

Fuzzy Ontology Based System for Product Management and Recommendation

James N.K. Liu

Abstract—Systems that are integrated with recommendation features to support decision processes can provide a more personalized and proactive retrieval experience for users to choose among retrieval alternatives and subsequent refinement of choices. Many real-world systems call for autonomous intelligent agents acting in the face of uncertain knowledge and limited computational resources. This paper explores the development of an intelligent system for book recommendation and retrieval. We focus on content management and the use of a broader range of search and retrieval tools including artificial intelligence, mobile intelligent agents, and fuzzy ontology, personal profile. The use of mobile agent technology provides a specific solution for overcoming the problem of being overloaded with too much information. The use of fuzzy ontology helps determine the association of concepts and relationships that can exist within an agent or a community of agents. The use of personal profile created by fuzzy ontology map helps reduce the search time. It facilitates search facilities suitable for comparing, evaluating and classifying the webagents behavior. The fuzzy decision-making is based on the interest level generated from user preferences, product feature selection and possible system recommendation. The simulation testing shows promising results.

Keywords—Fuzzy ontology, intelligent agents, content management, recommender system

I. INTRODUCTION

SYSTEMS that can provide recommendation for us to look at certain factors including what other people are doing as well as what we are doing are quite useful for information retrieval. This kind of information can be accessed from the Web, but the usefulness of this access is limited because users have no easy way to prioritize the information. While these systems are expected to help users navigate to products of preference and with more proactive and personalized information services, our ability to locate, filter, and analyse this information can be much improved by making use of intelligent techniques, user modelling with agent technologies [1]. One example of this might be the processing of quantitative data by automatic computer programs using artificial intelligence techniques such as fuzzy logic for knowledge representation and for making useful inferences or actions; expert systems for evidential and heuristic reasoning; neural

networks for classification and adaptive learning; genetic algorithms for solution optimisation; and data mining techniques for knowledge discovery (e.g. [2, 3]). Similarly, agent technologies now allow us to build software modules that can monitor, assist, and act on behalf of a user and to inter-operate with other agents (e.g. [4, 5]). Such technologies have also considerable potential for information retrieval.

There are many different book, movie and music recommender systems currently available online (e.g. Amazon, MediaUnbound). Most of them work by asking users to rate some sample items. Collaborative filtering algorithms are then used to process this input and to match the current user with others who share similar tastes and from that information to generate a list of recommendations. Traditionally, recommendations from friends have been the most helpful and trusted way for us to find out about new things and prioritize their relevance to our own interests and concerns. Currently, most of our systems conduct query searches using keywords and prioritize content based solely on recency: newest at the top, oldest at bottom. This is an information-poor approach that means that, for example, we see ads for new books in a much higher proportion than we do old books, even though old books might be just as appropriate or as important to us as the new ones.

Agent based systems contain software modules which can perform tasks on behalf of users. These modules, called intelligent agents or IAs, can handle separate tasks independently and autonomously as other agents also process tasks in parallel. An agent system is a platform providing the operations to the lifecycle of an agent [6]. Web content mining is the process of extracting knowledge from the content of Web documents. Typically, the user has to specify the area of the interests and we can opt for retrieving the book with recommendations from popular book publishers such as oreilly.com, books.mcgraw-hill.com. The proposed application is based on the intelligent agent-based platform, iJADE¹ from which a number of systems have already been developed including an authentication system, a surveillance system, a web mining system, a stock advisor system, a weather forecasting system, and a free walker which is a location-aware application utilizing the Global Positioning System to gather geographic information (e.g. [7, 8, 9, 10]). These systems are integrated with different intelligent agents to enhance the functionality of the application development.

Manuscript received October 22, 2006; Revised version received April 10, 2007. This work was supported in part by the departmental grant 1-ZV41 of The Hong Kong Polytechnic University.

James Liu is with the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (e-mail: csnkliu@comp.polyu.edu.hk).

¹ <http://www.ijadk.org>

On the other hand, a fuzzy set or distance can be used to classify information retrieval by the web agents. Using a fuzzy distance is flexible for us to modify the importance of the different values that we are using to classify the retrieved data by the agents. In the context of knowledge retrieval and recommendation, ontology is adopted to provide association of concepts and relationships that can exist within an agent or a community of agents [11].

In this paper we propose an intelligent system for book recommendation and retrieval. It focuses on content management and the use of a broader range of search and retrieval tools including artificial intelligence, intelligent agents, and fuzzy ontology. The remainder of this paper is organized as follows. Section 2 gives an overview of the basic concepts involved in the system development. Section 3 presents the system architecture and how the product recommendation is obtained. In Section 4, we describe the simulation tests and analysis. The last Section gives the conclusion and further work.

II. CONCEPT DEVELOPMENT

We review several core concepts in the design of our intelligent retrieval system which makes use of the mobile agent model, fuzzy logic, fuzzy ontology for book recommendation and retrieval.

