An Ontology-Supported User Modeling Technique with Query Templates for Interface Agents

SHENG-YUAN YANG
Dept. of Computer and Communication Engineering
St. John's University
499, Sec. 4, TamKing Rd., Tamsui, Taipei County 251
TAIWAN
http://mail.sju.edu.tw/~ysy

Abstract: - This paper describes an Interface Agent with ontology-supported, query template-based user modeling techniques which works as an assistant between the users and FAQ systems to retrieve FAQs on the domain of Personal Computer. It integrates several interesting techniques including user modeling, domain ontology, and template-based linguistic processing to effectively tackle the problems associated with traditional FAQ retrieval systems. Our work features an ontology-supported user modeling technique with query templates for developing interface agents; a nature language query mode, along with an improved keyword-based query mode; and an assistance and guidance for human-machine interaction. Our preliminary experimentation demonstrates that user intention and focus of up to eighty percent of the user queries can be correctly understood by the system.

Key-Words: - Ontology, Query Templates, User Modeling, Interface Agents

1 Introduction

With increasing popularity of the Internet, people depend more on the Web to obtain their information. Especially the use of the World Wide Web has been leading to a large increase in the number of people who access FAQ knowledge bases to find answers to their questions [12]. As the techniques of Information Retrieval [8,9] matured, a variety of information retrieval systems have been developed, e.g., Search engines, Web portals, etc., to help search on the Web. How to search is no longer a problem. The problem now comes from the results from these information retrieval systems which contain some much information that overwhelms the users. Therefore, how to improve traditional information retrieval systems to provide search results which can better meet the user requirements so as to reduce his cognitive loading is an important issue in current research [2].

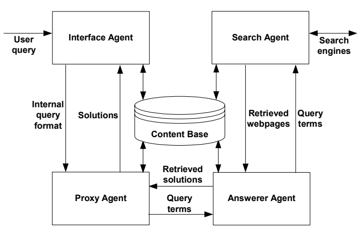


Fig. 1 System architecture for FAQ-master

The websites which provide Frequently Asked Questions (FAQ) organize user questions and expert answers about a specific product or discipline in terms of question-answer pairs on the Web. Each FAQ is represented by one question along with one answer and is characterized, to be domain-dependent, short and explicit, and frequently asked [4,6]. Traditional FAQ retrieval systems, however, provide only little help, became they fail to provide assistance and guidance for human-machine interaction, personalized information services, flexible interaction interface, etc. [2]. In order to provide high-quality FAQ answers from the Web to meet the user requests, we have proposed an *FAQ-master* as an intelligent Web information aggregation system, which provides intelligent information retrieval, filtering, and aggregation services [15,21]. By a high-quality answer we mean an answer which is *profound*, *up-to-date*, and *relevant* to the user's question. Fig. 1 illustrates the system architecture of FAQ-master. It contains four agents supported by a Content Base, which in turn contains a User Model Base, Template Base, Domain Ontology, Website Model Base, Ontological Database, Solution Library, and Rule Base. The Interface Agent captures user intention through an adaptive human-machine interaction interface with the help of *ontology-directed and template-based user models* [13,16,22]. It also handles user feedback on the suitability of proposed responses. The Search Agent performs in-time, user-oriented, and domain-related Web information retrieval with the help of *ontology-supported website models* [14,17,23]. The Answerer Agent works as a back end process to perform *ontology-directed information aggregation* from the webpages collected by the Search Agent [18,24,25]. Finally, the Proxy Agent works as an *ontology-enhanced intelligent proxy* mechanism to share most query loading with the Answerer Agent [19,20,24,25].

