Research on Component Retrieval Methods

Yan-pei Liu
Information Engineering College, Henan Institute of Science and Technology, Xinxiang, China
Email:32933415@qq.com

Yuesheng Gu and Chen Jun
Information Engineering College, Henan Institute of Science and Technology, Xinxiang, China
Email: {hz34567@126.com, 47594409@qq.com }

Abstract — With the development of the component technology and the expansion of component library, representing and retrieving components, the two core technologies of component library management have been a focus on research hot issues in researches. Based on the present widely-used component faceted classification description method and its features, and from the different requirements and ways of component retrieval that were used by component re-users, three component retrieval ways based on browser, facet and term, the five corresponding matching models and two matching algorithms are proposed. The theoretical analysis and experimental results show that the three kinds of retrieval methods used in large-scale component library component retrieval, will meet the component reuse of various searching requirements and that these ways are feasible in retrieval efficiency..

Index Terms — general attribute; Component retrieval; component library; retrieval efficiency; faceted classification;

I. Introduction

Component classification and retrieval technology are two key technologies of component library management system.. Component classification methods can be divided into artificial intelligence methods, hypertext methods and information science methods. Information science method has been a successful class in the actual reuseing projects application nowadays, and enumerates, facets, attribute values; keywords and text retrieval are more common. Among them, the expressions of component facets and the component retrieval technology based on those indications have attracted the attention of software reuse. REBOOT and NATO made their own reusable software component classification scheme. Domestic Jade Bird Component Library is made primarily of facet classification, multi-mode classification of combining method give a classification description of the components [1].

Component retrieval actually is the process of search condition and component information matches in component library. Therefore, the match between them is the key component retrieval. Currently, the component retrieval with the facet description mainly draws supports from traditional database retrieval, combined with the use of thesaurus and hierarchy of faceted term space, all of which will achieve are relaxed matching of component.[1, 2]. But the component library is a special structure of the database (using facet term of description method), and its object-oriented are not general users, but components' re-user, so it's retrieval method differ from those searches of general database or Library, The differences go as follows:

1. Component retrieval need to provide a variety of component retrieval methods to adapt to different levels of component re-user;
2. Component retrieval should be capable of dealing with the inquiry conditions proposed by the component re-users, and these abilities aimed to correct or extend the conditions of users' inquiries, so as to improve rates of fulfillment, precision and efficiency;
3. Component retrieval should have the matching methods of storage structure of component facet classification;
4. Component retrieval should have the processing power of searching results, and then can calculate the degree of matching search results, which need to be in proper order;
5. Component retrieval had better provide meaningful reference information for users of reused components.

By understanding the component reuse historical data and the feedback data of other re-users' component retrieval, component retrieval makes re-users understand the components better, and understand the focused components' information in practical application. Forbes Gibb introduced XML as a markup language of the component facets' description in their reusable software component research projects [3]. What's more, they use XML retrieval language, XML-SQL, to complete the task of component retrieval. Literature [4] also proposed the descriptive methods and retrieval method based on XML. Literature [5] and [6] proposed descriptive methods and retrieval methods based on ontology. Literature [2] and [7~10] proposed descriptive methods and retrieval methods based on facet. However, the data of XML storage not only lack index and data modification, data security access control, complete transactions and data consistency control, multi-user access mechanism, triggers, concurrency control and other features, but also existed shortage in storing efficiency compared with the technology of database. As a consequence, with regards to the requirements of the quantity of information, this large number of users, data integration and high

doi:10.4304/jsw.7.7.1633-1640

performance for the component library, making use of the technology of database is rather a better choice. Body structure has a huge challenge of owning a great number of attributes and association and the establishment of the body is complex, expensive and very time-consuming project [11, 12]. Currently component library mostly use facet description, therefore the described method and facet search are now research hotspot. However, the current component retrieval methods are mostly based on keywords. They do not exactly follow the rules of described facets and components for reuse in a variety of search methods are also inadequate.

This thesis combines the component facets descriptive methods and their features, According to component re-user with different search methods, it proposes three retrieval methods of component which are based on the view searching methods, faceted searching methods and terms retrieval methods, gives the five matching model and two matching algorithms and the experimental analysis are carried out through three main factors which affect the retrieval efficiency.