A. The Mobile Agent Model

There exist server-side and client-side recommender systems. The server-side systems help users navigate in a specific web site (e.g. [12]) while the client-side systems can better capture the interests of an individual user and help finding useful information among the entire web (e.g. [13]). Such information, if gathered collectively, can support recommendation and reference to users. With the advancement of agent technologies [14], for system that deals with book recommendation and retrieval, it may involve different types of agents responsible for retrieving web information, performing post-retrieval analysis, interacting with users, sharing information about user search sessions, performing profile matching and carrying out retrieval and analysis tasks according some schedule.

A mobile agent is a running program (including both code and data) that travels on someone's behalf from one computer to another in a network carrying out a task, such as collecting information, and eventually returning with the results. At each site it visits, a mobile agent may invoke local resources— for example, accessing individual database entries. This approach has lower overheads than the traditional architecture in which a static client invokes resources remotely and possibly transfers large amounts of data. Other advantages of using a mobile agent model include the fact that agents are mobile and as they travel through the network they can not only perform other tasks but can carry partial or intermediate results calculated at the previous host. Further, mobile agents are asynchronous and so they can execute applications and tasks with users at the same time without blocking each other.

B. Fuzzy Logic

Knowledge about the system contains ambiguity and vagueness. It is convenient to represent the knowledge using fuzzy sets and fuzzy implications. Linguistic variables or fuzzy numbers seem to be an effective way of expressing factors such as user preference, goods flexibility, and quality of the products, which are difficult to denote by either crisp values or random processes. The agent obtains and learns the knowledge from the agents' knowledgebase and sets a series of rules from this knowledge. The more the agent is equipped with knowledge, the richer the rule base will be. As a result, the more rules the agent sets, the more experienced and knowledgeable the agent will be.

Basically speaking, the fuzzy sets vocabulary defines the "semantic" (the knowledge) of the problem. The rules manipulating fuzzy sets define the way the knowledge is applied. Zadeh [15] says that rather than regarding fuzzy theory as a single theory, we should regard the process of "fuzzification" as a methodology to generalize ANY specific theory from a crisp (discrete) to a continuous (fuzzy) form. Thus recently researchers have also introduced "fuzzy calculus", "fuzzy differential equations". Our system involves fuzzy rules typically for information searching. Therefore, the antecedent of these rules will be a logic expression relating a set of fuzzy variables. The fuzzy variable is defined by the name of physical variable and a qualifier. The set of linguistic variables used in the system application involves $X_1 = \{\text{Fame}\}$, $X_2 = \{\text{Age}\}$, $X_3 = \{\text{Date}\}$, $X_4 = \{\text{Price}\}$. Each variable has different levels of description. For example, Fig. 1 shows the fuzzy set X_1 with descriptions, Fame = {Unknown, Notfamous, Normal, Famous, Wellknown}.

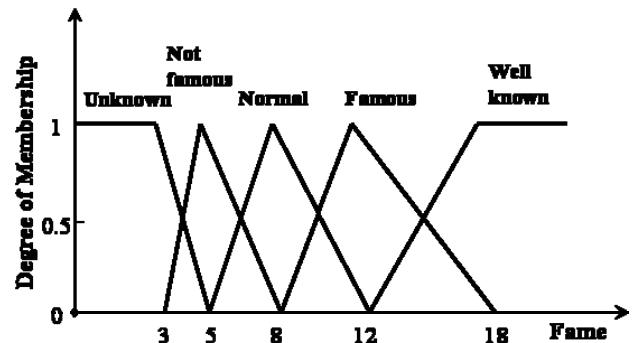


Fig. 1 The conversion table of fuzzy value for Fame(x)

Under FUZZIFICATION, the membership functions defined on these four variables are applied to their actual values, to determine the degree of truth for each rule premise. The antecedent contains the associated variables and all have their own degree of membership μ . For example, when user indicates the preference for a Fame ID of 2, the fuzzy description is "Not famous" and ranges between Fame = 3 and 8. Accordingly, we derive the membership values for Fame = 3, 4, ..., 8. For example, when

Fame=3, $\mu_{\text{Unknown}}(3) = 1$, $\mu_{\text{Notfamous}}(3)$, $\mu_{\text{Normal}}(3)$, $\mu_{\text{Famous}}(3)$ & $\mu_{\text{Wellknown}}(3) = 0$

Fame=4, $\mu_{\text{Unknown}}(4)$, $\mu_{\text{NotFamous}}(4) = 0.5$, $\mu_{\text{Normal}}(4)$, $\mu_{\text{Famous}}(4)$
& $\mu_{\text{Wellknown}}(4) = 0$

and so on. The fuzzy knowledge base has a series of set-level inferences with conditional and unconditional fuzzy rules. These can be used in carrying out approximate reasoning.