This paper discusses the Interface Agent focusing on how it captures true user's intention and accordingly provides high-quality FAQ answers. The agent features ontology-based representation of domain knowledge, flexible interaction interface, and personalized information filtering and display. Specifically, according to the user's behavior and mental state, we employed the technique of user modeling [13,16,22] to construct a user model to describe his characteristics, preference and knowledge proficiency level, etc. We also used the technique of user stereotype [7] to construct and initialize a new user model, which helps provide fast personalized services for new users. We built domain ontology [5] to help define domain vocabulary and knowledge and based on that to construct user models and the Interface Agent. We extended the concept of pattern match [3] to query template to construct natural language based query models. Our preliminary experimentation demonstrates that the intention and focus of up to eighty percent of the users' queries can be correctly understood by the system. The Personal Computer (PC) domain is chosen as the target application of our Interface Agent and will be used for explanation in the remaining sections.

2 Fundamental Techniques

2.1 Domain Ontology and Services

The concept of ontology in artificial intelligence refers to knowledge representation for domain-specific contents [1]. It has been advocated as an important tool to support knowledge sharing and reusing in developing intelligent systems. Although development of an ontology for a specific domain is not yet an engineering process, we have outlined a procedure for this in [13] from how the process was conducted in existent systems. By following the procedure we developed an ontology for the PC domain in Chinese using Protégé 2000 [5], but was changed to English here for easy explanation, as the fundamental background knowledge for the system.

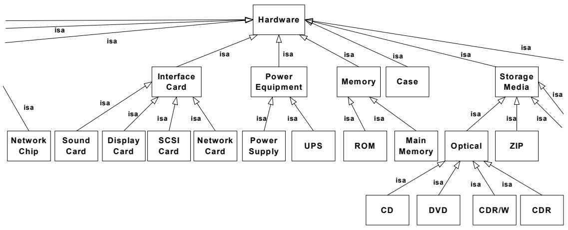


Fig. 2 Part of PC ontology taxonomy

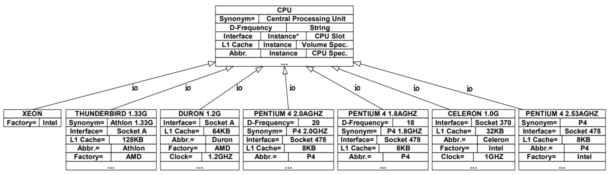


Fig. 3 Ontology of the concept of CPU

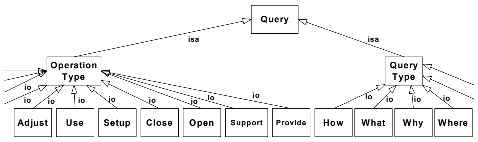


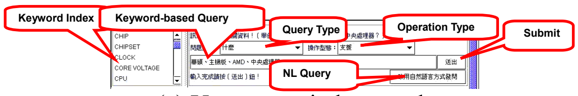
Fig. 4 Part of problem ontology taxonomy

Fig. 2 shows part of the ontology taxonomy. The taxonomy represents relevant PC concepts as classes and their parent-child relationships as *isa* links, which allow inheritance of features from parent classes to child classes. Fig. 3 exemplifies the detailed ontology of the concept of CPU. In the figure, the root node uses various fields to define the semantics of the CPU class, each field representing an attribute of "CPU", e.g., interface, provider, synonym, etc. The nodes at the lower level represent various CPU instances, which capture real world data. The arrow line with term "*io*" means the instance of relationship. The complete PC ontology can be referenced from the Protégé Ontology Library at Stanford Website (http://protege.stanford.edu/download/download.html).

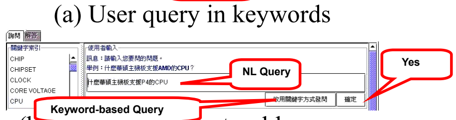
We also developed a Problem ontology to deal with query questions. Fig. 4 illustrates part of the Problem ontology, which contains query type and operation type. Together they imply the semantics of a question. Finally, we use Protégé's APIs to develop a set of ontology services, which provide primitive functions to support the application of the ontologies. The ontology services currently available include transforming query terms into canonical ontology terms, finding definitions of specific terms in ontology, finding relationships among terms, finding compatible and/or conflicting terms against a specific term, etc.