II. Helpful Hints

A. Faceted classification and coding

1. Faceted Classification and description of the tree faceted classification

A connected graph without rings called tree, expressed by T : $T=(V,E,root(T))$. Among them, V denotes a finite node set, $root(T)\in V$ is the root of the tree, E is the edge set, it is a binary relation on V . It satisfies the anti-reflexive, ant symmetry and transitivity. If $(u,v)\in E$, says u is v node of the parent, denoted as $u=parent(v)$ or $v=parent(u)$, all child nodes of u are denoted by $C(u)$. A tree must meet the following three conditions [2]:

(1) The root node not exist father node.

(2) In addition to other root, other node has only one parent among v .

(3) Each non-root $v\in V$, exists $root(V),v\in E^*$, among them, E^* is the transitive closure of E .If the two nodes v_1 v_2 and $(v_1,v_2)\in E^*$, then says v_1 is ancestor node of v_2 . Denoted by $v_1=ancestor(v_2)$ or $v_2=descendant(v_1)$. Ancestor of all nodes of u is denoted by $A(u)$, u all the descendants of node set is denoted by $D(u)$, let $u\in v$, among T , to u is the root of the sub-tree denoted $T[u]=(V',E',u)$, among them $V'=[u]\cup D(u), E'=E\cap(V'\times V')$.

For a faces classification scheme, facets and sub-facets are mapped to the corresponding node's parent and child nodes in the tree, and corresponding term mapped to leaf node. This paper will represent all the components information with tree structure of Figure 1, among them, Root node is a virtual node. It can be seen from the figure that faceted classification describes that tree has its particularity, mainly embodied in the following four:

(1) Term node, that leaf node is the virtual node, expressed by X_y , its meaning of term information under y facet, to wit $y=parent(X)$.

(2) Node of the label value is $lable(x)$, for example, a node label value recorded as $lable(a)$. Facet node of the label value is encoding, term node of label value is the specific term information.

(3) As the final layer of sub-facets under the terms only unique values, the term value may be empty or a specific value, so every last layer of sub-facets has only one leaf node, leaf node of the label value is empty or the specific value for the term.

(4) If $Root\in(ancestor(v))$ then v , $parent(v)$ and $A(v)$ form a facet tree is denoted by F , among them, v is a leaf node, as shown figure 1, the node $Root,a,c,X_c$ formed a facet tree.

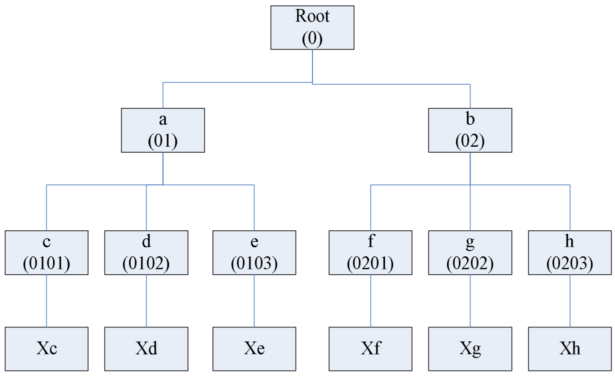


Fig.1 component facet classification of description tree and encoding

A component with the application environment, function, applications to describe the three facets, facets of the sub-application environment are the database and operating system terminology, the term of the sub-function File Store, applications, sub-terms for the MIS, a database of sub-terms for SQL Server, operating system, the sub-terms as Unix, the above methods can be constructed as shown in Figure 2.

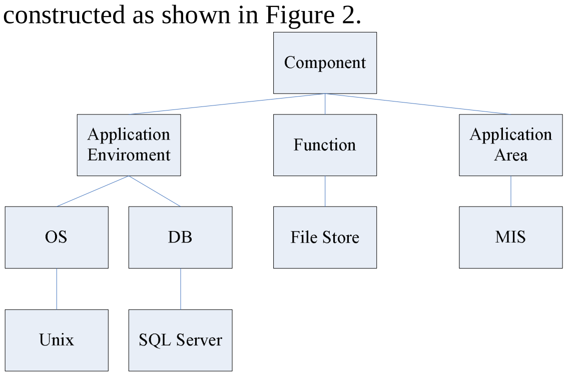


Fig.2 Description of the tree component facets

<Component>
<Application Environment>
<Operation System>UNIX</<Operation System >
<Database>SQL Server</ Database >

</Application Environment>
<Function>File Store</ Function >
<Application Area>MIS</ Application Area >
</ Component >

2. Encoding

The facets, sub-faceted and terms of component description have hierarchical relationship and components re-users can balance the acceptance and rejection towards the required abstract levels of components.. In order to facilitate the retrieval and its implementation, according to the hierarchy of faceted classification, the component re-users begin to encode the component tree. Encoding strategies are as follows:

(1) A one-to-one correspondence between facet and coding, coding is their unique identifier.

