International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 8 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT183883

Framework for Semantic Search Enabled Software Component Repository (SSESCR)

Vishnu Sharma[1]**, Prof. Vijay Singh Rathore**[2] **Prof. Chandikaditya Kumawat**[3]
[1]Research Scholar, Mewar University, Chittorgarh, Rajasthan, India
[2]Professor & Head, JECRC Engineering College, Jaipur, Rajasthan, India
[3]Professor, Mewar University, Chittorgarh, India

ABSTRACT

Software reuse can improve software quality with the reducing cost and development time. Systematic reuse plan enhances cohesion and reduces coupling for better testability and maintainability. Software reuse approach can be adopted at the highest extent if relevant software components can be easily searched, adapted and integrated into new system. Large software industries hold their own well managed component libraries containing well tested software component with the project category based classification .Access to these repositories are very limited. Software reuse is facing so many problems and still not so popular. This is due to issues of general access, efficient search and adoption of software component. This paper propose a framework which resolves all of the above issues with providing easy access to components, efficient incremental semantics based search, repository management, versioning of components.

Keywords : CBSE: Component Based Software Engineering, ANP: analytic network process, CBSS: Component Based Software Systems, SSESCR: Semantic Search Enable Software Component Repository

I. INTRODUCTION

Software reuse is a process of developing new software applications from existing software components. Once a component is developed and stored in repository it may be used for unlimited number of time. Component with less functionality may be replaced with new one without changing the configuration of the overall application. Software component may be source code, designs, specifications, architectures and documentation[1]. Source code is mainly reused. Most of the software component repositories use four different types of classification techniques Keyword based, enumerated, Attribute Value, and Faceted classifications. The main issue of software reuse in software industry is the lack of ability to find out and retrieve relevant software components from repositories. To resolve this obstacle, an important step is organizing and cataloging of the components, it may lead to a to a quick and efficient search for appropriate component for prospective reuse. This will be a obligatory support to the software developer[2].

Software reuse is a significant area of software engineering research that considerably improves software productivity and quality [3]. Reuse in software successfulness depend on wide assortment of elevated quality components, efficient classification and retrieval techniques. Efficient software reuse ensures that the users of the system have appropriate permission to the components repository. The user

CSEIT183883 | Received : 05 Dec 2018 | Accepted : 12 Dec 2018 | November-December-2018 [3 (8) : 253-256]

must be able to search, retrieve these components precisely and rapidly, and if required, he must be able to request to amend them. Component is defined as a well-defined entity of software that keeps a published interface and can be used as a integral part of a larger unit [4]. Reuse is capable to combine self-sufficient software components to form a larger software application module or complete application. For successful reuse developers have to first find out a reliable, secure and component rich repository with efficient search and download features.

Classification of software component make diversity of components organize into structures so that they may be searched and retrieved with no trouble[5]. Most of the Component retrieval techniques [16] need classification of the components for better search results. With the time classification schemes gets obsolete with new technology. So, the classification method should be updated from timely. This paper mainly focuses on developing a reliable, secure and component rich repository with efficient search and download features and understanding the requirement of the user in better way.

II. METHODS AND MATERIAL

1.0 Software Components characteristics

Every software component has 9 common characteristics, which were identified considering both the functional and non-functional perspective. These identified characteristics are as follows.
Component Name, its Functionality, its domain, Operating System, Algorithm, Its Implementation Language, Developer and Time Complexity.

Every characteristic identified is encoded to a string(binary), each characteristic requires four bits. So for all 9 characteristic require 36 bits which includes all the characteristics of software component. This obtained 36 bit string is used by genetic algorithms to find the component classifiers[11].

Bagheri[12] et.al. proposed a “Component Based Software Product Line Reliability Estimation Model” this model is capable to caters upper and lower reliability bounds it confirms a “software product line feature model”, its specializations and configurations.

R.A. Pyne et. al This paper demonstrates a method this method uses a “semantic syntactic approach” to retrieve components from a software component supermarket so called heterogeneous software repository [13].

2.0 Quality Based Software component selection

Component based software system is a reliable method in which well tested components are chosen that provide all the required functionalities. Selecting best qualified components is a critical task for ensuring the efficient functionality of the entire system. Most of the time component reusability decisions are made without planning or in an ad hoc manner, which results in crossing the schedule deadlines and worse the quality of the system. Nazir et.al presented The Analytic network process (ANP) technique for selection of best software component [15].

