ISSN: 2230-9926

Available online at http://www.journalijdr.com

IJDR

International Journal of Development Research
Vol. 5, Issue, 04, pp. 4259-4261, April, 2015

International Journal of
DEVELOPMENT RESEARCH

Full Length Research Article

ACCESSING SOFTWARE QUALITY FOR COMPONENT-BASED SOFTWARE THROUGH TRUSTWORTHINESS AND DEPENDABILITY ANALYSIS

[1*]Shambhu Kr. Jha and [2]Dr. R. K. Mishra

[1]Research Scholar Mewar University, Chittorgarh, India
[2]National Informatics Centre, India

ARTICLE INFO

Article History:

Received 06th January, 2015
Received in revised form
25th February, 2015
Accepted 15th March, 2015
Published online 30th April, 2015

Key words:

CBSD,
Software Component,
Software Reuse,
Trustworthiness,
Dependability.

ABSTRACT

We are witnessing an enormous growth in the use of software in business, manufacturing, administration, defense, medical and research. Software is not only significant in technical systems but also has become an important factor in lots of other disciplines. This has motivated the software community to search for flexible, timely, cost-efficient, trustworthy and dependable software development methodology. The kind of flexibility presented by Component-Based Software Development (CBSD), and the opportunity of reducing production costs and time through software reuse, has lead to an increased preference for component based development techniques. In component based development methodology a component plays a role of individual piece of software that includes enough functionality to be useful on its own. The components can also be further reused number of times to create various large and complex software systems. The feature of a component that enables it to be assembled with other components into a system is its built-in interface(s) which are usually based on standards. Trustworthiness and dependability plays significant role in growth of software system, especially when the software development is component based. Trustworthiness and dependability of complex information systems that are embedded in the infrastructure supporting advanced society has become a nationwide and world-wide concern of the highest priority.

INTRODUCTION

Traditional methods of software development involves too much of customization, multifaceted integration and deployment, lack of interoperability, complicatedness in dealing with changes, high development time and cost and lack of run-time flexibility. However software development has continuously evolved to deal with such challenges i.e. from structure programming approach, to object-oriented approach and object-oriented approach to component based approach. Component-Based Development (CBD) approach is to deal with the ever-present difficulties mentioned above.

Figure shown above is an example of a component based software system designed to fulfill the requirement of specific software application. This figure shows a system composed of five different components namely A, B, C, D and E. Each component of the system is capable of performing certain functionalities independently.

Corresponding author: Shambhu Kr. Jha
Mewar University, Chittorgarh, India

A Component is integrated with remaining components into the system through its built-in interface(s). As components and application software have separate lifecycles and different kinds of requirements, there is some risk that a component will not completely satisfy the application requirements or that it may include hidden characteristics which are unknown to application developers. This paper discusses the present situation of component-based development approach and some of its critical issues i.e. managing trustworthiness and dependability of the software components through software reuse. Our goal is to present a brief overview of the concepts and techniques that have evolved in the field of component based software methodology to analyze trustworthiness and dependability by improving component reusability.

Selecting Trustworthy and Dependable Software Component

In component based software development methodology, software developer or software component users (integrators) experience a lack of trust in third-party software components.

