Ontology Acquisition and Exchange of Evolutionary Product-brokering Agents

Steven Guan and Fangming Zhu

Department of Electrical and Computer Engineering
National University of Singapore
10 Kent Ridge Crescent, Singapore 119260
eleguans@nus.edu.sg

Agent-based electronic commerce (e-commerce) has been booming with the development of the Internet and agent technologies. However, little effort has been devoted to exploring the learning and evolving capabilities of software agents. This paper addresses issues of evolving software agents in e-commerce applications. An agent structure with evolution features is proposed with a focus on internal hierarchical knowledge. We argue that knowledge base of agents should be the cornerstone for their evolution capabilities, and agents can enhance their knowledge bases by exchanging knowledge with other agents. In this paper, product ontology is chosen as an instance of knowledge base. We propose a new approach to facilitate ontology exchange among e-commerce agents. The ontology exchange model and its formalities are elaborated. Product-brokering agents have been designed and implemented, which accomplish the ontology exchange process from request to integration.

Keywords: Software agents, ontology exchange, e-commerce, product brokering

1. INTRODUCTION

Intelligent agents are already on the Web, freeing us from some drudgework of searching and keeping us up to date automatically. For example, software agents may help users sift through the mass of data and make intelligent decisions. However, applications of software agents in e-commerce are just burgeoning. In the recent decade, agent-based e-commerce (Dignum and Cortés, 2001; Glushko, Tenenbaum and Meltzer, 1999) has emerged and attracted many efforts in academic and industrial fields. The motivation of introducing software agents into e-commerce is to overcome the arising barricades such as overload of information, difficulty in searching, lack of negotiation infrastructure, etc. Software agents have demonstrated tremendous potential in conducting various e-commerce activities, such as comparison-shopping, negotiation, payment, and auction (Guttman and Maes, 1999; Krishna and Ramesh, 1998). However, these novel e-commerce applications also bring up some new technical challenges to the agent technology, such as security, authentication, and privacy. Much research work has concentrated on these issues (Corradi, Montanari and Stefanelli 1999; Greenberg, Byington, and Harper, 1998), as they are essential in e-commerce applications.

Manuscript received: 8 October 2002
Communicating Editor: Denis Warne

The academia has not reached a generally accepted definition for software agents. In general, software 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. Agents differ from traditional software in that they are personalised and autonomous. They can be personalised to the end-users' preferences. Furthermore, they are adaptive and can learn from past experiences. In addition, in a multi-agent environment, agents can interact with each other. Therefore, coordination, cooperation, and communication have become the most important external properties for agents. In this paper, we explore the evolutionary features of software agents based on these basic properties.

Mobility is another exciting feature for agents, especially with the development of the Internet. Mobile agents can move from one machine to another, across different platforms or architectures. Adding mobility to software agents will improve the potential of their applications in e-commerce. With mobility, agents can now move from one e-commerce service provider (ESP) to another, and carry on their execution from where they were left off in the previous ESP. In this way, mobile agents can not only retrieve information or negotiate prices from one ESP, but also they can compare the prices from the other ESPs before deciding for the end-user. However, to take advantage of this feature, interoperability of agents across different platforms must be ensured.

When agents are initially created, they have little knowledge and experience. Although agent owners may give some basic knowledge or functionality to these agents, it is advantageous if they have the ability to learn and evolve. Furthermore, the Web environment also changes rapidly and agents should strive to adapt themselves in order to complete their tasks successfully.

Many issues are essential in agent evolution. First, evolution of agents depends on agents' knowledge and structure. Thus, a suitable agent structure for knowledge acquisition is the basic concern in agent evolution. Second, agents should have their own mechanisms to advance evolution. How to design strong and adaptive evolution mechanisms is another issue to be pursued. Third, in multi-agent systems, evolution of agents is also related with many issues, such as coordination, relationship, topology, and communication. Finally, agent owners should closely interact with the evolutionary process of their agents (Zhu and Guan, 2001).

Agents are both self-interested and social. They have their own goals, but they also seek for collaboration. The interaction with other agents should in some way or another help each individual agent to fulfill one or more of its goals. The motivation for collaboration can arise for the purpose of one temporary task, e.g. information retrieval, or for a long-term objective, e.g. co-evolution.

