



Design of Rank Based Reusable Component Retrieval Algorithm

Shweta Yadav

*M.Tech in Computer Science Engineering
Punjabi University, Patiala, India*

Kamaljeet Kaur Mangat

*Assistant Professor (Computer Science Engineering)
Punjabi University, Patiala, India*

Abstract: *In the recent past lots of work have been done on retrieving of software reusable component and various techniques have been developed, yet they all lack in reducing up the ambiguity in retrieving up the exact component. The component that are not necessary are also retrieved that lead in the confusion to the user choice. Based on this the algorithm and framework has been developed which not only solve the problem of ambiguity but is very user friendly. This algorithm is combination of two most popular retrieval techniques i.e. based on keyword based approach and semantic component retrieval techniques. Taking a user choice and the previous demand of user on the old component new component is ranked accordingly. This paper presents the efficient model for retrieving up the reusable component in more optimized manner.*

Keywords- *Component Retrieval, Keyword based approach, reusable component, and semantic approach.*

I. Introduction

Software reuse is gaining its importance due to its property of efficient use time and software productivity. Software reuse is the process of creating the software from the existing software rather than building the software from the scratch. Although this seems to be simple from the above mentioned statement but in actual scenario reusing software is quite a tedious task, it needs a reusable software component and retrieving up the component is the major concern. In order to retrieve up the reusable software component one need some approach or techniques to retrieve not only component but the exact component. Existing techniques retrieve up the software component but have major problem that are very hard to tackle. This paper focuses on the retrieval techniques that reduce up the ambiguity in the component and ranking up the component taking their user in mind. As the different user has different choices of component, we need to retrieve the component that not only satisfies up their need but as well as it's accurate. This framework for retrieving up the exact component uses the keyword and semantic based approach. Keyword based technique being the most traditional techniques but have the one of the major problem of the ambiguity so in this model we combine it by the semantic based approach that will reduce up the ambiguity in the keyword based retrieval technique

II. Previous work and research

Retrieval techniques have been developed in order to refine the software reuse process, so it is the one of the major step in software reusing cycle. Descriptive techniques are usually group of keyword that describe the software. Some of techniques that have been most widely used in retrieving up the component are discussed as follows

1. Keyword based retrieval technique: This technique is most primitive retrieval technique which retrieves the reusable component on the basis of the keyword present in the repository. In this all the possible component are retrieved and thus help in reducing the domain search. This technique is user friendly but as all possible components are retrieved, confusion is created in retrieving most relevant component.

2. Facet based retrieval technique: Facet classification is the most widely used method for the component classification. A facet schema has facets. Each facet is described as the essential characters of components. Every facet makes classification for the components in component library from different perceptive. Each facet has group of terms. The development of each facet is done by identifying important vocabulary in domain and grouping the relevant terms together into facet. This classification not only improves search and retrieval, but it is quite helpful in selecting and contributing in the development of the standard vocabulary for components and their attributes. The methodology used in faceted classification is to divide the scheme into several facets and each facet again has several terms.

3. Knowledge based retrieval technique: It is the retrieval technique that uses various kind of reasoning that relate the new query with an old one or that match the query with the components. [4]. This retrieval is based on the methodology that makes decision on the basis of the knowledge base present for the application domain. These techniques maintain a knowledge base which makes some kind of lexical, syntactic, semantic analysis of natural language specification of software component. It stores semantic information about application domain and about natural language. Knowledge based technique uses a text set which is checked whether it is null or not. If the text set is not null then we select the text for analysis by identifying the topic which is searched from the dictionary and the topic feature aggregation formula library. Some rules are applied on the procedure to identify the topic which results in output. [3]

4. Genetic algorithm based approach: This retrieval technique is based on the genetic algorithm which revolves around the concept of natural selection and natural genetics in a biological system. The basic concept of survival of the fittest coined by Charles Darwin. It is non- deterministic search algorithm that is used in problem that need high optimization and for model evolving system. This algorithm finds the accurate result to the problem by genetically breeding the population of individual. Following steps are followed in order to implement the genetic algorithm [3, 4]

- **SELECTION and REPRODUCTION:** Develop an initial population and select a individual that fit best in the whole population
- **CROSSOVER:** From the group of fit individual, crossing take place between the two fit individual. Here the exchanges of the bits and the string takes place in computational form.
- **MUTATION:** After the cross over there is the changing of the bits. For example consider the entire bit 1 is converted to 0 and the bit 0 is converted to the 1.

