Panel: Identifications of Concepts, Features, and Concerns in Source Code

Andrian Marcus and Václav Rajlich
Department of Computer Science
Wayne State University
Detroit, MI 48202
313 577 5408
amarcus@wayne.edu, rajlich@wayne.edu

Panelists

- *Arie van Deursen - CWI and Delft University of Technology, The Netherlands.*
- *Rainer Koschke - Universität Bremen, Germany*
- *Harry Sneed - ANECON GmbH, Austria*
- *Paolo Tonella - ITC-irst, Italy*

1. Introduction

Software development process transforms requirements into source code. These requirements are formulated as concepts from the problem domain or the solution domain; among them, features are the concepts that describe user selectable behavior.

During software evolution, existing concepts and features of the system are changed, deleted, or new ones are added. In order to solve any of these change tasks, the existing concepts need to be precisely identified in the source code. This identification process was initially defined as the *concept assignment (location) problem* [2]. It is of no surprise that searching and browsing the software artifacts with the goal to identify parts of the source code that implement a concept from the software domain is one of the most common activities during software evolution.

The existing methods of concept and feature location fall into two broad categories, based on the information from the software used during the location process: *static* [3] and *dynamic* [4].

Concerns of the software system are closely related to the concepts. A major issue during software design is the separation of these concerns [1] from one another through the modularization mechanisms available in the chosen program language. The goal is to implement each concern as one module in the software.

The limitations of existing programming paradigms and languages, often combined with the lack of design expertise, result in a sad reality, where concerns are implemented in several modules, often cross-cutting the primary decomposition of the system.

2. Discussion topics

The panelists, coming from both academia and industry, address several fundamental aspects of this area of research and practice:

- There is a need to define taxonomy of existing techniques for concern location, based on the type of software analysis, user interactions, and necessary knowledge needed. This must be complemented with empirical validation efforts (P. Tonella).
- The field of concept location is in the *era of consolidation*, when existing techniques are compared and benchmarks are developed (R. Koschke).
- Should concerns be documented? Perhaps the "why", but much less the "how". Instead, we should rely on and invest in concern exploration and reconstruction technology (A. van Deursen).
- Since no existing method is perfect, there is a need to combine them (A. Marcus).
- Do research efforts support the real needs of practitioners? (H. Sneed)

3. Acknowledgement

4. References

[1] Aksits, M., "Separation and composition of concerns in the object-oriented model", *ACM Computing Surveys*, vol. 28, no. 4es, December 1996.

[2] Biggerstaff, T. J., Mitbander, B. G., and Webster, D., "The concept assignment problem in program understanding", in Proceedings of International Conference on Software Engineering (ICSE'93), Baltimore, MD, 1993, pp. 482 - 498.

[3] Marcus, A., Rajlich, V., Buchta, J., Petrenko, M., and Sergeyev, A., "Static Techniques for Concept Location in Object Oriented Code", in Proc. of Int. Workshop on Program Comprehension, St. Louis, MO, May 15-16 2005, pp. 33-42.

[4] Wilde, N., Buckellew, M., Page, H., Rajlich, V., and Pounds, L., "A comparison of methods for locating features in legacy software", *The Journal of Systems and Software*, vol. 65, 2003, pp. 105-114.