We exploit agent evolution in the knowledge level in this paper, as we deem that knowledge base is the cornerstone of all emergent behaviours of agents. Although agents may gain new knowledge from self-reasoning, we concentrate on knowledge exchange among agents as we argue that this is the most applicable and efficient way to acquire knowledge in a trustworthy multi-agent environment. We use product ontology as a typical instance of knowledge to test our design on evolutionary agents.

The paper is organised as follows. We first introduce some background on agent-based e-commerce, agent evolution, and ontology in Section 2. Then, an agent structure with evolutionary features is introduced in Section 3. Section 4 presents the basic definitions of agent ontology. Section 5 builds a simple ontology exchange model and elaborates the ontology exchange formalities from request to integration. Section 6 presents the implementation work and Section 7 concludes the paper.

2. BACKGROUND

Although software agents have been under development since the last decade, only with recent advances in agent technologies, have their potential and capabilities been greatly enhanced. A number of intelligent agent-based systems have been developed for purposes of e-commerce. MIT

Media Lab's Kasbah (Chavez and Maes, 1998) is an online marketplace for buying and selling goods. A user can create a buyer agent, provide it with a set of criteria, and dispatch it into the marketplace. The Minnesota AGent Marketplace Architecture (MAGMA) (Tsvetovatyy, Mobasher, Gini and Wieckowski, 1997) is a prototype for a virtual marketplace targeted toward items that can be transferred over the Internet. Agents can register with a server that maintains unique identifiers for the agents. AuctionBot (Wurman, Wellman and Walsh, 1998) is a flexible, scalable, and robust auction server which supports software agents. In our previous work, we have proposed a SAFER (Secure Agent Fabrication, Evolution and Roaming) architecture, which aims to construct an open, dynamic and evolutionary agent system for e-commerce (Zhu, Guan and Yang, 2000). It provides a framework for agents in e-commerce and establishes a rich set of mechanisms to manage and secure them. We have already elaborated agent fabrication and roaming in Guan, Zhu and Ko (2000) and Guan and Yang (1999) respectively. Agent evolution is an integrated part of the SAFER architecture. We have proposed a model for agent life cycle and some evolutionary computation methods to evolve agents (Zhu and Guan, 2001).

Literature in DAI (Distributed Artificial Intelligence) and MAS (Multi-Agent System) has addressed agent evolution extensively. Haynes and Wainwright (1995) aim to evolve programs to control an autonomous agent capable of learning how to survive in a hostile environment. Namatame and Sasaki (1998) provide a model for investigating collective behaviours that emerge from local interaction among self-interested agents. Most of the existing research work on e-commerce agents focuses on evolving strategies for agents, extending the MAS approaches. Gimenez-Funes *et al* (1998) use possibility-based and case-based decision models to design the bidding strategies for agents in electronic auctions. Ritcher, Sheble and Ashlock (1999) develop bidding strategies which are used for electric utilities in the scenario of double auctions.

Sheth and Maes (1993) design a population of personalised information filtering agents. These agents will recommend news articles to the user based on the user's interest. Users will give each recommended article a certain score. A relevant article will receive a higher score than an irrelevant one. After a certain number of iterations, the more successful agent will be retained and be allowed to reproduce while the unfit ones are killed. Yu, Koo and Liddy (2000) describe a neuro-genetic approach to develop an MAS which forages as well as meta-searches for multi-media information in online information sources.

There are few research efforts dedicated to agent evolution in e-commerce applications from the perspective of agent interaction, especially from the viewpoints of ontology exchange. However, it is quite usual that agents are interacting with other agents, regardless of whether they have the same or different objectives and structures. Our research considers the evolutionary process in an agent group with a focus on ontology exchange.

In order to coordinate heterogeneous agents effectively across distributed network and facilitate communication between agents, some Agent Communication Language (ACL) and knowledge representation language have been developed, such as FIPA ACL (FIPA, 2003) and KQML (Knowledge Query and Manipulation Language) (Finin *et al*, 1994). KQML has become the de facto standard and is widely used in various agent-based systems. KQML provides an extensible set of primitives which defines the permissible operations that agents may attempt on each other's knowledge. It also allows agents to communicate views regarding the content and find agents relevant to process their requests.

