A Multi-Agent System for E-Insurance Brokering

Luís Nogueira[1] , Eugénio Oliveira[2]

[1] Instituto Superior de Engenharia do Porto,
Instituto Politécnico do Porto
luis@dei.isep.ipp.pt
[2] Faculdade de Engenharia,
Universidade do Porto, LIACC
eco@fe.up.pt

Abstract. Agent-based systems suitable for dealing with applications where the environment is both dynamic and populated with competitors demand for sophisticated characteristics including adaptation, negotiation and coordination. In this paper we propose an agent-mediated insurance brokering system using a flexible negotiation model that includes multi-attribute bidding as well as some kind of learning capabilities. Moreover, in the core of the provided brokering facility, we are using conceptual clustering procedures as an approach to better match customers and insurance product offers providing a valuable add-on to both customer's and sellers' sides. Intelligent agents engage themselves in a negotiation process by exchanging proposals and counter-proposals trying to convince opponents to modify their bidding values. We are now developing a Java based multi-agent infrastructure specifically dedicated to the insurance e-commerce domain, exploiting Toshiba's Bee-gent framework. For both acceptability and generalisation purposes, XML (including appropriate ontology-based messages) has been chosen as our agent communication format.

1 Introduction

In most current e-commerce applications, the buyers are generally humans who typically browse through a catalogue of well defined products (books, computer components, CDs) and make fixed price purchases. However, there are important differences between selling this type of goods and selling insurance and other financial products over the Internet. This paper explores some of those differences from both the insurers' and customers' perspectives and describes the development of a multi agent system through which products (and services) offered by insurance companies could be better evaluated and selected. From the customers side point of view, more interesting information can be found, even things that the customer did not think of before. On the other hand, the electronic market system, with its intimate knowledge of who the user is and what he wants, can shorten the time needed for finding an appropriate insurance product. Insurers can then use information automatically collected during negotiation to develop a more customer-directed kind of marketing strategy. Information about the customer can be used to find out what he is interested in and, therefore, a more personalised product could be offered.

2 Electronic Markets and Insurance

Despite e-commerce's huge impact on business in general, the insurance industry has yet to fully embrace it. The insurance industry faces several obstacles to e-commerce, including customer attitudes, complex insurance policies, state regulations and the traditional agency distribution system.

The insurance business is based on selling a service at a given risk. Insurers have to make premiums high enough to cover the forecast level of claims but also to keep them low enough to be attractive in an increasingly competitive market. The balance between profit and risk is fundamental to the success of any insurance business.

Insurance companies traditionally have segmented the policyholders into separated lines of business such as auto, life and business markets. Each segment would carry its own underwriting, claims and marketing strategy. This made it impossible to get a complete view of multiple relationships a customer might have with one company.

E-insurance involves the advertisement, recommendation, negotiation, purchase and claim settlement of insurance policies through the Internet. At present, most of these processes are not automated. Some insurance sites offering web-based policies are little more than passive catalogues of alternatives available to customers.

The success of the sale of an insurance policy depends on how good the requirements of the insurer have been matched with the terms of the policy. In the conventional insurance industry, the insurance company initially informs its customers through advertisements. Advertisements are made either through passive channels like newspapers, magazines, billboards, radio and television, or, through active channels like human insurance agents. E-insurance employs the Internet to reach customers through advertisements more effectively since it integrates the traditional passive and active channels of advertisement into one. Advertisement banners, e-mail notifications and coupons are used to replace passive media, while software agents replace their active human counterparts.

However, it's a much difficult task to match the insurance requirements of a customer with appropriate financial complex products than, for example, finding those shops that sell a specific book. An insurance policy has benefits, conditions and exclusions that add detail to the high level coverage features advertised by the insurer.

The availability of an insurance product may be determined according to the risk profile of the customer. Customers will be asked to supply enough information from which insurers have the ability to make a proposal. An electronic market will therefore need to provide a facility, capable of exploiting this bidirectional exchange of information.

3 Existing Online Insurance

Despite the increasingly rapid advances made in computer technology, companies in the insurance industry are making the most use of the new technical possibilities offered today in their internal operations only, while hardly employing them at all when negotiating with their customers. A recent comparative study of 25 web-based Internet sites offering comparative term life insurance information from Consumer Federation of America has showed us that not all sites are useful for getting quotes. Some of them are too difficult to use and others little more than referral services where you are put in touch with an agent, something you, most of the time, do not need the Internet to achieve. Several of the quote services do not include no commission insurance companies, because many of this sites make money through commissions on sales and do not show this companies in their service because it would reduce their incomes [9].

Most web sites offering online quotation and purchase of insurance products are implemented by the insurers and sell directly to the customer, excluding the broker. Brokers, however, provide a valuable service and are widely used by customers. Yet, online brokerage is rare.

Those sites, which do offer a brokerage service, do so by drastically simplifying the problem: they standardise offered products. The broker's role is then reduced to collect a standard set of information from the customer and negotiate standard coverages. This gives no advantage for any of the players because:

- customers are provided with a more limited choice of products, which may not meet all their requirements
- insurers have limited flexibility in product design, targeting and pricing
- brokers lose their traditional role

4 Insurance Brokerage

Insurance brokerage is a process involving three types of players:

4.1 Customers

The customer wishes to buy an insurance to cover certain risks. This requirement will usually be incomplete and uncertain and, possibly, the customer will not be aware of all the options available and may be prepared to compromise on certain aspects. The customer expects the broker to help and advise in defining his needs, to select an insurance product and to appropriately deal with insurance companies always taking customers' preferences into consideration.

