

Software Component Reuse with various Attributes

Navjot Kaur

Research Scholar, Department of Computer Science and Engineering

RIMT-IET College

Mandi Gobindgarh, Punjab, India

Dr.Sushil Garg

Professor & Principal

RIMT-MAEC College

Mandi Gobindgarh, Punjab, India

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. In our research we have provided efficient software component reuse search.

Keywords – Component Reuse, Software Development, Faceted Classification, Code level components, Functional Reuse.

I. INTRODUCTION

Software reuse is the process of implementing or updating software systems using existing software assets. Software assets, or components, include all software products, from requirements and proposals, to specifications and designs, to user manuals and test suite. Reuse allows us to efficiently create reuse of software components improves overall software quality, reduce software costs, and deliver software with fewer defects. Reuse allows us to efficiently create software systems from existing software artifacts rather than building software systems from scratch. The initial use of the software reveals any design and implementation faults. These are then fixed, thus reducing the number of failures when the software is reused. Reduced process risk if software exists, there is less uncertainty in the costs of reusing that software than in the costs of development. This is an important factor for project management as it reduces the margin of error in project cost estimation. The software reusable components need not be limited to source code fragments but also include design structures, specifications, documentation and so on. Based on this reuse can be categorized into four levels[2].

1. Code level components
2. Entire applications
3. Analysis level products
4. Design level products

Code level component reuse occurs most frequently (functional modules, procedures, subroutines, libraries, etc.). However it is criticized for lacking in reuse potential, as the level of abstraction is low for these components. Reusing entire applications often means using off-the-shelf packages, or minimal adaptation of a specialized product applied to a new customer, but it is not always feasible (e.g., real-time software environments). Perhaps the less represented areas are design products, which allow reuse of similar system implementation strategies, and analysis products, which allow reuse of knowledge about real world domains. The analysis level products are possibly the most powerful of all as these allow description and manipulation of real world domains. A simple example of a reusable software part is Reusable software components can be simple like familiar push buttons, text fields list boxes, scrollbars, dialogs . every thing visible in Java interface are reusable components [3].

II. COMPENET REUSE WITH FACETED CLASSIFICATION

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 various approaches for software reuse are explained below.

2.1 Application Framework

Collections of concrete and abstract classes that can be adapted and extended to create application systems. It is used to implement the standard structure of an for a specific development

environment. A framework is a incomplete implementation plus conceptually complete design. Application frameworks became popular with the rise of, since these tended to promote a standard structure for applications.

2.2 Application Product Lines

Application product lines, or Application development, refers to methods, tools and techniques for creating a collection of similar product line systems from a shared set of software assets using a common . An application type is generalized around a common architecture so that it can be adapted in different ways for different customers. A type of application system reuse. Adaptation may involve component and system configuration; selecting from a library of existing components; adding new components to the system; or modifying components to meet new requirements [3]

III. PROPOSED METHOD

Effective storage and retrieval of software components is much essential in software components reuse process. But present system are not that effective, the main problem is that these system don't has 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 allow user to query the system and repository stores the information regarding component.

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.

IV. METHODOLOGY

In our research we will start 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. A basic interactive engine will be 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 consider 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.

V. CONCLUSION

The paper shows various types of software component reuse process and techniques used in the efficient retrieval of the searched component. The main thing about the component reuse is to save resources and utilization of already available resources of the software development. This is our on going research and we have proposed a efficient mechanism for component reuse with advance search of the component based on faceted attributes.

REFERENCES

- [1] Dr.S.S.V.N.Sharma," Building Reusable Software Component For Optimization Check in ABAP Coding", International Journal of Software Engineering & Applications (IJSEA), Vol.1, No.3, July 2010.
- [2] Michael L. Nelson,"Barriers to Software Reuse and the Projected Impact of World Wide Web on Software Reuse", 1996, pp 1234-1237.

[3] B.Jalender, Dr A.Govardhan, Dr P.Premchand," Designing Code Level Reusable Software Components", International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.1, January 2012.

[4] V. Subedha," Process Model for Reusability in Context-Specific Reusable Software Components", Indian Journal of Computer Science and Engineering (IJCS), Vol. 3 No. 1 Feb-Mar 2012.

[5] Ajay Kumar," Measuring Software Reusability Using SVM Based Classifier Approach", International Journal of Information Technology and Knowledge Management, January-June 2012, Volume 5, No. 1, pp. 205-209.

[6] Rubén Álvarez-González," Position Paper: Improving Source Code Reuse Through Documentation Standardization", International Book Series "Information Science and Computing, 2011.

[7] Deepti Aggarwal," Software Reuse - A Compendium", IJRIM Volume 2, Issue 2, February 2012.

[8] Wouter Spoelstra," Software Reuse in Agile Development Organizations - A Conceptual Management Tool", SAC'11, March 21-25, 2011, TaiChung, Taiwan.