C. Fuzzy Ontology

We propose the use of fuzzy ontology to collect the fuzzy membership values between each class in the ontology [16]. The ontology is a hierarchical relationship between concepts within a domain, which can be viewed as a graph. It is developed based on the ontology graph and fuzzy logic. Fig. 2 illustrates the graph of fuzzy book ontology. Therefore, if a user is looking for some book to read, the system with this ontology graph should automatically consider all categories of books available for reading, i.e. fiction, comic, novel, etc. without requiring the user to enter all possible information. The system will consider all the concepts of subsumed by the concept “book” according to the ontology. In Fig. 2, the fuzzy membership value describes the similarity between two concepts. For instance, C1 representing the category of books being fiction is similar to R: book with fuzzy membership value 0.5. C2 being of comic category is similar to R: book with fuzzy membership value 0.8. This means that C2: comic would be more similar to R: book than C1: fiction. Each category C_i captures a set of book features for user preference matching. These fuzzy values are important for book prioritization and form the recommendation of the system which collectively gathers user preferences from node U of the ontology graph. This might include recommendation from popular book publishers such as oreilly.com and amazon.com. Therefore, user can make use of the ontology graph via the system to get recommendation for the book selection.

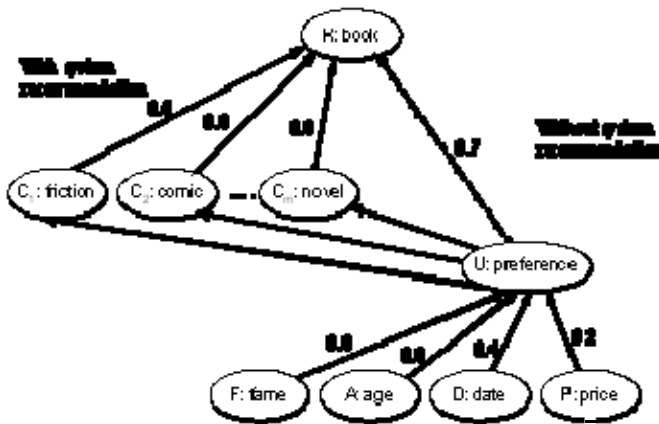


Fig. 2 A typical graph of fuzzy book ontology

On the other hand, user can by pass the recommendation of the system and rely on specific preference for book retrieval. For instance, nodes F, A, D and P correspond to the preference features respectively:

- F: the descriptor for the famous book
- A: the descriptor for the age group that the book best suits

- D: the descriptor for the duration period that the book has been published
- P: the descriptor for the price of the book

The fuzzy inference based on user input preference will be resolved and goes to R for book retrieval accordingly. In any case, the user information will be sent to the recommender system for prioritization update.

D. Personal Profile

To allow the ontology to be reused by others, the book ontology was modeled in Web Ontology Language (OWL) by using Protégé [17]. Protégé is a free, open-source platform with a friendly user interface that provides a set of tools for constructing domain model and knowledge-based applications with ontologies. The book ontology could have many classes. It is difficult for the user to search or find a particular book in such size. To solve this problem, we proposed a personal profile [18], which created by Fuzzy Ontology Map (FOM) [19, 20] to reduce the searching time. Given a partial OWL of the book ontology in Fig. 2, the FOM in the personal profile, we defined the fuzzy set as the interest level. We defined the interest level: most interested (1.0-0.8), very interested (0.8-0.6), interested (0.6-0.4), not very interested (0.4-0.2) and not interested (0.2-0.0). The fuzzy membership values can be captured in FOM through interaction of users via the recommender system in terms of preferences or analyzing the user browsing history.

III. SYSTEM OVERVIEW AND DESIGN

The proposed system is built on iJADE, an agent platform that implements the FIPA Agent Management Specification using pure Java 2 Standard Edition (JDK1.4). iJADE provides comprehensive intelligent agent based APIs and applications for e-commerce and its agent technologies and intelligent algorithms. The proposed system uses four main agents (Fig. 3), a stationary agent (servletAgent) in a Tomcat servlet server, a stationary agent (fuzzyAgent) in an iJADE server at port 2000, a mobile agent (fetchAgent) dispatching between the iJADE server at port 3333 and a web server on the Internet and, fourth, a mobile agent (dbAgent, updatedbAgent) dispatching between the iJADE server and a database server.

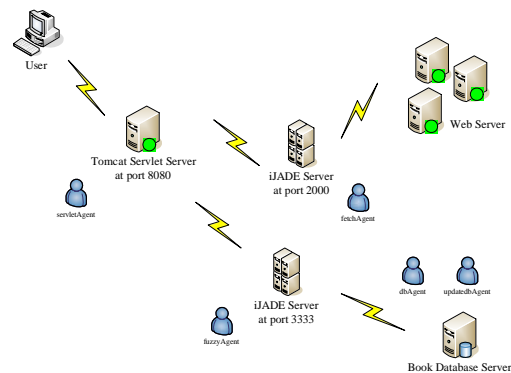


Fig. 3 System diagram

A. The System Architecture

The system adopts an iJADE architecture (Fig. 4) consisting of four layers:

1. **Application Layer:** This is the uppermost layer and contains various intelligent agent-based applications. This layer accepts data from the conscious layer and is connected to external applications.
2. **Conscious Layer:** This is an intelligent layer that includes Sensory, Logical Reasoning and Analytical Areas.
3. **Technology Layer:** This provides all the necessary mobile agent implementation APIs for the development of intelligent agent components in the 'Conscious Layer'.
4. **Supporting Layer:** This provides a programming language and protocols to support the development of the 'Technology Layer'.