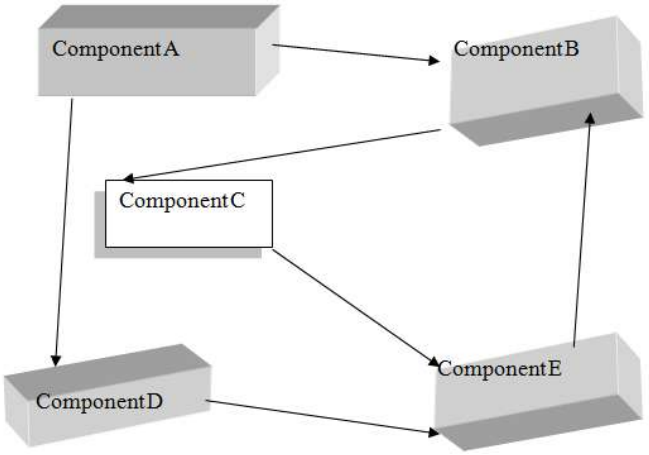

Figure 1. Component-based systems are built from software components

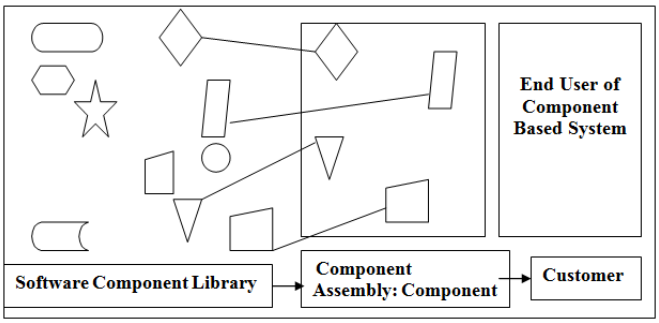

Figure 2. Stakeholders of the CBSD paradigm

Trust requirements for software integrators – the frontline component consumers or user are quite different from the end-users of an application software. The foremost requirement of any trustworthy and dependable component would be self-disclosure of identity, origin, and security properties that a component has. However, other non-technical factors have tremendous influences in forming trust as well. In this endeavor, we focus on only those factors that are related to the technical aspects of software products. To manage the trustworthiness and dependability in the component based software system we have to ensure two things at the component level.

i) An individual component of the component based software system is performing exactly in the same way as it is required to perform.
ii) Each component of the software system is individually not performing anything which is not expected from the components to perform.

Software Component Reuse and Reusability

Through component reuse it is possible to apply formerly developed software components which are functionally suitable for different application program in different operating environment. Component reuse not only helps in designing and developing new software but also is a major contributor for analyzing trustworthiness and dependability of the component

Component Based Reuse can be broadly classified into two types

Reuse of previously developed component without any change.
Reuse of previously developed component with change.

Reuse of an existing software component is simply based on selecting suitable software component from a software component database and plugging it into new software application being developed. However reusing an existing component without any change is a very difficult choice for the component users (integrators). The main reason is because of difference in programming language used or operating environment for the new software being developed. On the other hand reuse of an existing software component with change is also a very difficult and challenging task because of its black-box nature. Another challenging task in case of reuse with change in software component is to identify those parts of the components which really require change. Even after changes the modified components need to be thoroughly tested before plugging it into new system for ensuring the trustworthiness and dependability. Selection of the best suitable components for reusability is decided on the basis of which component is more easily adapted for development of new software applications.

General guidelines for predicting reusability are

1. Simple structure,
2. Small size of code
3. Good documentation.
4. Need of less Customization
5. Easily Testable

Above guidelines may be used in developing trustworthy and dependable component based systems through reusable software component and to rank candidate components for reuse with the assumption that two components have similar functionality.

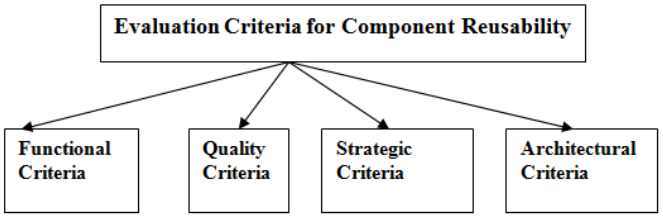

Figure 3. Factors effecting the selection of reusable components

Analyzing Dependability and Trustworthiness of Software Component

Software systems are not always trustable and dependable because they do not always work in a desired manner under different operating environments. In case of component based software development trustworthiness and dependability issues are very important and critical as we have to identify functionally suitable components from component library for different application programs which can work in different operating environment.

