Optimized Software Component Reuse with Various Attributes for Efficient Search

Navjot Kaur[1], Dr.Sushil Garg[2]

[1] Research Scholar, Department of Computer Science and Engineering, RIMT-IET College, Mandi Gobindgarh, Punjab.,
[2] Professor & Principal, RIMT-MAEC College, Mandi Gobindgarh, Punjab.,

ABSTRACT: Component reuse process is always play important role in software component reuse which uses the existing component in software. With rise of usage in the component based systems, storage of various component of the system become a challenging task for the developers. The components that are identified as reusable are stored in a repository so that other teams can use them to serve in to get quality product. In this research we have proposed a component retrieval system for reuse process with help of facet attributes with multiple facet attributes for fetching process. Meta Data repository integrates expert knowledge of correlative domains and generalizes crucial concepts and relations among concepts in these domains. We will fetch the interfaces, and function based attributes is the use of existing software components to build a new software system. Effective storage and retrieval of software components is much essential in software components reuse process. The researchers have developed a number of software components reuse techniques for storage and retrieval of software components. No one technique is complete in its own; every technique has its own merits and demerits. This paper presents a meta-data model and faceted classification for storage and retrieval of software components that considers domain semantic information based on ontologies and texonomies. In contrast to most existing repositories, which only retrieve a limited set of components, the proposed metadata model makes possible the recommendation of interrelated components, as ontology and taxonomies characteristics were incorporated. The software component retrieval based on facet classification is a method which has been widely applied in software component retrieval, but the precision of software component retrieval is poor as a result of subjective factor in faceted classification retrieval. The architecture of software component retrieval system and the model of software component retrieval system were designed, the corresponding match algorithm was provided. According to the relation of facet and term space, meta-data repository was established and abstracted from domain knowledge which formed coherent retrieval in the domain and was applied to software component retrieval process. These terms in the meta-data repository were then used to match software components which described in the software component description repository with facet classification, related software components were retrieved from the software component repository. The results of application show that the new software component retrieval method can evidently improve the component retrieval precision and take care of the full-scale of the searching results. Finally the results are good as compared to various other mechanisms available for software component reuse process.

KEYWORDS: Software Component Reuse, Software Development, Faceted Classification, Code level components, Functional Reuse, Graphical User Interface.

1. INTRODUCTION

Software development (also known as application development, software design, designing software, software application development, enterprise application development, or platform development) [3] is the development of a software product. The term "software development" may be used to refer to the activity of computer programming, which is the process of writing and maintaining the source code, but in a broader sense of the term it includes all that is involved between the conception of the desired software through to the final manifestation of the software, ideally in a planned and structured process.[4] Therefore, software development may include research, new development, prototyping, modification, reuse, re-engineering, maintenance, or any other activities that result in software products.[5]

Software can be developed for a variety of purposes, the three most common being to meet specific needs of a specific client/business (the case with custom software), to meet a perceived need of some set of potential users (the case with commercial and open source software), or for personal use (e.g. a scientist may write software to automate a mundane task). Software component reuse is the use of existing software components to build a new software system. Effective storage and retrieval of software components is much essential in software components reuse process. The software reusable component is nothing but a component development in a product and used in the development of other new product. The software reuse is meant to reduce cost, effort to develop a new product and also increase the quality of newly developing product. The components that are identified as reusable are stored in a repository so that other teams can use them to serve in to get quality product. Component Based Software Engineering (CBSE) proposes the reuse of software components, which can be retrieved and assembled into applications of specific domains. CBSE is a process that aims to design and construct software systems using reusable software components. In order to build these applications successfully, it is fundamental to choose appropriated software components. Thus, it is desirable to have a repository that supports the storage, query and retrieval of software components and makes reuse possible. Most existing software component repositories only

retrieve a limited set of software components and some do not satisfy user queries. Interrelated software components may exists and would be useful, but the user either does not know about them or is unable to retrieve them because the query is defined too narrowly. The schema of the repository itself often does not consider semantic relationships among software components and thus omits important retrieval information. A technique to software component repositories is needed that provides the retrieval and recommendation of semantically interrelated software components. This technique can be mentioned as Faceted Classification based on Meta data repository and component repository for storage and retrieval of software components.

A software component is classified by each facet from different profiles, a component can be described by many facets and many terms in a facet, different facet can describe a component from different angle of views. There are a set of terms in a facet, structured term space is formed by common and special relation. The value of a term can be only attained from given facet. It is helpful to understand correlative domain for the reused that travel in term space, the term space can be evolved. The method of faceted classification is most accurate to express information of a software component and can be easily understood by users in various methods of software component retrieval, therefore, if the method of faceted classification can be provided in some software component meta-data and component repositories which include many methods of software component retrieval, then it will achieve the best effect that the method of faceted classification is used.