The procedure of the genetic algorithm being represented above is applied for solving up the problems.

5. Semantic based component retrieval technique: This retrieval technique retrieves the component on the basis of description about the component of the domain that it focuses. The semantic component retrieval technique can retrieve the component on the basis of their domain or features. This component is retrieved from the repository according to the matching by their meaning given for every component present in repository. Systems make analysis without having a complete knowledge of description. This analysis is based upon the knowledge base which store semantic information.

6. Hypertext based retrieval: This retrieval technique organises or designs the information non-linearly into the network of nodes and links. [4] Hypertext retrieval techniques is designed on the concept of interconnection of nodes and links where each node is a unit of textual information and connection that is being established in between two nodes is represented as the link. The point from where the link starts is known as the parent node or anchor and the ending point is termed as the destination or the final node. The links formed up can be either single directional or multidirectional. Stored information can be accessed by navigating along the links. Links represented here are basically pointer from one node to another. [12]

GAPS IN EXISTING TECHNIQUES

Hypertext based retrieval

This retrieval technique is easy to use and retrieving up of the exact component is also quick but the major disadvantages in this technique is that hypertext network requires the well good study of the application domain and it is very difficult to add a new component as require proper study of the link between the old and the new component. In hypertext based retrieval technique user gets lost during navigation. [1, 7]

Faceted classification

This classification is very easy and cost efficient and the source code is very low. Here the meaning of the keyword is usually assigned by preferences of programmer. The dividing of the facet and its meaning assignment is quite a tedious task. A disadvantage of facet classification is identifying important vocabulary in domain is quite a tedious task. [1, 2]

Knowledge based retrieval

This retrieval technique retrieves the new component from the knowledge base of application domain and for making the decision. The biggest flaw in these retrieval techniques is it requires more human resource for the development of knowledge base.

Keyword based retrieval

Keyword based retrieval techniques retrieves the component with their ambiguous nature. For the similar name different component is retrieved which lead to ambiguity in retrieving the exact component.

Genetic based retrieval

Genetic based component retrieval technique that is based on genetic algorithm is prohibitive. Repeated fitness function evaluation for complex problem is most prohibitive.

This is quite expensive technique as for finding the most optimal solution to the complex problem requires expensive fitness function evaluation.

III. Design

Retrieval of component from the repository is the biggest challenge in the software reuse. For exact retrieval of the component from the repository there is the need of proper retrieval technique. Various techniques have developed but very few techniques are user friendly. From the literature survey it was concluded that classical retrieval technique that is based on keyword although it is very easy for user to operate on, it gives most ambiguous component. Previous classical retrieval techniques lead to confusion in user choice. The design process of this work starts with the designing an algorithm that remove the flaw from the keyword based retrieval technique. The key issue of this work is to get optimized solution. To get an optimized solution there is need of some other retrieval technique that could further refine up the results; this problem was solved by introducing the semantic based component retrieval approach. [12]

The selection of the exact component from the retrieved set of component is difficult task which is simplified by ranking up the component on the basis of the frequency of the retrieval that is in how many times that component is retrieved in past and on the description that has been specified by the component. [1, 15]

