

## **André Marburger: Reverse Engineering of Complex Legacy Telecommunication Systems**

**Promotion:** RWTH Aachen

**Erstgutachter:** Prof. Dr.-Ing. Manfred Nagl, RWTH Aachen

**Zweitgutachter:** Prof. Dr. Jürgen Ebert, Universität Koblenz

**Datum der Prüfung:** 17. Dezember 2004

**Veröffentlichung:** 420 S., Shaker-Verlag, 2005

### **Kurzfassung:**

Reverse and reengineering of large and complex software systems is a difficult task. As a result, many methods and tools for reverse and reengineering have been developed so far. However, the work in this field has concentrated on sequential, and untimed systems, mainly for business applications. The majority of the approaches deals with decomposing monolithic systems, decoupling user interface/presentation from application logic and data handling/database management, or with identifying reusable components. In particular, numerous approaches have addressed the migration of legacy business applications to an object-based or object-oriented architecture. To a large extent, the corresponding methods are data-centered since they focus on structuring the data maintained by an application. Another stream of research has dealt with migration to code of programming languages such as C++ and Java which already provide language support for object-oriented programming.

Reverse and Reengineering of process-oriented applications has been addressed less extensively. For example, a telecommunication system is composed of a set of distributed communicating processes which are instantiated dynamically for handling calls requested by the users of the system. Such a system is designed in terms of services provided by entities which communicate according to protocols. Understanding a telecommunication system requires the recovery of these concepts from the current source code and other sources of information. Furthermore, analyzing and visualizing the dynamic behavior is a key to system understanding.

This dissertation describes the concepts and the implementation of integrated tools for rever-

se and reengineering of telecommunication systems which were developed in close cooperation with ERICSSON in the ECARES project (Ericsson Communication ARchitecture for Embedded Systems). The concepts are based on studies and evaluation of a real telecommunication system - Ericsson's Mobile-service Switching Center (MSC) for GSM networks called AXE10. These studies led to specific requirements. These requirements and an abstract system structure are described within a conceptual framework, which specifies the problem domain and identifies and interrelates the necessary concepts, thus building the terminological and conceptual foundation of this dissertation.

To guarantee the suitability and applicability of the methods and tools developed in this thesis, tool support was developed step by step in response to the requirements and questions stated by telecommunication experts. This approach implied an iterative and incremental analysis and development process. Each pass of this process concentrates on a subset of the overall functionality and delivers appropriate analysis functionality and result documents, thus providing another portion of the final reverse and reengineering environment.

The essential contributions (concepts, methods, and tools) to reverse engineering of telecommunication systems are as follows:

- Structural, control flow, and data flow information is seamlessly integrated into a modular graph scheme. In addition, the scheme organizes static (e.g., from code analysis) as well as dynamic information (e.g., from trace analysis). This enables complex analysis operations incorporating all kinds of information.
- The structural analysis part combines aggregation, condensation, and lifting of information from different sources with deduction of additional information and multi-level abstraction and visualization.
- Static and dynamic behavioral analyses are utilized to realize link chain analysis, state machine extraction, and trace analysis. For trace analysis, a comprehensive simulator is introduced.

- 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.