# Semantic eBroker

Bjorn Azzopardi and Matthew Montebello

Department of Artificial Intelligence, University of Malta

**Abstract.** The world of e-commerce presents ample opportunity to fully utilise the capability of intelligent Agents. The highly dynamic, fast-moving and information-rich environment can often be overwhelming for the human participant. Agents can intelligently assist users by mimicking human behaviour and adapting themselves to their client's specification. This thesis presents an e-commerce framework that would introduce negotiation techniques which allows sellers and buyers to trade using Case Base Reasoning techniques as well as being proactive in remembering users' requests and autonomously monitoring vendor sites for new items that might match the users' needs and preferences. It observes the users whilst shopping and learns their preferences with respect to various features that characterise shopping items.

## 1 Background

This paper introduces a thesis with a vision of creating an e-commerce system which would facilitate the buying process in today's competitive world. There currently exist a lot of price comparison sites which compare several products but however these are subject to the website having predefined vendors available.
There has been substantial research in the area of shopping agents. BargainFinder was presented by Andersen Consulting as part of the SMART STORE initiative. It allowed users to compare prices of music CDs from several stores. Ringo was another example which later evolved to FireFly which was based on collaborative filtering. It suggested music basing its suggestions on what people with similar interest chose. The ShopBot was an agent that could learn how to submit queries to e-commerce sites and interpret the resulting items to identify lowest priced items.
The eBroker continues to evolve on these ideas with a vision of being more helpful. Using new technologies it is possible to build a framework for e-commerce comparison sites.
Semantic web gives the possibility for machines to read data within the site. Developing vendor sites using this markup would benefit the readability of the data it contains. This would allow agents to browse through the website without the need to build vendor specific modules.
Software agents assist users and will act on their behalf. An agent is often based on fixed pre-programmed rules and multiple agents can coexist together to achieve several goals.

Learning through experience is one of the main drivers to human know how. Case base reasoning is a proven area in artificial intelligence which uses past cases to devise a strategy for the future. This can be useful in being able to choose a negotiation strategy.

Inductive machine learning methods extract rules and patterns out of massive data sets. The system will learn user behaviour by observing user preferences whilst requiring no or minimal user feedback. The system will be give the ability to adapt to the users preferences and suggest the products the user is most likely to buy.

## 2 Introduction

The architecture of the system would be that of having multiple agents with diverse goals and capabilities to solve specific problems. Using an effective platform for coordination and cooperation amongst themselves would allow them to achieve one specific goal which would be to offer a client a service. The system would be made up of two external entities namely the client and the vendor. For each client, there are a number of lists for different items which are ordered according to the user preferences. The lists are gathered according to the interests the user has had in the past whilst also being able to suggest the most likely product the client would like to purchase. A buyer can specify the type of product, the number of lots, the maximum purchasing price, expiry date and time, etc. The vendor, through his website or application can specify the product, the number of lots, the minimum selling price, expiry date and time, etc. The seller group of agents residing on the vendor's server, communicate product information, negotiation details, etc to the E-Broker group of agents residing on the E-Broker server.

The system maintains a database of outstanding requests from prospective buyers towards products various vendors are offering. The system would then enrol in a negotiation process between the buyer and the seller trying to arrive to a price that suits both parties. An agent (the buyer negotiator agent) makes an offer to another agent (the seller negotiator agent). If the seller accepts the offer the negotiation process ends, and the buyer can buy the product. If the seller does not accept the offer then the buyer can make a counter offer. The bidding proceeds in this fashion until an offer is accepted or a maximum number of offers have been rejected.

In such a brokerage system, AI techniques are applied to the short-listing and the negotiating stages and the amount of computation grows rapidly with a large number of clients and requests. The focus of research is therefore on efficient strategies and algorithms so that the system can respond to clients' requests within a short time. An e-commerce trading system is expected to complete a great number of transactions every day and the system has to respond in a matter of seconds. Thus the demand on response time is very critical.

## 3 System Architecture

Figure 1 illustrates the Privacy and the eBroker server. The privacy agent allows the user to take on a shopping persona and hides all identifying information. A shopping persona is unique identity for a particular user and its external details are independent to the owner's identity, therefore the privacy of the user is kept intact. Each user can have many personae for different purposes, even outside the scope of this system. This will allow the eBroker system to learn a different profile for each persona.

