Software Reuse and Reusability Involving Requirements, Product Lines, and Semantic Service Specifications

Dr. Hermann Kaindl, Professor
Vienna University of Technology, ICT
Gusshausstr. 27-29
A-1040 Vienna, Austria
Phone: +43 1 58801-38416
Fax: +43 1 5 99666-938416
Email: kaindl@ict.tuwien.ac.at
Web: http://www.ict.tuwien.ac.at/kaindl
**Title**
Software Reuse and Reusability Involving Requirements, Product Lines, and Semantic Service Specifications

**Presenter**
Dr. Hermann Kaindl, Professor
Vienna University of Technology, ICT
Gusshausstr. 27-29
A-1040 Vienna, Austria
Phone: +43 1 58801-38416
Fax: +43 1 5 99666-938416
Email: kaindl@ict.tuwien.ac.at
Web: http://www.ict.tuwien.ac.at/kaindl

Hermann Kaindl joined the Institute of Computer Technology at the Vienna University of Technology in early 2003 as a full professor, where he also serves in the Senate. 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 170 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, CADUI-IUI’04, INCOSE’04, RE’04, HICSS-38, IRMA’05, INCOSE’05, AAAI’06, HCI’06, OOPSLA’06, HICSS-40, ICONS’07, INCOSE’07, AAAI’07, IFIP Interact’07, OOPSLA’07, HICSS-41, ICCCI’08, RE’08, ICSEA’08, ICIW’09, IFIP Interact’09, SMC’09, HICSS-43, ACHI’10, ACM EICS’10, ICSEA’10, TdSE’10, HICSS-44, ACM SAC’11, INCOSE’11, AAAI’11, RE’11, ICSEA’11, HICSS-45, ACM SAC’12, ACM CHI’12, PROFES’12, BCS HCI’12, IEEE APSEC’12, HICSS-46, ACM SAC’13, NexComm’13, PROFES’13, ICSoft’13, IEEE Africon’13, IEEE APSEC’13, HICSS-47, ACM SAC’14 and WEB’14.

Some of these tutorials were related to the one proposed here, most strongly one given at IEEE APSEC’13.

**Duration**
Half day

**Abstract**
Software reuse and reusability is often just addressed at the level of code or low-level design. In contrast, this tutorial explains them starting from requirements. It integrates and presents three approaches co-developed by the presenter over more than a decade, which also involve product line technology, case-based reasoning and semantic service specification.

One approach deals with requirements reuse and reusability 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 shows how requirements for a product line can be modeled, selected and reused to engineer the requirements for innovative new products. Another 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. In fact, these two approaches can be usefully combined. Yet another approach involves *semantic service specification*, which facilitates automated generation of service composition. In the context of business software reuse and reusability, this formal specification facilitates automated verification, and validation including business rules as well.

These approaches have different key properties and trade-offs between costs of making software artefacts *reusable* and benefits for *reusing* them. These will be particularly explained in this tutorial.

**Motivation, target audience, and interest for the SAC community**

The primary motivation for giving this tutorial is to improve software development in practice regarding the difficult and important issue of reuse and reusability.

The target audience is software development practitioners, such as requirements engineers, software designers and project leaders. Also educators will benefit from this tutorial. The assumed attendee background is primarily some interest in requirements or software reuse and reusability based on services.

The application area of this tutorial is clearly software development. It closely relates to the SAC 2015 Tracks RE and SE as well as partly to SOAP and even to SWA.

**Outline**

This outline is adopted from previously held tutorials, primarily from one given at APSEC’13:

- Introduction and background
  - Requirements
  - Use cases / scenarios
  - Business processes
  - Software reuse and reusability

- Requirements reuse and reusability in product lines
  - Product lines
  - Modeling requirements for a product line
  - Selecting single-system requirements for reuse
  - Verifying consistency

- Software reuse involving case-based reasoning
  - Software cases
  - Requirements-based reuse utilizing similarity
  - Partial requirements specification / scenario
  - Reuse and reusability of requirements

- Reusing business knowledge and software
  - Business use case — business process
  - Automated adaptation based on business rules
  - Semantic service specification
  - Automated generation of service composition
o Business process verification and validation

- Contrasting these approaches
  - Reusable artifacts
  - Reuse approaches
  - Trade-offs

- Summary and conclusion

Specific goals and objectives
This tutorial has the primary objective to provide the participants with a better understanding of reuse and reusability. They will learn about different approaches to address them involving requirements, product lines, and semantic service specifications. They will also learn about their respective trade-offs between costs of making software artefacts reusable and benefits for reusing them.

Expected background of the audience
The assumed attendee background is basic familiarity with issues of reuse and reusability in software systems development. There are no pre-requisites such as knowledge on specific approaches, however.

Audio Visual equipment needed for the presentation
For the instructor, a computer screen projector (to be connected with my laptop computer) will be needed, capable of at least 1024 x 768. In addition, a flip-chart with pens of various colors is required.

For the course attendees, paper for the flip-chart and a sufficient number of pens are needed for the group exercises.

Selected publications of the proposer related to this tutorial


