A Review on Reusability of Component Based Software Development

Shambhu Kr. Jha, Dr. R. K. Mishra

•

Research Scholar Mewar University, Chittorgarh, India
skjha2@amity.edu
National Informatics Centre, New Delhi, India
R.k.mishra@nic.in

Abstract

People across software community have marked the true endeavor to go round development of software through component based software development (CBSD) practices. Reusability of software component has a very positive impact on development time, cost, reliability and marketability of the software. In this paper we are discussing about bottlenecks of software component reusability and trying to provide some useful guidelines to improve the reusability for the component based software developer which can further improve the productivity, reduce the development cost. This paper also discusses the steps for effectively storing the software component in component repository that is helping the component user in finding the most eligible component for reuse which will progress the component based software development.

Keywords: CBSD, Component Repository System (CRS), Component Classification (CC), COTS.

I Introduction

Reusability of software Component is considered as best answer to multiple problems faced by software developer community. It is highly accepted and appreciated for faster development of large and multifaceted system, reducing development expenditure and improving overall quality of software. Many software developing organizations that are functioning in this domain have already started developing their own component library and claimed considerable benefits from it. They are trying to develop such software component which is reusable in multiple implementations without any change or only with minor changes. Such components are thoroughly documented and certified in order to make it portable across different hardware and operating system. Other software developing organizations have supported their reuse policy with Commercial off-The Shelf (COTS). Customer for such software component has no access to source code in most of the cases therefore they are only looking for certified software component which is reliable and trustworthy. As the size of component library is increasing due to their increasing demand, component selection process is another big challenge for component reusability. Therefore many software organizations are spending a large amount of time in development of appropriate component repository and efficient retrieval of reusable component. Since the selection of the suitable components has a foremost impact on the software project and finally the resulting products. Researchers in this domain have suggested classifying the component repository into three different categories based on availability of source code for individual component to improve the reusability. This will also make the retrieval and selection of software component easier and faster. A global survey conducted recently by the research community has observed that almost two third of the software developing organizations are practicing component based approach in course of software development.

Component integrator lacks in trust for a software component developed by a third party vendor during deployment of Component as per defined component architecture. This is again a huge challenge for component reuse. The reason behind mistrust and lack of confidence among component integrator is that majority of the component is available without source code and without proper documentation. A component developed in a particular programming language is available for reuse in different operating environment with different prospective of component user. This characteristic of the component is again causing a big hurdle for component reuse. A component of large size is again not very friendly to component user due to its complexity and interoperability.

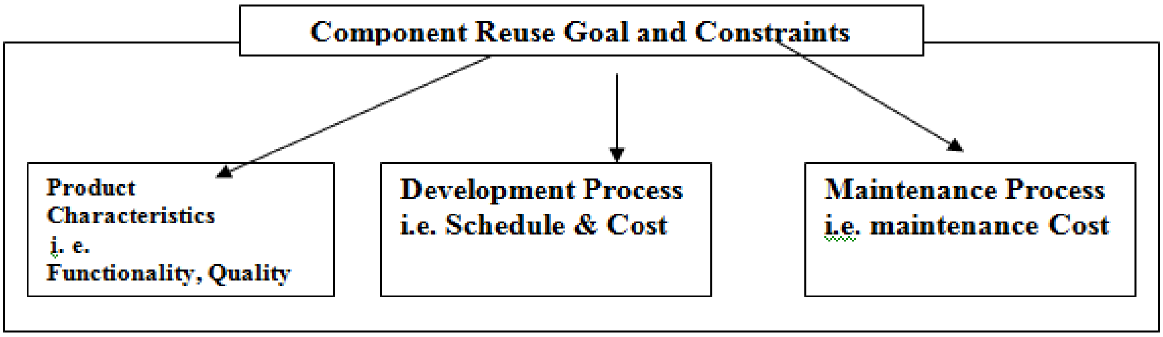
Component Reuse Goal and Constraints
Product Characteristics i. e. Functionality, Quality
Development Process i.e. Schedule & Cost
Maintenance Process i.e. maintenance Cost