Mapping ANP and ISO/IEC 25010:2011 In the proposed method the selection of software components is based upon the quality criteria. We also know that for system quality criteria ISO/IEC 25010:2011 standard is used . Figure 1 shows dependency of software quality over components, building blocks in component based development. ANP is based on (1) goal, (2) criteria, and (3) alternatives.

Mathematical software component selection is represented as selection = $\sum_{i \in P} Pi$ where P={effectiveness, efficiency, satisfaction, safety, usability}. According to Saaty's algorithm, the fundamental scales for judgment are given that show us which component is more important than the other.

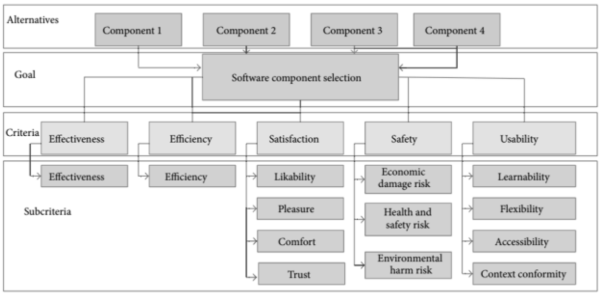


Figure 1 : Software component selection model based on ISO/IEC 25010:2011 using ANP[14]

3.0 Role of Repository in Component Based Software Development

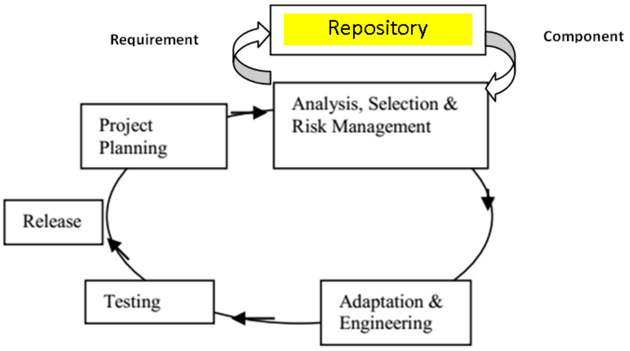


Figure 2. Role of component repositories in CBSD

Component repositories play a vital role in successful component based development. As figure 2 shows after project planning the first step is to identify a competent reliable repository, second step is to search component and third step component is integrated in the system.

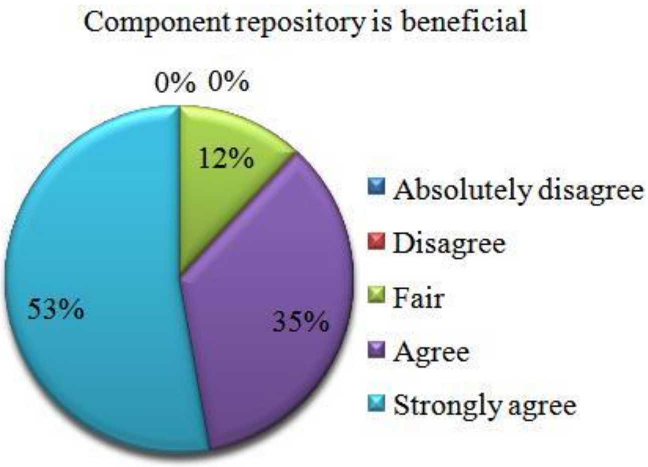


Figure 3. Survey conducted on Software Component Repository[17]

As per the survey conducted[17] over 130 software developers to find out the requirement of software component repository and its prospective benefits. Figure 3 show the certain results. So its very important to work upon software repository to get highest potential of the component based development.

Here are the 2 essential and important divisions of the repository

3.1 Component repository Management

Component repository contain heterogeneous types of software components. These components may be code, design, document or a test plan. Management of these components is a very tedious task. Few important tasks of a software component repository are as follows

a. Admin and User Account Management
b. Adding components
c. Verification for functionalities and threats
d. Classification
e. Ensuring Availability
f. Downloading of required component
g. Collection of feedback
h. Gathering change requirement
i. Version management of the components
j. Removal of Obsolete components

3.2 Component Search Engine and Retrieval

a. Easy interface for searching component
b. Efficient searching algorithms
c. Obtaining minimum number and most appropriate component list
d. Ensuring best qualified component retrieval