4.2 Insurers

Each insurance company offers a number of insurance products, covering different risks and aimed at different groups of customers. Each product is only available to customers who satisfy a complicated set of rules, designed to minimise the risk of the insurer. The product is usually configurable, allowing the customer to select the amount of each type of cover and a number of other different optional extras. The premium charged by the insurer takes into account the characteristics of the customer and the risk insured, as well as the configuration of the product.

4.3 Broker

The broker mediates between customers and insurers, attempting to supply each customer with a product appropriate to his needs at an acceptable price to both parties. To accomplish this role, a broker needs to execute four different, although related tasks:

- To gather and filter information about available products
- To discuss customers' requirements and provide adequate advice
- To negotiate product details with insurers on behalf of the customer
- To establish the final contract involving both the customer and the selected insurance product provider

5 Agent Mediated Insurance Brokerage

5.1 Our Proposal

It is our belief that intelligent agents are well suited to deal with the insurance brokering problem in a distributed manner. By configuring a society of intelligent agents, each one charged with autonomously carrying out different specific functionalities, the insurance broker system will not only be able to analyse products being offered, but will also deduce useful information regarding the current state of the market.

We here present a distributed, intelligent agent-based system making it possible the electronic commerce of insurance products. Our approach for an agent-based insurance products assisted electronic market includes an agent representing each of the insurers, an agent representing the customer and a broker agent for intelligent brokering services. Each insurer has full ownership of its agent, ensuring that all strategic information remains confidential. The goal is to support distribution of a full range of insurance products from several different insurers without the need to modify or constrain them in standard rigid formats for electronic commerce purposes.

By increasing the degree and the sophistication of the automation process, commerce becomes much more dynamic, personalised and context sensitive. These changes can be of benefit of both customers and insurers. From the customers' perspective, it is desirable to have software that could search all the available offers to find the most suitable one and that could then go through the process of actually purchasing the product.

From the insurer's perspective it is desirable to have software that could vary its own offering depending on the customer it is dealing with, what its competitors are doing and the current state of its own business.

We use a common descriptive language for describing the terms, conditions and relationships of insurance products. These terms can define any aspect of the market. Relationships can be used to define dependencies between terms and values of these terms. A product is then defined as an encapsulation of an attribute set to define an individual item or service for sale by an individual insurance company in the market. A requirement is the specification of what a customer wishes to buy. It stores attributes about the customer and the details of his insurance requirements.

Given a customer's details and requirements, the broker must find a match with the list of insurance products proposed by insurers. The customer will have specific requirement constraints and preferences as to his ideal product. Insurance products will have a set of eligibility criteria which may exclude some customers or require further information to be determined. In our system both products and requirements have attributes, constraints and preferences.

Customers need to supply enough information from which insurers have the ability to make a proposal. Different insurers will require different information in order to make a decision. The simplest solution would be to display all the required questions to answer. However, this may prove to be too extensive list to be acceptable to most customers and eligibility often involves dependencies rather than straightforward constraints.

Communities of users can be used to improve the negotiation of insurance products. In this paper we examine *unsupervised learning* for the acquisition of user communities. The question is whether there is any meaning in the generated communities, that is if they associate users with a limited set of common interests. For this reason we use a metric to decide which preferences are most representative for each community. This approach allows the insurance company to target product configurations at specific market segments, and avoids the need to ask all customers the same typically large number of product specific questions.

In order to reach an agreement about a particular insurance product a negotiation process is started by the broker. This negotiation process comprises several rounds, starting when the broker sends and announcement for all the insurer agents in the market. The negotiation ends when a deadline is reached or a satisfactory proposal is received. At each negotiation round bids are evaluated. Bids evaluation is done through a multi-issue function that encodes the customer's preferences.

A Customer Agent coordinates the dialogue between the customer and the Broker Agent, passing on information as appropriate. It offers the customer a flexible navigation tool that allows the exploration of the received proposals. This is particularly useful because the customer wish to express product feature preferences and view the corresponding proximity of each offered product. The result is a ranking of products, which can be tuned by the customer by varying the preferences and viewing the consequent effect on the ranked list [18]. Such a navigation tool encourages the user to consider non-price related features and helps the customer to explore the trade-off between product features and price. This is not just of benefit to the customer, because insurers have also the means of drawing attention to their products' distinguishing features other than price [7]. This helps the customer to make an informed purchase decision.