(2) Regarding the double-digit (from 01 to 99) as a hierarchy in the coding, every two bit represents the relative position in the hierarchical tree, in which the Root node is encoded as 0.

(3) The sub-faceted with the same parent faceted can be encoded according to the order of the increasing1.

(4) Because the term is virtual child node, its label value can be replaced by its code identification, and it is not encoded.

According to the above coding strategy, we can derive component code as shown in Figure 1.

3. Component model

Component model has the external interface and internal structure:

(1) External interface: Its external interface is the reuse of components to provide basic information, including component name, function description, foreign function interface, the required components, parameterized attributes. External interface components with the outside world is a set of interaction points, indicating that the components provided by those services (messages, operations, variables).

(2) Internal structure: The internal interface consists of two aspects. The internal members include the specific members and virtual members, including membership among members of the association, as well as the interface between the internal and external members of the interconnection.

Component library entity relationship diagram as shown in Figure 3.

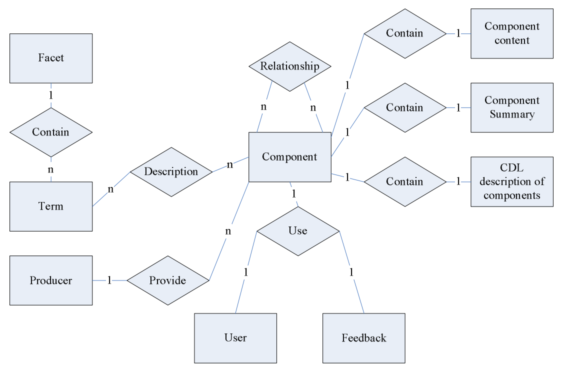


Fig. 3 Component library entity relationship

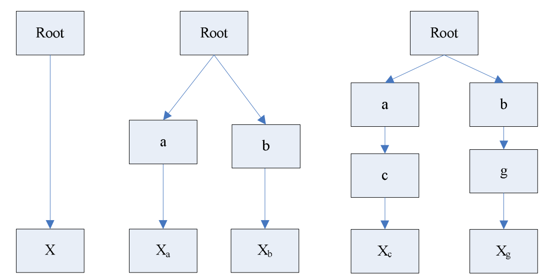
B. Search method

1. Construction and expansion of search criteria

At present retrieval system has the general support of AND, NOT, OR Boolean logic operation and combinations of expression. Users can do not select facet, but direct input inquires the conditions according to their needs, or select the corresponding facets and enter the search conditions in each facet. However, when the users input search conditions, they can not directly be retrieved because the problem expressed of the users expression of computer have some differences, which need to pass the system treatment and expansion to become a real retrieval conditions. By analyzing the user's query processing, get some sense of an independent search words, the facets, sub-surface appearing in the query, should be engraved into the corresponding levels of parent-child node, and search keywords should be mapped to a leaf node labels. When users retrieve components based on viewing searching methods, they do not need to select the facets, just input inquire conditions. Based on faceted search method the users need to select some of non-final layer sub-facets (Figure 1 facet c to h), then enter each facet of the search condition. Based on terms retrieval method they needs to select final layer sub-facets (Figure 1 facets c, d, e, f, g, h), and then enter each child facets of inquires conditions. Three kinds of search methods correspond to retrieval tree structure as shown in figure 4(a, b, c). Search criteria after extended will get n isomorphism retrieval tree, as shown in figure 5. Among them, X_y^n is term X_y of the n-th extension node.

2. Component retrieval matching model

Now assume that Q is shown in Figure 4 for a search tree. T is shown in Figure 1 describes the facets of a component tree.



