International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:11, 2011

WebGD: A CORBA-based Document Classification and Retrieval System on the Web

Fuyang Peng, Bo Deng, Chao Qi and Mou Zhan

***Abstract*—**This paper presents the design and implementation of the WebGD, a CORBA-based document classification and retrieval system on Internet. The WebGD makes use of such techniques as Web, CORBA, Java, NLP, fuzzy technique, knowledge-based processing and database technology. Unified classification and retrieval model, classifying and retrieving with one reasoning engine and flexible working mode configuration are some of its main features. The architecture of WebGD, the unified classification and retrieval model, the components of the WebGD server and the fuzzy inference engine are discussed in this paper in detail.

***Keywords*—**Text Mining, document classification, knowledge processing, fuzzy logic, Web, CORBA

I. Introduction

COMPUTERIZED information service is becoming popular in our information society. With the rapid development of networking technology, particularly the popularity of Internet, traditional computer information systems will be transitioned into networked and distributed ones.

The Internet browser/Web server computing model is a simple but effective model. Its simplicity lies in its use of hypertext model and HTTP protocol. Its effectiveness lies in the fact that huge number of distributed information resources can be organized as a hypertext web and you can get any information you want with the browser tool. This computing model is a static one in essence. It is not suitable to some complicated applications. CGI can relieve the problem to certain extent, but as a stateless technique, CGI has its limits too. Worse still, CGI implementations do not rigidly follow a standard, making portability a problem. The invention of Java injects vigor into Internet community. As any node which has installed the Java virtual machine can downloads and runs the Java bytecodes of applications, the passive status of the browser is totally changed. The browser now can downloads not only the static files but also the executable code. The downloaded Java code can communicate with remote Java program using RMI(remote method invocation). This is a model for global distributed computing and application integration. The limit of this model is due to the tight binding of RMI and Java. From the viewpoint of implementation languages of distributed systems, this model is a homogeneous distributed computing model.

OMG CORBA is aimed to the resolution of portability and interoperability problems of distributed applications, it guarantees the transparencies of applications to object location, platform used, network protocol and implementing language[1], [2]. Its IIOP protocol is an object interoperation protocol over Internet and will be another mainstream protocol for future Internet.

In this paper we will present the design and implementation of WebGD, a CORBA-based document classification and retrieval system on the Web. WebGD has a client/server structure and combines the merits of Web, CORBA and Java applets. The server of WebGD performs fuzzy document classification and retrieval operation using a knowledge-based approach. Web browser user can download the WebGD client Java applet which then requests WebGD server's service via CORBA ORB using the IIOP protocol. Results of the service are sent back to browser in the form of hypertext files.

The structure of the paper is as follows. Section 1 is the introduction. Section 2 gives the architecture of WebGD and shows how it works. Our emphasis is on the document classification and retrieval server which is discussed in section 3. A formal model for unified document classification and retrieval is presented which has the basis on fuzzy logic. The core of the server is an engine that implement the model. Section 4 is the conclusion.

II. WebGD Architecture

Fig.1 shows the architecture of WebGD. The overall framework is an Internet browser/Web server architecture using HTTP protocol to communicate between them (see the upper part of Fig.1). The WebGD system has two parts, the client and the server. The client and the HTML file that the client code is anchored are shown at the upper box of the WebServer in Fig.1 and the server components are shown at the lower box of the WebServer in Fig.1.

The client of WebGD consists of three parts: (1) the WebGD client application code; (2) the IDL generated client stub which provides down-call interface with the ORB engine (runtime kernel) to appication code; (3) the ORB engine which performs real communication between client and server. The client program is pure Java code, and thus can be downloaded onto any machine supporting Java virtual machine(JVM).

The server program of WebGD also has three main parts: (1) the WebGD server implementation code, i.e., the wrapper code and the WebGD server proper; (2) the IDL generated server skeleton which provides upcall interface with the wrapper code to the ORB engine; (3) the ORB engine which performs real

Fuyang Peng, Bo Deng and Chao Qi are with the Software Division at Beijing Institute of Systems Engineering, Beijing 100101, China (e-mail: fuyang_peng@ sina.com).

communication between client and server. The server is implemented in C++.

