

ABSTRACT

Software maintenance usually consumes about 60% of the budget of software development in enterprises. The most common types of maintenance are the evolutive and adaptive. Information about modified artifacts during a maintenance must be registered in order to allow evaluating the impact of future maintenances. In the case of existent systems that are already implemented and updated without up-to-date documentation, usually named legacy systems, reverse engineering can help in the recovery of this information. It allows revealing points in the code where a use case scenario is implemented. Through the extraction of behavioral models representing two or more scenarios of the same use case in different software versions or similar use cases, it is possible to discover where a specific functionality is implemented or if it has suffered adaptive or evolutive maintenance between software versions, detecting the impacts of a past maintenance. Although it is not always possible to trace the maintenance's rationale, this analysis can help in revealing points where use cases use to be modified. Thus, this work proposes an approach for the comparison of behavioral models for existing Java systems that represent use case scenarios, aiming at supporting system maintenance. A tool for the comparison of models representing method calls for use cases is developed. It uses as input XML files generated by tools of binary code instrumentation, providing through algorithms of difference calculation, i.e. *Diff*, the modifications between use case scenarios.

Keywords: software maintenance, XML *Diff*, use case scenario, reverse engineering, object orientation.