Incremental Software Construction

Ralph-Johan Back

A central problem in constructing large software systems is how to manage complexity. The larger the system becomes, the more difficult it also is to extend the system and adapt it to changing requirements. For this purpose, we have been experimenting with an incremental software architecture and software construction process that we refer to as *stepwise feature introduction* [2] , where a software system is built in very thin layers. Each layer adds just one new feature (unit of functionality) to the system, without destroying the features that already are present. The extension mechanism is class inheritance. A layer will usually consist of more than one class, where most classes are extensions of corresponding classes in earlier layers. The guiding principle has been to keep the class invariant as simple as possible, and to use *superposition refinement* as correctness criteria for the extension. Stepwise feature introduction allows an evolutionary approach to software construction, where the software can be adapted to changing requirements during the construction process.

We have used this approach in a number of practical experiments in software construction, and the experience has been very encouraging. Stepwise feature introduction fits well with the short iteration cycles that are championed in extreme programming, and we apply a combination of these two techniques in our software projects. We will look at both theoretical and practical aspects of stepwise feature introduction, focusing in particular on

- software architecture and design aspects [2, 1],
- correctness concerns, expressed in the refinement calculus [6, 4, 5, 2], and
- software processes and practical experiences [1, 3]

References

[1] Heikki Anttila, Ralph-Johan Back, Pekka Ketola, Katja Konkka, Jyrki Leskelä, and Erkki Rysä. Combining stepwise feature introduction with user-centric design. Technical Report 495, TUCS - Turku Centre for Computer Science, www.tucs.fi, Dec 2002. An extended abstract was published at the International Conference on Human Computer Interaction, Creete, 2003.

[2] Ralph-Johan Back. Software construction by stepwise feature introduction. In Bowen J.P. Henson M.C. Robinson K. Bert, D., editor, *ZB 2002: Formal Specification and Development in Z and B, proceedings of the 2nd International Conference of B and Z Users*, LNCS, pages 162–183, Grenoble, France, January 2002. Springer Verlag. Also appeared as TUCS Technical Report 496.

[3] Ralph-Johan Back, Luka Milovanov, Ivan Porres, and Viorel Preoteasa. XP as a framework for practical software engineering experiments. In *Proceedings of the Agile Processes in Software Engineering (XP2002)*, May 2002. Also appeared as TUCS TR 451.

[4] Ralph-Johan Back and Joakim von Wright. *Refinement Calculus: A Systematic Introduction*. Springer-Verlag, 1998.

[5] Ralph-Johan Back and Joakim von Wright. Enforcing behavior with contracts. In A. McIver and C. Morgan, editors, *Programming Methodology*. Springer-Verlag, 2002.

[6] Ralph-Johan Back and Joakim von Wright. Compositional action system refinement. *Formal Aspects of Computing*, 15(2):103–117, 2003.