Abstract

Component-Based Software Engineering offers the feature of reusability which is a distinctive paradigm of Software Engineering. It promotes the development of software systems by picking suitable and appropriate pre-existing, pre-built, pre-tested and reusable software work-products called ‘components’. Further, these components are assembled and integrated with well-defined architectural design. Rather than to focus only on coding, Component-Based Software development enables the application developers to concentrate on better design and optimized solutions of the problem, since coding objects are available in the repository in the form of components. Component-Based Software Engineering accentuates ‘development with reuse’ as well as ‘development for reuse’. Reusable components interact with each other to provide and to access functionalities and services to each other. These interactions and integrations of heterogeneous components raise issues including, the suitable and efficient reusability of components, complexities produced during the interaction among components, testing of Component-Based Software, and the overall reliability of the application under-development.

In this thesis, these issues are addressed through a model driven approach. This approach is very simple, suitable and comparatively efficient to resolve the above issues effectively. To explore and analyze the static behavior of components, ‘Reusability-metric’ for all categories of components including partially modifiable, fully modifiable as well as for off-the-shelf components are identified and described. Using Reusability-metric, a ‘Reusability-matrix’ containing the reusability ratios of all the different classes of components are defined. Reusability-metric can be stored as an attribute along with the other performance coefficients for future use. This thesis introduces a selection and verification criteria for components by using the reusability features of Component-Based Software.

To capture the complexities generated due to interactions and integrations among multiple components, this work proposes two methods namely, Cyclomatic complexity and In-Out Interaction complexity computation for Component-Based Software systems. Cyclomatic complexity method is helpful to compute the complexity of interactions as well as to count the number of test cases in the context of White-Box testing.

Integration-Effect Graph technique is developed to compute the number of test cases for Black-Box testing of Component-Based Software.
This work also proposes a method of reliability estimation for Component-Based Software by introducing metrics named ‘Reusability-Ratio’ and ‘Interaction-Ratio’ as a factor of reliability estimation. The role of Interaction-ratio of individual components to estimate the execution time of the Component-Based Software is presented in this thesis. These metrics are not only easy to compute, but they are informative to analyze the performance of the components as well as the Component-Based Software.

A model driven approach has been used to analyse and show the results of proposed measures and metrics. Suitable scenarios and case studies are constructed to model the Component-Based environment. Comparative results are shown in respective chapters and on the basis of these achieved results, efficiency of proposed metrics are proved.