An ontology is an explicit specification of a conceptualisation. The term 'ontology' is borrowed from philosophy, where an ontology is a systematic account of existence. In the context of artificial intelligence, ontology is referred to as a special kind of knowledge. Much research work has been

dedicated to explore ontology-based knowledge management, which includes (1) ontological engineering to build an ontology of the subject matter, (2) characterising knowledge in terms of ontology, and (3) providing intelligent access to knowledge. Huhns *et al* discuss the requirements of ontologies for software agents (Huhns and Singh, 1997; Huhns and Stephens 1999). Luke *et al* (1997) describe a set of HTML ontology extensions for web agents to annotate web pages with semantic knowledge, and Mahalingam and Huhns (1997) describe how query formation can be made simple and less complicated by using ontologies. Our paper focuses on the product ontology for product-brokering agents, and we try to demonstrate that agents can build efficient product ontologies via collaboration and exchange with other fellow agents.

3. AGENT STRUCTURE WITH EVOLUTIONARY FEATURES

As mentioned earlier, agent structure is a basic concern in agent evolution. Figure 1 shows an agent structure with evolutionary features which include some hierarchical knowledge structure and the evolutionary modules.

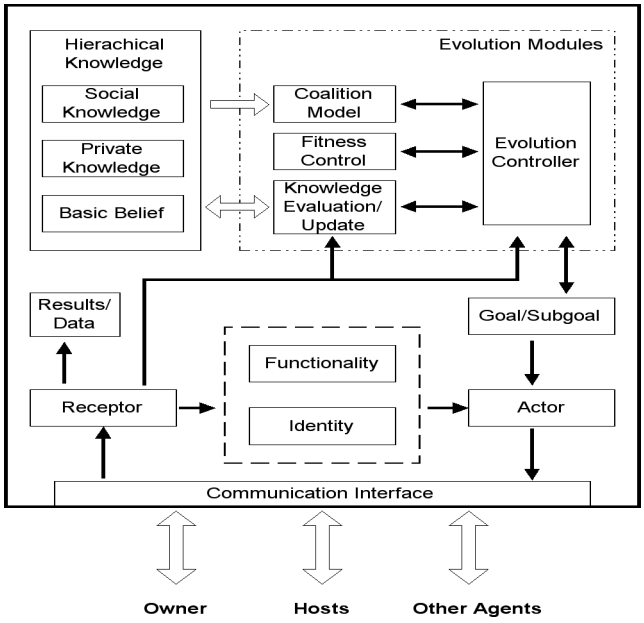

Figure 1: An agent structure with evolutionary features

The communication interface is responsible for communication with the owner, hosts, and other agents. Each agent is also equipped with a Receptor and an Actor. The Receptor receives inputs from the outside world and passes information to the functionality modules or saves it in the result/data modules, while the Actor enforces decisions made by the agent. The identity module contains basic elements regarding the identity of an agent, such as agent ID, certificate, timestamp, and agent-digest. The functionality modules provide standard functions for agent actions and specific functions for individual purposes.

The hierarchical knowledge base stores the agent knowledge base for actions of reasoning, analysis, and decision-making. It is divided into three layers, from basic beliefs to private knowledge then social knowledge. Basic beliefs can be abstracted and integrated into private

knowledge. Typical social knowledge includes agent relationship, topology of merchant hosts and other agents.

The evolutionary modules include evolution controller, coalition model, fitness control, and knowledge evaluation/update. They are responsible for evolution management. Apart from the goals which are initially set by the agent owner, evolution controller can generate new subgoals or adjust goals according to the process of evolution.

Similarly to natural ecosystems, agent evolution is not an isolated process. Agents coexist in a shared environment and pursue their respective goals in the presence of other agents. A collection of agents interacting with each other can be organised as an agent group. We regard agent group as the fundamental unit for the evolution process. Users can organise agent groups in terms of their preferences or let agents decide which groups they are willing to join. Thus, the criteria for group formation can be varied, and agents in one group can be homogeneous or heterogeneous.

4. AGENT ONTOLOGY

According to the nomenclature of Maes' group in the MIT Media Lab (Guttman and Maes, 1999), the common commerce behavior can be described with the Consumer Buying Behaviour (CBB) model, which consists of six stages, namely, need identification, product brokering, merchant brokering, negotiation, purchase and delivery, and product service and evaluation. There are various types of agents existing for different stages. Therefore, the knowledge base for each type of agent is different.