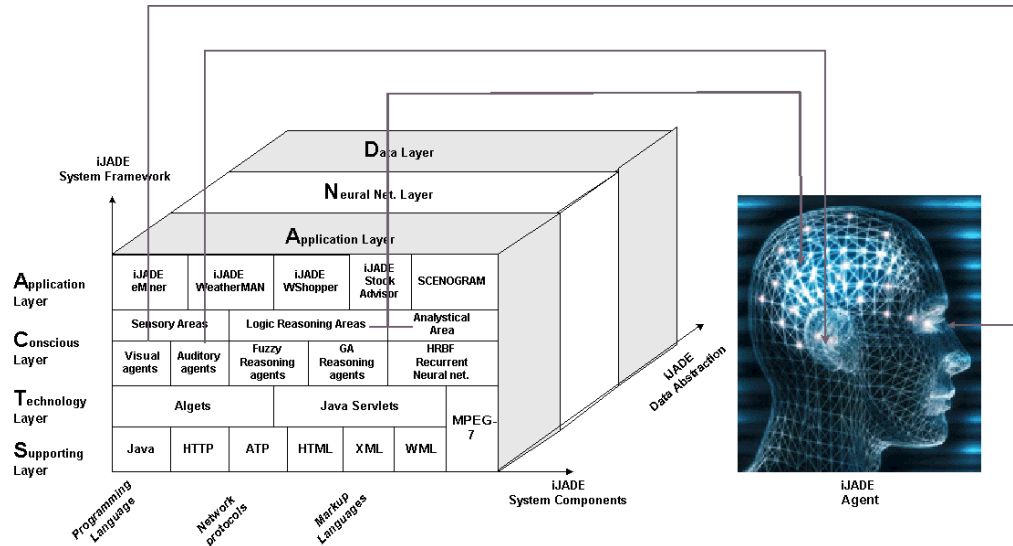


Fig. 4 The iJADE architecture

B. ServletAgent

The servletAgent is a stationary agent with built-in intelligence. It has a knowledge base that it uses to create request objects. Its main function is to automatically create a request object in response to a request and to provide the following output content to users: names of book, category, price, and short description. It also acts as an interface between a user and the iJADE server. When a user clicks the Java Servlet Webpage, Java Servlet will invoke servletAgent and pass a simple message, e.g. [booktype, fame, age, date, price, keyword, action]. Then the agent will send the request to other agents with action type. If it is a request for recommendation without any specific preferences, it passes to the dbAgent to retrieve the related recommendation.

If it is a request with specific preferences, then it passes to the fuzzyAgent to convert the id value (for example, id value for booktype = 0, fame = 2, age = 2, date = 2, price = 2) to fuzzy value and then passes to dbAgent to get the related information from ontology graph in the system database. In addition, it handles the output content for user with the following features - names of the book, category, price and short description (see Fig. 5).

The servletAgent is responsible for three tasks: Automatic Login and register; Update user preference; and Choosing query mode. Automatic Login and register is the autologin protocol that agents use to detect if the user is a newcomer by

automatically detecting the user's IP address if the user is not listed in the database. It will create the new account with default value for this newcomer. If the user has visited that before, it will use the user's stored information. There are four query modes: General query without keyword (i.e. no user preference); General query with keyword; Fuzzy query without keyword; and Fuzzy query with keyword. Users can make query for the book using categories and keyword from a simple dictionary. Stop-word removal can exclude high-frequency words such as "a", "an", "the", "of", etc. and the remaining words are used to get the recommended book with information about the book title, author, and description, Users can also ask for a book using the same parameters as in a general query as well as additional fuzzy parameters such as fame, age, date and price.



Fig. 5 Output content that handled by servletAgent

C. FuzzyAgent

The fuzzyAgent is a stationary agent that is used to calculate the fuzzy value for each book. It uses the information [fame, age, date, price] provided by the servletAgent to search for similar books in the database. It can handle four input properties. In this case they are the degree of fame of the book, reader age, publication date and price: Fame(x) = (unknown, not famous, normal, famous, well-known); Age(x) = (baby, young, middle, teen, adult); Date(x) = (long before this year, before this year, this year, after this year); Price(x) = (very cheap, cheap, normal, expensive, very expensive)

Users can set the book preference (e.g. a typical criterion with corresponding crisp value ID): Fame ID = 2, Age ID = 1, Date ID = 4, Price ID = 5, and derive the fuzzy values of these criteria accordingly. For example, Fame involves different fuzzy sets - Unknown, Notfamous, Normal, Famous, Wellknown. Fig. 6 gives the maximum points of the fuzzy membership functions of the variable throttle as {0, 0.2, 0.4, 0.6, 0.8} corresponding to different fuzzy sets. Given Fame ID = 2 which denotes "Not famous" and ranges between 3 and 8, we can derive the associated description with respect to these fuzzy sets:

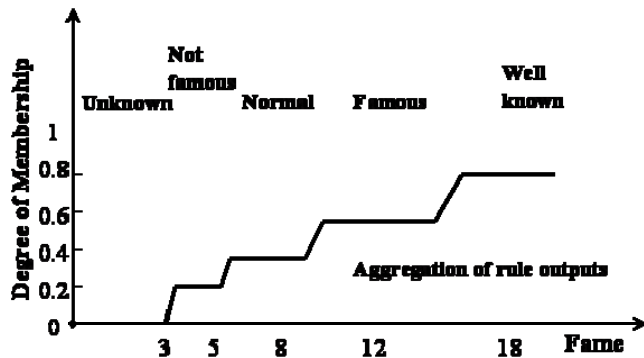


Fig. 6 Aggregation of rule outputs

The defuzzification is done using the centroid method (e.g. [20]),

$$\text{Fuzzy_fame } \beta_1 = \frac{\sum_{i=1}^6 CF_i * \overline{\mu Y_i}}{\sum_{j=1}^6 CF_j}$$

Therefore, the corresponding fuzzy values are: fuzzy_fame β_1 is 0.1083, fuzzy_age β_2 is 0.025, fuzzy_date β_3 is 0.50, fuzzy_price β_4 is 0.3666

All of these crisp values will be output as preference score for finding the relevant book from the ontology graph in the database. The update fuzzy value will be:

$$\text{UPDATE book SET book.fuzzy} = \frac{1}{4} \sum |CF_i - \beta_i| = 0.187$$

Then the query is processed by using the following equation.

select * from book ORDER BY fuzzy ASC

(select all from "book" ontology database ordering by the term "fuzzy" in ascending order)

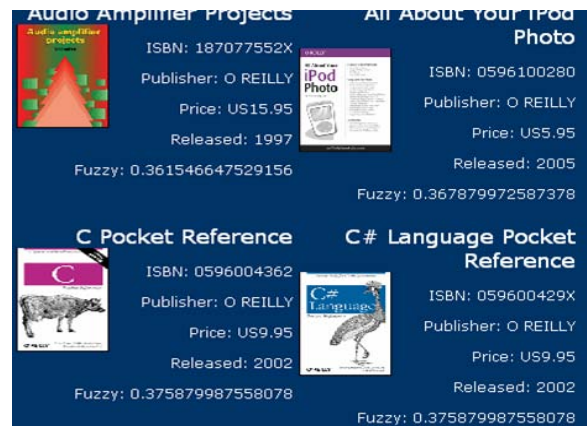


Fig. 7 Output result sorted by fuzzy value in ascending order

So, the book with lower fuzzy value will be displayed first. The outputs are sorted in ascending order (see Fig. 7). The information helps build up the level of interest of books recommended to user upon any preference information captured in the FOM system.

D. fetchAgent

Fig. 8 shows the fetchAgent which is a mobile agent for searching books from popular book publishers such as oreilly.com and amazon.com so as to get the updated book context from there. The fetchAgent uses pattern matching to determine the boundary of the book content page and title. It loads the URL of content page from the object file and fetches the content page from the web site of publisher to the local system. Then it reads the fetched content page and checks the anchor string pattern and the information is used to update the book ontology graph.

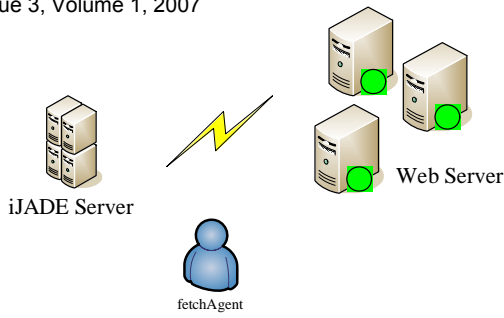


Fig. 8 An overview of fetchAgent

E. dbAgent and updateAgent

Fig. 9 shows the dbAgent and updatedbAgent which are mobile agents to handle the information processing between iJADE server and database server. This agent is to communicate between the iJADE server and database server. It receives the request from other agent such as fuzzyAgent, servletAgent. For instance, servletAgent sends a request to dbAgent so that dbAgent retrieves the information for servletAgent to display to user accordingly.

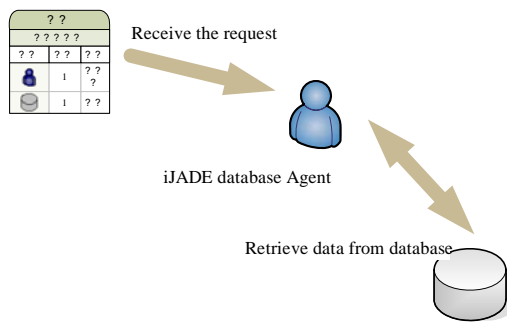


Fig. 9 An overview of dbAgent / updatedAgent

IV. EXPERIMENTAL RESULTS AND ANALYSIS

Structural information was collected from a number of online bookstores. After filtering and grouping similar terms, we used the information to model our book ontology. For simplicity, we only model the upper-level of book ontology representing the basic concept of commonly used categories with various similarity scores. Here with we describe three tests of different query features that were conducted for getting the product recommendation and retrieval:

- 1) Basic Fuzzy Query without any user preference
- 2) Basic Fuzzy Query with user preference
- 3) Fuzzy Query with user preference and comparison

A. Test 1

This test demonstrated the accuracy of the fuzzy query by using only the preference for the fuzzy parameter price. As shown in Fig. 10. The ranking of the books starts from US\$80 to US\$40. The price is found almost from high to low except that for US119.95. The experiment shows that the fuzzy system can work out the matching scores and have recommendation

ranked using preference information provided by user.