Figure: 1 (Reuse Goal and Constraints)

II Identifying reusable software Component

Concept of software reuse is continuously improving due to rapid demand of new software in less time and less cost. In general Software Company is continuously facing both technical and economical challenges due to ever changing business requirement. Effective software reuse can always help in improving business profitability and reducing the development time of the software. Component based software development is solely depending on reuse of preexisting software component. The purpose of CBSD is to built once and reuses any number of times with no modifications or minor modifications. Therefore reuse policy for Component Based software development can be generally categorized into:

Reuse of preexisting software component without any change.

Reuse of preexisting software component with change.

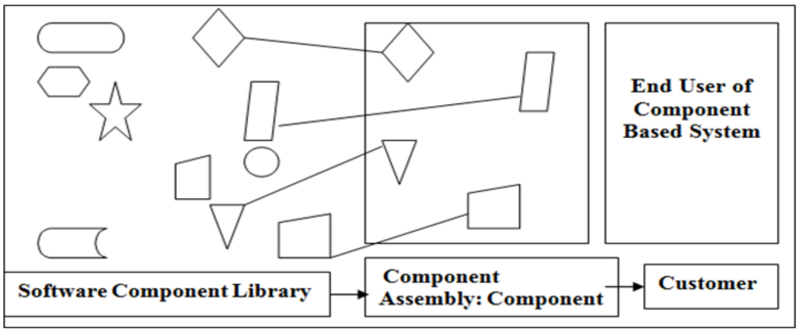
End User of Component Based System
Software Component Library
Component Assembly: Component
Customer

Figure 2: Component assembly process used by component integrator

As we already discussed about unavailability of source code for most of the software component due to its black box nature. Therefore its implementation is entirely hidden from component user. It has to be used as plug and play devices. At the time of reuse we cannot make any change into it. They are directly used into the application and integrated via component interface.

On the other hand if the software component considered for reuse needs some minor modifications due to little discrepancy between component functional and non functional characteristics and characteristics of component based software which is presently going to develop then we prefer to have software component which are developed in-house. In that case source code of the component is available for required modifications. Even after change the modified components need to be thoroughly tested before plugging it into new system.

III Guidelines for improving reusability

Reuse goals of software component and its process of integration for development of component based software is mentioned in Figure 1 & 2.Component reusability has very high impact on overall quality of the component based software. On the basis of review of various aspects of component reuse we are suggesting following guidelines to the different stakeholders of software component which can improve component reusability:

Table 1: Findings and Observations for improving component reusability

S. No	Guidelines
1	Language used for development of new component must be platform independent.
2	Code structure used for development of component must be of moderate size and easy to test.
3	During component development it must be thoroughly documented for easy to use and future implementation.
4	Certification of each component must be considered on a serious note by component developer
5	Design specification of component must be optimized for next implementation.
6	Development of new component must be based on its portability across different hardware and operating system .
7	Applying suitable reuse metrics for better understanding of reusable components
8	Standardization of component retrieval process

IV Repository Development of software component

Most of the component users are facing plenty of challenges in effectively retrieving the reusable components to develop component based software as per specific user requirements. Large number of components in the component repository makes the component retrieval process more and more challenging and it takes lots of time and consumes more resources. Proper classification, cataloging and certification of the component need to be performed for every component before placing it into component repository. It supports the component user not only in faster retrieval of component but also in finding most suitable component in application development. Component repository is most valuable asset in the process of developing component based software. Repository for the software component must be upgraded on regular basis. Budgeting is another important aspect which must be communicated to the component used time to time. A systematic approach for developing component repository is mentioned in figure 3. Software component information is stored into the repository using tools for Component Repository System (CRS) as mentioned in the figure. Searching of component is based on using suitable keyword. This method of searching helps in faster retrieval of component without much knowledge about it.

