane Q 4001

[12] B. J. Buckham and C. Lambert. Simulated annealing applications. Retrieved September 2003 from the World Wide Web: http://www.me.uvic.ca/~zdong/courses/mech620/SA_App.PDF.

[13] S. Russell and P. Norvig. Arti_cial Intelligence: A Modem Approach. Prentice Hall, Saddle River, New Jersey, 1995.

[14] Simulated Annealing and Combinational Optimization. Surendra Nehar, Sartaj Shani, Eugene Shragowitz. 23rd Design Automation Conference.

[15] B. T. Luke. Simulated annealing. Retrieved September 2003 from the World Wide Web: http://fconyx.ncifcrf.gov/~lukeb/simann1.html.

[16] Distributed Artificial Intelligence MIT press.

[17] M.E. Bratman, D.J. Israel, and M.E. Pollack. Plans and resource-bounded practical reasoning. Computational Intelligence, 4:349-355, 1988.

[18] D. C. Luckham, J. J. Kenney, L. M. Augustin, J. Vera, D. Bryan, and W. Mann. Specification and Analysis of System Architecture Using Rapide. *IEEE Trans. on Software Engineering*, pp. 336-355, April 1995.

[19] J. Magee and J. Kramer. Dynamic Structure in Software Architectures. In *Proc. of the 4th Symposium on the Foundations of Software Engineering* (FSE4), pp. 3-14, San Francisco, CA, October 1996.

[20] D. Garlan, R. Allen, and J. Ockerbloom. Exploiting Style in Architectural Design Environments. In *Proceedings of SIGSOFT'94: Foundations of Software Engineering*, pages 175–188, New Orleans, Louisiana , USA, December 1994.

[21] M. Shaw, R. DeLine, D. V. Klein, T. L. Ross, D. M. Young, and G. Zelesnik. Abstractions for Software Architecture and Tools to Support Them. *IEEE Trans. On Software Engineering*, pp. 314-335, Apr. 1995.

[22] R. Allen and G. Garlan. *Formal Connectors*. Technical Report, CMU-CS-94-115, Carnegie Mellon University, March 1994.

[23] D. Garlan, R. Monroe, and D. Wile. ACME: An Architecture Description Interchange Language, In *Proc. of CASCON 97*, Toronto, pp. 169-183 Jan. 1997.

[24] P. C. Clements. A Survey of Architecture Description Languages. In *Proc. of the Eighth International Workshop on Software Specification and Design*, Paderborn, Germany, March 1996.

[25] D.Kinny and M.Georgeff. Commitment and effectiveness of situated agents. In proceedings of Twelfth International joint conference on Artificial Intelligence (IJCIA-91), pages 82-88, Sydney, Australia, 1991.

[26] www.yahoo.com

[27] www.ebay.com

[28] www.amazon.com

[29] www.ubid.com

[30] www.priceline.com

[31] Baird, L. C. (1995). Residual Algorithms: Reinforcement Learning with Function Approximation. In Armand Prieditis & Stuart Russell, eds. *Machine Learning: Proceedings of the Twelfth International Conference*, 9-12 July, Morgan Kaufmann Publishers, San Francisco, CA

[32] Tuomas W. Sandholm and Robert H. Crites.On Multiagent Q-Learning in a Semi-competitive Domain, 1994.University of Massachusetts at Amherst.