(a) Construction retrieval tree based on view search method (b) Construction retrieval tree based on faceted search method (C) Construction retrieval tree based on terms retrieval method

Fig.4 Three kinds search methods of search tree

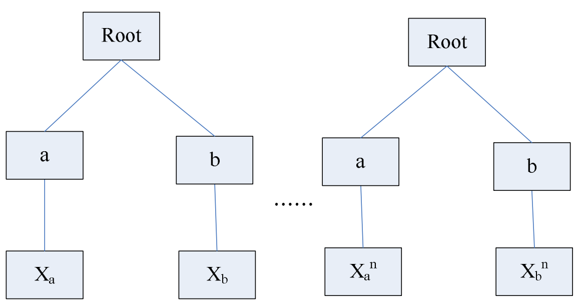


Fig.3 expansion of search tree based on faceted search method

Definition 1: X node in Q and all nodes in F_{tsub} matching.

Definition 2: X_y node in Q and all leaf nodes in F_{tsub} matching.

Definition 3: Given: U is the set of nodes F_{qsub}, V is the set of nodes F_{tsub}. If $U \subseteq V$, then F_{qsub} and F_{tsub} equal to the tree structure

Instruction: By definition 2 and 3 we can learn figure 2 (b) of retrieval facets tree formed by $Root, a, X_a$ and figure 1 of three facets tree structure composed by $Root, a, c, X_c$, $Root, a, d, X_d$ and $Root, a, e, X_e$ are equal.

Definition 4 (based on browser retrieval method matching) if meet the following conditions, then called Q and T are matching based on browser retrieval method.

$$lable(T_t) \approx lable(Q_t) \tag{1}$$

Among it, F_{tsub} leaf node is T_t, F_{qsub} leaf node is the Q_t, $\approx$ indicates the label value is similar of two leaf nodes, to wit fuzzy equal.

Definition 5 (based on faceted search method of partial matching (FPM)) if meet the following conditions, then called Q and T are partial matching based on faceted search method.

$$\exists F_{qsub} = F_{tsub} \tag{2}$$

$$lable(T_t) \approx lable(Q_t) \tag{3}$$

Definition 6 (based on faceted search method of total matching (FTM)) if meet the following conditions, then called Q and T are total matching based on faceted search method.

$$\forall F_{qsub} \in F_q, \exists F_{tsub} = F_{qsub} \tag{4}$$

$$lable(T_t) \approx lable(Q_t) \tag{5}$$

Instruction: By definition 2 and the structure of Q can be obtained based retrieval methods match needs of faceted tree in the same context, then the corresponding matching terms. If any F_{qsub} in the F_q can be found the structure equal of F_{tsub} in the F_t, then it is the completely match, otherwise it is part of the match. Figure 2 (b) and Figure 1are the facet retrieval methods completely match.

Based on Term retrieval method and based on faceted search method of matching only differ on the structure of F, can be seen as its special case. Therefore, TTM (based on term retrieval method of total matching) and FTM are the same definition; TPM (based on term retrieval method of partial matching) and FPM are the same definition. Figure 2 (b) and Figure 1 are the completely matching based on based on faceted search method.

Definition 7 (matching cost) Tree editing operations have their own editorial costs. In the tree matching process, the cost of various operations, and will form the tree matching cost. The following is the calculation of matching:

$$c = \Sigma\lambda(delete) + \Sigma\lambda(insert) + \Sigma\lambda(relable) \tag{6}$$

Among them, $\Sigma\lambda(operate)$ means to match the price in the sum of all edit operations. As the operator can modify the label is equivalent to a combination of insert and delete operations, so (6) can be simplified as (7):

$$c = \Sigma\lambda(delete) + \Sigma\lambda(insert)) \tag{7}$$

The smaller the value matching the price usually indicates higher accuracy of matches. It is noteworthy that the editing operations corresponding to the editing system allows the maximum cost and the cost of the system is designed to match the personnel or management setting. When the tree is less than the cost of system settings match the maximum time, it's a successful match. To sum up，It is not difficult to see with the same keyword search, the relationship between them is: $TTM \subseteq TPM \subseteq FTM \subseteq FPM \subseteq BM$. The five kinds of matching the constraint conditions of longitudinal from parent relationship (facets and sub-facets) relax to the ancestors (facets and terms) relationship, Horizontal from facets tree structure completely equal constraints (TTM and FTM) relax to facets tree structure of partial equality constraint(TPM and FPM), until relaxation of the unconstrained (BM). The five matching components constitute rich layers of facets described match models; it can well meet components re-user of the different retrieval needs.