End user uses an Internet browser such as Netscape Navigator and Internet Explorer and gives the URL of the WebGD system. The Home page WebGD.html in which there is an anchor (an Applet tag) is downloaded. This applet tag is anchored to a CORBA-enabled Java applet. After the home page is loaded❶, the browser recognizes the Applet tag and requests the JVM to download and run the WebGD client application code❷. When interpreting the code, JVM automatically downloads the IDL client stub❸ , and finally downloads the ORB classes❹ , thus getting support of CORBA ORB core. All these are transparent to end user and code downloading is a standard "class loader" function of the Java interpreter. The CORBA-enabled client applet communicates with the WebGD server with IIOP protocol❺❻。WebGD server generates a html result file to inform the result of a classification or retrieval request. The file name is returned as a function return value to the event procedure on the client side and the client code downloads the file to the browser for displaying according to the current context. ❼。

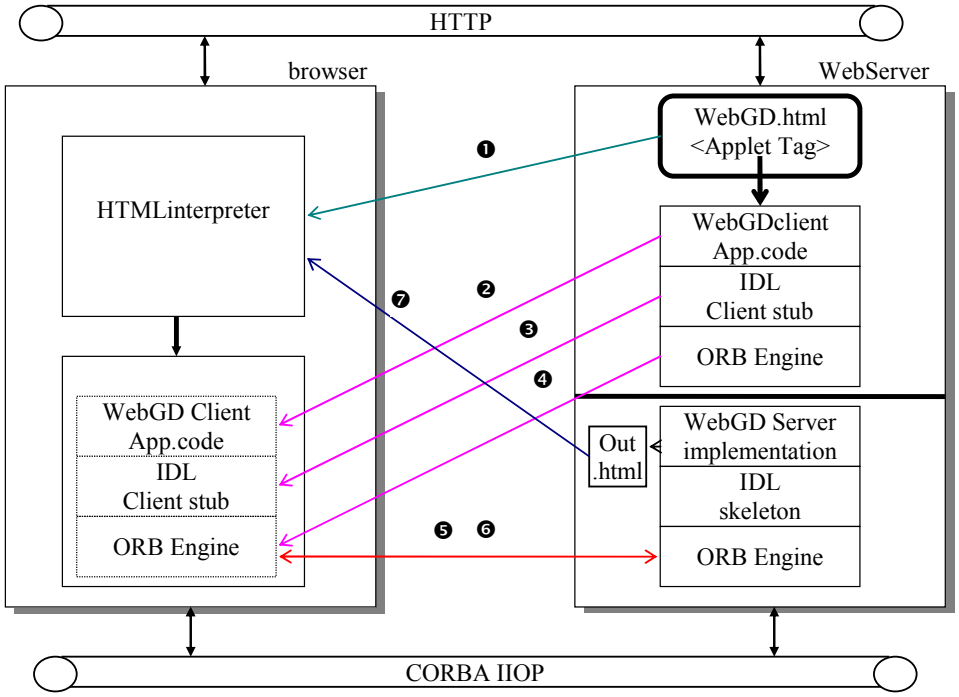

Fig.1 The WebGD architecture

III. Document Classification and Retrieval Server

Theoretical basis of the WebGD server is the unified classification and retrieval model we propose. The basic idea of the model is: (1) the knowledge used in classification and retrieval process is organized as a conceptual hierarchy, a fuzzy bounded algebraic lattice; (2) the document text is preprocessed to get the initial evidences to support or deny concepts; (3) the classification and retrieval process is modeled as an evidence-driven, conceptual hierarchy knowledge-based fuzzy reasoning process; (4) results of classification and retrieval are represented as fuzzy sets on different base sets or the cut set of them [3], [4].

A. Unified Classification and Retrieval Model

Let $\Sigma = \{ a_1, a_2, ...a_n \}$ be the set of Chinese characters, $f_w: \Sigma^* \to \{0,1\}$ be the function to decide whether a Chinese character string is an acceptable Chinese word, and $W=\{\alpha \mid \alpha \in \Sigma^*, f_w(\alpha)=1\}$ be the set of all acceptable Chinese words. Suppose $D = \{d_1, ..., d_m\}$ be the set of documents to be processed. To observe whether a given document is relevant to a concept, We can follow a systematic approach. First a Chinese word segmentation procedure is executed and the evidences are extracted from documents with the help of a system word list (if any). Then an inference procedure is carried out with the conceptual hierarchy (detailed later) as the knowledge base and the evidences extracted from the documents as evidence base. And finally relevance value of concept c to document d_i is

obtained. As relevance value is not accurate, fuzzy inference technique is used in our system. The WebGD model is based on the basic idea above and formally defined as follows.

