Abstract

Recent trend in systems architecture and design is component-based. A system is designed as a set of mutually supporting components that provide the intended services. The requirements models such as business type models and use case models are often used for deriving the targeted component-based architecture. The component interfaces are derived via sequence diagrams, collaboration diagrams and context diagrams. As the business model evolves, it becomes vital that the system also needs to match the business evolution whether it involves changing business rule set or growth in volume of business transactions. Timely reengineering of systems is profitable to any organization. The systems reengineering can be conducted in a pragmatic manner via component by component or a selected set of components; it becomes manageable and cost-effective to maintain the system and to train only a smaller sample of affected users. This paper offers a methodology for system reengineering via component replacement and model-view-control framework for component refinement and evolution in order to achieve a reengineered system that reflects upon the latest requirements in business domain.


Heineman G. T. and Councill W.T (2001), Definition of a Software Component and its elements, Ch.1, Component Based Software Engineering, Addison-Wesley.


Index Terms

Computer Science
Software Engineering

Keywords
Information System
Reengineering
Business Type model
Use Case
Component-based model
Interface model

Context model

MVC framework.