2.2 Ontology-Supported Query Templates for User Query Processing

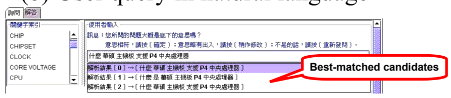
Fig. 5 illustrates two ways in which the user can enter Chinese query through Interface Agent. Fig. 5(a) shows the traditional keyword-based method, enhanced by the ontology features as illustrated in the left column. The user can directly click on the ontology terms to select them into the input field. Fig. 5(b) shows the user using natural language to input his query. In this case, Interface Agent first employs MMSEG [11] to do word segmentation, then applies the template matching technique to select best-matched query templates as shown in Fig. 5(c) [16,22], and finally trims any irrelevant keywords in accord with the templates [18,24,25].



(a) User query in keywords



(b) User query in natural language



(c) Best-matched templates for user query in natural language

Fig. 5 User query through our Interface Agent

Table 1 Question types

Question Type	Intention
是否 (A-NOT-A)	Asks about can or cannot, should or should not, have or have not
如何 (HOW)	Asks about solving methods
什麼 (WHAT)	Enumerates related information
何時 (WHEN)	Asks about time, year, or date
哪裡 (WHERE)	Asks place or position
為什麼 (WHY)	Asks reasons

Table 2 Examples of intention types

Intention Type	Description
ANA_CAN_SUPPORT	Asks if support some specifications or products
HOW_SET	Asks the method of assignment
WHAT_IS	Asks the meaning of terminology
WHEN_SUPPORT	Asks when can support
WHERE_DOWNLOAD	Asks where can download
WHY_SETUP	Asks reasons about setup

To build the query templates, we have collected in total 1215 FAQs from the FAQ website of six famous motherboard factories in Taiwan and used them as the reference materials for query template construction. Currently, we only take care of the user query with one intention word and at most three sentences. These FAQs were analyzed and categorized into six types of questions as shown in Table 1. For each type of question, we further identified several intention types according to its operations. Table 2 illustrates some examples of intention types. Finally, we define a query pattern for each intention type. Table 3 illustrates the defined query patterns for the intention types of Table 2. Table 4 explains the syntactical constructs of the query patterns.

Now all information for constructing a query template is ready, and we can formally define a query template. Table 5 defines what a query template is. It contains a template number, number of sentences, intention words, intention type, question type, operation type, query patterns, and focus. Table 6 illustrates an example query template for the ANA_CAN_SUPPORT intention type. Note here that we collect similar query patterns in the field of "Query patterns," which are used in detailed analysis of a given query.

Table 3 Examples of query patterns

Question Type	Operation Type	Intention Type	Query Pattern
是否 (If)	支援 (Support)	ANA_CAN_SUPPORT	<S1 是否 支援 S2>
		GA-7VRX 這塊主機板是否支援 KINGMAX DDR-400？ (Could the GA-7VRX motherboard support the KNIGMAX DDR-400 memory type?)	
如何 (How)	安裝 (Setup)	HOW_SETUP	<如何 在 S1><安裝 S2>
		如何在 Windows 98SE 下，安裝 8RDA 的音效驅動程式？ (How to setup the 8RDA sound driver on a Windows 98SE platform?)	
什麼 (What)	是 (Is)	WHAT_IS	<S1 是 什麼>
		AUX power connector 是什麼？ (What is an AUX power connector?)	
何時 (When)	支援 (Support)	WHEN_SUPPORT	<S1 何時 支援 S2>
		P4T 何時才能支援 32-bit 512 MB RDRAM 記憶體規格？ (When can the P4T support the 32-bit 512 MB RDRAM memory specification?)	
哪裡 (Where)	下載 (Download)	WHERE_DOWNLOAD	<S1><哪裡 可以 下載 S2>
		CUA 的 Driver CD 遺失，請問哪裡可以下載音效驅動程式？ (Where can I download the sound driver of CUA whose Driver CD was lost?)	
為什麼 (Why)	列印 (Print)	WHY_PRINT	[S1]<S2 無法 列印>
		為什麼在 Win ME 底下，從休眠狀態中回復後，印表機無法列印。 (Why can I not print after coming back from dormancy on a Win ME platform?)	