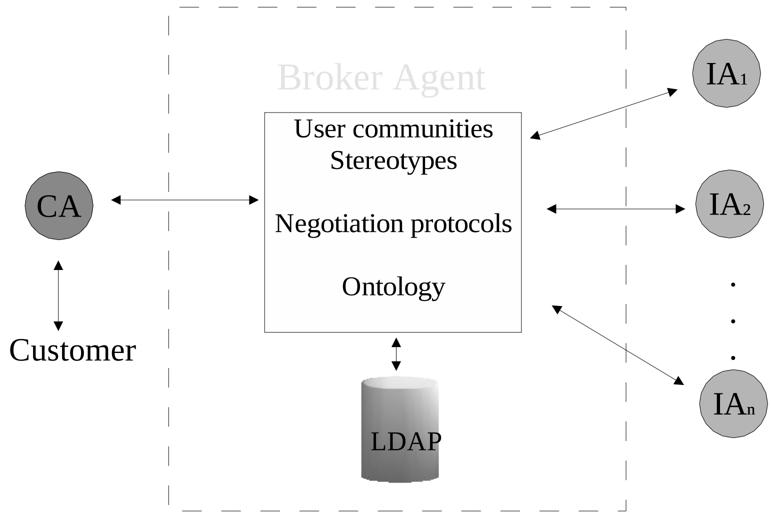

Fig. 1. Agent based architecture for the Insurance Brokering System

5.2 Stages of the Brokering Problem

Fig. 1 briefly depicts the overall architecture including needed agents as well as agent interaction links in our agent-based insurance brokering system. Notice that, besides relevant services like negotiation protocols and ontology-based services, BA, the Broker Agent, provides a facility to build up, memorise and exploit User's Stereotypes. Although this concept will be elaborated later on section 5.2, the reader can see a stereotype as a classification of a set of features and characteristics associated with a specific sub-set of customers.

Our model for the insurance brokering activities divides the interaction process aiming at solving the problem into five stages as follows:

In **Stage 1**, the user defines the allowed range for the insurance product attributes' values. This definition includes attaching a degree of importance (weight value) for each one of the product's attributes (in a range from *low* to *high*) and the increasing order of preference for the attributes' values.

Stage 2 , starts when the Customer Agent (CA), instructed by the user, sends a request to the Broker Agent (BA). This request is constructed by selecting the most important value for each one of the user-defined attributes. When it is possible to obtain, according to the defined values, the direction in which the user is willing to relax the value's constraint, this direction is also sent in the initial request. The purpose of all that information is helping the BA evaluate the received proposals.

The BA replies with a stereotype to CA. Based on this received stereotype, the user can refine its initial request. CA re-sends a, possibly improved, request to BA. Now, BA also asks user's preferences to CA.

In **Stage 3**, BA sends an announcement to each Insurer Agent (IA), starting a negotiation process. Each IA replies with bids to BA, which are then evaluated according to CA preferences, extracting relevant features from these bids. At each negotiation round bids are evaluated, and the one with the greatest evaluation value is considered the winner in current round.

When this negotiation process ends, BA starts **Stage 4**, by starting a new interaction with CA. This interaction will ultimately direct the system to a suitable solution through a constraint satisfaction process.

If this conversation has produced a valid number of alternatives, BA initiates **Stage 5** by ranking selected proposals according to user utility function results, and send them plus relevant information to CA. The user either rejects or agrees with one of the received proposals.

If the user has selected one of the proposed insurance products, BA resumed to **Stage 6**, establishing a contract with the winning IA.

6 Learning About User Communities and Stereotypes

6.1 User Communities

Machine learning methods have been applied to user modelling problems mainly for acquiring models of individual users interacting with a system, e.g. [1][2][16]. Recently, other authors like [14] have approached this subject including a higher level of generalisation of the users' interests and leading to the identification of different user communities in a population of users.

The choice of a learning method or algorithm largely depends on the kind of training data that is available. The main distinction in machine learning algorithm paradigm is between *supervised* and *unsupervised* learning. Supervised learning requires the training data to be preclassified. This means that each example is assigned a unique label, signifying the class to which the item belongs. Given these data, the learning algorithm builds a characteristic description of each class, covering the examples of this class. The important feature of this approach is that the class descriptions are built conditional to the preclassification of the examples in the training set. In contrast, unsupervised learning methods do not require preclassification of the training examples. Through these latter methods clusters of examples are built up, which share common characteristics. When the cohesion of a cluster is high, i.e, the examples described in it are similar, such cluster defines a new class.

User communities can be automatically constructed using an unsupervised learning method. Unsupervised learning tasks have been approached by a variety of methods, raging from statistical clustering techniques to neural networks and symbolic machine learning. The branch of symbolic machine learning that deals with this kind of unsupervised learning is called *conceptual clustering* and a popular representative of this approach is the COBWEB algorithm [3]. Conceptual clustering is a type of learning by observation that is particularly suitable for summarising and explaining data. Summarisation is achieved through the discovery of appropriate clusters, which involve determining useful subsets of an object set. In unsupervised learning each example is an object. Explanation involves determining a useful concept description for each cluster.

COBWEB is an incremental clustering algorithm that employs the concept of *category utility* [5] to create a clustering that maximises inter-cluster dissimilarity and intra-cluster similarity. The category utility of a partition is measured by the following equation:

$$CU = \frac{\sum_k \left(P(C_k) \left[\sum_i \sum_j P(A_i = V_{ij} | C_k)^2 - \sum \sum P(A_i = V_{ij})^2 \right] \right)}{k} \quad . \tag{1}$$

where k is the number of categories or classes, C_k is a particular class, A_i refers to one of the I attributes and V_{ij} is one of the J values for attribute A_i.

