Abstract

Regarding modularised software development, Aspect-oriented programming (AOP) identifies and represents individually crosscutting concerns during the software development cycle's programming stage. This article proposes and applies OOAspectZ to formal Aspect-oriented requirement specifications for prior stages of the software development cycle. It particularly concerns requirement specification and the structural design of data and behaviour, along with describing and applying Aspect-oriented UML class diagrams to designing classes, aspects and associations among classes and aspects during Aspect-oriented software development (AOSD). OOAspectZ is a language integrating both Object-Z and AspectZ formal languages whereas Aspect-oriented UML class diagrams represent AOP code, object class and crosscutting concern class structure by means of stereotypes. This article shows and applies the main OOAspectZ and AO UML class diagram characteristics to Aspect-oriented software modelling (AOSM) using a classic example of AOP. Ideas for future work concerning an actual AOP version are also indicated.

Keywords

Aspects, OOAspectZ, UML class diagram, crosscutting concern.