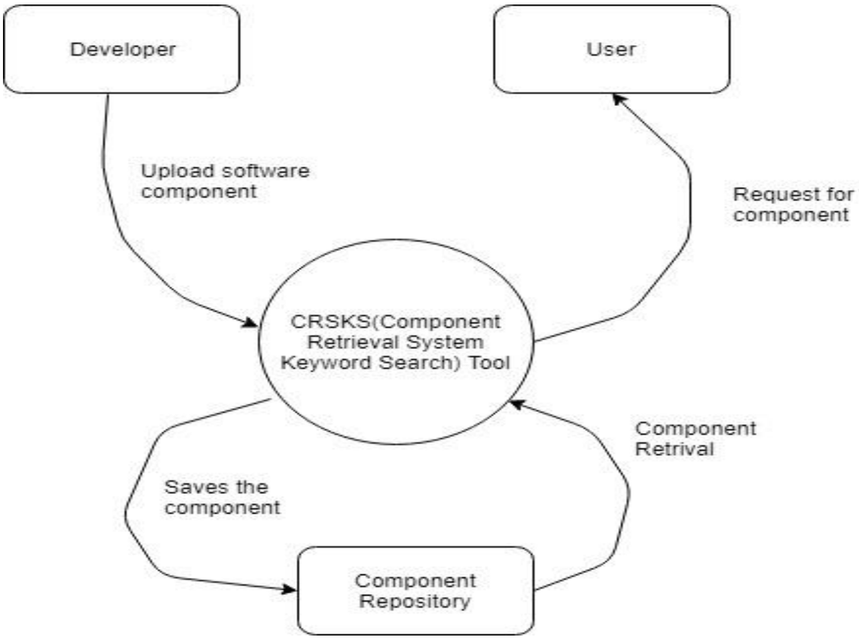

Figure 3: Repository Development for Software Component

V Conclusion and Future Work

In this paper we have discussed about various reusability constraints of component based software development and further made effort to provide some useful tips about improving component reuse to the different stakeholders of component-based software. This paper also gives an insight view of development of component repository which is helping the component user in effective and faster retrieval of software component without much knowledge about it. Retrieval of component is based on keyword search technique which can be further improved based on some advance search algorithm. This algorithm may help the component user in improving component reusability and faster retrieval of components.

References

[1] Shambhu Kr Jha, R.K.Mishra, ″Multi criteria-based retrieval techniques for reusable software components from component repository″ International Journal of Engineering Applied Sciences and Technology(IJEAST) ISSN: 2455-2143[online], Vol 1, Issue 6, Page 88-91, Apr 2016 (DOI 10.7615/IJEAST).

[2] Arun Sharma, Rajesh Kumar, P S Grover, "Investigation of reusability, complexity and customizability for component-based systems", ICFAI Journal of IT, Vol.2 Issue. 1, June 2006.

[3] V. R. Basili, G. Caldiera, and H. D. Rombach."GoalQuestionMetric Paradigm," Encyclopedia of Software Engineering, ed. J. J. Marciniak. New York John Wiley & Sons, 1994.pp. 528-532.

[4] T. Birgerstaff and C. Richter, "Reusability Framework, Assessment, and Directions, IEEE Software, vol. 4, March. pp. 41-49, 1987.

[5] B. W. Boehm, J. R. Brown, and M. Lipow "Quantitative Evaluation of Software Quality," pp. 592-605, 1976. Proceedings of the Second International Conference on Software Engineering. IEEE.

[6] Shambhu.Kr Jha, Mishra R.K. "Predicting and Accessing Security Features into Component-Based Software Development: A Critical Survey". Software Engineering. Advances in Intelligent Systems and Computing, 2018 vol 731. Springer, Singapore. ISBN: 978-981-10-8848-3. https://doi.org/10.1007/978-981-10-8848-3_28 Page: 287-294.

[7] Nasib Singh Gill, "Reusability Issues in Component-based Development", ACM SIGSOFT SEN Vol. 28 No. 6, pp. 30.