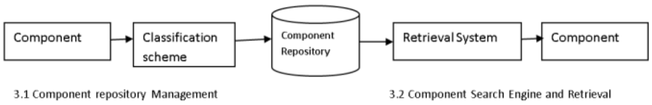
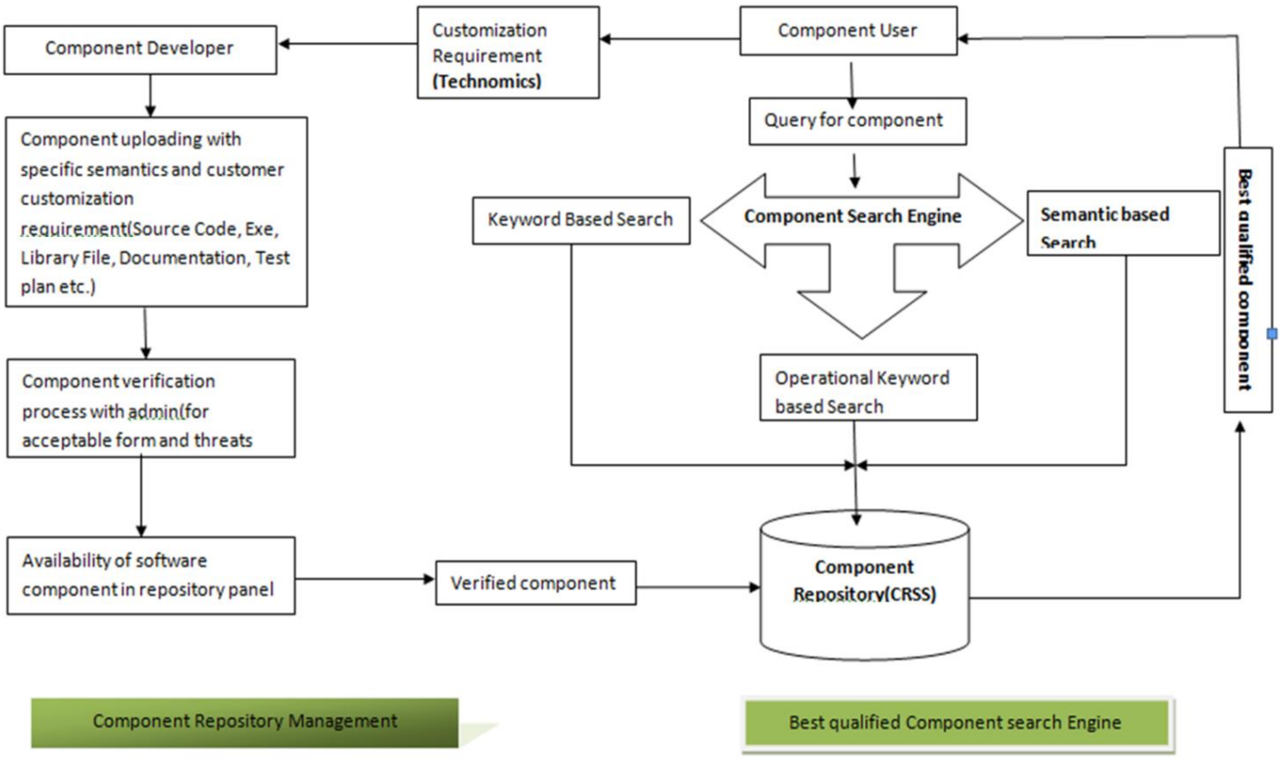

Component
Classification scheme
Component Repository
Retrieval System
Component
3.1 Component repository Management
3.2 Component Search Engine and Retrieval

Figure 3 : Essential Division of the Component Repository Architecture

4.0 Semantic Search enabled software component repository(SSESCR)

SSESCR suggests a framework which keeps all the features of a efficient repository. This is a web based repository which allows all the developers to search, submit, download, replace and emend components. Figure 4 shows this frame work . Besides having component repository management and retrieval management. It contains a efficient translation tool technomics [17] also which provides freedom to the component user to change the component properties up to a certain level.