Table 4 Definition of pattern symbols and descriptions

Symbol	Description
<>	Means single sentence and considers the sequence of intention words and keywords
[ ]	Means at least one sentence and only consider appeared keywords but sequence
Si	Means the variable part of a template which is any string consists of keywords
Intention Word	Means the fixed part of a template which can help the system distinguish between user intentions
Keyword	Means the concepts in domain ontology which are usually domain terminologies
Focus	Means if the variable part of a template is the user query key point, we called the focus

Table 5 Query template specification

Field	Description
Template_Number	Template ID
#Sentence	The number of sentences: 1 for one sentence, 2 for two sentences, and 3 for three or more sentences
Intention_Word	Intention words must appear
Intention_Type	Intention type
Question_Type	Question type
Operation_Type	Operation type
Query_Patterns	The semantic pattern consists of intention words and keywords
Focus	The focus of the user

Table 6 Query template for the ANA_CAN_SUPPORT intention type

Template_Number	304
#Sentence	3
Intention_Word	是否(If)、支援(Support)
Intention_Type	ANA_CAN_SUPPORT
Question_Type	是否(If)
Operation_Type	支援(Support)
Query_Patterns	[S3]<S1 是否 支援 S2> [S2]<是否 支援 S1>
Focus	S1

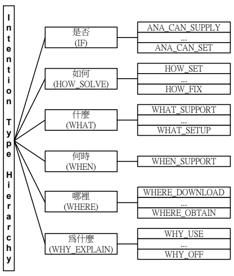


Fig. 6 Intention type hierarchy

According to the generalization relationships among intention types, we can form a hierarchy of intention types to organize all FAQs. Currently, the hierarchy contains two levels as shown in Fig. 6. Now, the system can employ the intention type hierarchy to reduce the search scope during the retrieval of FAQs after the intention of a user query is identified. Table 7 shows the statistics of query templates in our system. Currently, we have in total 154 intention words which form an intention word base.

Table 7 Statistics of query templates

Question Type	#Intention Type	#Template	#Pattern	#FAQ (%)
是否 (if)	19	53	113	385(31.7 %)
如何 (how)	18	44	121	265(21.8 %)
什麼 (what)	6	18	19	91(7 %)
何時 (when)	1	1	3	3(0.2 %)
哪裡 (where)	3	4	4	4(0.3 %)
為什麼 (why)	25	199	771	467(38.4 %)
Total	69	319	1031	1215

3 System Architecture

3.1 User Modeling

A user model contains interaction preference, solution presentation, domain proficiency, terminology table, query history, selection history, and user feedback, as shown in Fig. 7.

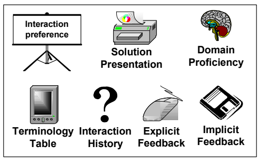


Fig. 7 Our user model

The interaction preference is responsible for recording user's preferred interface, e.g., favorite query mode, favorite recommendation mode, etc. When the user logs on the system, the system can select a proper user interface according to this preference. We provide two modes, either through keywords or natural language input. We provide three recommendation modes according to hit rates, hot topics, or collaborative learning. We record recent user's preferences in a time window, and accordingly determine the next interaction style.

The solution presentation is responsible for recording solution ranking preferences of the user. We provide two types of ranking, either according to the degree of similarity between the proposed solutions and the user query, or according to user's proficiency about the solutions. In addition, we use a Show_Rate parameter to control how many items of solutions for display each time, in order to reduce information overloading problem.

The domain proficiency factor describes how familiar the user is with the domain. By associating a proficiency degree with each ontology concept, we can construct a table, which contains a set of <concept proficiency-degree> pairs, as his domain proficiency. Thus, during the decision of solution representation, we can calculate the user's proficiency degree on solutions using the table, and accordingly only show his most familiar part of solutions and hide the rest for advanced requests. To solve the problem of different terminologies to be used by different users, we include a terminology table to record this terminology difference. We can use the table to replace the terms used in the proposed solutions with the user favorite terms during solutions representation to help him better comprehend the solutions.