C. Component retrieval matching algorithm and analysis

1. Component retrieval matching algorithm

This paper gives retrieve the matching algorithm 1 and algorithm 2, which Algorithm 1 for BM, FPM and TPM, algorithm 2 for FTM and TTM, the specific algorithm is as follows.

Algorithm 1. Input: Search tree M
Known conditions: Component tree set M
Output: Match the result set M

```
1   M = φ
2 do
3 {
4     for (p=1;p<=| F_q |;p++){
5       if ( F_stub = F_qsub && lable(T_t) ≈ lable(Q_t) )
6             {Put  T_i  into M; // i is the i-th component tree
7                 i++;
8             }
9        }
10     }
11     While (i<=| T |)
12     out M;
```

The algorithm for tree Q and tree T of the facet tree matching and similarity matching term labels (line 5), if a match successful, it will be recorded into the set M. The algorithm time complexity maximum is $O(m \bullet n \bullet f)$, the minimum is $O(n)$. Among them, m is the number of search tree facet tree, n is the number of component tree in the component library, and f is the number of facet tree for each component tree.

Algorithm 2. Input: Search tree M
Known conditions: Component tree set M
Output: Match the result set M

$M=\varphi$
bool sign=1;
do
{sign=1;
for (p=1;p<=| F_q |;p++)
{
if ($F_{stub} = F_{qsub}$ && $lable(T_t) \approx lable(Q_t)$)
{i++;
}
else
{sign=0;
p=| F_q |+1;
}
}
if (sign==1)
{
Put T_i into M;
}
}
While (i<=|T|)
out M;

The algorithm woks for the facet trees' matching and similarity of term labels' matching for tree Q and tree T_i, if both of them match well, it will be recorded into the set M. Minimum time complexity of the algorithm is $O(n)$, it means the T in the component tree does not match Q the minimum is $O(m \bullet n \bullet f)$.

2. The sort of the retrieval results

In the search results, due to the diversity of component libraries they must be retrieved in more similar results, so the need to propose a sorting mechanism to retrieve search results[11-14] in accordance with the conditions of similarity to low order. Sorting search results is by calculating the priority. By defining the components in some facet of the important components of setting the weights to calculate the priority and descending order for users to choose.

Assuming retrieve the result set one component C, its faceted set of nodes as FC, Q faceted search conditions set for the node FQ, Participate in similarity calculation of the node set for $FP \cap FC \cap FQ$, Node FP_i weight notes for PF_i, Then C is denoted by P(C) is the priority [15, 16]:

$$P(C) = \frac{\sum_{i=1}^{n} PF_i}{n} \tag{8}$$

Selection criteria for the search facet can not serve as the basis for calculating the priority. Because if the retrieval facets as the priority basis, they will go against wills of the re-users. For example, component reuse will name as search criteria, you want the weight of the larger component name, so it can not be set by the system's weight in the calculation.

3. Retrieval Model

For a problem space, component re-users enter the query to select query mode based on their understanding of the problem space; and then system will query can be expanded into a component library system understand the search criteria; by the matching algorithm on the search conditions in the public component library or a business component library to retrieve the components match or approximate match; the returned search results are sorted, the final results presented to the user. A component retrieval process model shown in Figure 6:

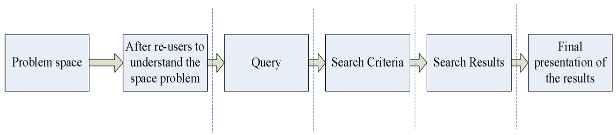


Fig. 6 Component retrieval process model

Figure in the level of understanding of the problem space and the accuracy of the expression of demand depends reuse the skills and experience, and component classification coding, retrieval conditions and extended coding accuracy and efficiency of the matching algorithm is a component library and search tools need to consider.

4. Algorithm Analysis

Through the algorithm analysis above, we can draw some the features of the algorithm which are as follows:

(1) The algorithm divides the matching process to two steps. Firstly, match facet tree structure, if the match passes, then the similarity match of leaf node label will be conducted. According to coding properties of the facets and Definition 2.3, facets tree match actually is the similar matching of Q and T leaf node of parent prefix, which can reduce the unnecessary matches between the ancestors of the node, and will be able to improve the matching efficiency.