COBWEB performs its hill-climbing search of the space of possible taxonomies and uses the expression above for category utility to evaluate and select possible categorisations. It initialises the taxonomy to a single category whose features are those of the first instance. For each subsequent instance, the algorithm begins with the root category and moves through the tree. At each level it uses category utility expression to evaluate the taxonomies resulting from the following four steps algorithm:

1. Classifying the object with respect to an existing class
2. Creating a new class
3. Combining two classes into a single class (merging)
4. Dividing a class into several classes (splitting)

Using COBWEB on a small data set generated a concept hierarchy presented in Fig. 2. An important property of the hierarchy is the balanced split of objects in different branches. Therefore the underlying concepts are of similar strength.

6.2 Stereotypes of Customer's Communities

Our insurance brokering system is then using COBWEB as a tool for grouping potential customers in meaningful classes we call stereotypes.

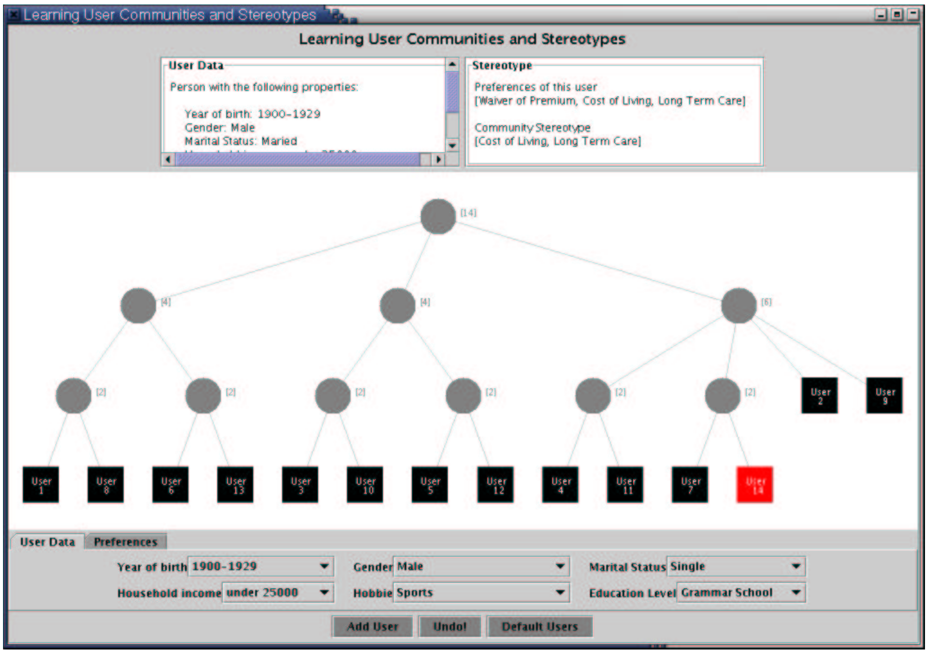

Fig. 2. Learning user communities and stereotypes

The clusters generated by COBWEB, should represent well-defined customer communities. Besides the customer data used for incrementally build up customer's communities, a customer is characterised by its own preferences and by the insurance configuration chosen in the negotiation process. Thus, the natural way to define meaningful stereotypes associated to customer's communities is by trying to identify patterns that are representative of the participating users. We try to construct a prototypical model for each community, which is representative of its users and significantly different from other communities of users.

In order to build appropriate stereotypes, our system is using a specific metric to measure the increase in the frequency of a specific preference or negotiation result within a given community, as compared to the default frequency in the whole number of available observations [15]. In [14] and [4] the increase in frequency was used as an indication of the increase in the predictability of the feature (a given preference, for example) within the community. Given a component c (a user preference), with the default frequency f_c, if the frequency of this component within a community i is f_i, the frequency increase is defined as a simple difference of the squares of the two frequencies:

$$FI_c = f_i^2 - f_c^2 \ . \tag{2}$$

When FI_c is negative there is a decrease in frequency and the corresponding component is not representative of the community. A community's representative characteristic is found through $FI_c > \alpha$, where α is pre-established as the required threshold for considering that frequency increase enough relevant.

The hierarchy generated in Fig. 2 has showed us that the insurance attributes that are chosen by either to few or to many users do not appear in the constructed community stereotype. In the former case the algorithm ignores them during learning and in the latter case, they correspond to such general interests, that they cannot be attributed to particular communities. Filtering out this to types of preferences is a positive feature of the used metric [15].

7 Negotiation

7.1 Negotiation Protocol

Most of the well-known negotiation protocols deal with a single dimension (usually the price). Moreover,[20] also says that the preferences' reduction to the price single figure is what characterises the market. However, this seems to be a bit simplistic and unrealistic for the insurance domain.

Some authors [19][10] advocate the use of auctions protocols for agent-mediated electronic commerce, arguing that they are widely recognised by economists as the most efficient way of resolving one-to-many negotiations.

Other authors point out the limitations of auctions protocols and look for more flexible negotiation models [7]. They stress out that online auctions are in fact less efficient and more hostile than it would be desired. For example, the winner's curse, i.e. "the winner pays more than the real value of the product", seems to be a consequence of auctions and, therefore, they propose more cooperative multi-attribute decision analysis tools and negotiation protocols using distributed constraint satisfaction policies to support it.

However, both lines of research are aware of the need of enhancing simple price-based auction mechanisms, transforming them into new protocols encompassing multi-dimensional issues to be negotiated among the market participants.