Definition 1. The WebGD model is the ten-tuple given below.

$$< D, W, SL, KB, OP, V, Q, f_{seg}, f_{ev}, f_o >$$

where *D* is the set of documents to be processed by the WebGD system,

W is the set of all legal Chinese words,

SL is the set of system word lists, *KB* is the set of knowledge bases in the form of conceptual hierarchy,

OP is the option set for fuzzy inference, the element of which has the form of (*and_op, or_op, detach_op, thd*) with *and_op* representing conjunct operator, *or_op* disjunct operator, *detach_op* detachment operator , and *thd* threshold value,

V=[0,1],

Q is the set of concepts,

$f_{seg} : D \to P(W)$ is the Chinese word segmentation function,

$f_{ev} : P(W) * SL \to P(W)$ is an evidence acquisition function,

$f_o : P(W) * Q * KB * OP \to V$ is a concept evaluation function.

Now we give further explanation to this definition.

There are two kinds of system word list. One is called insignificant word list(IWL), the other significant word list(SWL). The behavior of function f_{ev} is closely related to this list. Assume all the words in document *d* make the set *Wd*, system word list used is *sl*. function f_{ev} is defined as follows.

$$f_{ev}(Wd, sl) = \begin{cases} Wd - sl & \text{if the type of } sl \text{ is IWL} \\ Wd \cap sl & \text{if the type of } sl \text{ is SWL} \\ Wd & \text{if } sl \text{ is empty} \end{cases}$$

Relevance value of concept *c* to document *d* is

$$\alpha_{cd} = f_o(f_{ev}(f_{seg}(d), sl), c, kb, op)$$

where $d \in D, sl \in SL, c \in Q, kb \in KB, op \in OP.$

At the time we perform document retrieval and classification, *sl*, *kb*, and *op* usually are invariant, so the above formula can be rewritten in the following convenient form.

$$\alpha_{cd} = f_o'(c, d)$$

Let all the fuzzy sets defined on D be $\widetilde{S}$ (D), all the fuzzy function defined on C be $\widetilde{S}$ (C), we have the following definitions for document retrieval and classification.

Definition 2. Concept retrieval $\widetilde{f}_r$: Q→ $\widetilde{S}$ (D) is defined as

$$\widetilde{f}_r(q) = \{f_o'(q, d_1)/d_1, ..., f_o'(q, d_m)/d_m\}$$

Definition 3. document classification $\widetilde{f}_c$: D→ $\widetilde{S}$ (C) is defined as

$$\widetilde{f}_c(d) = \{f_o'(c_1, d)/c_1, ..., f_o'(c_n, d)/c_n\}$$

We have mentioned of conceptual hierarchy in the above discussion. Now we give a formal definition for it.

Definition 4. Suppose (***N***, <) is a bounded lattice and the number of elements in ***N*** is not less than 3,let *u* and *l* be the whole upper bound and whole lower bound respectively. We define

Nodes=N-{u,l},

Edges={(*a,b*)|*a,b*∈*Nodes*, *a*≠ *b*, *a*<*b*.and for all *c*∈*Nodes*, that *a*<*c* and *c*<*b* can not hold}.

Then *Nodes* is called the term set, and *Edges* the link set.

Definition 5. Conceptual hierarchy *CH* is a quadruple *(Nodes,Edges,Cluster,f)* where

(1) *Nodes* is a term set

(2) *Edges* is a link set

(3) $Cluster \subseteq Nodes * P(Nodes)$, which satisfies condition: if *(a,s)*∈*Cluster*, *s* cannot be empty.

(4) *f: Cluster→ [0,1]* is a mapping.

Definition 6. Let *CH=(Nodes,Edges,Cluster,f)* be a conceptual hierarchy. n∈ Nodes is called a concept if there exists an *m*∈*Nodes* that makes *(n,m) in Edges*.

Definition 7. Let*CH=(Nodes,Edges,Cluster,f)* be a conceptual hierarchy, n∈Nodes is called a keyword if for all m∈Nodes, that (n,m)∉Edges.

B. Structure of the WebGD Server

The WebGD server is designed as a four tiered architecture. The first tier is a support layer, including a database management support sub-system DBMS/GD. DBMS/GD provides such functions as database schema definition, database manipulation and index management.