Finally, we record the user's query history as well as FAQ selection history and corresponding user feedback in each query session in the Interaction history, in order to support collaborative recommendation. The user feedback is a complicated factor. We remember both explicit user feedback in the selection history and implicit user feedback, which includes query time, time of FAQ click, sequence of FAQ clicks, sequence of clicked hyperlinks, etc.

In order to quickly build an initial user model for a new user, we pre-defined five stereotypes, namely, expert, senior, junior, novice, and amateur [13], to represent different user group's characteristics. This approach is based on the idea that the same group of user tends to exhibit the same behavior and requires the same information. Fig. 8 illustrates an example user stereotype. When a new user enters the system, he is asked to complete a questionnaire, which is used by the system to determine his domain proficiency, and accordingly select a user stereotype to generate an initial user model to him. However, the initial user model constructed from the stereotype may be too generic or imprecise. It will be refined to reflect the specific user's real intent after the system has experiences with his query history, FAQ-selection history and feedback, and implicit feedback [2].

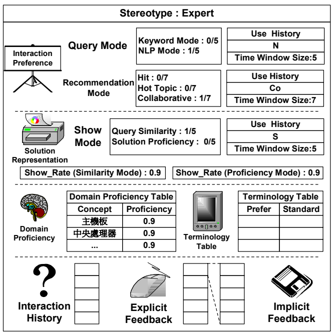


Fig. 8 Example of expert stereotype

3.2 System Overview

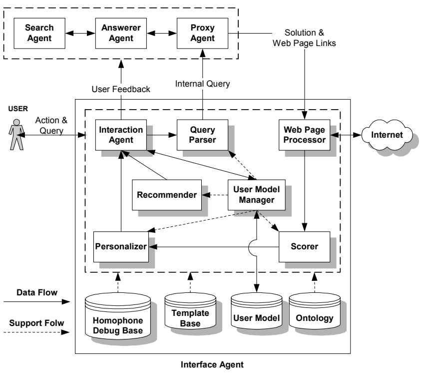


Fig. 9 Interface agent architecture

Fig. 9 shows the architecture of the Interface Agent. The Interaction Agent provides a personalization interaction, assistance, and recommendation interface for the user according to his user model, records interaction information and related feedback in the user model, and helps the User Model Manager and Proxy Agent to update the user model. The Query Parser processes the user queries by first segmenting word, removing conflicting words, and standardizing terms, followed by the recording of the user's terminologies in the terminology table of the user model. It finally applies the template matching technique to select best-matched query templates, and accordingly transforms the query into an internal query for the Proxy Agent to search for solutions and collect them into a list of FAQs, which each containing a corresponding URL. The Web Page Processor pre-downloads FAQ-relevant webpages and performs some pre-processing tasks, including labeling keywords for subsequent processing. The Scorer calculates the user's proficiency degree for each FAQ in the FAQ list according to the terminology table in his user model. The Personalizer then produces personalized query solutions according to the terminology table. The User Model Manager is responsible for quickly building an initial user model for a new user using the technique of user stereotyping as well as updating the user models and stereotypes to dynamically reflect the changes of user behavior. The Recommender is responsible for recommending information for the user based on hit count, hot topics, or group's interests when a similar interaction history is detected.

4 System Demonstration and Experiments

Our Interface Agent was developed using the Web-Based client-server architecture. On the client site, we use JSP (Java Server Page) and Java Applet for easy interacting with users, as well as observing and recording user's behavior. On the server site, we use Java and Java Servlet, under Apache Tomcat 4.0 Web Server and MS SQL2000 Server running Microsoft Windows XP. In this section, we first demonstrate the developed Interface Agent, and then report how better it performs.