In order to reach an agreement about a particular insurance product, customers and brokers usually engage themselves in a sequential negotiation process composed of multiple rounds for exchanging proposals and counter-proposals. A negotiation protocol should then be defined in order to select the participants in the electronic market that, based on its capabilities and availability, will be able to make the optimal deal according to its own goals.

Real negotiation in the insurance domain implies taking into consideration not only one, but multiple attributes for defining the insurance under discussion. For instance, although the policy premium is an important attribute, the length of the coverage, the renewability and convertibility of the policy are complementary issues to include in the decision about whether to buy or not a specific insurance product.

Attaching utility values to different issues, helps to solve the problem of multi-issue evaluation. However, in some cases, it can be a very difficult task to attach absolute values to issues' utilities. A more natural and realistic situation is to simply impose a multi-issue evaluation based on a qualitative, not quantitative, measure.

One way of enhancing agent's autonomy in dynamic environments, is to endow its architecture with learning capabilities. Agent learning process may include several different facets, from simple, step by step, adaptation to the changes in a dynamic environment, to heavier and more sophisticated processes of gathering new knowledge about the environment and other agents based on the past history.

This capability is included, in our negotiation protocol, through a Reinforcement Learning algorithm. Reinforcement learning algorithms support continuous, on-line learning during the negotiation process itself by making decisions according to the environment reactions in the past. The history of a negotiation is a crucial information to be considered when to decide what to do in the next negotiation round. Q-Learning also includes not only exploitation but also exploration facilities. In dynamic environments or in the presence of incomplete information, exploration, i.e., trying out different possibilities different from the obvious ones, becomes a powerful technique.

We also have adapted the Q-Negotiation algorithm [11][12] to the insurance brokering problem. This algorithm uses a reinforcement learning strategy based in Q-learning for the formulation of new proposals. The Q-learning algorithm is a well-known reinforcement learning algorithm that maps values (Q-values) to state/action pairs.

The Q-Negotiation algorithm has the ability to maintain information private to individual agents, and at the same time, includes the capability to evaluate multi-attribute proposals, to learn how to make better proposals during the negotiation process and to resolve attributes' inter dependencies.

Intelligent agents technology is a flexible paradigm suitable for dynamic and open environments, since agents can effectively cope with the complexity and large amount of information. Intelligent agents technology seems to be an appropriate paradigm to use in our case, since electronic commerce environments are both very dynamic and complex.

7.2 Negotiating with Insurers

Generally, an evaluation of a received proposal is a linear combination of the attributes' values weighted by their corresponding importances. In this way, a multi-attribute negotiation is simply converted in a single attribute negotiation, once there is a single result of the combined evaluation.

The multi-attribute function presented in the following formula encodes the attributes' and attributes values' preferences in a qualitative way and, at the same time, accommodates attributes intra-dependencies.

$$Evaluation = \frac{1}{Deviation} \ . \tag{3}$$

$$Deviation = \frac{1}{n} * \sum_{i=1}^{n} \frac{i}{n} * dif(PrefV_i, V_i) \ . \tag{4}$$

$$dif(PrefV_i, V_i) = \begin{cases} \frac{V_i - PrefV_i}{max_i - min_i} & \text{if continuous domain .} \\ \frac{Pos(V_i) - Pos(PrefV_i)}{nvalues} & \text{if discrete domain .} \end{cases} \tag{5}$$

where n is the number of attributes that defines a specific insurance product component.

A proposal's evaluation value is calculated by the Broker Agent, as the inverse of the weighted sum of the differences between the optimal $PrefV_i$ and the proposed value V_i of each one of the attributes. The proposal with the highest evaluation value so far is the winner, since it is the one that contains the attributes' values more closely related to the preferred ones from the customer point of view.

The function $dif(PrefV_i, V_i)$ quantifies for an issue i, the degree of acceptability of the current value V_i proposed by a specific Insurer Agent when compared to its preferable value $PrefV_i$.

If the insurance product component's values domain is of continuous values, this quantification simply is a normalised difference between the two values V_i and $PrefV_i$.

If the insurance product component's values domain is of discrete values, the difference is now calculated as a normalised difference between the preference attached to V_i and $PrefV_i$. This difference can be calculated considering the relative position of the two values in the ontology's enumerated domain values specification.

The negotiation process is considered as a set of rounds where Insurer Agents concede, from round to round, a little bit more trying to approach the customer preferences, in order to be selected as the winning insurance company. The winner bid in the current round is selected as the one that presents the highest evaluation value, since it is the solution that contains attributes values the closest to the preferable ones. The winner bid in the current round is compared with the bid in all past rounds, and the best on is selected. The Broker Agent helps Insurer Agents in their task of formulating new proposals by giving them some hints about the direction they should follow in their negotiation space. The Broker Agent gives this hints as comments about attributes' values include in current proposals.

This qualitative feedback reflects the distance between the values indicated in a specific proposal and the best one received so far, and is formulated as a qualitative comment on each of the proposal' attributes values, which can be classified in one of three categories: sufficient, bad or very_bad. The Broker Agent compares a particular proposal with, not its optimal, but the best one received so far because its more convincing to say to an Insurance Agent that there is a better proposal in the market than saying that its proposal is not the optimal one.

Insurer Agents will use this feedback information about its past proposals, in order to formulate, in the next negotiation rounds, new proposals trying to follow the hints included in the feedback comments.

