Chapter 2
Component Certification Process and Standards

Parminder Kaur
Guru Nanak Dev University, India

Hardeep Singh
Guru Nanak Dev University, India

ABSTRACT

Component-based technology deals with the process of assembling existing software components in an application such that they interact to satisfy a predefined functionality. The success of component-based software depends on system integrator’s ability to select the most suitable software components for their intended application (Leavens and Sitaraman, 2000; Voas and Payne, 2000; Crnkovic et al., 2001). One persistent and largely unaddressed challenge is how the consumers of software components can obtain a meaningful level of trust in the runtime behaviour of software components. The most frequently cited concerns are centered on issues of security and component behaviour (Wallnau, 2004). Certification is a practical, proven means of establishing trust in various sorts of things in other disciplines and is, therefore, a natural contender for developing trust in software components. This is only possible if component suppliers have clear and reliable information about their component’s functional and non-functional properties. The component attributes allow system integrator to better classify the components. The reliability of a component-based software system is dependant on the reliability of the components that is made of. Proper search, selection, and evaluation process of components is considered as cornerstone for the development of any effective component-based system (Alexandre, 2010). This chapter defines certain properties of a component, which are helpful during their classification process along with component certification/accreditation process, which further helps component suppliers to issue the guarantee of claimed functional properties and quality attributes. Component certification framework is also discussed to evaluate the quality of software components with the help of component quality model and measurement mechanism.

DOI: 10.4018/978-1-4666-2958-5.ch002
INTRODUCTION

Component-based technologies simplify functional decomposition of complex systems and support building of re-configurable compositions and tuning of component compositions for the particular context they are used in. Various component models like CORBA from Object Management Group, COM/COM+/DCOM/.NET from Microsoft, and Enterprise JavaBeans from Sun Microsystems and OpenDoc from Apple (John, 1994), [App] focus on functional features of the components and entire component systems and are unable to estimate the non-functional properties of the components and their compositions.

One of the most compelling reasons for adopting component-based technology is the principle of reuse. The idea is to build software from existing components for reduced development time and improved product quality. The top objective is to avoid reinvention, redesign and reimplementation when building a new product, capitalizing on previously done work that can be immediately deployed in new contexts (Jacobson et al, 1997).

Reliable and high-quality software systems can build only by using components of high quality and reliability. Reused components must be free of design and implementation flaws. Usually components benefit from multiple reuses in that they are thoroughly tested. The idea behind the certification of components is to guarantee that a specific set of quality guidelines has been met (Michael and John, 1993). Traditional quality certification standards e.g. ISO/IEC 12207, Capability Maturity Model – CMM and Capability Maturity Model Integrated – CMMI are focused on the software development process. However, the certification of components for safety and business critical application must consider both the development process and the product intrinsic quality. Several ISO standards like ISO/IEC 12119, ISO/IEC 9126, ISO/IEC 25051 are now focusing on component certification.

RELATED WORK

Certification has the goal to offer a general scheme that indicates the quality or compatibility of a component in respect to certain properties. (Wohlin and Runeson, 1994) has presented a usage modeling technique, which can be used to formulate usage models for components. This technique will make it possible not only to certify the components, but also to certify the system containing the components. The usage model describes the usage from a structural point of view, which is complemented with a profile describing the expected usage in figures. The failure statistics from the usage test form the input of a hypothesis certification model, which makes it possible to certify a specific reliability level with a given degree of confidence. This certification model becomes the basis for deciding whether the component can be accepted, either for storage as a reusable component or for reuse.

In early years, software products were based on related monolithic building blocks whereas now Component-Based Development (CBD) appeared as a new perspective for software development, aiming at breaking monolithic blocks into interoperable pre-tested components with the promise of reduced development complexity as well as its cost (Sametinger, 1997). (Voas, 1998, 1999) suggested that independent agencies such as software certification laboratories should assume the role of software product certification and suggests that the only approach that consumers can trust is the certification provided by agencies that are completely independent from the software product providers. A certification methodology using automated technologies, such as black-box testing and fault injection to determine whether the component fits into a specific scenario is defined. This methodology uses three quality assessment techniques to determine the suitability of a candidate COTS component. Black-box component testing is used to determine whether the component quality is high enough.
Related Content

Classification of Bug Injected and Fixed Changes Using a Text Discriminator
[www.igi-global.com/article/classification-of-bug-injected-and-fixed-changes-using-a-text-discriminator/121547?camid=4v1a](www.igi-global.com/article/classification-of-bug-injected-and-fixed-changes-using-a-text-discriminator/121547?camid=4v1a)

Is Lean Agile and Agile Lean?: A Comparison between Two Software Development Paradigms
[www.igi-global.com/chapter/lean-agile-agile-lean/51967?camid=4v1a](www.igi-global.com/chapter/lean-agile-agile-lean/51967?camid=4v1a)

Enhancing ERP System with RFID: Logistic Process Integration and Exception Handling
[www.igi-global.com/article/enhancing-erp-system-rfid/58513?camid=4v1a](www.igi-global.com/article/enhancing-erp-system-rfid/58513?camid=4v1a)

Person Identification Using Top-View Image with Depth Information
[www.igi-global.com/article/person-identification-using-top-view-image-with-depth-information/89776?camid=4v1a](www.igi-global.com/article/person-identification-using-top-view-image-with-depth-information/89776?camid=4v1a)