The second tier is the kernel of WebGD server. It is composed of three major sub-systems, i.e., the document preprocessing and evidence acquisition sub-system (DPEA), the knowledge base management sub-system for conceptual hierarchy(KBCH), and the concept evaluation sub-system based on fuzzy logic(FCES). DPEA first conducts word segmentation, that is, transforms the string-form text into word list (for Western languages which have explicit delimiters between words, this step is unnecessary), then recognize the words and their features, such as paragraph number, sentence number, word number, whether they are in the title, whether they have significant implications. After this, DPEA decides if the words extracted should be inserted into the evidence base as evidences according to the type and contents of the system word list. KBCH is in charge of transformation of knowledge base from external form to internal form of conceptual hierarchy, controlling the loading of KB, and management of core image for KB. FCES gives the fuzzy reasoning mechanism. It conducts

fuzzy inference and computation via the KB, making use of the evidences in evidence base.

The WebGD server provides means for selecting fuzzy operators and controlling outputs. Two types of optimization are employed. one is a macro-level heuristic approach, and another is a micro-level memory-based approach.

The third tier consists of function modules and tools. Functions include document registration, document classification, and document retrieval. Tools include listing tools, browsers, editors, user management tools, options management tools, index re-construction, and on-line help facility.

The fourth tier is the wrapper tier for CORBA objects. The functions and global variables provided above are stored in a dynamic link library(DLL). To make these functions and variables invoked as CORBA object attributes and methods, they must be wrapped into CORBA object form according to the WebGD.idl interface definition file. The fourth tier performs all this stuff.

C. Knowledge-based Classification and Retrieval

Text classification and retrieval are the main functions of WebGD. As the model in section 3.1 shows, unlike traditional Boolean retrieval systems, the result of classification or retrieval operation of WebGD is a fuzzy set, which we think reflects the inherent inaccuracy of these operations much better.

The classification and retrieval functions in WebGD are uniformly supported by a conceptual hierarchy based fuzzy evidential reasoning engine. The engine takes as input the evidential knowledge of the text, the conceptual hierarchy lattice and the classification criteria set, computes and propagates the relevance value according to the fuzzy operators selected. The only difference between classification and retrieval implementation lies in the control structure which controls the invocation of the inference engine, the classification criteria set being used as loop control variable in classification and the document set processing as loop control variable in retrieval. In the following emphasis is put on the fuzzy inference engine of the WebGD server.

As we mentioned above, knowledge-based evidential reasoning is employed in WebGD. We take certain text patterns (initial evidences) as the inputs to the leaf nodes of the conceptual hierarchy (knowledge base), the engine then computes and propagates the relevance value of evidence from bottom up toward the lattice top. If a node of the lattice has a computed relevance value of ***rv***, we can say the concept are supported by the text evidences to a degree ***rv*** (wrt the knowledge base used). A concept is ***rv***-supported if the relevance value ***rv*** of the concept node is above a set threshold value, otherwise the concept is said to be ***rv***-denied.

The reasoning process in WebGD is actually an evaluation process of the conceptual hierarchy. The process consists of three distinct parts, i.e., the evidence primitive evaluation, the rule evaluation, and the concept evaluation.

C.1 Evidence primitive evaluation

Evidence primitives are represented in the conceptual hierarchy lattice as nodes with zero out degree. There are two types of evidence primitives: keywords and operators. Keywords are text words or phrases appeared in the documents being processed and taken as evidences. Operators include such operators as logical operators, range operators, positional operators, words count operators and free match operators. Evaluation of evidence primitives is implemented by access procedures which directly access the evidence base.

The value of a keyword is defined as the weighted sum of the keyword's occurrence frequencies at the various positions.

Evaluation of an operator is much more complex. Theoretically it is equivalent to a join or join-like operation of variable number of relations. There are two key issues to be solved to implement. The first issue is tackled with a cursor locking technique combined into traditional relational database engine. The second issue is solved by properly controlling the recursive evaluation process of multi-relations join (the relations joined may be the same in some instances).

C.2 Rule evaluation

Conjunct operators should meet the criteria of triangular norm[3], [5]. In our implemented system we have taken the following three.