(2) Expansion of the searching conditions can increase the precision and recall rates of the components. In order to achieve this point, according to the literature [13], it will be needed to build terms semantic library in component library. And the term conditions which has been expanded is the "or" relationship, search criteria will be extended to connect a new search conditions with the keyword "or", one-time comparison, to avoid a multiplicity of statements.

(3) It is easy to see that the search sub-condition for "or" relationship corresponding toalgorithm1, the retrieval sub-condition corresponding to algorithm 2 is "and" relations. If the search condition contains both common "and" and "or" , then the two algorithms will be used in combination, in other words, firstly, make "and" conditions which are around "or" form a new facet tree search conditions, and then carry on the " or" search terms. Search conditions do not convert multiple search tree not going against the wills of the component

re-users it is more convenient for the implementation for the realization of the retrieval methods.

Test the effectiveness of the improved algorithm is as follows: In the prototype system, the VC runtime, MFC and STL library components in the 110 (including functions, classes, templates) for the experiment, the two methods in Table 1 are a simple comparison of time performance, from which further illustrates the matching algorithm, especially better after the application of the algorithm in practicing the availability and effectiveness.

Tab1. Algorithm's time performance comparison

Algorithms	Mean time	Variance	Query times in experiment
General algorithms	0.406	0.01214	68
Improved algorithms	0.342	0.00185	76

This algorithm is described as facets of the component tree when the construction is completed in the component storage, and stores it into the appropriate database. Making future construction need not be repeated for each search, which makes the cost to the system minimum, they do not give users an additional burden.

D. Experimental Analysis

1. Efficient verification

In order to prove that the proposed retrieval method is more efficient, We can describe the components of the program description of facets by Prieto-Diaz's proposal. The proposed scheme two main features of facet(01) (01 facets for the encoding, the same below) and environment (02). Functional facet include role (0101), object (0102) and media (0103) three sub-facets. Environmental facets contain applications (0201), system type (0202) and customer type (0203) three sub-facets. Not considering the conditions of hardware, first of all, make the five kinds of matching model form the SQL statement. The left conditions of "or" satisfy in the SQL conditions, do not continue to judge right conditions, if it does not satisfy, continue to determine the right conditions. Conditions "and" will select conditions in turn. Because the characteristics of "or" and "and", the complexity of the algorithm conditions of the five ways may be the same, (Such as, when the search result is set to 0, the complexity of the algorithm of the five matching models are the largest $O(m \bullet n \bullet f)$. Through the complexity of functional analysis, we can be drawing the factors that affect the retrieval efficiency. And are the number of components in the component library, faceted classification number and the number of expansion terms. In this dissertation, in the case of Inter (R) Pentium (R) 4 processor hardware environment and Algorithmic Time Complexity Maximum $O(m \bullet n \bullet f)$ of the test. Experimental data in Table 2 (the number of facets to 6, each term expansion of the number 1), Table 3 (number of elements is 10000, each term extended to 1) and 4 (the number of elements is 10000, the number of facets 6).

As can be seen from the table the number of components' influence on the efficiency of component retrieval is very small, it can be described as a basic linear growth. The number of facets on the efficiency of component retrieval is also very small. The expansion of the number of terms has a greater impact.. However, this study is done under circumstances of the largest complexity of algorithm, so in practice, the spending time will be reduced under the same conditions. Thus, in the large scale of component library, as long as a reasonable extension of the term, the proposed retrieval method is feasible in efficiency [17-18].

Tab.2 Effect of components number on efficiency

components number(piece)	take time (ms)
1000	15
2000	31
4000	62
6000	93
8000	125
10000	156

Tab.3 Effect of facets number on efficiency

facets number (piece)	take time (ms)
1	31
2	62
3	93
4	109
5	140
6	156

Tab.4 Effect of term extension on efficiency

term extension (piece)	take time (ms)
1	156
2	312
3	468
4	625
5	781
6	937

2. Recall rate and precision validation

To further verify the effectiveness of the algorithm, the experiments select 134 members to describe information; they originate from 512 component libraries. All the components of the library indexing, retrieving 40 components from the system are shown in Figure (1) show the components of the recall and precision rate. The recall rate of components = retrieve relevant collection / library of all the components related to the collection;

The precision rate of components = retrieve related collections / library collection of all relevant components. A is based on facet the path query precision and recall rate, B is based on the keyword query precision and recall rates. It can be seen from the figure for the keyword search, even though there is a high recall rate, but the precision rate is very low, for the facets of the retrieval methods, the average precision rate of 0.94, the average recall rate of 0.92. The experiment above shows that the proposed retrieval method based on facets of the components to ensure a high recall rate and precision rate. This method is feasible in practice.