2. SIGNIFICANCE OF COMPONENT REUSE

There are so many ways to reuse the component in software. But the three major ways to reuse software, the first way is you can use the component in its original from in multiple systems, the second ways is you can extend component functionality as needed for individual systems, and the last way is you can restrict component functionality as needed for independent systems. In brief the first way involves the technique for writing reusable software components and identifying those components, the second way involves the covering of the steps required for extending reusable software components; finally the last way addresses testing and deploying your extensions and wrappers for reusable software components[8][9]. When using a component for reuse that must be meet requirements. We need to remember that to inheritance means to derive a new component from original component to extend the required functionality [10]. All most all the reusable software components comes in the one of the above three forms. In the first way code samples are copied and pasted among systems. In the second way you have a string parsing routine that your coworkers find useful. You email that code to them and they perhaps embed or modify it to a new method. Recipes are an extension of code samples by which a way to reproduce some behavior is described in terms of consuming an existing component [11]. In the last way you can reuse the binaries distributed on local or remote systems without distributing them with each product [12].

3. PROPOSED WORK

Effective storage and retrieval of software components is much essential in software components reuse process. This proposed model makes possible the recommendation of interrelated components by using the method of faceted classification which can be provided in some software component meta-data and repositories. The main reasons for using this technique for storage and retrieval of software components are due to most existing repositories can only retrieve a limited set of components and method of faceted classification is most accurate to express information of a software component and can be easily understood by users in various methods of software component retrieval. The graphical panel will be providing the good view of research. The graphical view is given in figure 1 below.

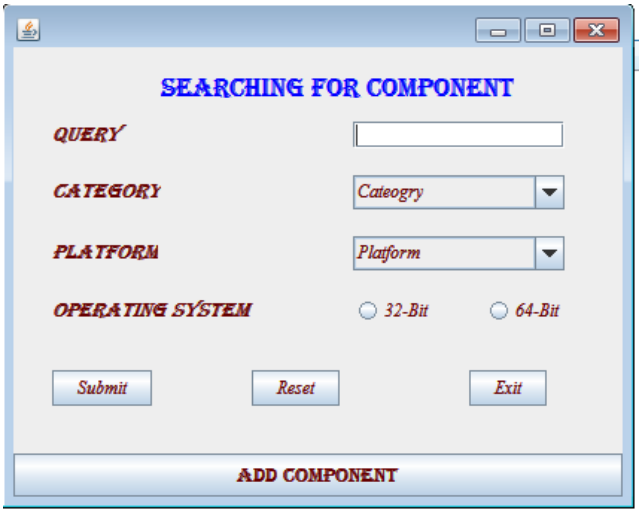

Figure 1: Graphical panel for Component Searching

Reusable component are very important part of software development. It is cheaper and less time consuming to use a pre-existing component for your software instead of developing the whole component from the scratch. In the past few years many organization researched and understood the importance of reusable components. But it is not always a simple task to use a pre-existing component in your software project. Finding the right component for your need can be the biggest challenge. Over the time many Reusable Component Retrieval Systems are developed and proposed. But present systems are not that effective, the main problem is that these systems don't have the necessary information that is needed for the retrieval of the component. Most system store and search components according to keywords. But most of the time only keywords are not enough to describe a software component. Our system is divided into interface and

repositories. Interface allows user to query the system and repository stores the information regarding component.
Our research is focused on providing a facet based Reusable Component Retrieval System. Facet based approach allow to describe a component in a better way. In this approach we try to describe component in different profiles. Describing component in all possible ways allows better and meaningful component search. Moreover to further refine the search user may select the category and execution environment of the component.

4. EXPERIMENTATION

The proposed work has been considered in building the component retrieval system with java platform. Basic parameters used for experimentation are explained in table 1 below. Some of the experimentation done for checking the behavior of component retrieval based on facted classification.

Parameters	Value
Simulator	Eclipse
Language Used	JAVA

Table 1: Parameters used for experimentation

Result obtained from this research by fetching component search based on various attributes is given below.

