

Software Reuse based on Business Processes and Requirements

Dr. Hermann Kaindl, Professor
Vienna University of Technology, ICT

Short Biography

Hermann Kaindl joined the Institute of Computer Technology at the Vienna Univ. of Technology in early 2003 as a full professor. Prior to moving to academia, he was a senior consultant with the division of program and systems engineering at Siemens AG Austria. There he has gained more than 24 years of industrial experience in software development and human-computer interaction. He has published five books and more than 150 papers in refereed journals, books and conference proceedings. He is a *Senior Member* of the IEEE, a *Distinguished Scientist* member of the ACM, a member of the AAAI, and is on the executive board of the Austrian Society for Artificial Intelligence.

He has previously held tutorials at CAiSE'00, RE'01, RE'02, HICSS-36, INCOSE'03, RE'03, IUI- CADUI'04, INCOSE'04, RE'04, HICSS-38, IRMA'05, INCOSE'05, AAAI'06, HCI'06, OOPSLA'06, HICSS-40, ICONS'07, IRMA-07, INCOSE'07, AAAI'07, IFIP Interact'07, OOPSLA'07, HICSS-41, ICCGI'08, RE'08, ICSEA'08, ICIW '09 , IFIP Interact'09, SMC'09, HICSS-43 ACHI'10, EICS'10, ICSEA'10, TdSE'10, HICSS-44, SAC'11, INCOSE'11, AAAI'11, RE'11, HICSS-45, SAC'12, ACM CHI'12, PROFES'12, BCS HCI'12, APSEC'12, HICSS-46, SAC'13, NexComm'13 and PROFES'13.

Tutorial purpose

The primary objective of this tutorial is to improve software development in practice regarding the difficult and important issue of reuse. The participants will get a better understanding of this issue and learn about different approaches to address software reuse based on business processes and requirements.

Tutorial description

Software reuse is often just addressed at the level of code or low-level design. In contrast, this tutorial will explain software reuse based on business processes and requirements. It will present and compare three approaches co-developed by the presenter over more than a decade.

The first of these approaches deals with requirements reuse in the context of *product lines*. It makes the relations among product line requirements explicit, so that single system requirements in this product line can be derived consistently. A key issue is commonality and variability across different products. This tutorial will show how requirements for a product line can be modeled, selected and reused to engineer the requirements for innovative new products.

The second approach for software reuse involves case-based reasoning. Instead of explicit relations between requirements (or other artifacts), similarity metrics are employed for finding the most similar software case in a repository to a given set of requirements. This even works when a single envisioned usage scenario is specified yet, and it allows reusing also requirements from retrieved

cases. The major point, however, is to facilitate reusing software design (including architecture) and code from similar software cases.

The third approach (still under development) strives for (partly) automating software development for certain business applications through reusing business knowledge and software, where both are tightly connected. It involves automated reuse of business processes, and software executing them, based on ontological knowledge. A key point is closing the representational gap between procedurally represented business processes and declaratively represented concepts and their relations, taxonomies, paronomies, etc. So, this is an ontology-based approach for (partly) automated software development guided by business models.

Selected publications of the proposer related to this tutorial

H. Kaindl, M. Mannion, Verification of Selection from Product Line Requirements, in *Proceedings of the 15th Annual International Symposium of the International Council on Systems Engineering (INCOSE 2005)*, Rochester, NY, USA, 2005.

H. Kaindl, M. Smialek, W. Nowakowski, Case-based Reuse with Partial Requirements Specifications, in *Proceedings of the 18th IEEE International Requirements Engineering Conference (RE 2010)*, 2010, pp. 399–400.

H. Kaindl, D. Sventinovic, On confusion between requirements and their representations. *Requirements Engineering*, vol. 15, 2010, pp. 307–311.

H. Kaindl, P. Wagner, A Unification of the Essence of Goal-oriented Requirements Engineering, in *Proceedings of the Fourth International Conference on Software Engineering Advances (ICSEA 2009)*, Porto, Portugal, 2009, IEEE.

M. Mannion, H. Kaindl, Using Parameters and Discriminants for Product Line Requirements. *Systems Engineering*, vol. 11, no. 1, 2008, pp. 61–80.

Mannion, M., Keepence, B., Kaindl, H., and Wheadon, J., Reusing Single System Requirements from Application Family Requirements, in *Proc. Twenty-first International Conference on Software Engineering (ICSE-99)*, Los Angeles, CA, May 1999, 453–462, ACM.