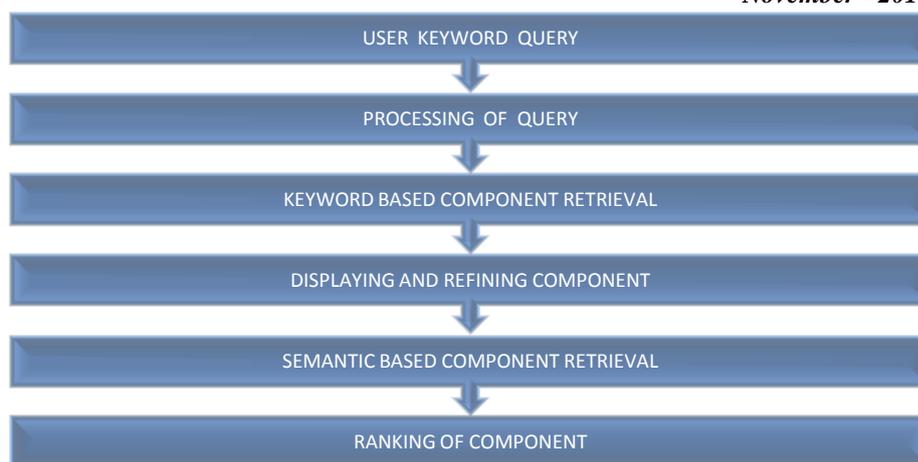


Figure 1: Flowchart of system

In the figure 1, flowchart of system is represented which shows that the user keyword query is processed. This processing is done within the system by the particular keyword provided by user. In the repository, keyword name is matched with the component present. This step is followed by the displaying the component name on their different features. This refined component is then further refined by using the technique based on the semantic. Component retrieved on the basis of keyword is ambiguous in nature. In order to refine the retrieved component semantic retrieval technique is used. In the phase for every component present in the repository there is semantic meaning attached with it. The component retrieved on the basis of keyword is refined by the meaning that is more resemblance with the description given by user. The new retrieved components are ranked on the basis of their frequency of occurrence in past and on the result obtained after semantic matching.

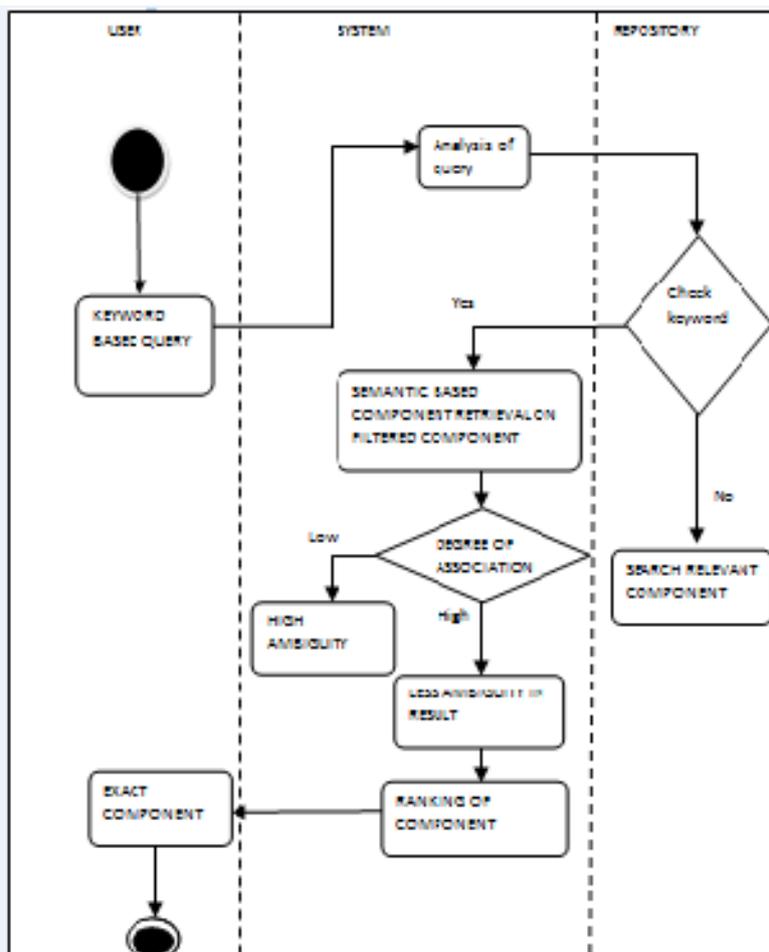


Figure 2: Activity diagram of system

The above figure depicts the activity diagram of system. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system and the control flow from one system to another.

The above diagram shows the six main activities:

1. Keyword based query
2. Analysis of query
3. Semantic based component retrieval on filtered component
4. Ranking of component
5. Exact component.

Proposed model

This model is based upon the descriptive retrieval techniques which are used to process the component in two different phases, each phase refine up the component resulting in the further refinement up the component at every step. The phases which are being followed are

1. Extraction of component.
2. Ranking of component.

Component is extracted in two different steps. The first step extracts the component on the basis of the keyword. Keyword based retrieval technique is descriptive retrieval technique which uses the keyword as the basis for matching and retrieving up the component.