The most widely known e-commerce agents are product- brokering agents, which are also called comparison-shopping agents or shopping-bots. The main function of these agents is to help users find their favorite products with certain constraints such as price, quality, and delivery time. Such agents will first assist users to construct their queries with current knowledge about product definitions, which is called ontology here. Then they will be able to locate thousands of merchants carrying specific products or services, and scour through millions of items. Finally they may sort the results or offers and present them to users.

There are two main bottlenecks for deploying agents in such applications. One is that most Internet stores do not provide facilities to support or serve visiting agents. Either the Internet stores do not allow agents to access product information, or agents do not have such capability to acquire information from the stores. Another point is that agents may return with irrelevant or overwhelming information, as they cannot filter out useful information due to lack of proper knowledge of certain products. This paper tackles the second case with a focus on product ontology.

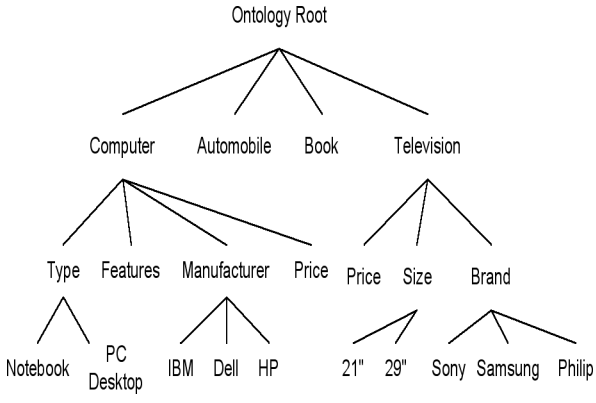

Figure 2: A typical product ontology

A product ontology is a conceptualisation of the real-world products. It consists of a specification of concepts to be used for expressing knowledge, including the types and classes of entities, the kinds of attributes and properties they can have, the relationships and functions they can participate, and constraints that hold. A typical ontology for a product-brokering agent is shown in Figure 2 with a suitable tree structure for representation.

An agent aims to carry sufficient and useful ontology information within its body. There are three approaches for an agent to build its product ontology. The first choice is to approach some ontology definition provider to gain a suitable ontology definition, this may incur time and cost. The second choice is to enhance its ontology by analysing the data collected online, but it highly relies on the agent capability and intelligence, and may also need some interaction from its owner. The last choice is to interact with other agents for ontology acquisition or exchange, and this is the focus of this paper.

Ontology rules are a basic representation format for agent ontology and are suitable to be exchanged among agents. Actually, ontology rules can be inferred from the ontology tree. Here we give some definitions for ontology rules. Table 1 lists some examples of ontology rules.

<Ontology Rule>::= <Subject> <Relationship> <Object>
<Subject>::=<Entity_Grp>
<Entity_Grp>::=<Entity><Entity>…
<Entity>::=<string>
<Relationship>::=<Rlt-Element>
<Rlt-Element>::=<string>
<Object>::=<Entitity_Grp> | <Attr_Grp>
<Attr_Grp>::=<Attribute> <Attribute>…
<Attribute>::=<string>

Rules	Entities	Relationship Element	Entities/Attributes
1	Convertible	is a type of	car
2	Car	has attributes of	engine, color
3	Color	has types of	red, white, black, blue

Table 1: Some examples of ontology rules

Ontology is an essential knowledge base for software agents. For the above-mentioned product-brokering agents, product ontology can help query reformulation with several factors:

1. Sharpen a user request by query expansion along the downward direction of the ontology tree.
2. Widen a query scope context along the upward direction of the ontology tree.
3. Disambiguate a user request by finding the context relevant to the request.
4. Reformulate a query within the same context or even by going from one context to a different context that fits better with respect to the actual wishes of the user.

5. ONTOLOGY EXCHANGE MODEL AND FORMALITIES

We assume that agents are egoistic, but they are willing to collaborate with each other to achieve their objectives. Ontology exchange is a practical way for agents to enhance and accumulate their ontologies. There are three basic ontology exchange models, namely, unilateral, bilateral, and multilateral, based on the number of agents involved. Here, a model of ontology exchange between one agent

and several agents is shown in Figure 3. In the figure, agent S contacts agents A_1 to A_n for certain ontology definitions. First, agent S sends a request to other agents, encapsulating some existing ontology terms/rules in the request. Then, each corresponding agent may respond with some ontology definitions. Upon receiving responses, agent S will determine whether and how they are related with the existing ontology and which ontology pieces will be integrated.