The negotiation process ends when:

- Broker Agent receives a bid which has a satisfactory evaluation value. This is the winner bid.
- A deadline is reached. The winner bid is the one that presents the highest evaluation value among all bids received until then.

7.3 Interacting with the Customer Agent

When the negotiation process ends it is possible that the received bids do not satisfy all the constrains imposed by the customer Agent. In that case a conversation with customer Agent is initiated. This interaction takes the form of a sequence of questions whose aim is to reduce alternatives rather than simply sort them.

We use the schematic format (*performative sender_ agent receiver_ agent conversation_ operator content*). The outermost performative, or message type, represents the general class of a message. In our system, we make use of the classes *request*, *query* and *inform*.

According to [8], we may view a request as having an associated level of commitment. A request performative is tightly coupled with advancing the task of selecting an insurance product, as it always concern constraints. When a request is made the Broker Agent is making a pre-commitment to how the progress on the selection of an insurance product might be accomplished and it prompts the Costumer Agent for information in order to do this. The Costumer Agent must respond with an appropriate *inform* message.

A query performative is about exchanging information. Its objects of discourse are the domain ontology and relevant information to the current state of the interaction. When a query is sent by the Costumer Agent, the Broker Agent must respond with an *inform* followed by an appropriate conversation operator.

The first operator available to the Broker Agent, CONSTRAIN-ATTRIBUTE, involves asking the Customer Agent to provide a value for an attribute that does not yet have one. In some cases, the process of introducing a constrain can produce a situation in which no insurance products are satisfactory. When this occurs, the Broker Agent applies RELAX-ATTRIBUTE, which asks whether the user wants to drop a particular constrain.

Another operator, SUGGEST-VALUES, informs the Customer Agent of possible values for an attribute. In this case, the Broker Agent sends only the most adequate options rather than all possible choices. A similar operator, SUGGEST-ATTRIBUTES, informs the Customer Agent about the possible attributes for an insurance product.

Once the conversation has produced a valid number of alternatives, the Broker Agent invokes RECOMMEND-INSURANCE, an operator that proposes a complete insurance product to the Customer Agent.

Table 1. Broker Agent Conversation Operators

Performative	Conversation Operator	Description
Request	CONSTRAIN-ATTRIBUTE	Asks a question to obtain a value for an attribute
	RELAX-ATTRIBUTE	Asks a question to modify a value for an attribute
Inform	SUGGEST-VALUES	Suggests a set of possible values for an attribute
	SUGGEST-ATTRIBUTES	Suggests a set of unconstrained attributes
	RECOMMEND-INSURANCE	Recommends an insurance product that satisfies the constrains

Now let us discuss the operators available to customer Agent. The action, PROVIDE-CONSTRAIN, involves specifying the value of some attribute. This can be a value for the attribute just asked by the Broker Agent, a value for a different attribute, or a replacement for a previously specified value. If the proposed value for an attribute is found to be inappropriate by the customer Agent or less relevant than some other factor, it can reject the attribute or even replace it with another. The REJECT-SUGGESTION is used for explicit rejection.

In addition, the customer Agent can explicitly accept or reject other proposals that the Broker Agent makes, for relaxing a certain attribute (ACCEPT-RELAX or REJECT-RELAX), or for a complete insurance product (ACCEPT-INSURANCE or REJECT-INSURANCE). The customer Agent can also query the Broker Agent about available attributes (QUERY-ATTRIBUTES) or about possible values of an attribute (QUERY-VALUES).

Table 2. Customer Agent Conversation Operators

Performative	Conversation Operator	Description
Inform	PROVIDE-CONSTRAIN	Provides a value for an attribute
	REJECT-SUGGESTION	Rejects the proposed attribute
	ACCEPT-RELAX	Accepts the new value of an attribute
	REJECT-RELAX	Rejects the new value of an attribute
	ACCEPT-INSURANCE	Accepts proposed insurance product
	REJECT-INSURANCE	Rejects proposed insurance product
Query	QUERY-ATTRIBUTES	Asks broker for information about possible attributes
	QUERY-VALUES	Asks broker for information about possible values for an attribute

8 Agent Communication Using XML

8.1 Agent Communication

Autonomous agents cooperate by sending messages including concepts from an appropriate domain ontology. A standard message format with meaningful structure and semantics has become a key issue in agents understanding each other. Furthermore, the message format should be accepted not only by the agent research community, but also by all information providers. As XML is fast becoming the standard for data interchange on the Web, we choose XML as the message format for agent communication.

Agents send and receive information through XML encoded messages. We use a FIPA ACL-like format, encoded in XML. XML tags markup the information and break up the data into parts, with meaningful structure and commonly agreed semantics.

The power of XML, its role in e-commerce, and even the use of XML for agent communication, have been recognised. However, although XML is well structured for encoding semantically meaningful information, it must be based on an appropriate ontology.

Generally speaking, a domain ontology provides a set of concepts that can be asked for, advertised and used to control the agent cooperation behaviour. These concepts can be marked using XML tags, and then a set of commonly agreed tags, underlie message interpretation. The structure and semantics of the message used in a particular problem domain are represented by the corresponding DTDs.

8.2 XML Benefits in Insurance

Standardisation in information representation and transfer is crucial to both B2B and B2C e-commerce. XML is platform and application independent, and vendor-neutral mechanism. XML relies on other technologies, in particular, SGML for syntax, URIs for name identifiers, EBNF for grammar, and Unicode for character encoding, which are all standards.