This technique is not efficient way for the retrieving the component as lots of irrelevant component is retrieved but due to its ease of the understand ability keyword based approach is used for retrieving. This component by basically the process of matching apart from the keyword matching there are following parameters on the basis of which component get retrieved. This way it shortens up our retrieving method. The various parameters that are used in keyword based techniques are

1. Platform
2. Return type
3. Application Area
4. Functionality

Each of the parameter is assigned a value and then added together on the basis of priority where the higher priority is assigned to the platform and then to the application area and last priority is given to the return type of the component. The component having maximum value is assigned in increasing order.

Higher acquired valued component goes to the next phase where the further refinement is done on the basis of their semantic characteristics. This step form up the most important part in the extraction of exact component. In this the ambiguous data extracted up by the keyword based retrieval technique get removed by semantic matching. Semantic based component retrieval algorithm matching is used to find the strongest semantic relation between software component objects and queries; depending on a semantic basis; to retrieve the relevant components. Semantic search seeks to improve search accuracy by understanding searcher intent and the contextual meaning of terms as they appear in the searchable data space, whether on the Web or within a closed system, to generate more relevant results. The same semantic structure as well as the content in the semantic structures is related by synonymy relationship. Thus in order to retrieve the most semantically correct component we need to get those components that has got the most degree of similarities in between them.

Degree of similarity is done by using up the two entities and their relationship is calculated by the coefficient for the every retrieved component of the first phase. Thus the one with the higher degree is highly matched and thus it is given up the component and the ranking is performed on the basis of their higher degree of similarities which satisfy the above query and the one in which the user have previously used.

This will help in retrieving up the unambiguous component as well as taking user as its priority and helping the user to get its most suitable component. If the user makes the new choices apart from the previous choice, this query and the user result is added to the repository.

The retrieved component is then again ranked on the basis of the two type of the classification approach:

- Component Description: It uses the description that has been used by user to rank up the component fully or partially.
- Component Usage: In this rank based phase, the component being retrieved is ranked on their previous demand and use. The component is ranked above who have been used up more frequently.

Working of algorithm

In order to achieve higher recall and precision value, following algorithm is designed There is two different phases in the algorithm that work on keyword based retrieval technique and another work on the semantic based retrieval technique. Basically the refined component given by the keyword retrieval technique is again refined on the basis of their semantic which reduces the ambiguity. The process of retrieving up the component on the basis of algorithm is divided up in the following phase:

PHASE1

Step1: Match the user keyword from the keyword present in the repository. Keyword selected is described further by the following properties and values are assigned to them by checking it from the repository.

- Component that satisfy the function Kr are searched where Kr form the sum of all the value assigned to the component retrieved by the keyword.

$$Kr = (wt+ wp+wa+we+wf)$$

Where wt=component type value

wp= platform value
 wa =application area value,
 we= application environment value,
 wf= return type value,

- These values are assigned to the component on the basis of matching up of the component from the repository.
- Assign the value 3 for the every matched platform and 2 for application area values and 1 for the return type value and 0 for the unmatched properties. These values are assigned on the basis of priority where application area is assigned highest priority and return type is assigned the lowest priority.

Step 2: Repeat the step for the every retrieved keyword from the repository Kr1...Kr n.

Step 3: According to the values arrange Kr1...Kr n in increasing order.

Kr1>Kr2>.....>Kr n.

- Retrieve the component according to the value of Kr.

PHASE 2

This phase works on the semantic retrieval of the above retrieved component in order to further refine up the result. a formal representation of the basic terms that comprise the vocabulary of a specific domain, relations that form association between terms, and the set of axioms; which are the rules and constraints for combining terms and relations to define extensions to the vocabulary.

The steps that are to be followed as:

Step1: The association between the component retrieved (E1) and the predefined component (E2) present in the repository are calculated according to the association coefficient. To illustrate how these coefficients are calculated, assume two entities E1 and E2, represented by semantic feature vectors indicating the presence or absence of a meaning. The similarity between E1 and E2 can be compactly represented by a table as shown below:

Table 1: Semantic Matching

		E1	
		1	0
1 E2	A		
	B		
	C		
	D		

In the above table 'a' represents the count of semantic feature present in both E1 and E2, 'b' represents the total number of semantic feature present in E1 but absent in E2, 'c' represents the total number of semantic feature present in E2 but absent in E1, and 'd' represents the number of semantic feature that are absent in both E1 and E2. The following association coefficients can then be defined: [36]

Jaccard coefficient $J = a/(a+b+c)$

The interpretation of the given query is expanded with closely related concepts in order to achieve match with a conceptual description of an intended components rather than specific words or concepts.