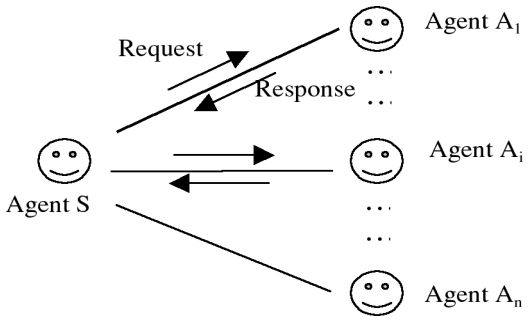

Figure 3: An agent contacting other agents for ontology definitions

Here, we only consider the terms in the ontology, as we deem that the ontologies acquired from respondents are trustworthy. Assuming that:

O_s denotes the ontology of agent S;
O_i denotes the ontology response from agent A_i;
T_r denotes the set of terms included in the request;
T_s denotes the set of terms which are currently in O_s;
T_i denotes the set of terms included in O_i

Define $_{Sim}[O_s,O_i] = _{Sim}\lfloor T_s,T_i \rfloor = N_e / N_s$ to be the similarity between two ontologies O_s and O_i, which means that the similarity of ontology is measured by the term set in the corresponding ontologies. In this formula, N_e represents the number of terms which are the same or synonymous in both ontologies O_s and O_i, and N_s represents the total number of terms in O_s.

With the above definitions, agents can determine the criteria for ontology exchange and integration. For example, when agent S is deciding which ontology piece from a respondent should be integrated, it can use a criterion such as choosing the one which has the maximum similarity with the existing ontology. If agent S decides to integrate more than one piece of ontology, then it can choose those similarity values with the highest values.

In order to facilitate ontology exchange, communication channels should be established. There exist some standard Agent Communication Languages (ACLs) for this purpose. KQML (Finin *et al*, 1994) is the most widely used ACL. KQML provides an extensible set of performatives, which defines the permissible operations between agents. The reserved performative categories include basic informative, response, query, networking, capability-definition, etc. The following is a typical message in KQML which embeds ontology rules:

```
( tell  :sender        Agent A1
        :receiver      Agent S
        :language      Specified
        :ontology      Product
        :content       "Convertible is a type of car"
)
```

Figure 4 shows the flowchart of the ontology exchange process in an individual agent. The KQML messages coming from some other agents are received, and ontology rules in KQML messages are extracted and parsed. The resultant ontology elements will be fused into the ontology database. As for the reverse direction, some ontology rule elements are selected from the ontology database, constructed into a rule and embedded into a KQML message, and then sent to other agents. Other functional modules of agents include a representation to construct an ontology tree from the ontology database and an initiator to raise requests or enquiries.

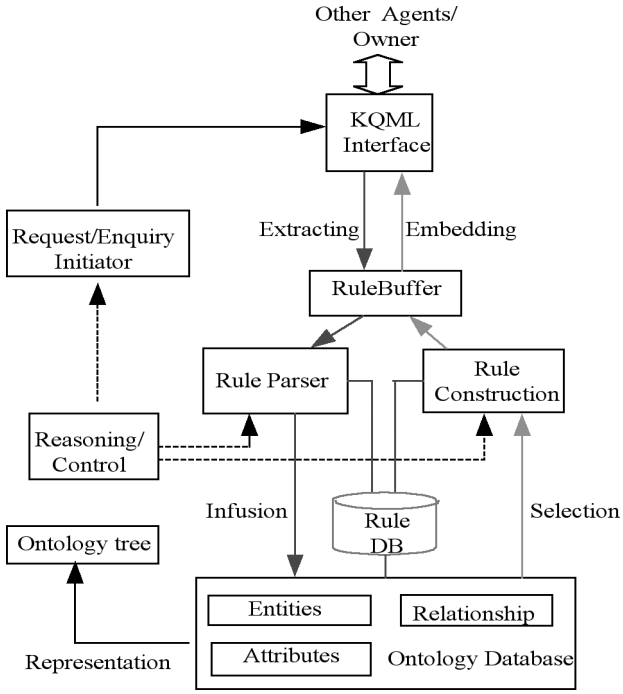

Figure 4: Ontology exchange process

After an agent receives ontology pieces from other agents, it will use some algorithms to integrate them with its own ontology. As shown in Figure 5, an agent is integrating the incoming ontology (O_i) into its own ontology (O_s). First, the agent will check the database of synonyms and hyponyms to establish the relationship map between two ontologies. If some facts of the relationship contradict each other, the part in contradiction will be rejected or placed into a temporary set for future settlement. The compatible parts will be inserted into certain position of the old ontology. As indicated in Figure 5, the agent finds that A_{22} and B_{11} are synonyms, so B_{11} is merged with A_{22} in the resultant ontology, and O*i* is inserted as the child nodes under A_{11}. The database of synonyms and hyponyms may be updated after each integration process.