The advantage of data being independent of any particular platform, application or vendor, is that it can be transformed to produce different types of outputs for different media devices (Web browser, paper, CD-ROM) without the need to modify the original content. When modifications are required, only the original version of the content need to be edited before republishing to the various target media. This leads to efficiency and ease-of-maintainability, without the inherent problems of version control and the effort required in making modifications in medium-specific document versions.

Conducting e-commerce requires communicating with other companies and often poses a challenge. XML simplifies business-to-business communication, particularly in vertical industries for the following reasons: (1) The only thing that is to be mutually agreed upon is the XML vocabulary that will be used to represent data. (2) Neither company has to know how the other's back-end systems (platforms, operating systems, programming languages) are organised, which does not put any extra technical burden while keeping the privacy. All that is required is that each company develop the mapping to transform XML documents into the internal format used by the back-end systems. (3) XML-based solution is scalable: If there is an addition of another partner, there is no need by the host company to interact with the systems of the new company. All that is required is that they follow the protocol (the XML vocabulary).

A major advantage of conducting business on the Web is that it broaden the customer-base towards globalisation, without the necessity of having physical office locations. However, in order to communicate, a business must still "speak" the language of the region in context. With the Unicode support in XML, Web sites can be multi-lingual. XML also includes a method to signal what language and encoding is being used.

XML provides companies opportunities for customer services that did not exist previously. Corporate data that was previously stored in disparate sources and considered to be non-integrable, can be transformed in an XML format. By consolidating different data sources, opens the doors for the companies to make a variety of such data available to be explored. It gives insurance companies a powerful way to transact, manage and share data over the Web.

XML is far less expensive than other data exchange alternatives, such as Electronic Data Interchange (EDI), opening the door for small, low-tech companies to participate in online data exchange.

9 Knowledge Representation

Sharing common understanding of the structure of information among people or software agents is one of the more useful and common goals in developing ontologies [6]. Adopting a common ontology guarantees information consistency and compatibility for a community of agents. The information consistency is satisfied when each specific expression has the same meaning for every agent in the market. The information compatibility is verified when any concept is described by the same expression, for all the agents. This is why knowledge representation becomes an important issue in the context of agent-based insurance brokering as well.

Fundamentally we need a common descriptive language that all insurers can agree on. These include the terms, conditions, relationships and categorisations of insurance products. Such definition must be extensible in that it can support new terms and relationships being added later. Given a standard definition language, there needs to be agreement on how to define an insurance product so that it can be searched, displayed and its terms negotiated over. These include defining attributes, constraints, eligibility criteria, preferences and negotiable aspects of the product.

In our system a class is described by a set of slots, and slots are described by a set of attributes, which are instantiated with values. The schema we are using for an ontology is then the knowledge representation scheme suitable for properly identify classes, slots, attributes and values, together with relationships that map classes to slots, slots to attributes and attributes to values. Such ontology can be represented by the following structure:

$$Ont = \{Class, Slot, Attr, Val, CS_r, SA_r, AV_r, Deps\}$$

where $Class$ is the set of item's identifiers, $Slot$ is the set of component's identifiers, $Attr$ is the set of attributes' identifiers, Val is the set of attribute values' identifiers, IC_r is a relationship that assigns to each class in $Class$ a set of slots in $Slot$, AV_r is a relationship that assigns to each attribute in $Attr$ a specific value in Val and $Deps$ is a set of relationships defining the dependencies between attributes' values.

Each value is represented by the tuple $Val_i = \{Type, Domain\}$ where $Type = \{integer, real, string\}$ and $Domain = \{continuous, discrete\}$.

Each relationship CS_r is represented by $Class_i \rightarrow \{Slot\}, \forall_{Class_i} \in Class$.

Each relationship SA_r is represented by $Slot_i \rightarrow \{Attr\}, \forall_{Slot_i} \in Slot$.

Each relationship AV_r is represented by $Attr_i \rightarrow Val_k, \forall Attr_i \in Attr, \exists^1 Val_k \in Val$.

Each dependency in $Deps$ is represented by $Dep_{ij} = f(Val_{ki}, Val_{mj}), \forall Attr_i, Attr_j \in Attr$.

10 Related Work

In [17] a system was proposed to support a market place for complex products and services such as insurance. Such an electronic market place is represented by a network of match-maker objects or traders, which perform a symmetric match-making between product features and customer's requirements. Each tier of traders is responsible for determining the suitability and availability of products following each successive stage of the dialogue between insurers and their customers.

This paper proposes a distributed multi-agent system where customers are grouped together, exploiting user modelling and machine learning techniques. Although clustering seems a computationally expensive task compared to the case-based approach, this is not a real problem since communities change far less often than individual user models. In the other hand communities can be very useful since they can be used to tailor products offering to the needs of customers. However, this can be done effectively, only when the generated communities are meaningful. That's why our system is using a specific metric to characterise the generated communities of users.

We propose a common format for describing products and requirements that can be agreed upon by all participants and that is rich enough to incorporate all the current and future functionality of the system. Thus, if the product definition ontology is agreed, then it becomes relatively easy for submit new products at any time or to make changes in existing product terms and conditions and make these changes automatically and instantly.