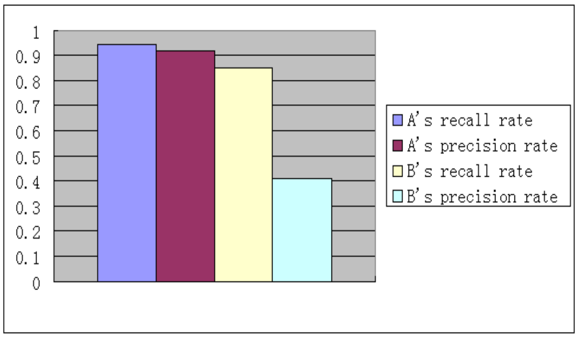


Fig.7 Component recall and precision rate statistics

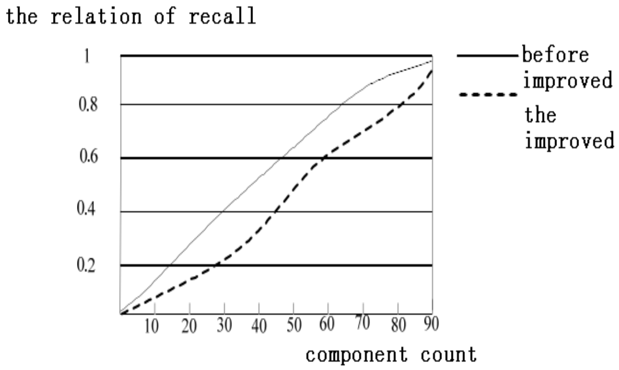


Fig.8 the relation of recall and component count

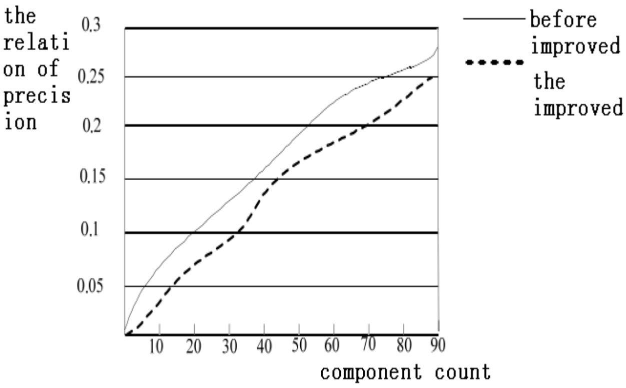


Fig.9 the relation of precision and component count

III. Conclusion

This paper aims at the components based on facets description, draws supports from the ideas of tree matching, and combines the coding characteristics of description of the faceted classification, proposes three component search methods and the corresponding five kinds of matching models, and gives the matching model algorithm and experimental analysis. In practical applications, these three retrieval methods for large-scale component library based on facet description for retrieval will on the one hand meet various user needs to retrieve, on the other hand, they can ensure the efficiency of retrieval. In order to further improve the retrieval algorithm precision and recall rates, and how to extend the term dictionary; how to expanded retrieval condition in case of non-modify the program; How to design good retrieval interface and efficient retrieval platform for component reuse are next steps of retrieving the key issues.

References

[1] Chang JC, Li KQ, Guo LF, Mei H, Yang FQ. Representing and retrieving reusable software components in JB (Jadebird) System [J]. Electronica Journal, 2000, 28(8):20~24.

[2] Wang Yuan-feng, Xue Yun-jiao, Zhang Yong. A Matching Model for Software Component Classified in Faceted Scheme [J], Journal of Software, 2003, 14(3).

[3] Gibb K, McCartan C, O'Donnell R, Sweeney N, Leon R. The integration of information retrieval techniques within a software reuse environment [J]. Journal of Information Science, 2000, 26(4):520~539.

[4] Zeng Yi, Liu Yuan-yong, Guo Yong-lin. A United Component Matching Technology Based on XML [J]. Computer Science, 2007, 34(3).