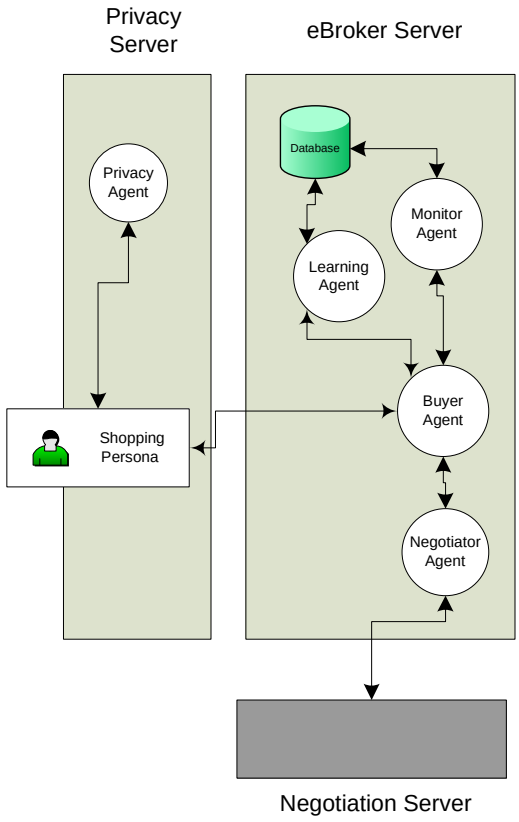


Fig. 1: Privacy Server, eBroker Server and Negotiation Server

The remaining module in figure 1 is the eBroker which is hosted on a separate server and is used for the user to interact with. The buyer agent takes user requests, displays results for searches and communicates with vendors for products. The learning agent takes user requests through the buyer agent and saves

them on a local database. It also monitors users' actions and adjusts the profile accordingly. When a user submits a search through the Buyer agent the learning agent parses this request and returns results according to the profile learnt in the past.
The monitor agent periodically queries vendors for outstanding user requests. The negotiator agent acts as a client to the negotiation server where all the negotiations between the buyer and server will occur.
Figure 2 illustrates the Vendor Server which contains a semantic enabled website containing all the products the vendor is selling. The seller agent communicates with prospective clients and accepts requests for various products. Using the Browsing agent a list of products matching to the criteria are retrieved and returned to the buyer. The browsing agent is an agent that can traverse semantic websites searching for specific queries, which in this case are various products.

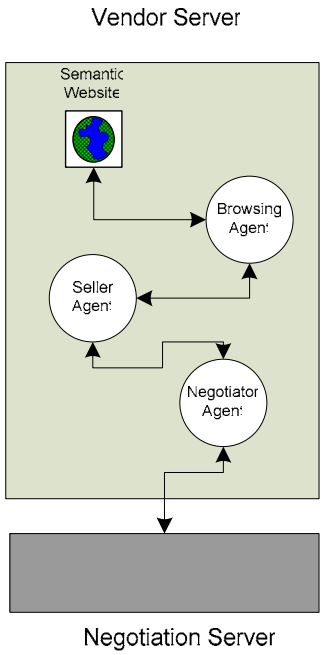


Fig. 2: Vendor Server and Negotiation Server

The seller agent can also accept a negotiation request which is passed on to the negotiator agent which in turn communicates with the negotiation server. The negotiation will commence on the negotiation server.
The negotiation server is a module where all negotiations between the respective vendors and buyers will occur. The negotiation server holds a case repository which is used for case based reasoning. This consists of a repository of successful and unsuccessful transactions attached to a user profile. The case base repository

is then used to extract similar negotiations in the past and use that past knowledge to come to an accord. The user profile is saved in order for the case base reasoning filter to extract only cases that have occurred to that specific user. This creates different negotiation skills to different sellers and buyers, allowing vendors to be better than others, etc.