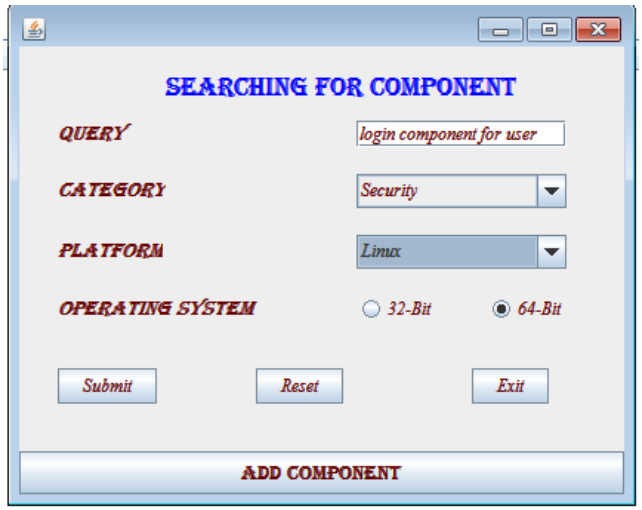

Figure.2: Main interface with a user query

User will add a component description in query, will select the category for the component, will select platform and operating system and will submit the query to the program.

Proposed work specially focused on this process of selection due to fact that huge time spent on the deciding the platform at the later part of search by traditional search process. So we have tried to reduce the time and refined filtered results with this platform selection procedure.

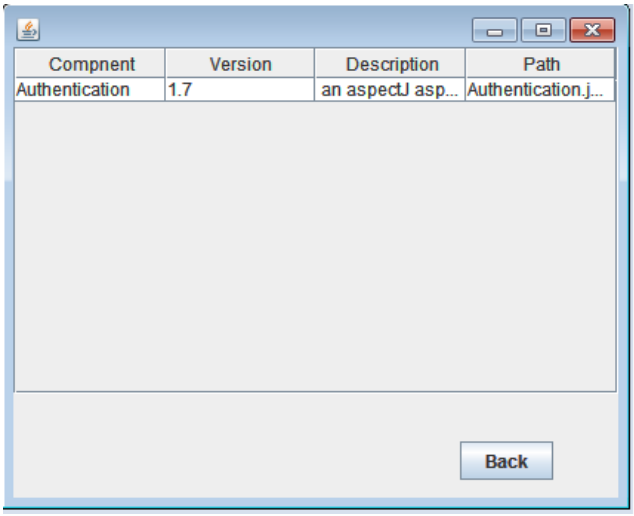

Figure 3: Successfully fetched query with search

If search is successful then a table will be displayed to the user that will show all the components that matches user query. Component name, version, a description about the component and a path that specify the location of the component will be displayed to the user.

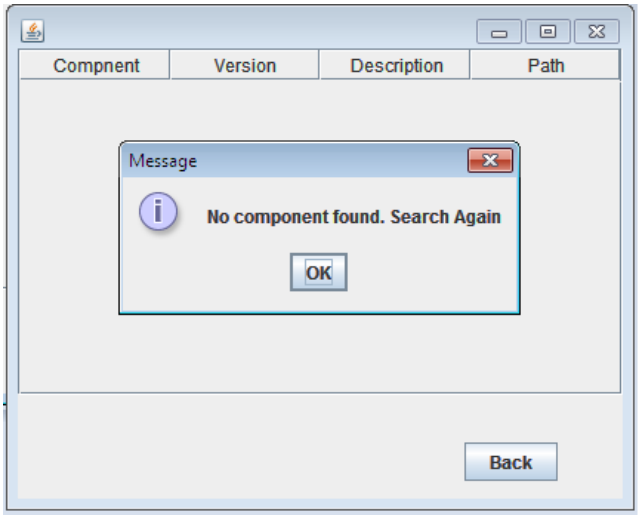

Figure 4: Demonstration of no component matching

If no component is found then user will be displayed with a No component found message. Pressing Ok will take user back to main interface, where user may search for the component with different query.

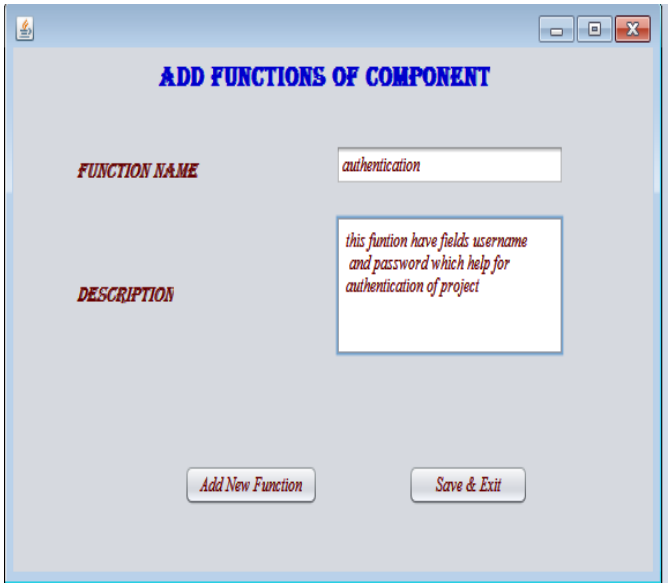

Figure 5: Functionality for adding of component with added component

This snap shot is showing add function interface with field values. In function name we will give the name of function, it could be anything that will help to understand the functionality and in description we will give a description to explain the functionality provided by function.