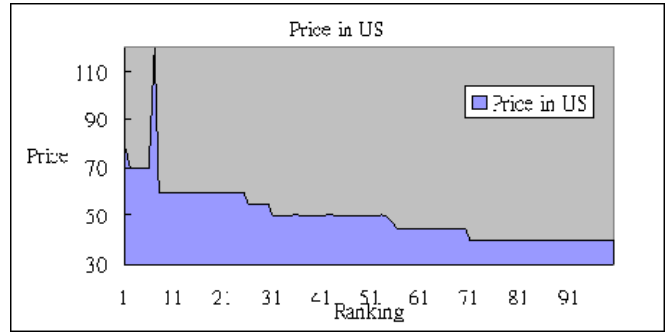


Fig. 10 Result of Test 1

B. Test 2

This test demonstrated the accuracy of the fuzzy query by using all fuzzy preference information including fame, age, date and price with parameter specifications given in Table 1. The test result (Figs. 11 – 12) shows that the overall ranking between 1 and 79 are correct. However, the ranking 80 has date 2006. Therefore, the fuzzy query without keyword can be used to find the ranked recommendation quite accurately.

Table 1. Detailed description of the fuzzy parameters of test 2

Parameter	Value	Description
Frame	Famous	Range between 8 and 18
Age	Adult	Larger 18
Date	Before Year	Range between 1956 and 2006
Price	Normal	Range between HK\$100(US\$12.5) and HK\$400(US\$50)

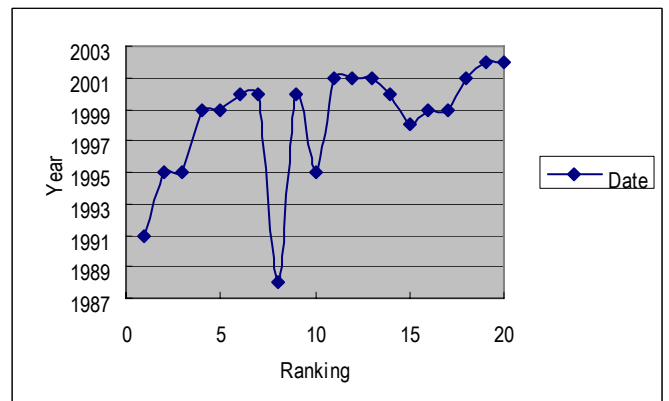


Fig. 11 Result of Test 2 (date)

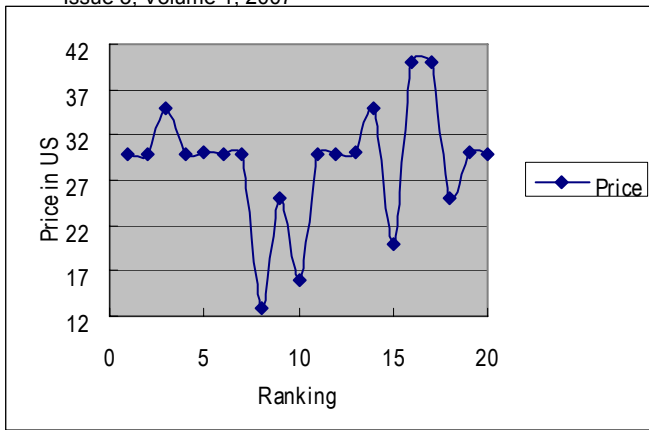


Fig. 12 Result of Test 2 (price)

C. Test 3

This test demonstrated the accuracy of the fuzzy query by using all fuzzy information including fame, age, date and price as well as keywords for limiting the outcomes. Compared to the last experiment, test 2, this test involves query with keywords. It shows that only 11 recommendations are returned compared to more than 100 ones from last experiment. Moreover, all recommendations are matched to the preference parameter. Because of the use of keyword “linux”, the following words “linux”, “open-source”, “unix” and “operating” from the book ontology are used for retrieving book recommendation. Therefore, more relevant and useful result can be obtained (see Figs. 13-14).

Moreover, we do the comparison with www.oreilly.com using the same keyword “Linux” without ontology information illustrated in Fig. 15.

