ABSTRACT

Software development organizations believe that Component-Based Software Engineering (CBSE) approach will improve their productivity and quality by selecting pre-existing software components. CBSE is an approach which is used to enhance the reusability because reusability is a way to improve efficiency and productivity of software systems. Component-Based Software Development (CBSD) with reusable component not only reduces the time to market but also brings down the cost of development heavily.

For very large and hardly complex application, some components need to be developed separately specifically tailored to the need of the application and some components are selected from the third party repositories. So CBSE is latest technology which is mainly objective to increase the reusability functionality with the development of CBS from the COTS software components.

CBSE is a technology used to develop a large software system with the assembly of reusable software components according to the client-specific requirement. When researchers and practitioners talk about reusable software components then components selection process and algorithm are play a main role in CBSD. Component selection for CBSD is very active and challenging field for researchers and practitioners. So this thesis presents a new proposed component selection process and new proposed algorithm to select the optimal component sets for client-specific requirement which make a complex software system with the composition of optimal reusable software component.

This thesis work is categorized into nine chapters. Chapter 1 explains software engineering and software reuse in general and evolution of CBSE, software component, interface and the purpose of CBSE. Chapter 2 describes about the study and review of various approaches and process models for CBSE. Chapter 3 explains the CBSE in general term, the process of CBSE and component technologies. Chapter 4 presents the challenge like performance, time, components size, fault tolerance, reliability, components functionality, components compatibility and available component subset which are faced by developer during component selection. Chapter 5 presents the software metrics measurement for quality software. Chapter 6 explains the slicing
of components. Chapter 7 explains a new optimal process for optimal component selection from third party repositories and component selection algorithm. Chapter 8 explains the implementation of an algorithm. Chapter 9 is summarized with conclusion and looks ahead to the future work.