Chapter 1

Introduction

I had rather feel compunction than understand the definition thereof.

Thomas a Kempis

The goal of Component-Based Software Engineering (CBSE) is to enable the assembling of software systems from existing, independent, components. Component-Based Software Engineering (CBSE) is a natural progression in the evolution of software engineering as the use of prefabricated components is fundamental to any mature engineering discipline. Despite the current successes in software components, there are many barriers to the widespread adoption of software components.

In the market, most components are shipped in a binary form that leads to a 'black-box' effect – a system developer has no knowledge of the component’s internal mechanism. Without an access to the sources code, component users have to rely on documentation to answer questions about important information including the component’s behavior and resource requirements. Unfortunately, component documentation often does not contain enough detail to overcome the ‘black-box’ effect. CBSE has a number of limitations that create a question mark on concepts of quality with itself [Bucanac1999]. This can be in matter of reusability, security, trustworthiness, reliability, component retrieval, certification, testing, etc.
To address these problems, we have developed a framework for providing detailed, concise specifications of component based software. Based upon formal specifications, the framework provides both structural and behavioral reasoning capabilities.

This thesis describes the Quality Assurance Framework for building Component Based Software Systems which are flexible and are for high-performance.

1.1 Motivation and objective

Quality Assurance [Gao2003] is crucial issue in the component based software development. By building quality assurance models at design time risks can be minimized. This quality assurance models at design time helps to identify any potential design problems and can be used as a feedback to software component integrators.

Component based software development is becoming more generalized representing a considerable market for the software industry. The perspective of reduced development cost and shorter life cycles acts as a motivation for this expansion. However, several technical issues remain unsolved before software components industries, even at the stage of maturity. Problems such as the component selection by their integrators and the uncertain quality of third-party developed components bring new challenges to the software research community. On the other hand, the software component quality area is still immature and further research is needed in order to obtain well defined model or standard for Quality Assurance. In this way, we
aim to propose a component quality model, describing consistent and well defined characteristics, quality attributes and related metrics for the components evaluation.

Most of the research dedicated to software component is focused on their functional aspects (i.e, Component Specification, Development, test, etc). In our ongoing research, we are concerned with the evaluation of component based software systems quality at the early design phase of life cycle. It lowers the development cost as well as the run time risks. To accomplish this, building predictive quality assurance framework is of utmost importance.

This evaluation should be performed using a component quality model, which specifies the quality characteristics that the software component has to attend. However, there are several difficulties in the development of such a model, such as: (i) which quality characteristics should be considered (ii) how we can evaluate them and (iii) who should be responsible for such evaluations.

In recent years, Component Based Software Engineering (CBSE) has drawn more and more interest in both academic and industrial communities. Most of the work is focused on functional aspects with relatively less reported on non-functional aspects such as performance, reliability, security etc. One difficulty is that a software component could be deployed in different environments. The need to consider its potential execution environments and incorporate them into the systems modeling makes the problem more difficult.

Yacoub et. al. proposed a way to characterize a software component, in which they mentioned non-functional attributes [Yacoub1999]. Sitaraman et. al. argued that
Chapter 1

performance specification of software components and assemblies is a basic problem that must be solved to assemble the systems from components [Sitaraman2001]. It would be very helpful if quality sub-models could be built and the quality attributes of each software component could be described in the sub-models. Then a system model can be built by assembling these sub-models. Therefore, the system quality prediction can be done by solving these system models. It is desirable if there are tools or methodologies which can automate these processes. This thesis work has been motivated by this thought.

Very often, when a system is planned, in order to evaluate the quality, the model has to be built from scratch even though it may have some pre-existing components. For large and complex systems, this is error-prone and tedious work. If the sub-models can be reused, it would be much easier to build system-level models. By using these sub-models, system models can be built more quickly for many different configurations which are tied to software configurations. Through these system models the capacity of system can be predicted. Meanwhile, using these models, the quality sensitive attributes are easily measured.

1.2 The Heart of the Problem

The quality assurance in the traditional software system is complex and quality evaluation at design time in component based software system is much more complex.

One of the major problems in building component based software systems is predicting the reliability of the eventual solution before the application has been built.
Chapter 1

[Cheung2008]. Middleware offered by various technologies does not provide any guarantee for the fulfillment of quality assurance requirements. These technologies only support the assembling of components and provide means to connect components together, but they don’t provide support for predicting the quality of the assembly or an application. When developing a software quality assurance model, one must first identify factors that strongly influence software quality and the no. of residual errors. Unfortunately, the degree of influence is imprecise in nature.

1.3 Thesis Goals

The context of my work is Component-based Software Engineering, rather than Component-based Systems. Some of the questions mentioned in the “Heart of the problem” have been addressed in the propose Quality Assurance Model. Quality Assurance Techniques can be extended to detect reliability predication in the early design phase of Component Based Software Engineering. A framework has been developed for providing detailed concise specification.