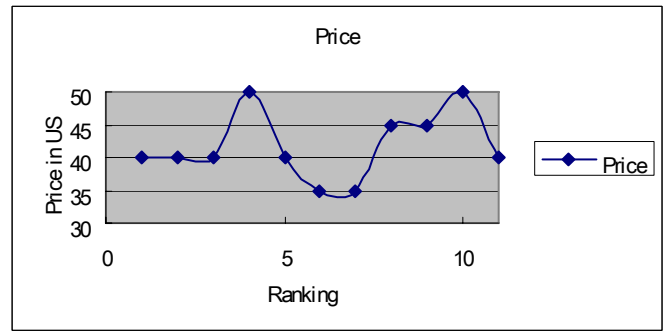


Fig. 13 Result of Test 3 (date)

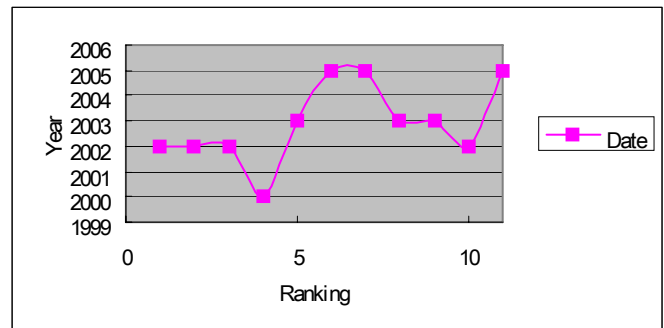


Fig. 14 Result of Test 3 (price)

For comparison, 379 recommended books are returned from oreilly.com. This represents about 30% (i.e. 379/1200) of books in its database. With the use of ontology information, we are able to reduce it to about 6% of recommendations via our system. The result is quite promising and our system can take less time for filtering the query result.

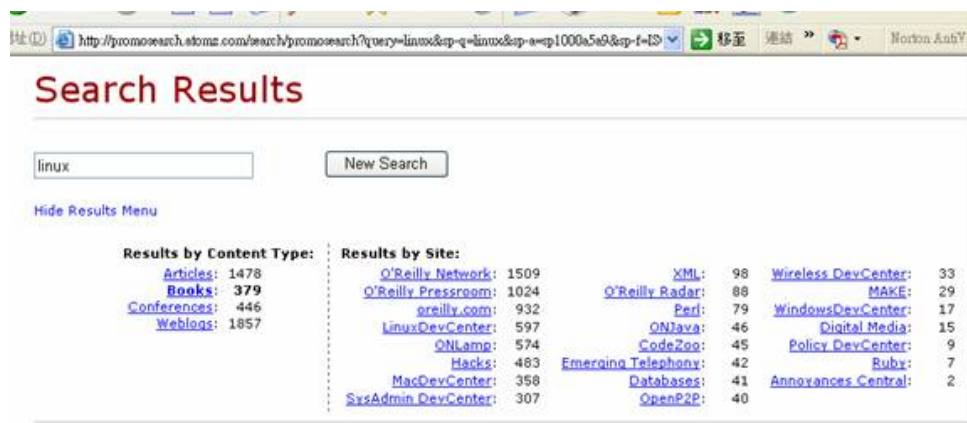


Fig. 15 Query result from oreilly.com

V. CONCLUSION AND FUTURE WORK

Fuzzy reasoning and ontology information is found to be quite useful for book retrieval with recommendation. With fuzzy logic and user preference information, we can use fuzzy

input terms such as fame, price as well as keywords for query and obtain the most perfectible recommendation. This paper shows promising result not only demonstrating the usefulness of proposed system, but also offering practical and helpful

advice to users, with a user-friendly interface as well as without too much pre-requisite knowledge about the system.

The present paper emphasizes on the use of mobile agents, fuzzy logic and ontology for product retrieval with recommendation. It caters for user preferences which could have been induced from online bookstores or peer inference in terms of some recommendation. In situation where recommendations are to be collected from different sources, we need good means of extracting, organizing, categorizing and digesting them like what we did for news digest (e.g. [21]). The fuzzy ontology structure requires detailed analysis of the domain behaviour and concept association with different fuzzy descriptors. For recommender system, and as shown from the fuzzy ontology graph in Fig. 2, association mining can be performed within each book category to find user-to-user correlation and book-to-book correlation. We can calculate similarity between the different users and provide the user with the product recommendation selected by similar users (e.g. collaborative filtering systems [22]). This is via the generation of personal profiles for different users. Furthermore, existing attribute features of iJADE agent can be applied, like collaboration to improve the system operation efficiency. All these will be the subject of future research.