6. IMPLEMENTATION

We have implemented a group of product-brokering agents, and the formalities of ontology exchange from request to integration have been successfully implemented. Users can set up their queries and dispatch agents to search for product information. Java is chosen as the language for implementation, because it has important features including robustness, security, and portability.

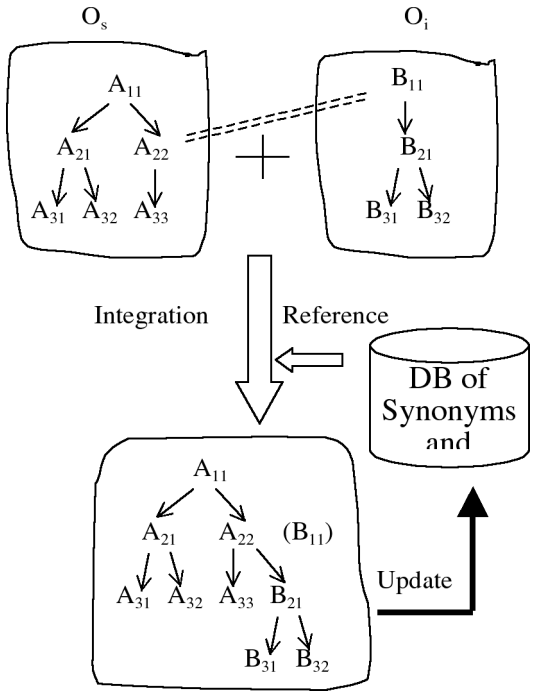

Figure 5: Ontology integration

The implemented agent uses a tree structure to display ontology, and provides an interface for a user to view and edit the ontology tree, which includes the operations of adding a node, deleting a node, saving the revised ontology, as shown in Figure 6.

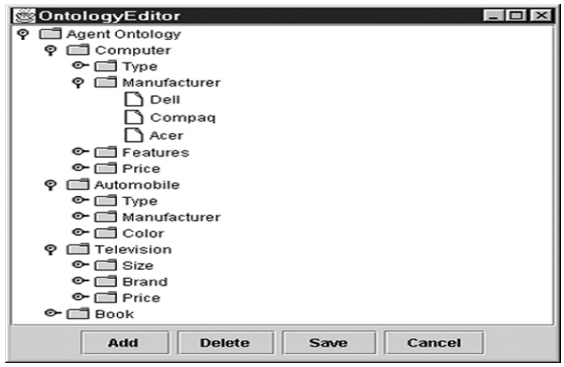

Figure 6: A screenshot of interface for viewing and editing ontology

Figure 7 illustrates the process of integrating ontology. The top left window shows the original ontology of the agent, and the top right window shows the ontology incoming from the other agent as the exchanged component. The resultant ontology is shown in the bottom window, which shows that the agent has merged the acquired ontology definitions about computer. Actually, the whole process is completed by the agent automatically.

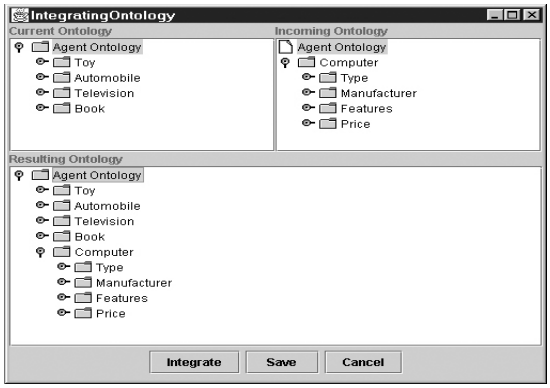

Figure 7: A screenshot of integrating ontology

We have also implemented the query construction process based on product ontology, as shown in Figure 8. A user specifies his queries with the aid of ontology. The interface includes three components: ontology helper, constraint setting, and query window. The user can click the ontology tree to choose query terms, and the corresponding terms and available selection choices will appear in the constraint setting part. Then the user can make selection from the available set or construct new constraints. At last, these constraints can be combined to form a query, which is displayed in the query window.