### 3.1 Learning Agent

One of the main features of the eBroker system is that the system can adapt to user preferences. The system gathers information whilst requesting minimal feedback from the user, therefore it studies user actions. For example if a user shows interest in expensive items then the system learns that this persona likes expensive items and starts giving more importance to expensive items. If the user ignores or removes an item from a list then the persona has shown that it is not interested in this type of item therefore showing that they dislike the features of this product. In time the system will learn the likes and dislikes of the user.
Inductive machine learning requires the extraction of features from the items in question. In this case the features which have been chosen in the current model are the price of the item, the number of bids that have been made towards this item, the time remaining for the auction and the similarity of the user query with the description of the item.
For each feature, e.g. price, a range of distribution temperatures are maintained across the range of values, e.g. low, medium and high. The temperature of a feature and value pair should indicate the users past desirable features.
The actions the user conducts maintain the temperatures for a feature value pair. The four possible reactions are to buy, browse, ignore and remove. Each reaction contains a predefined weight. For example buy contains a change in temperature weight of $+0.5$ since the reaction is considered to be a strong and positive one. When an item is removed a change in temperature weight of $-0.5$ is applied since it is considered a strong negative reaction.

$$T(t+1) = \alpha_1 T(t) + \alpha_2 \Delta T \quad \text{where } \alpha_1 = \alpha_2 = 1 \tag{1}$$

Items are then listed according to a simple sum of their temperature. User interactions then cause an update in temperatures and a re-ranking of the items based on updated temperatures.

### 3.2 Monitor Agent

The monitoring agent is a background agent which continuously queries vendors for any requests the user has which have not yet expired. The queries are done at intervals specified in the profile, and update the list of available items for the user. Updates could also include change in prices or product descriptions, whilst also adding new items that might have been added by the vendor. The next time the user will log on an amended list will appear.

This means that the system is autonomous by browsing the several vendors available for any products that the user might want then giving a result within a predefined timeframe.

### 3.3 Buyer Agent

The buyer agent acts in the name of the actual user. Its main tasks are that of coordinating all surrounding agents allowing them to work together and interface with any external systems which in this case would be the vendor and the privacy agent.

### 3.4 Negotiator Agent

The negotiator agent is in charge of communication with the negotiation module. This agent exists on both the vendor module and the eBroker module. When a user decides that a product can be bought then the negotiation agent is notified to start the negotiation process with the seller. In the meantime the buyer agent also notifies the seller agent that a negotiation process can commence which instructs the negotiation agent to commence negotiations through the negotiation server.
All negotiations between buyer and seller will occur on the negotiation server and when an accord on price has been reached the respective negotiation agents are notified together with any relevant information.

### 3.5 Privacy Agent

The privacy agent is used to hide the real identity of the user. It creates a persona which is used by the seller agent to communicate with other components. Using this agent none of the external components would know the real identity of the user.

### 3.6 Browsing Agent

As will be discussed further in this document the main aim behind semantic websites is that machines can read the information they contain. This agent would be used to browse semantic websites and search for specific queries the buyer might require.
This would allow the development of one agent that could be used for all vendors having semantic websites rather then developing vendor specific modules.

### 3.7 Seller Agent

The seller agent acts in the name of the actual vendor and coordinates with the all surrounding agents. It is also used as an interface for the vendor module and is used for all communications with the seller.

### 3.8 Semantic Website

Humans are able to browse the web and extract useful information however an agent cannot achieve this due to the fact that web pages are built for humans to read. The semantic web provides a common standard called the Resource Description Framework (RDF) to publish relevant information in a machine readable form. RDF is a markup language used for describing information and resources on the web.
Using the semantic web as a requirement for our vendor sites will give agents the ability to browse through the website and extract data which on this case are products that will interest the buyer. This will remove the need to develop vendor specific modules to extract information from various websites.

### 3.9 Negotiation Module

A good negotiation skill in humans seems to come from experience, which comes to the reason why Case Base Reasoning was chosen. Past negotiation situations are used as strategies for future similar situations.
Figure 3 shows all the negotiation modules needed to develop the negotiation server.

