Abstract

Component-based software development (CBSD) is an important area in the software engineering field. It has become one of the preferred streams for developing large and complex systems by integrating prefabricated software components which have not only made easier the process of software development but has also changed the ways for software professionals to develop software applications. A number of attempts have been made by software development teams, developers as well as researchers to improve component-based software development through improved measurement tools and techniques i.e. through an effective metrics. To manage and to study the different forms of dependencies that might occur in component based systems there has been little effort. The Component-based systems (CBS) are built up by integrating a number of these components in the system thus known as component assembly. In a CBS interaction can be in the form when one component interacts with another component, like one component provides an interface and other components use it, and also when an event is submitted by a component and it is received by other component. These interactions results in dependencies. Higher dependency has adverse effects thus leading to a complex system, resulting in poor understanding and higher maintenance cost. Dependency is represented in many forms one is by an adjacency matrix used in graph theory. However, this representation can check only for the presence of dependency between
components and does not consider the type of interactions between these components. Interaction type can have a significant contribution to the complexity of the system. In this paper we present a technique in which component-based system is modeled using UML tool like ArgoUML 0.32, this tool is an open source and has the capability to export the component model in XMI (XML Meta-data Interchange) file format, this XMI file stores all the information related to a model, by applying parsing technique through a tool developed in Java, this technique can store the dependency along with other information like, provided interface (UML:Dependency.supplier) and required interface (UML:Dependency.client). This paper also presents the results of an experiment of the proposed approach and measures the interaction densities and dependency level of an individual component and for the system.

References

- http://tutorials.jenkov.com/ood/understanding-dependencies.html, last access: 12-02-2012
- Larsson, M., 2007. Applying Configuration Management Techniques to Component -

Index Terms

Computer Science
Web Technology

Keywords
Component-based System
Unified Modeling Language