The ambiguous senses in natural language can be overcome using the semantic based approach. Concepts are closely related when they have high degree of similarity and that are positioned closely together. These algorithm form the basis for the part of the extracting the most probable component. These components are ranked accordingly on the basis of their usage.

Algorithm

Working of proposed retrieval algorithm is given below as:

Step 1: Find the user defined values.

Step 2: Match the user defined properties from the repository.

Step 3: For every matched property assign the values as defined in step 4-6.

Step 4: Initialize the value 3 for matched platform feature

Step 5: Initialize the value to 2 for matched application area feature

Step 6: Initialize the value to 1 for matched return type.

Step 7: Add the total values Kr for the every component present in the repository.

Step 8: Repeat the step 1-7 for the every retrieved keyword from the repository Kr1...Kr n.

Step 9: Arrange the component in order of increasing value Kr.

Step 10: Initialize E1 to retrieved component.

Step 11: Initialize E2 to predefined component.

Step 12: Compare the retrieved component with the predefined component by using the association coefficient $j = a/(a+b+c)$

a→ The count of semantic feature present in both E1 and E2.

b→ The count of semantic feature present in E1 but absent in E2.

c→ The total number of semantic feature present in E2 but absent in E1.

d→ The count of semantic feature absent in E1 and E2.

Step 13: Repeat the step 10-11 for the every retrieved component.

Step 14: Retrieve the component on the basis of high matching value.

IV. Implementation

Algorithm was successfully implemented on Rank based retrieval software designed on application interface Java Development Kit 7.0 and repository was developed on MySQL 5.3.

First of all, code was created on JDK Net Beans 7.0 that retrieves the reusable component on the basis of the algorithm. The repository used for the system contains 500 components that can be entered manually or automatically. According to the user query, the component are retrieved on the basis of the keyword that is given by the user and on the semantic that describes the meaning of every word set which are arranged in the order of their priority of matching of the pattern and words. Description of the keyword given by the user is matched from the repository in such a way that it represents the set of set which means every word is described by the words meaning so that the component is efficiently checked. The components in the repository are indexed in such a manner that every time the component is retrieved the frequency of retrieval is incremented in the repository. After retrieving the component they are ranked on the basis of the frequency and on the mat

Results and discussions

In this section, in order to very this method of retrieving component. The experimental results are shown and discussed with the different condition and different number of component.



Figure 3: User Interface of RBRT

This screenshot is of the RBRT: Rank Based Retrieval Technique software model where the function name, application area, description, platform and the return type form up the features of the particular reusable component. Function Name is the name of the keyword according to which the retrieval will take place. In this it is mandatory for user to enter the function name and its description.

