

- The architecture recovery facility introduced allows to inspect software systems on architecture level described in ROOM notation.
- Explicit usage of domain knowledge in the development of concepts, methods, and tools guarantees suitability and user acceptance.
- The analysis of multi-language software systems is possible due to the realization of a seamlessly integrated yet open, modular, extensible, and flexible reengineering system.

Not all of these concepts are new. There are a number of mature techniques that have already proved to work quite well, especially for the static structural analysis of software. So, there was no need to re-invent them but rather to adapt and incorporate them. However, the combination of these techniques within an open, modular, and flexible reengineering system leads to synergetic effects. These result in considerable improvements with respect to power, applicability, and integration within the reengineering system implementation showing the desired functionality and behavior described by the conceptual framework.

Maximilian Störzer: Impact-Analyse für AspectJ - Eine kritische Analyse mit werkzeuggestütztem Ansatz

Promotion: Universität Passau

Erstgutachter: Prof. Dr. Gregor Snelting, Universität Passau

Zweitgutachter: Prof. Dr. Christian Lengauer, Universität Passau

Datum der Prüfung: 11.Mai 2007

Veröffentlichung: <http://www.opus-bayern.de/uni-passau/volltexte/2007/89/>

Kurzfassung:

Aspekt-Orientierte Programmierung (AOP) wird seit Jahren als "die Lösung von Software-Modularisierungsproblemen propagiert, die in der Literatur als die Tyrannei der dominanten Dekomposition" bekannt sind. Unterzieht man AO Sprachen jedoch einer kritischen Untersuchung, so zeigt sich, dass diese pauschale Aussage bezweifelt werden muss.

Diese Arbeit trägt zwei Punkte in diesem Kontext bei: Zum Einen werden AO Sprachkonstrukte kritisch analysiert, um Programmierer auf problematische Anwendung dieser Konstrukte aufmerksam zu machen. In diesem Kontext wird demonstriert, dass AOP - ganz im Gegensatz zu seinen Zielen - leicht in schwerer verständlichem, schlecht wartbarem und fehleranfälliger Quellcode resultieren kann.

Zum Anderen werden zur Unterstützung von Programmierern Software Werkzeuge basierend auf statischer und dynamischer Programm-analyse vorgeschlagen. Im Detail wird untersucht, wie Techniken der Change Impact Analyse" verwendet werden können, um die Effekte von Aspekten und problematische Anwendung von AO Sprachkonstrukten automatisch zu ermitteln sowie um die Auswirkungen von Systemevolution handhabbar zu machen. Die Arbeit stellt ebenfalls eine Analysetechnik vor, um potentielle Interferenz von Aspekten automatisch zu ermitteln.

Die Dissertation schließt mit einer Übersicht über verfügbare Open Source AspectJ Systeme, die zur Evaluierung der vorgestellten Techniken verwendet wurden, und einer auf software-technischen Grundlagen und den in der Arbeit gewonnenen Erkenntnissen beruhenden Bewertung von AOP.