Figure 4 :Semantic Search enabled software component repository(SSESCR)


Component Developer
Customization Requirement (Technomics)
Component User
Query for component
Component uploading with specific semantics and customer customization requirement(Source Code, Exe, Library File, Documentation, Test plan etc.)
Keyword Based Search
Component Search Engine
Semantic based Search
Best qualified component
Operational Keyword based Search
Component verification process with admin(for acceptable form and threats
Availability of software component in repository panel
Verified component
Component Repository(CRSS)
Component Repository Management
Best qualified Component search Engine

4.0 Major division of SSESCR

4.1. Component Repository Management

Its functions are

a. Adding new components with essential documents
b. Verification process
c. Classifying components
d. Update components
e. Version Management
f. Remove components

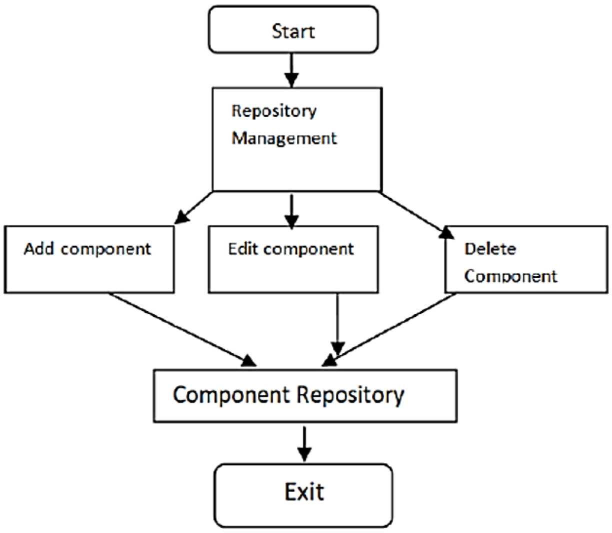


Figure 5. Component Repository Management[18]

4.2. Component Retrieval Management

a. Searching component
b. Retrieval Process

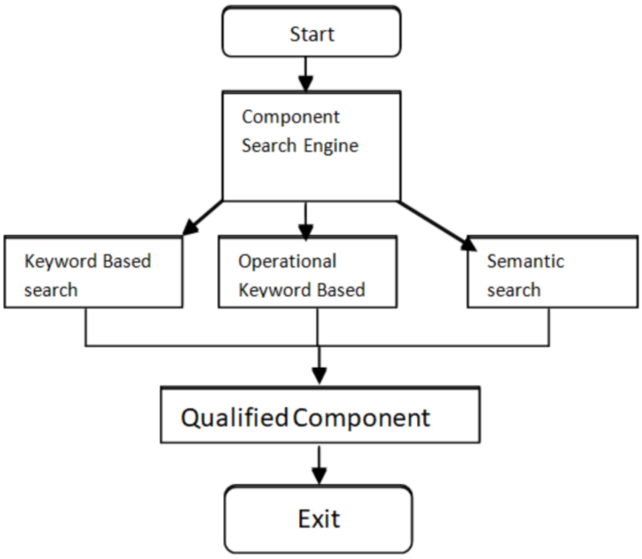


Figure 6 : Component Retrieval Management[18]

4.3 Technomics Translator[17]

There are four modules
i. User Interface
ii. Database
iii. Technomics, and
iv. Check Authenticity

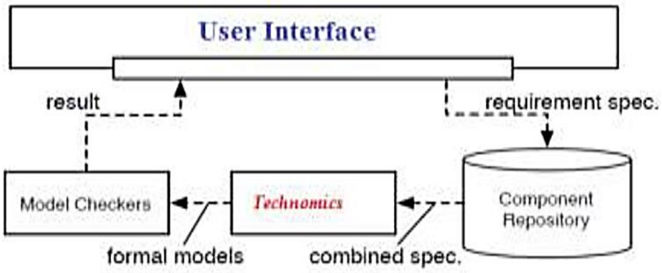


Figure 7 : Translation of user requirements specification to qualified component[17]

III. CONCLUSION

SSESCR is the best solution from the problems recent component based software is facing. Proposed framework provide solution for access of component libraries for public domain, expert groups and developers as well. On other side semantic search engine supports component user to find best qualified component[18]. Semantics may be enhanced with the availability of new more specialized properties. It will not only increase the belief of the software professionals in component based development but also give a hike in software productivity. Qualified component search will reduce the time and efforts. SSESCR is equally important as it will let software professional for add component and download components for better software development time.