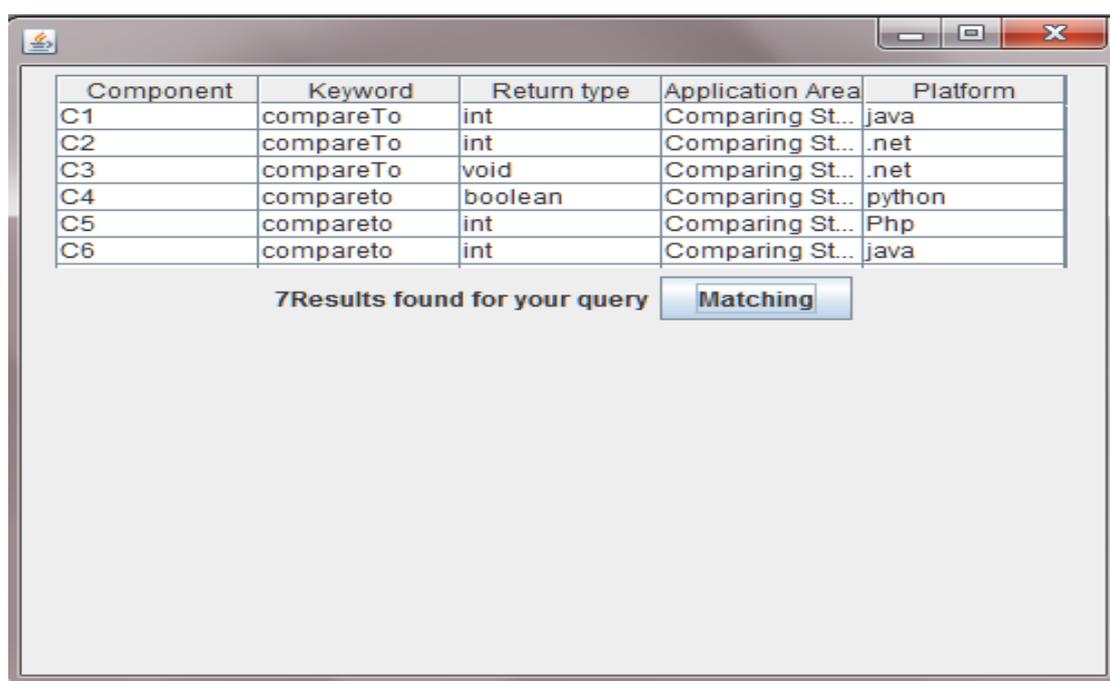
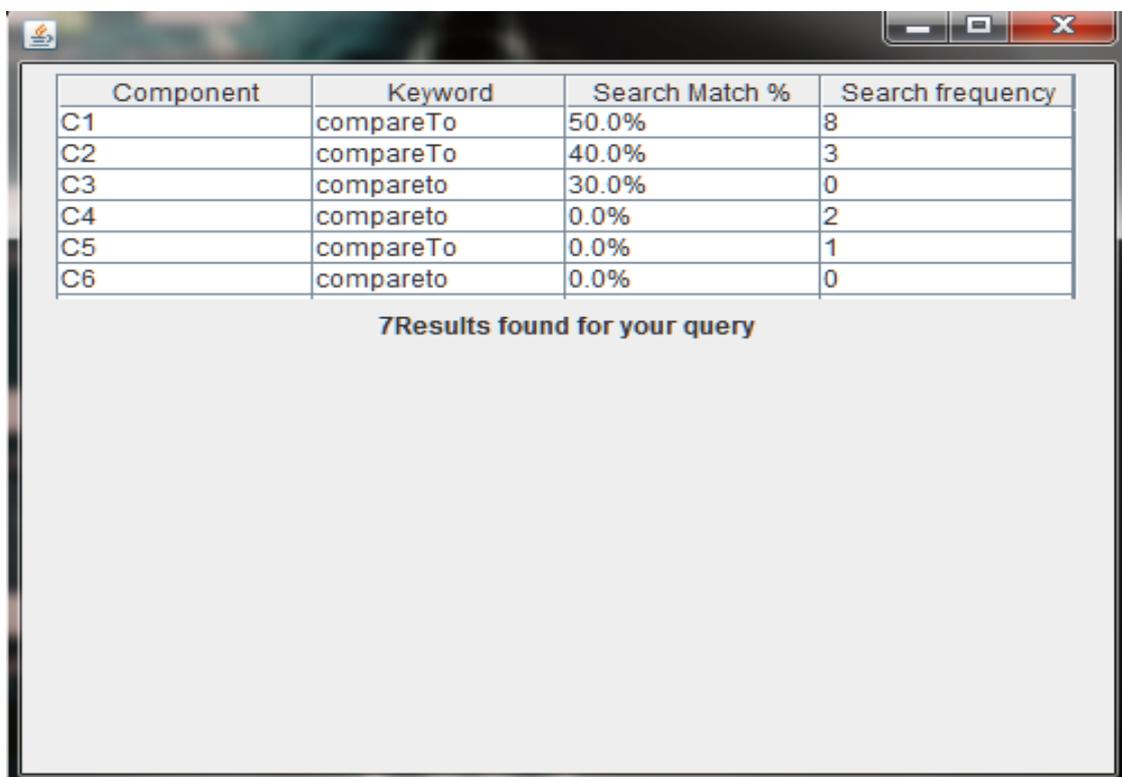


Figure4: Interface of RBRT in comparison mode.

This figure specifies screenshot shows the matched component on the basis of both semantic and keyword based retrieval technique. The component being retrieved as c1, c2....., c6 are component that matched the keyword or function name compareTo and is order according to the values they have attained after the matching of their features. Description given by user is also used for further refinement in which there is association between the component retrieved and the actual component. Over here the components are retrieved according to the basis of increasing order.