4.1 System demonstration

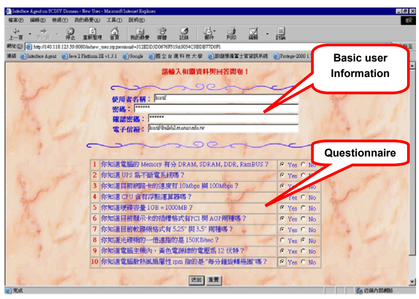


Fig. 10 System register interface

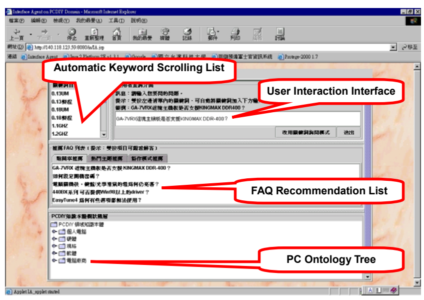


Fig. 11 Main tableau of our system

When a new user enters the system, the user is registered by the Agent as shown in Fig. 10. At the same time, a questionnaire is produced by the Agent for evaluating the user's domain proficiency. His answers are then collected and calculated in order to help build an initial user model for the new user.

Now the user can get into the main tableau of our system (Fig. 11), which consists of the following three major tab-frames, namely, query interface, solution presentation, and logout. The query interface tab is comprised of the following four frames: user interaction interface, automatic keyword scrolling list, FAQ recommendation list, and PC ontology tree. The user interaction interface contains both keywords and NLP query modes as shown in Fig. 5. The keyword query mode provides the lists of question types and operation types, which allow the users to express their precise intentions. The automatic keyword scrolling list provides ranked-keyword guidance for user query. A user can browse the PC ontology tree to learn domain knowledge. The FAQ recommendation list provides personalized information recommendations from the system, which contains three modes: hit, hot topic, and collaboration. When the user clicked a mode, the corresponding popup window is produced by the system.

The solution presentation tab is illustrated in Fig. 12. It pre-selects the solutions ranking method according to the user's preference and hides part of solutions according to his Show_Rate for reducing the cognitive loading of the user. The user can switch the solution ranking method between similarity ranking and proficiency ranking. The user can click the question part of an FAQ (Fig. 13) for displaying its content or giving it a feedback, which contains the satisfaction degree and comprehension degree. Fig. 14 illustrates the window before system logout, which ask the user to fill a questionnaire for statistics to help further system improvement.