Several other researchers have also addressed learning algorithms for agents in electronic commerce negotiation. Reinforcement learning algorithms have been used in the work reported in [13], where simulation results showed that learning agents performed better than their non-learning competitors agents in different market situations. However the algorithm only addresses single-issue type of negotiations. Learning in multi-issue negotiation is addressed in [21], where negotiation occurs in a one-to-one basis. Our work combines both one-to-many and multi-issue negotiation characteristics. Learning is an important characteristic that should be available in automated e-commerce negotiation. Past proposals can, and should, constraint the value of the next insurer proposal. Moreover, in open systems, as it is the case of e-commerce, learning becomes a powerful capability for deal with the environment dynamics. Therefore, we consider both multi-issue and adaptive negotiation features as being of great importance for agent-mediated electronic commerce.

11 Conclusions

The system described in this paper presents a new approach to insurance products brokering and has the potential to improve the quality of customer service by ensuring that individual customer needs are reflected in the products offered.

Different communities of users can be identified and used to improve the exploitation of an insurance brokering service. The construction of those communities is achieved using an unsupervised learning technique. We also use a specific metric to decide which are the representative preferences of a user's community.

The proposed ontology includes multi-attribute definition as well as attributes' intra and inter dependencies. This scheme is suitable for properly identify items, components, attributes and values of a insurance product, together with relations that map items to components, components to attributes and attributes to values.

We also have adapted an advanced negotiation protocol, suitable for multi-issue negotiation in electronic commerce activity. A learning capability was also included enabling agents to become more effective in a dynamic market by learning with past experience through the qualitative feedback received from their opponents. Our platform for automatic insurance brokering services is now ready to be exploited for future experiments in several realistic scenarios.

References

1. Marko Balabanovic and Yoav Shoham. Learning information retrieval agents: Experiments with automated web browsing. In *Proceedings of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Resources*, pages 13–18, 1995.
2. P. Chiu. Using c4.5 as an induction engine for agent modelling: An experiment for optimisation. In *Proceedings of the User Modelling Conference UM'97*, 1997.

3. D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.
4. J.B. Weinberg G. Biswas and D. Fisher. Iterate: A conceptual clustering algorithm for data mining. *IEEE Transactions on Systems, Man and Cybernetics*, 28:100–111, 1998.
5. M.A. Gluck and J.E. Corter. Information, uncertainty and the utility of categories. In *Proceedings of the 7th Conference of the Cognitive Science Society*, pages 283–287, 1985.
6. T.R. Gubber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
7. R. Guttman, A. Moukas, and P. Maes. Agent-mediated electronic commerce: A survey. *Knowledge Engineering Review*, 13(2):147–159, 1998.
8. A. Haddadi. Communication and cooperation in agent systems: A pragmatic theory. *Lecture Notes in Computer Science*, 1056, 1996.
9. J. Robert Hunter and James H. Hunt. *Term Life Insurance on the Internet: An Evaluation on On-line Quotes.* Consumer Federation of America, http://consumerfed.org/termlifeinsurance.pdf, 2001.
10. J. Moore. Implementation, contracts and negotiation in environments with complete information. *Advances in Economic Theory*, 1, 1992.
11. Eugénio Oliveira and Ana Paula Rocha. Agents advanced features for negotiation in electronic commerce and virtual organisations formation process. *Lecture Notes in Computer Science*, 1991:77–96, 2001.
12. Eugénio Oliveira and Ana Paula Rocha. Electronic institutions as a framework for agent's negotiation and mutual commitment. *Lecture Notes in Artificial Intelligence*, 2258:232–245, 2001.
13. Eugénio Oliveira, José M. Fonseca, and Nick Jennings. Learning to be competitive in the market. In *American Association of Artificial Inteliggence Workshop on Negotiation*, Orlando, USA, July 1999.
14. G. Paliouras, V. Karkaletsis, C. Papatheodorou, and C. Spyropoulos. Exploiting learning techniques for the acquisition of user stereotypes and communities. In *Proceedings of the International Conference on User Modelling (UM '99).*, 1999.
15. G. Paliouras, C. Papatheodorou, V. Kakaletsis, C. Spryropoulos, and V. Malaveta. Learning user communities for improving the services of information providers. *Lecture Notes in Computer Science*, 1513:367–384, 1998.
16. B. Raskutti and A. Beitz. Acquiring user preferences for information filtering in interactive multimedia services. In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence*, pages 47–58, 1996.
17. Y. Hoffner S. Field. Vimp - a virtual market place for insurance products. *EM - Electronic Markets*, 8(4):3–7, 1998.
18. M. Stolze. Soft navigation in electronic product catalogs. *International Journal in Digital Libraries*, 3(1):60–66, 2000.
19. Nir Vulkan and Nick Jennings. Efficient mechanisms for the supply of services in multi-agent environments. In *Proceedings of the First International Conference on Information and Computation Economics*, Charleston, South Carolina, October 1998.
20. M. Wellman and P. Wurman. Market-aware agents for a multiagent world. *Robotics and Autonomous Systems Journal*, 24:115–125, 1998.
21. Dajun Zeng and Katia Sycara. How can an agent learn to negotiate. In *Proceedings of the Third Internation Workshop on Agent Theories, Architectures and Languages*, Budapest, Hungary, August 1996.