[5] Chen Ying, Shen Jun. Description and retrieval of component based on ontology [J].Computer Applications and Software, 2007, 24(7).

[6] Li Yong, Zhang Zhi-gang. Semantic Retrieval Research Based on Ontology [J], Computer Engineering & Science, 2008, 30(4).

[7] Zhong Chun-ping, Guo Guo-ping, Zheng You-cai. Research on Matching Algorithm for Faceted-based Software Component Query through Different Component Repositories [J], Computer Engineering, 2006, 32(21).

[8] Yan Rong-jie, Feng Li, Song Yu. The Research on the Classification and Retrieval Method of Reusable Component [J], Computer Engineering and Applications, 2003(6):85~87.

[9] Ma Liang, Xie bing, Yang Fu-qing. The Unified Facet-Based Method to Retrieve Component in Multi-Library [J], Acta Electronica Sinica, 2002, 30(2A).

[10] Wang Yuan-feng, Zhang Yong, Ren Hong-min. Retrieving Components Based on Faceted Classification [J]. Journal of Software, 2002, 13(8).

[11] Benjamin M. Good, Erin M. Tranfield and Poh C. Tan, etc. Fast, Cheap and Out of Control: A Zero Curation Model for Ontology Development. Pacific Symposium on Biocomputing, 2006.

[12] Suk-Hyung Hwang,Hong-Gee Kim, and Hae-Sool Yang.A FCA-Based Ontology Construction for the Design of Class Hierarchy.ICCSA 2005,Springer LNCS 3482,2005.

[13] Fu Cui-yun. Research and Implementation of component retrieval technology based on faceted classification [D]. Master's degree paper, GuangDong: Sun yatsun university, 2008, 39~44.

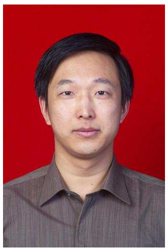
[14] Weifei Fan, Floris Geerts, Wouter Gelade, Frank Nevent, Antonella Poggi. Complexity and composition of synthesized web services. Maurizio Lenzerini, Domenico Lembo, (Editors) PODS. ACM, 2008, 231-240

[15] Liang-Jie Zhang, Nianjun Zhou, Yi-Min Chee, Ahamed Jalaldeen, Karthkeyan Ponnalagu, Renuka R. Sindhgatta, Ali Arsanjani, Fausta Bernardini. SOMA-ME: A platform for the model-driven design of SOA solutions. IBM system Journal, 2008, 47(3):397-413

[16] Olaf Zimmermann, Uwe Zdum, Thomas Gschwind, Frank Leymann. Combining Pattern languages and Reusable Architectural Decision Model into a Comprehensive and

Comprehensible Design Method. WICSA. IEEE Computer Society, 2008, 157-166
[17] Ateret Anaby-Tavor, David Amid, Aviad Sela, Amit Fisher, Kuo Zhang, Ou Tie Jun. Toward a Model Driven Serivice Enginneering Process. SERVICES'08: Proceedings of the 2008 IEEE Congress on Services-Part 1. Washington, DC, USA: IEEE Compter Society, 2008,503-510
[18] Anatoly Gorbenko, Alexander Romanovsky, Vyancheslav Kharchenko, Alexey Mikhaylichenko. Experiment with exception propagation mechanisms in service-oriented architecture. WEH'08: Proceedings of the 4th international workshop on Exception handling. New York, NY, USA: ACM, 2008, 1-7

Yanpei Liu (1982-) was born in Luoyang, China. She received her B.S degree in 2006 form the computer science and technology, Luoyang normal university, her M.S degree in 2010 form School of computer application, Nanchang Hangkong University. Currently she is a professor in the School of Information Engineer, Henan Institute of Science and Technology, Henan Xinxiang, China.



Yuesheng Gu, birth 1973, male, vice professor of Henan institute of science and technology. His current research area is compute rnetwork technology and artificial intelligence.

Jun Chen was born in1979 in Xinyang City, Henan Province, China. She received her B.S. degree in computer application technology from Henan Normal University in 2001. She received her M.S. degree in computer application technology from Guangxi university in 2008. Her research interests include distributed database, grid computing, etc. She is a lecturer at Henan Institute of Science and Technology.