1. $rv(A \wedge B)=\max[0,rv(A)+rv(B)-1]$
2. $rv(A \wedge B)=rv(A).rv(B)$
3. $rv(A \wedge B)=\min[rv(A),rv(B)]$

There are five kinds of detachment operators in our system. They are:

1. $rv(B)=\min[rv(A),rv(A=>B)]$
2. $rv(B)=\begin{cases}\min[rv(A),rv(A=>B)] & \text{if } rv(A)+rv(A=>B)>1\\ 0 & \text{otherwise}\end{cases}$
3. $rv(B)=rv(A).rv(A=>B)$
4. $rv(B)=\max[0,rv(A)+rv(A=>B)-1]$
5. $rv(B)=\max[0,(rv(A)+rv(A=>B)-1)/rv(A)]$

C.3 Concept evaluation

Disjunct operators should be triangular conorm[5],[6]. In our current implementation, the following three are taken.

1. $rv(A \vee B)=\min[1, rv(A)+rv(B)]$
2. $rv(A \vee B)=rv(A)+V(B)-rv(A).rv(B)$
3. $rv(A \vee B)=\max[rv(A),rv(B)]$

IV. Conclusion

This paper presents the design and implementation of the WebGD, a CORBA-based document classification and retrieval system on the Web. The WebGD makes use of such techniques as Web, CORBA, Java, NLP, fuzzy technique, knowledge-based processing and database technology. Unified classification and retrieval model, classifying and retrieving with one reasoning engine and flexible working mode configuration are some of its main features of WebGD. Document classification and document retrieval share the same

core engine. The engine is knowledge-based, but it can work with an empty rule base(in this case, the system is much like a full text retrieval system). The rule base can be incrementally constructed, with a structure having common KB and domain KB separated. The rule base can be incrementally loaded or overlapped dynamically. The work presented in [7] is also based on conceptual KR, but it is designed for document retrieval only and is not flexible. The knowledge base in WebGD is ontology-based, similar to [8-10].Organization of the knowledge base is two-layered, one Common, one Specific, which is very different from [11].

Initial experiment shows very positive results. We choose the written motions from the National Political Consultative Committee for experiment. The 400 motions are to be sorted into 13 topic categories. Using our system, 91% of the motions are correctly or nearly correctly classified. As the ontological knowledge base is improved, the result is expected to be more accurate.

ACKNOWLEDGMENT

The authors would like to thank Professor Xingui He for his guidance in the research direction of this work. We also acknowledge the fund support by the Chinese Natural Science Foundation and Project 863, the National High-tech Project of China.

REFERENCES

[1] Vinoski, S., 1997. CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments, *IEEE Communications Magazine*, 14(2).

[2] Peng,F.,1998. Distributed Applications Develop-ment Using CORBA , *Technical Report.*

[3] Peng, F. and He, X., 2002, Conceptual Hierarchy: Formalism, Implementation and Applications, *Proc. Int. Conf. on Chinese Information Processing*,Vol.1.

[4] Peng, F. and He, X., 2005. Text Analysis and Information Retrival, *Proc. The Nat'l Jt. Conf. On Computation Linguistics*, 2005 .(in Chinese).

[5] Mizumoto,M.,et al., 1982. Comparison of fuzzy reasoning methods, *Fuzzy sets and systems*, Vol.8,No.3.

[6] Yager,R.R.,et al., 1987. *Fuzzy sets and applications: selected papers by L.A.Zadeh,* John Wiley & Sons,Inc.

[7] Tong,R.M.,et al., 1989. A knowledge repre-sentation for conceptual IR, *Int. J. Intelligent Systems*, 4(3), pp.259-284.

[8] Pablo Castells, Miria Fernandos, David Vallet, Phivos Mylonas and Yannis Avrithis, Self-tuning Personalized Information Retrieval in an Ontology-based Framework, Lecture Notes on Computer Science, Volume 3762, 2005, 977-986.

[9] Gabor Nagypal, Improving Information Retrival Effectiveness Using Knowledge Stored in Ontologies, Lecture Notes on Computer Science, Volume 3762, 2005, 980-989.

[10] Dolf Trieschnigg, Proof of Concept: concept-based biomedical information retrieval, ACM SIGIR Forum, Vol.44, No.2, December 2010.

[11] Ivan Cantador, Alejandro Bellogin, and Pablo Castells, A Multilayer Ontology-based Hybrid Recommendation Model, AI Communication, Vol.21, No.2-3, 2008.