REFERENCES

- [1] T. H. W. Lam, R. S. T. Lee and J. N. K. Liu, "An Ontology-based Intelligent Mobile System for Tourist Guidance," accepted to appear in *Computational Intelligent Handbook*, 2008, Springer-Verlag.
- [2] J. W. Han and K. C. C. Chang, "Data Mining for Web Intelligence," *IEEE Computer*, pp. 64 – 70, 2002.
- [3] S. C. K. Shiu, J. N. K. Liu, J. N. C. Lam and B. Feng, "A data mining approach for branch and ATM site evaluation," *LCNS 3755*, Springer, 2006, pp. 303-318.
- [4] E. C. L. Chan and R. S. T. Lee, "iJADE Reporter – An Intelligent Multi-Agent Based Context – Aware News Reporting System," in *Proc. Int'l Conf. Knowledge-based and Intelligent Information Engineering Systems (KES2005)*.
- [5] J. N. K. Liu, M. Wang and B. Feng, "iBotGuard: An Internet-based Intelligent Robot Security System Using Invariant Face Recognition Against Intruder," *IEEE Trans. Systems, Man and Cybernetics (Part C)*, vol. 35, no. 1, pp. 97-105, 2005.
- [6] S. Murugesan, "Intelligent Agents on the Internet and Web," in *Proc. IEEE Int'l Conf. Global Connectivity in Energy, Computer, Communication and Control (TENCON)*, 1998, vol. 1, pp. 97 - 102.
- [7] R. S. T. Lee, "iJADE Surveillant - An Intelligent Multi-resolution Composite Neuro-Oscillatory Agent-based Surveillance System," *Pattern Recognition*, vol. 36, pp.1425-1444, 2003.
- [8] R. S. T. Lee and J. N. K. Liu, "iJADE Web-miner: An Intelligent Agent Framework for Internet Shopping," *IEEE Trans. Knowledge and Data Engineering*, vol. 16, no. (4), pp. 461-473, 2004a.
- [9] R. S. T. Lee and J. N. K. Liu, "iJADE WeatherMAN: A Weather Forecasting System using Intelligent Multi-Agent Based Fuzzy-Neuro Network," *IEEE Trans. Systems, Man and Cybernetics (Part C)*, vol. 34, no. 3, pp. 369-377, 2004b.
- [10] R. S. T. Lee and J. N. K. Liu, "iJADE Stock Predictor – An intelligent multi-agent based time series stock prediction system," in *Proc. the 2nd Asia-Pacific Conf. Intelligent Agent Technology (IAT-2001)*.
- [11] T. R. Gruber, *A translation approach to portable ontologies. Knowledge Acquisition*, Academic Press, 1993.
- [12] R. Baraglia and F. Silvestri, "An Online Recommender System for Large Web Sites," in *Proc. IEEE/WIC/ACM Int'l. Conf. on Web Intelligence (WI'04)*, pp. 199-205.
- [13] T. S. Zhu, R. Greiner and G. H'auabl, "An Effective Complete-Web Recommender System," in *Proc of the 12th Int'l. World Wide Web Conference (WWW2003)*, Budapest, Hungary.
- [14] R. S. T. Lee, *Fuzzy-Neuro Approach to Agent Applications (From the AI Perspective to Modern Ontology)*, Springer, 2006.
- [15] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.
- [16] T. H. W. Lam and R. S. T. Lee, "iJADE FreeWalker – An Intelligent Ontology Agent-based Tourist Guiding System," *Studies in Computational Intelligence*, vol. 72, pp. 103-125, 2007.
- [17] Protégé, <http://protege.stanford.edu/>, 2007
- [18] S. E. Middleton, N. R. Shadbolt and D. C. Roure, "Ontological user profiling in recommender systems," *ACM Trans. Information Systems*, vol. 22, no. 1, pp. 54-88, 2004.
- [19] A. Pena, H. Sossa and F. Gutierrez, "Ontology Agent-Based Rule Base Fuzzy Cognitive Maps," *LNAI 4496*, Springer, 2007, pp. 328-337.
- [20] J. N. K. Liu, "An Intelligent System Integrated with Fuzzy Ontology for Product Recommendation and Retrieval," in *Proc. of the 8th WSEAS Int'l Conf. on Fuzzy Systems (FS'07)*, pp. 180-185.
- [21] J. N. K. Liu and M. K. Ho, "An Intelligent Agent-based Systems for Multilingual Financial News Digest," in *Proc. the 2nd KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications (AMSTA 2008)*, Korea, March 27-28, 2008, Springer Verlag.
- [22] W. Y. Lin, S. A. Alvarez and C. Ruiz, "Efficient Adaptive-Support Association Rule Mining for Recommender Systems," *Data Mining and Knowledge Discovery*, vol. 6, no. 1, pp. 83-105, 2003.

James N.K. Liu received the BSc(Hons) and MPhil degrees in mathematics and computational modeling from Murdoch University, Perth, Australia, in 1982 and 1987 respectively. He received his Ph.D. degree in artificial intelligence from La Trobe University, Melbourne, Australia, in 1992. While working on his degree, he worked as a computer scientist at Defence Signal Directorate in Australia from 1988 till 1990. He joined the Aeronautical Research Laboratory (ARL) of Defence Science and Technology Organization in Australia as a research scientist in 1990. At ARL, he helped perform AI research in areas of human factors and mission enhancement

Dr Liu joined the Hong Kong Polytechnic University, as a faculty member in 1994, and currently holds the position of Associate Professor in the Department of Computing. He has published technical papers on subjects in expert system verification, forecasting systems, pattern recognition, web technology and e-Commerce applications. His current interests include telerobotics, ontology-based agent systems, intelligent business computing, forecasting applications and data mining. He is a senior member of IEEE and member of AAAI