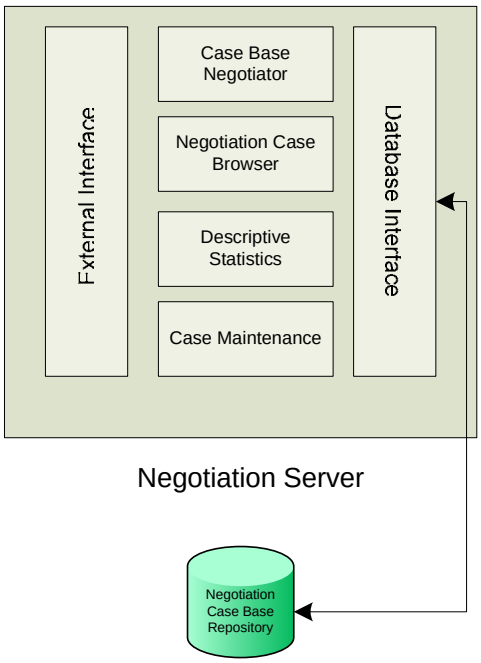


Fig. 3: Negotiation Server Modules

- The External Interface would be the interface that the negotiator agents would be using to communicate with the negotiation server.
- The Database Interface would be the module with which the server can communicate with the negotiation case base repository database server.
- The Case Base Negotiator helps the clients to negotiate between each other. Each user will have its own set of cases or can be set to use all past cases. This module will also save any new negotiations to its repository.
- The Negotiation Case Browser allows user to browse past cases.
- The Descriptive Statistics gives useful statistics on the case base repository.
- The Case Maintenance allows negotiation experts to amend the case base repository and fine tune it.

Negotiation is an iterative process of offers and counter offers where a seller and a vendor would both have pre-defined goals and with this process their goals would converge to an agreement. Given an offer by a buyer, the negotiation engine will evaluate it to decide whether the offer is acceptable. If the seller decides the offer is not acceptable then the buyer has to decide a strategy for a counter offer.
Case Base Reasoning techniques are used to extract strategies from previous similar negotiations to propose a new strategy. Similarity/matching filters are used to filter out the best matching past cases. Any new strategies are saved to increase the knowledge of the case base repository.

## 4 Conclusion

This paper introduced the eBroker system which would give shoppers the ability to buy products with as little human intervention as possible. The system would also learn human user preferences without any feedback from the user and also allowing the system to negotiate a desirable price whilst learning how to negotiate as time goes by.
The current market shows that e-commerce is rapidly growing and there is a need for such a system. The e-commerce market is so vast that products are available all over the place and there is no way that a user would have seen all products before coming to a decision. Currently there exist comparison websites which have predefined vendors programmed within, however using this framework, will give the user the ability to view all the products available within the query.

## References

[3WC] World Wide Web Consortium. `http://www.w3.org`.

[BBS] Phillip Bradford, Herb Brown, and Paula M. Saunders. Pricing, Agents, Perceived Value and the Internet. *Online Journal on the Internet*, Special Issue 6: Commercial applications of the Internet. `http://www.firstmonday.org/issues/issue6_6/bradford/index.html`.

[BdL+98] Martin Beer, Mark d'Inverno, Michael Luck, Nick Jennings, Chris Preist, and Michael Schroeder. Negotiation in Multi-Agent Systems. In *UK-MAS'98*, 1998.

[BF] Bargain Finder. http://crowston.syr.edu/papers/icis97present/wacked/agent.htm.

[CBR] Case Base Reasoning. http://www.i-rescue.com/links/cbr.html.

[DEW97] Robert Bo Doorenbos, Oren Etzioni, and Daniel So Weld. A Scalable Comparison-Shopping Agent for the World-Wide Web. In *ShopBot*, 1997.

[Dum00] Edd Dumbill. The Semantic Web: A Primer, November 2000. http://www.xml.com/pub/a/2000/11/01/semanticweb/index.html Last Accessed 15th October 2007.

[Hen] James A. Hendler. An example of Semantic markup, deconstructed. http://www.cs.umd.edu/users/hendler/sciam/walkthru.html Last Accessed 15th October 2007.

[IA] Intelligent Agent. http://en.wikipedia.org/wiki/Intelligent_agent.

[PDS07] Pantelis M. Papadopoulos, Stavros N. Demetriadis, and Ioannis G. Stamelos. Case Base Instruction on the Web. In *ITiCSE07*, 2007.

[RDF] Resource Description Framework. http://www.xml.com/pub/a/2001/01/24/rdf.html.

[RS] Recommendation Systems. http://imscdmim.usc.edu/publications/121new.pdf.

[SOA] Simple Object Access Protocol. http://www.w3.org/TR/2000/NOTE-SOAP-20000508/.

[SSL] Secure Sockets Layer. http://info.ssl.com/article.aspx?id=10241.

[SW] Semantic Web. http://www.w3schools.com/semweb/default.asp.

[XgPwY03] Bing Xu, Zhi geng Pan, and Hong wei Yang. Agent-based Model for Intelligent Shopping Assistant and its Application. In *ICAT2003*, 2003.