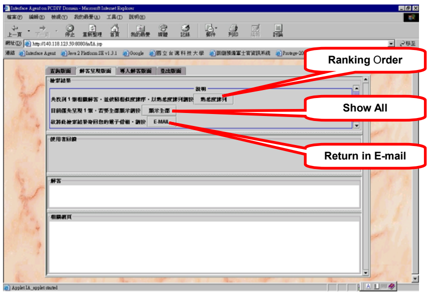


Fig. 12 Solution presentation

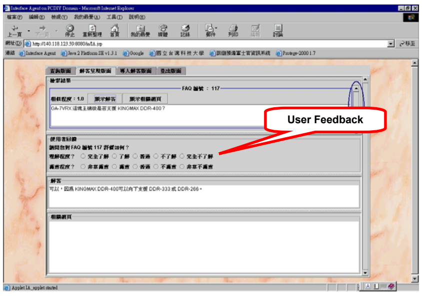


Fig. 13 FAQ-selection and feedback enticing

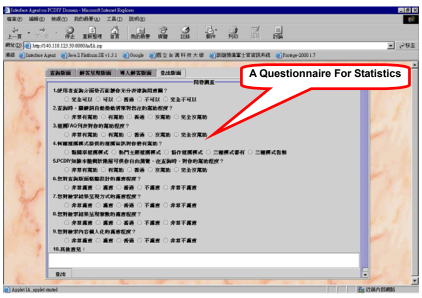


Fig. 14 System logout

4.2 System Evaluation

The evaluation of the overall performance of our system involves lots of manpower and is time-consuming. Here, we focus on the performance evaluation of the most important module, i.e., the Query Parser. Our philosophy is that if it can precisely parse user queries and extract both true query intention and focus from them, then we can effectively improve the quality of the retrieved. Recall that the Query Parser employs the technique of template-based pattern matching mechanism to understand user queries and the templates were manually constructed from 1215 FAQs. In the first experiment, we use this same FAQs for testing queries, in order to verify whether any conflicts exist within the query. Table 8 illustrates the experimental results, where only 33 queries match with more than one query patterns and result in confusion of query intention, called "error" in the table. These errors may be corrected by the user. The experiment shows the effectiveness rate of the constructed query templates reaches 97.28%, which implies the template base can be used as an effective knowledge base to do natural language query processing.

Table 8 Effectiveness of constructed query patterns

#Testing	#Correct	#Error	Precision Rate (%)
1215	1182	33	97.28 %

Our second experiment is to learn how well the Parser understands new queries. First, we collected in total 143 new FAQs, different from the FAQs collected for constructing the query templates, from four famous motherboard factories in Taiwan, including ASUS, GIGABYTE, MSI, and SIS. We then used the question parts of those FAQs for testing queries, which test how well the Parser performs. Our experiments show that we can precisely extract true query intentions and focuses from 112 FAQs. The rest of 31 FAQs contain up to three or more sentences in queries, which explain why we failed to understand them. In summary, 78.3% (112/143) of the new queries can be successfully understood.

5 Discussions and Future work

We have developed an Interface Agent to work as an assistant between the users and systems. It is also used to retrieve FAQs on the domain of PC. We integrated several interesting techniques including user modeling, domain ontology, and template-based linguistic processing to effectively tackle the problems associated with traditional FAQ retrieval systems. In short, our work features an ontology-supported, template-based user modeling technique for developing interface agents; a nature language query mode, along with an improved keyword-based query mode; and an assistance and guidance for human-machine interaction. Our preliminary experimentation demonstrates that user intention and focus of up to eighty percent of the user queries can be correctly understood by the system. In the future, we are planning to employ the techniques of machine learning and data mining to automate the construction of the template base. As to the allover system evaluation, we are planning to employ the concept of usability evaluation on the domain of human factor engineering to evaluate the performance of the user interface.

Acknowledgements

The author would like to thank Yai-Hui Chang and Ying-Hao

Chiu for their assistance in system implementation. This work was supported by the National Science Council, R.O.C., under Grant NSC-95-2221-E-129-019.

References:

[1] Chandrasekaran, B., Josephson, J.R., and Benjamins, V.R., What Are Ontologies, and Why Do We Need Them? *IEEE Intelligent Systems*, Vol. 14, No. 1, 1999, pp. 20-26.

[2] Chiu, Y.H., *An Interface Agent with Ontology-Supported User Models*, Master Thesis, Department of Electronic Engineering, National Taiwan University of Science and Technology, Taiwan, R.O.C., 2003.

[3] Hovy, E., Hermjakob, U., and Ravichandran, D., A Question/Answer Typology with Surface Text Patterns, *Proc. of the DARPA Human Language Technology conference*, San Diego, CA, USA, 2002, pp. 247-250.

[4] Lee, C.L., *Intention Extraction and Semantic Matching for Internet FAQ Retrieval*, Master Thesis, Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan, R.O.C., 2000.

[5] Noy, N.F. and McGuinness, D.L., Ontology Development 101: A Guide to Creating Your First Ontology, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Tech. Rep. SMI-2001-0880, 2001.

[6] OuYang, Y.L., *Study and Implementation of A Dialogued-Based Query System for Telecommunication FAQ Services*, Master Thesis, Department of Computer and Information Science, National Chiao Tung University, Taiwan, R.O.C., 2000.

[7] Rich, E., User Modeling via Stereotypes, *Cognitive Science*, Vol. 3, 1979, pp. 329-354.

[8] Salton, G., Wong, A., and Yang, C.S., A Vector Space Model for Automatic Indexing, *Communications of ACM*, Vol. 18, No. 11, 1975, pp. 613-620.