5. CONCLUSION

In this paper, we have started with detailed briefing about the component retrieval system with query optimization process as an initial stage. Database will be selected according to system required and cleaning process of database will be considered. An accurate requirement of users will be reflected to a describing repository of software component based on faceted classification by a module of accurate query processing, appropriate software components will be searched. Priority in the search will also be considered based on information used in meta data for particular repository. Meta Data repository integrates expert knowledge of correlative domains and generalizes crucial concepts and relations among concepts in these domains. We have fetched the interfaces, and function based attributes is the use of existing software components to build a new software system. Effective storage and retrieval of software components is much essential in software components reuse process. The researchers have developed a number of software components reuse techniques for storage and retrieval of software components. No one technique is complete in its own; every technique has its own merits and demerits. The software component retrieval based on facet classification is a method which has been widely applied in software component retrieval, but the precision of software component retrieval is poor as a result of subjective factor in faceted classification retrieval. The architecture of software component retrieval system and the model of software component retrieval system were designed, the corresponding match algorithm was provided. According to the relation of facet and term space, meta-data repository was established and abstracted from domain knowledge which formed coherent retrieval in the domain and was applied to software component retrieval process. These terms in the meta-data repository were then used to match software components which described in the software component description repository with facet classification, related software components were retrieved from the software component repository. The results of application show that the new software component retrieval method can evidently improve the component retrieval precision and take care of the full-scale of the searching results. A basic interactive engine has been used to fetch the accurate details. Search will be based on meta data and component in the repository will be calculated along with accurate component received. Storage of component and reusability can also be considered for the part of experimentation. The precision and recall values will be consider for checking the performance of the proposed architecture. Java runtime environment will be used for experimentation of component retrieval system.

REFERENCES

[1]http://ptolemy.eecs.berkeley.edu/publications/papers/02/embsoft/embsoftwre.pdf.

[2] Azoff, Michael. "Embedded software is car manufacturing's biggest challenge." Ovum, 2 August 2011.

[3] http://www.bitpipe.com/tlist/Application Development.html. "Application Development White Papers (Development of Software, Software Design, Designing Software, Software Engineering, Software Application Development, Enterprise Application Development, Platform Development, Software Development, Applications Development, Development) Software Downloads, Definition and Webcasts".

[4] "Application Development (AppDev) Defined and Explained", http://www.bestpricecomputers.co.uk/glossary/application-development.html.

[5] DRM Associates (2002). "New Product Development Glossary", http://www.npd-solutions.com/glossary.html.

[6] Maxime Perrotin, Eric Conquet, Julien Delange, Thanassis Tsiodras," TASTE: An open-source tool-chain for embedded system and software development", European Space Agency, ESTEC, Keplerlaan, 2010.

[7] Joseph M. Morris,"Software Industry Accounting", pp 1.10, 2001.

[8] Alan M. Davis," Great Software Debates", Wiley-IEEE Computer Society Press, pp 125-128, October 8, 2004.

[9] Alok Kumar Gupta, Vinod Kumar Yadav, Satendra Kumar, Vishal "A Robust Retrieval Scheme for Software Component Reuse" International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 1, July 2012.

[10] Gowtham Gajala, "Implementation of attribute value and faceted value classification scheme for constructing Reuse" International Journal of Computer Trends ad Technology, vol.4, Issue 1, 2013.

[11] Kudikala Mahesh Babu, Kudikala sravana Kumari, Pulluri Srinivas Rao "Searching Technique in Retrieving Software reusable Components from a Repository" International Journal of Scientific and Research Publications, February 2012.

[12] Nisha, Karambir "Factors Affecting Retrieval of best Component From Repository" IJCST Vol.3, Issue 3, July-Sept 2012.

[13] P.Niranjan, Dr.C.V.Guru Rao "A Mock-Up Tool for Software Component reuse Repository" International Journal of Software Engineering & Applications (IJSEA), Vol.2, No.2, April 2010

[14] Pankaj Rakheja, Prabhjot kaur, Performance Analysis of RIP, OSPF, IGRP and EIGRP Routing Protocols in a Network, International Journal of Computer Applications, 2012.

[15] Ruben Prieto Diaz "Implementing Faceted Classification for Software reuse", The Contel Technology Center 15000 Conference Center Drive Chantilly, VA 22021.