One of the original motivations for explicitly modeling the component based software was to early evaluate its non-functional properties. Currently three approaches for quality assurance exist. These are:-

(a) Informal and Manual Approaches

(b) By analytical Model

(c) Simulation Based
1.4 Approach

To solve the problem of quality assurance in component based software, we first find the attributes which directly or indirectly influence the quality of component based system and then looks for the existing quality assurance approaches. We prune some of the solutions based on time, complexity and computing resource requirements. Since our quality assurance solution is intended for use in third party component, we also focus on the soft computing techniques, which are the closest to the fuzzy characteristic of component.

1.5 Contributions

The main contributions of this study are:

- The creation of Quality Matrix to show the relationship among various quality attributes
- The development of CBQM: Component Based Quality Model to represent quality characteristic
- The development of Umbrella Life Cycle Model to define importance of testing.
- GA Based Component Retrieval from reusable repository has been proposed.
- Intra-Component Certification has been introduced.
- Measurement of the component reliability with usage profile.
Construction of testing framework.

1.6 Thesis Organization

The thesis has been organized as follows. Chapter 2 is a brief insight on the Component based software engineering with its existence from past to present. In this chapter we also introduce the problems in Component based software systems and overview of the active component based technologies. Chapter 3 is insight of related research survey. The following chapter 4 presents the component based quality model showing the relationship among various Quality Attributes. Additionally, we propose Umbrella life cycle Model. The Chapter 5 is based on the concept of component retrieval with creation of Genetic Algorithm. The chapter 6 discusses the part of security in CBSE and proposes an intra component certification. The following chapter 7 proposes testing framework in component based software development. Additionally, the relationship of component reliability with its usage profile has also been discussed. Finally the observation and conclusions this research have been complied in the last chapter with future direction.

The subsections below outline the contents of each chapter.

Chapter 2: Component Based Software Engineering: A state of Art

This chapter describes the background study of my research: software reuse, Component Based Software Engineering (CBSE). The chapter mainly explains the various software methodologies with their advantages and disadvantages. It introduces the concepts of software reuse. Barriers and problems in component based
software engineering have also been discussed in this chapter. Finally, it formulates the major research questions to be addressed in the project and to be answered in this thesis.

Chapter 3: A Systematic Research Survey on Component Based Quality Assurance

Both research community and industry have invested substantial effort in developing methods and tools for the early evaluation of quality attributes for component-based software systems. This chapter provides a systematic review of quality assurance in reliability, security, maintainability and certification. The review uses approximately 150 component based software quality assurance papers. For each direction, the achievements and drawbacks of the existing approaches has been summarized. The role of our research has been discussed with respect to the advantages and disadvantages of the contemporary methods.

Chapter 4: Quality Assurance Framework for Component Based Software Development

This chapter describes a method for evaluating the quality attributes of component based software systems, based on the properties of its constituents. The chapter identifies relevant factors influencing the quality attributes of the software and various models for specification of quality attributes e.g., ISO 9126, McCall are exemplified and compared.
Then, it attempts to rationalize component-based development process. It further recognizes that the construction of a software system is a complex multifaceted activity and involves user, developer and integrator. These activities, among others, are encompassed by a component based software quality process model, named as CBQM (Component Based Quality Model) model, put forward in this study. The CBQM model provides guidance for the major party and quality attributes that has to be followed under its umbrella. The quality matrix as well as Umbrella Life cycle has also been proposed.

Chapter 5: Component Retrieval Using Genetic Algorithm

This chapter mainly highlights GA approach for retrieval of component from reusable repository. It also introduces XML Modeling for retrieval process.

Chapter 6: Security Assurance in Component Based Software Systems

This chapter deals with one of the key problem of component based software that is concept of security. Security policy has been viewed to assure security of component at different levels. Apart from this we also propose an intra component certification mechanism.

Chapter 7: Quality Verification and Prediction

This chapter proposes a framework for quality assurance in component based software systems which covers component reliability and testing or maintainability. In this chapter first of all we have classified component based testing in two levels i.e, (i) developer oriented and (ii) integrators oriented, and then provide a XML based
Framework. The main contribution of this study is the presentation of relation between component based software systems reliability and component usage ratio.

**Chapter 8: Conclusions**

This chapter summarizes both theoretical and practical results of our research. The main achievements of our research are the methods for evaluation of static and dynamic quality attributes. Different proposed models help the architects in gaining architectural insight, in producing reliable quantitative estimates, and in trading estimation effort against estimation accuracy. The chapter highlights the advantages of models and compares them with the approaches there existed in the architecting field so far.

This chapter also provides directions for future research.