IV. REFERENCES

[1]. A. Orso, M. J. Harrold, and D. S. Rosenblum, "Component Metadata for Software Engineering Tasks", Proc. 2nd Int. Workshop on Engineering Distributed Objects, 2000, pp. 126-140.

[2]. P. Vitharana, F. Zahedi, and H. Jain, "Knowledge-Based Repository Scheme for Storing and Retrieving Business Components: A Theoretical Design and an Empirical Analysis",

IEEE Trans. on Soft. Engineering., vol. 29, issue 7, July 2003, pp. 649-664.

[3]. OMG, Reusable Asset Specification – Version 2.2, 2005.

[4]. G. Elias, M. Schuenck, Y. Negócio, J. Dias, and S. Miranda, "X-ARM: An Asset Representation Model for Component Repository", 21st ACM Symposium on Applied Computing, 2006, pp. 1690-1694.

[5]. W. Meier, "eXist: An Open Source Native XML Database", NODe 2002 Web and Database-Related Workshops on Web, Web-Services, and Database Systems, 2002.

[6]. R. Goldman and J. Widom, "DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases", 23rd Int. Conf. on Very Large Data Bases, 1997, pp. 436-445.

[7]. T. Brito, T. Ribeiro, and G. Elias, "Indexing Semi-Structured Data for Efficient Handling of Branching Path Expressions", 2nd Int. Conf. on Advances in Databases, Knowledge, and Data Applications, 2010, pp. 197-203.

[8]. M. P. Paixão, L. Silva, T. Brito, and G. Elias, "Large Software Component Repositories into Small Index Files", 3rd Int. Conf. on Advances in Databases, Knowledge, and Data Applications, 2011, pp. 122-127.

[9]. Chintakindi Srinivas, Vangipuram Radhakrishna, C.V. Guru Rao. Clustering Software Components for Program Restructuring and Component Reuse Using Hybrid XNOR Similarity Function. Procedia Technology, 2014 ; (12): 246-54.

[10]. Radhakrishna.V, C.Srinivas, C.V.GuruRao. High Performance Pattern Search algorithm using three sliding windows, International Journal of Computer Engineering and Technology, 2012; 3: 543-552.

[11]. P. Niranjan , C. V. Guru Rao , "A Model Software Reuse Repository with an Intelligent Classification and Retrieval Technique", Computer Science and Engineering, Vol. 1 No. 1, 2011, pp. 15-21. doi: 10.5923/j.computer.20110101.03.

[12]. Bagheri, E., and F.Ensan. ." Reliability estimation for component-based software product lines", Canadian Journal of Electrical and Computer Engineering,2014 37(2):94-112 .

[13]. R. A. Pyne and E. K. Mugisa, "Retrieving Software Components from a Heterogeneous Repository: a Semantic Syntactic Approach," 13th Annual IEEE International Symposium and Workshop on Engineering of Computer-Based Systems (ECBS'06), Potsdam, 2006, pp. 479-480.

[14]. S. Wagner, "Quality models," in Software Product Quality Control, Springer, Berlin, Germany, 2013

[15]. Nazir S, Anwar S, Khan SA, Shahzad S, Ali M, Amin R, et al., editors. "Software component selection based on quality criteria using the analytic network process. Abstract and Applied Analysis" Hindawi Publishing Corporation.; 2014.

[16]. Vishnu Sharma, Vijay Singh Rathore,"Three Phased Component Retrival Technique (TPCRT) for Best Qualified Component"International Journal of Applied Sciences & Engineering (IJASE) 1(2): October, 2013: 69-73.

[17]. Vishnu Sharma, Vijay Singh Rathore, Technomics: "Approach towards best qualified components"International Journal of Advanced Studies in Computer Science and Engineering, volume 5, issue 7, 2016

[18]. Hasan Kahtan, Nordin Abu Bakar et.al, "Awareness of embedding security features into Component-based software development model: a survey" Journal of Computer Science 10 (8): 1411-1417, 2014 ISSN: 1549-3636 © 2014 Science Publications doi:10.3844/jcssp.2014.1411.1417 Published

Online 10 (8) 2014 (http://www.thescipub.com/jcs.toc)

[19]. Vishnu Sharma, Vijay Singh Rathore," Recent trends in Component Based software development and Efficiency analysis of Semantic search based component" International Journal of Scientific Research in Computer Science, Engineering and Information Technology © 2018 IJSRCSEIT Volume 3 Issue 8 ISSN : 2456-3307