Table 1. User ratings for Component A1, A2, B1, B2, C1 and C2

Component Name	Trustworthiness and dependability				
	Conformance	Precision	Testability	Fault-tolerance	Safety
A1	9.1	9.2	8.9	9.0	9.2
A2	9.5	9.8	9.7	9.8	9.7
B1	9.9	9.8	9.5	9.5	9.7
B2	9.3	9.1	8.9	9.0	9.1
C1	9.7	9.9	8.9	9.6	9.9
C2	9.0	8.9	9.7	8.9	9.2

Faults and failures of the software components arise frequently causing directly or indirectly heavy losses for component integrator or end users. Software trustworthiness and dependability can be analyzed comprehensively by considering the characteristic attributes of conformance, precision, testability, fault-tolerance and safety. For our empirical study we have considered six software components A1, A2, B1, B2, C1, and C2 developed using different software environment by different software developers. Among them we have to select one component out of Component A1 and A2, one component out of B1 and B2 and one component out of C1 and C2. This is based on the fact that Component A1 is performing same functionality as A2, similarly Component B1 and B2 has similar functionality and same is the case of component C1 and C2. We have distributed these software components among 10 different component users and allowed them to rate each software component considering its conformance, precision, testability, fault-tolerance and safety on scale of 10.

On the basis of rigorous usage of these components by 10 different components user data compiled is represented graphically as bellow:

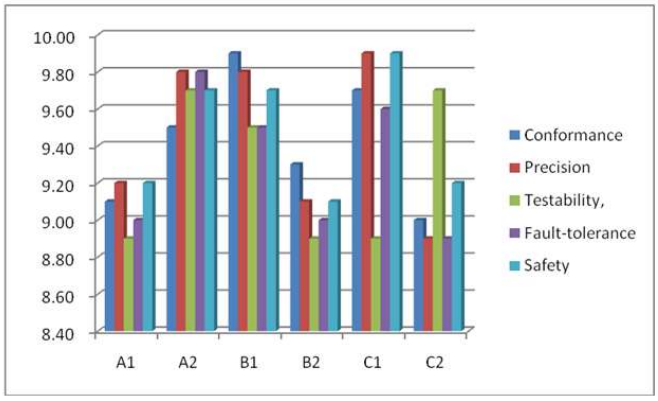

Figure 4. Result Analysis for Component Selection

After analyzing the above graph it is quite understandable that among component A1 and A2, A2 is the better choice. Similarly if we compare between Component B1 and Component B2 or Component C1 and Component C2, we can conclude that B1 and C1 should be considered for component based software having high trustworthiness and dependability.

Conclusion

Component integrators or component users should very cautiously handle the task of selecting functionally suitable software components from component library or COTS for the development of component based software. Components with high degree of reuse have high potential of conformance, precision, testability, fault-tolerance and safety. Each time the components are being used in different application software under different operating environments, higher is the probability of components being more dependable and trustable. However the above facts cannot always be taken for granted because the functional or non functional requirements of the component at individual level is different from the application level.

REFERENCES

Diaconescu, A. and J. Murphy, 2003. "A Framework for Using Component Redundancy for Self Optimizing and Self-Healing Component Based Systems", WADS workshop, ICSE'03, USA.

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

Boehm, B. and Abts, C., 1999. "COTS Integration: Plug and Pray", IEEE Computer, 32(1).

Jha., S. K. 2007. "Software Quality Model for Component Based Software",

John R. Blanchette Sr, 2005. "Pros And Cons Of Using Cots Products", IEEE.

Nasib Singh Gill, 2000. "Importance of Software Component Characterization for Better Software Reusability", ACM SIGSOFT SEN Vol. 31.

Jha, S. K., 2012. Quality Management for Component Based Software System, LAMBERT ACADMIC PUBLICATION GERMANY

Singh, Y., P. K. Bhatia and O. P. Sangwan, 2011. "Software Reusability Assessment Using Soft Computing Technique "ACM SIGSOFT Software Engineering Notes, Volume 36.

Nikita Soni and S. K. Jha, 2014. "Component Based Software Development: A New Paradigm", International Journal of Scientific Research and Education, Volume 2, Issue 6.

****