Component	Keyword	Search Match %	Search frequency
C1	compareTo	50.0%	8
C2	compareTo	40.0%	3
C3	compareto	30.0%	0
C4	compareto	0.0%	2
C5	compareTo	0.0%	1
C6	compareto	0.0%	0

7 Results found for your query

Figure 5: Output of RBRT for the Keyword compareTo.

This screenshot show that the ranking of the component on the basis of component description i.e. the matching percentage and on the frequency of the component. The frequency of the component compareTo is retrieved according to the number of the time component is retrieved in the previous past. Figure 5.3 shows the result of the search frequency as 8 and percentage of matching as 50%.

V. Conclusion

This work present a rank based retrieval software system framework for retrieving up the component. In this work key concept of reusable principle and methods of reuse was taken into consideration. This work present the study of the previous retrieval techniques and there flaws which need to be overcome in order to retrieve a component with the high recall and precision value. In the classical traditional method, the user query was processed only on basis of matching of keyword to retrieve the component. Their performance is limited because of the conceptual meaning of the keywords. The major contribution of this work is integration of keyword and semantic based retrieval techniques. Using the concept of semantic based retrieval technique the component being retrieved was greatly influenced as the percentage of exact component was retrieved more by using semantic approach. Ambiguity was reduced by use of semantic based retrieval techniques in comparison to keyword based retrieval. This technique guides the user in choosing the component of their demand as the ranking is retrieved on the basis of frequency and the description of the component. In the rank based retrieval component the precision and recall value was better than the other classical retrieval techniques.

Future work

Rank based retrieval technique is the efficient approach for retrieving up the reusable component yet in the future this work can be extended to get better retrieval approach. This work can be extended as follows in the future.

1. Measure and compare this model by using large component repository.
2. Semantic based retrieval techniques can be combined with other retrieval classification technique in order to get more optimized and efficient techniques.
3. Ranking of the retrieved component can be done using different methodology.

References

- [1] Mili Hafedh and Mili Ali and Mili Fatma., "Reusing software: Issues & Direction", *IEEE Transaction on Software Engineering*, Vol 21, No 6, June 1995.
- [2] Smolarova Maria and Navrat Pavol , " Software Reuse : Principles, Patterns, Prospects"[online] <http://www.dcs.elf.stuba.sk/smolar>,<http://www.elf.stuba.sk/navrat>.