[9] Salton, G. and McGill, M.J., *Introduction to Modern Information Retrieval*, McGraw-Hill Book Company, New York, USA, 1983.

[10] Soubbotin, M.M. and Soubbotin, S.M., Patterns of Potential Answer Expressions as Clues to the Right Answer, *Proc. of the TREC-10 Conference*, NIST, Gaithersburg, MD, USA, 2001, pp. 293-302.

[11] Tsai, C. H., MMSEG: A word identification system for Mandarin Chinese text based on two variants of the maximum matching algorithm, Available at http://technology.chtsai.org/mmseg/, 2000.

[12] Winiwarter, W., Adaptive Natural Language Interface to FAQ Knowledge Bases, *International Journal on Data and Knowledge Engineering*, Vol. 35, 2000, pp. 181-199.

[13] Yang, S.Y. and Ho, C.S., Ontology-Supported User Models for Interface Agents, *Proc. of the 4th Conference on Artificial Intelligence and Applications*, Chang-Hwa, Taiwan, 1999, pp. 248-253.

[14] Yang, S.Y. and Ho, C.S., A Website-Model-Supported New Search Agent, *The 2nd International Workshop on Mobile Systems, E-Commerce, and Agent Technology*, Miami, FL, USA, 2003, pp. 563-568.

[15] Yang, S.Y. and Ho, C.S., An Intelligent Web Information Aggregation System Based upon Intelligent Retrieval, Filtering and Integration, *Proc. of the 2004 International Workshop on Distance Education Technologies*, Hotel Sofitel, San Francisco Bay, CA, USA, 2004, pp. 451-456.

[16] Yang, S.Y., Chiu, Y.H., and Ho, C.S., Ontology-Supported and Query Template-Based User Modeling Techniques for Interface Agents, *2004 The 12th National Conference on Fuzzy Theory and Its Applications*, I-Lan, Taiwan, 2004, pp. 181-186.

[17] Yang, S.Y., Wang, C.M., and Ho, C.S., How Do Ontology-Supported Website Models Help Web Search? Submitted to *Web Intelligence and Agent Systems: An International Journal*, 2005.

[18] Yang, S.Y., Chuang, F.C., and Ho, C.S., Ontology-Supported FAQ Processing and Ranking Techniques, Accepted for publication in *International Journal of Intelligent Information Systems*, 2005.

[19] Yang, S.Y., Liao, P.C., and Ho, C.S., A User-Oriented Query Prediction and Cache Technique for FAQ Proxy Service, *Proc. of the 2005 International Workshop on Distance Education Technologies*, Banff, Canada, 2005, pp. 411-416.

[20] Yang, S.Y., Liao, P.C., and Ho, C.S., An Ontology-Supported Case-Based Reasoning Technique for FAQ Proxy Service, *Proc. of the Seventeenth International Conference on Software Engineering and Knowledge Engineering*, Taipei, Taiwan, 2005, pp. 639-644.

[21] Yang, S.Y., FAQ-master: A New Intelligent Web Information Aggregation System, *International Academic Conference 2006 Special Session on Artificial Intelligence Theory and Application*, Tao-Yuan, Taiwan, Mar. 2006, pp. 2-12.

[22] Yang, S.Y., An Ontology-Supported and Query Template-Based User Modeling Technique for Interface Agents, *2006 Symposium on Application and Development of Management Information System*, Taipei, Taiwan, Jun. 2006, pp. 168-173.

[23] Yang, S.Y., An Ontology-Supported Website Model for Web Search Agents, Accepted for presentation in *2006 International Computer Symposium*, Taipei, Taiwan, Dec. 2006.

[24] Yang, S.Y., An Ontology-Supported Information Management Agent with Solution Integration and Proxy, *Proc. of the 10th WSEAS International Conference on Computers*, Athens, Greece, Jul. 2006, pp. 974-979.

[25] Yang, S.Y., How Does Ontology Help Information Management Processing, *WSEAS Transactions on Computers*, Vol. 5, No. 9, Sep. 2006, pp. 1843-1850.