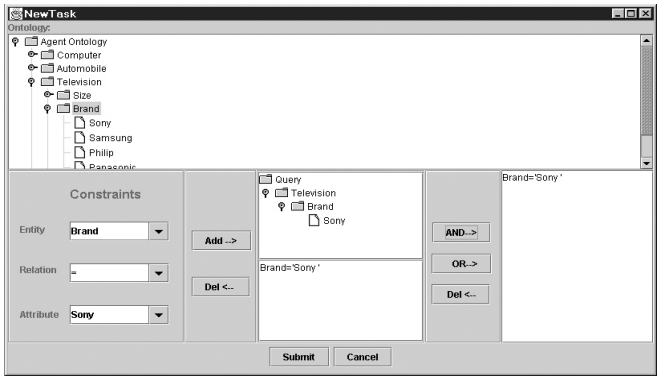

Figure 8: A screenshot of query construction with the aid of ontology

The resultant query is also visually represented as a tree like the ontology. Only terms specified with constraints will be preserved in the query tree, and the constraints will be added into the ontology tree as leaf nodes. Users can edit the query tree freely before submission. We simulate some databases for agents to search for product information. Figure 9 shows an instance of searching results with the query created in Figure 8.

7. CONCLUSIONS

This paper addresses issues of evolving software agents in e-commerce applications with a focus on ontology acquisition and exchange. An evolutionary agent structure with hierarchical knowledge base is proposed to facilitate the evolutionary process. We propose a new approach for ontology

Figure 9: A screenshot showing searching results

exchange among e-commerce agents. The ontology exchange model and its formalities have been elaborated. Product-brokering agents have been designed and implemented. They accomplish the ontology exchange process from request to integration.

We are improving our approaches and implementation in two aspects. Firstly, as checking the relevance of two ontologies and identifying the integration points are much more complicated than discussed in the paper, we are designing further approaches to outfit agents with higher intelligence and reasoning capability on these issues. Secondly, an enhanced message format embedded with XML (eXtensible Markup Language) will be defined to add meaningful structure and semantics into ontology rules.

REFERENCES

CHAVEZ, A. and MAES, P. (1998): Kasbah: an agent marketplace for buying and selling goods. *Proc. of First International Conference on Practical Application of Intelligent Agents and Multi-Agent Technology*, London, 75–90.

CORRADI, A., MONTANARI, R. and STEFANELLI, C. (1999): Mobile agents integrity in e-commerce applications. *Proc. of 19th IEEE International Conference on Distributed Computing Systems*.

DIGNUM F. and CORTÉS, U. (eds) (2001): Agent-mediated electronic commerce III: current issues in agent-based electronic commerce systems. Lecture notes in artificial intelligence 2003. Berlin, Springer.

FININ, T., FRITZSON, R., MCKAY, D. and MCENTIRE, R. (1994): KQML as an agent communication language. *Proc. of the Third International Conference on Information and Knowledge Management*, 456–463.

FIPA (2003): Foundation of Intelligent Physical Agents, http://www.fipa.org, accessed 29 July 2003.

GIMENEZ-FUNES, E., LGODO, L., RODRIQUEZ-AGUILAR, J.A. and GARCIA-CALVES, P. (1998): Designing bidding strategies for trading agents in electronic auctions. *Proc. of International Conference on Multi Agent Systems*, 136–143.

GLUSHKO, R.J., TENENBAUM, J.M. and MELTZER, B. (1999): An XML framework for agent-based e-commerce. *Communications of the ACM* 42(3): 106–114.

GREENBERG, M.S., BYINGTON, J.C. and HARPER, D.G. (1998): Mobile agents and security. *IEEE Communications Magazine* 36(7): 76-85.

GUAN, S.U. and YANG, Y. (1999): SAFE: secure roaming agent for e-commerce. *Proc. the 26th International Conference on Computers & Industrial Engineering*, Melbourne, Australia, 33–37.