- [3] Kaur Navneet, "Retrieving Best Component From Reusable Repository", Thapar Institute of Engineering and Technology, Patiala, India,-Jun 2005.
- [4] Rao D. Vijay and Sarma V.V.S , "Rough Fuzzy Approach For Retrieval of Candidate Component For Software Reuse", *ELSEVIER*,2003, pp. 875-886.
- [5] Saiyd N. A and Said I.A and Takrori A.H., "Semantic Based Retrieving Model of Reuse Software Component", *International Journal of Computer Science & Network Security*, Vol 10, No 7, July 2010.
- [6] Gupta Suresh Chand and Kumar Ashok, "*Smart Environment for Component Reuse*" Vol 2, International Journal of Advanced Research in Computer Science Software Engineering, February 2012.
- [7] Niranjana .P and Rao C.V. Guru , "A Model Software Reuse Repository with Intelligent Classification and Retrieval Technique". Computer Science Engineering, Kakatiya Institute of Technology & Science, Warangal, India - November 2011.
- [8] Lattanzi Mark and Henry Sallie, "Software reuse using C++ classes:The question of inheritance"vol.43, *Journal Of System Software*",1998, pp. 127-132.
- [9] Jilani Labeled Lamia and Mili Rym and Ali Mili," Approximate Retrieval: An Academic Exercise or a Practical Concern", *Proc.8th Annual Workshop on Software Reuse*,1997.
- [10] Martins A.C and Garcia V.C and Meira S.R. and Almeida E.S,"Suggesting Software Components for Reuse in Search Engines Using Discovered Knowledge Techniques"*Proc. 35th Euromicro Conference on Software Engineering and Advanced Applications*,2009.
- [11] Bruce W. Weide, William F. Ogden, Stuart H. Zweben, "Reusable Software Components", 1991.
- [12] Rajesh K Bhatia, Mayank Dave, R.C Joshi, "*Retrieval of most relevant reusable Component using genetic algorithms*", Software Engineering Research and Practice 2006, pp. 151-155.
- [13] Rajesh K Bhatia, Mayank Dave, R.C Joshi, "*A Hybrid Technique for Searching a Reusable Component from Software Libraries*", DESIDOC Bulletin of Information Technology, Vol.27, No.5, September 2007, pp. 27-34.
- [14] Rajesh K Bhatia, Mayank Dave, R.C Joshi, "*Ant Colony Based Rule Generation for Reusable Software Component Retrieval*", Proceedings of the 1st Conference on India Software Engineering Conference, pp 129-130, Feb 19-22, 2008, Hyderabad, India.
- [15] Jiang Guo, Luqi. "*A Survey of Software Reuse Repositories*", 7th IEEE International Conference and Workshop on the Engineering of Computer Based Systems, 2000, p. 92.
- [16] Arun Sharma, Rajesh Kumar and P .S. Grover, "*A Critical Survey of reusability aspects for component-based systems*", Proceedings of World Academy of Science, Engineering & Technology, Vol. 21, Jan 2007.
- [17] Boehm, B "*Managing software productivity and reuse*", IEEE Computer 16(9), 111- 113, 1999.
- [18] Burton, B. A., Aragon ,R. W. , Bailey ,S .A., Koehler ,K .D., and Mayer ,L .A, "The Reusable software library", IEEE Software 4, 4, 25-33, 1987.
- [19] Henninger, S "*An Evolutionary Approach to constructing effective software reuse Repositories*", ACM Transactions on software engineering and methodology 6(2), 111-140, 1997.
- [20] Isakowitz,T and R,J Kauffman , "*Supporting Search for Reusable Software Objects*", IEEE Transactions on Software Engineering 22, 6, 407-423, 1996.
- [21] Jiang Guo, Lqui, "*A Survey of Software Reuse Repositories*", p-92, 7th IEEE International Conference and Workshop on the Engineering of Computer Based Systems, 2000.
- [22] Jun-Jang Jeng, Betty H. C. Cheng, "*Specification Matching for Software Reuse: a Foundation*",ACM, pp 97-105, (1995).
- [23] Larry Smith, "*Software Reuse Technologies and Applications*", (December 1994).
- [24] M.R. Girardi And B. Ibrahim, "*An Approach to Improve the Effectiveness of Software Retrieval*",(1993).
- [25] M.R. Girardi And B. Ibrahim, "*Automatic Indexing of Software Artefacts*", (1994).
- [26] Mehrdad Dianati, Insop Song, Mark Treiber, "*An Introduction to Genetic Algorithms and Evolution Strategies*".
- [27] Michael Gordon, "*Probabilistic and Genetic Algorithms for Document Retrieval*", Communications of the ACM, Volume 31, Number 10, pp 1208-1218, (1988).
- [28] Oualid Khayati, Jean-Pierre Giraudin, "*Components Retrieval Systems*" (2001).
- [29] Panos Constanto poulos and Martin Dorr, "*Component Classification in the Software Information Base*", Chapter 7, pp. 177-200, Prentice Hall, (1995).
- [30] Paul V. Biron, Donald H. Kraft, "*New Methods for Relevance Feedback: Improving Information Retrieval Performance*", ACM, pp 482-487, (1995).
- [31] Praveen Pathak et. al., "*Effective Information Retrieval Using Genetic Algorithms Based Matching Functions Adaptation*, Proceedings of the 33rd Hawaii International Conference On System Sciences", pp 1-8. (2000).
- [32] Dr. Nedhal A. et. al., "*Semantic Based Retrieving Model of Reuse Software Component*", International Journal of Computer Science & Network Security, VOL10 NO 7, July 2010.
- [33] Xiong Lirong et.al., "*The Approach of Credit Domain Component System Framework Supporting Interoperation and Component Retrieval*" ,ISECS International Colloquium on Computing, Communication, Control, and Management,2008.
- [34] Li NIE and Luo ZHONG , "*Component Retrieval Based on Domain Ontology and User Interest*", IEEE Computer, 2009.
- [35] Maqbool. O and Babri. H.A., "*The Weighted Combined Algorithm: A Linkage Algorithm for Software Clustering*" Computer Science Department, Lahore University of Management Sciences, DHA Lahore, Pakistan