GUAN, S.U., ZHU, F.M. and KO, C.C. (2000): Agent fabrication and authorization in agent-based electronic commerce. *Proc. of International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce*, Wollongong, Australia, 528–534.

GUTTMAN, R.H. and MAES, P. (1999): Agent-mediated negotiation for retail electronic commerce. *In Agent-Mediated Electronic Commerce: First International Workshop on Agent Mediated Electronic Trading*, NORIEGA, P. and SIERRA, C. (eds). Berlin: Springer, 70–90.

HAYNES, T.D. and WAINWRIGHT, R.L. (1995): A simulation of adaptive agents in a hostile environment. *Proc. of the 1995 ACM Symposium on Applied Computing*, ACM Press.

HUHNS, M.N. and SINGH, M.P. (1997): Ontologies for agents. *IEEE Internet Computing* 1(6): 81–83.

HUHNS, M.N. and STEPHENS, L.M. (1999): Personal ontologies. *IEEE Internet Computing* 3(5): 85–87.

KRISHNA, V. and RAMESH, V.C. (1998): Intelligent agents for negotiation in market games, part1: model. *IEEE Trans. on Power Systems* 13(3).

LUKE, S., SPECTOR, L., RAGER, D. and HENDLER. J. (1997): Ontology-based Web agents. *Proc. of the 1st International Conference on Autonomous Agents*, 59–66.

MAHALINGAM, K. and HUHNS, M.N. (1997): An ontology tool for query formulation in an agent-based context. *Proc. of the Second IFCIS International Conference on Cooperative Information Systems*, 170–178.

NAMATAME, A. and SASAKI, T. (1998): Competitive evolution in a society of self-interested agents. *Proc. of IEEE World Congress on Computational Intelligence*.

RICHTER, C.W., SHEBLE, G.B. and ASHLOCK, D. (1999): Comprehensive bidding strategies with genetic programming/finite state automata. *IEEE Trans. on Power Systems* 14(4).

SHETH, B. and MAES, P. (1993): Evolving agents for personalized information filtering. *Proc. of the Ninth Conference on Artificial Intelligence for Applications*, 345–352.

TSVETOVATYY, M., MOBASHER, B., GINI, M. and WIECKOWSKI, Z. (1997): MAGMA: an agent based virtual market for electronic commerce. *Applied Artificial Intelligence* 11(6): 501–524.

WURMAN, P.R., WELLMAN, M.P. and WALSH, W.E. (1998): The Michigan Internet AuctionBot: a configurable auction server for human and software agents. *Proc. of the Second International Conference on Autonomous Agents*, Minneapolis, USA, 301–308.

YU, E.S., KOO, P.C. and LIDDY, E.D. (2000): Evolving intelligent text-based agents. *Proc. of the Fourth International Conference on Autonomous Agents*, 388–395.

ZHU, F.M. and GUAN, S.U. (2001): Towards evolution of software agents in electronic commerce. *Proc. of the Congress on Evolutionary Computation 2001*, 1303–1308, Seoul, Korea.

ZHU, F.M., GUAN, S.U. and YANG, Y. (2000): SAFER E-Commerce: Secure Agent Fabrication, Evolution & Roaming for e-commerce. In *Internet Commerce and Software Agents: Cases, Technologies and Opportunities*, RAHMAN, S.M. and BIGNALL, R.J. (eds). Idea Group, PA, 190–206.

BIOGRAPHICAL NOTES

Steven Guan

Steven Guan received his MSc and PhD from the University of North Carolina at Chapel Hill. He is currently with the Electrical and Computer Engineering Department at National University of Singapore. Professor Guan has worked in a prestigious R&D organisation for several years, serving as a design engineer, project leader, and manager. He has also served as a member on the R.O.C. Information and Communication National Standard Draft Committee. After leaving the industry, he joined Yuan-Ze University in Taiwan for three and half years. He served as deputy director for the Computing Center, and also as the chairman for the Department of Information and Communication Technology. Later he joined La Trobe University in Australia with the Department of Computer Science and Computer Engineering where he helped to create a new Multimedia Systems stream.

Fangming Zhu

Fangming Zhu received his BSc and MEng degrees from Shanghai Jiaotong University, China, in 1994 and 1997 respectively. He is now a PhD candidate in the Department of Electrical and Computer Engineering at National University of Singapore. His current research interests include evolutionary computation, pattern classification, and intelligent agents.