Tutorial Proposal for
The 21st Annual ACM Symposium on Applied Computing
Dijon, France
April 23 -27, 2006

Designing Software Product Lines with UML 2.0:
From Use Cases to Pattern-Based Software Architectures

Instructor: Dr. Hassan Gomaa

Department of Information and Software Engineering
George Mason University
Fairfax, Virginia 22030,
USA

Phone: (703) 993-1652
Fax: (703) 993-1638
E-mail: hgomaa@gmu.edu

Proposed Duration: Full Day (more concise half-day version also possible)

Level of Tutorial: Intermediate – some basic knowledge of software product lines and object-oriented concepts is expected.

Aims and Objectives

This tutorial is intended for information technologists who wish to understand how to perform object-oriented requirements, analysis, and design modeling for software product lines using the industry standard for object-oriented development, the Unified Modeling Language (UML) 2.0.

Intended Audience

Intended for a wide audience of academic and industrial professionals including researchers, academic faculty, graduate students, software developers, systems analysts, software designers, and project managers.

Purpose and Scope

This one-day tutorial addresses how to develop object-oriented requirements, analysis, and design models of software product lines using the Unified Modeling Language (UML) 2.0 notation. During requirements modeling, the tutorial covers how to develop kernel, optional, and alternative use cases for defining the software functional requirements of the system. The tutorial also describes the feature model for capturing product line requirements and how it relates to the use case model. During analysis, the tutorial covers how to develop static models for defining kernel, optional, and variant classes and their relationships. It goes on to describe how to create dynamic models in which statecharts define the state dependent aspects of the product line and interaction models describe the dynamic interaction between the objects that participate in each kernel, optional, and alternative use case. The tutorial then covers how to develop the OO software architecture for the product line, in which the system is structured into component-based subsystems. The tutorial gives an overview of the structural architecture patterns and communication patterns that can be used in designing component based distributed product lines. The tutorial is illustrated by means of several examples. The tutorial is based on a book by the author, “Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures", Addison Wesley Object-Oriented Technology Series, 2005.

Outline of Topics

Object-Oriented Software Life Cycle for Software Product Lines; Object-Oriented Requirements Modeling; Object-Oriented Analysis Modeling, Object-Oriented Design Modeling, Incremental software construction and integration.

Requirements Modeling for Software Product Lines. The use case modeling approach for defining functional requirements. Kernel, optional, and alternative use cases and actors. Modeling variability with use case parameterization, variation points, and extension points.

Feature Modeling for Software Product Lines. Feature as a reusable requirement. Functional, non-functional, and parametric features; Feature dependencies. Feature sets – mutually exclusive, one and only one, one or more of a set. Modeling features with use cases; relationship between features and use cases. Feature conditions.

Analysis Modeling for Software Product Lines. Static modeling: objects, classes, and relationships. Object and class structuring; class categorization using stereotypes. Kernel, optional, and variant classes. Modeling commonality/variability with abstract classes and hotspots. Using inheritance to support variant classes; abstract classes to model common aspects of a product line class, specialization to address variations in product line classes. Feature/class dependencies.

Statecharts for Software Product Lines. Kernel, optional and variant statecharts. Mutually exclusive variant statecharts and co-existing variant statecharts.
Hierarchical statecharts: high-level statechart to capture generalization of multiple variants. Modeling variability in statecharts.

Dynamic modeling for Software Product Lines. Developing object interaction models for kernel, option, and alternative use cases. Developing collaboration model for different scenarios addressing use case variability.

Software Architectural Design for Product Lines. Developing the overall software architecture. Separation of concerns in subsystem design. Component-based structuring criteria. Structural and communication patterns for software product line architectures.

Application configuration. Configuring individual members of the product line from the OO software product line model. Using the product line feature model to configure the requirements, analysis, and design models for the application.

Software Product Line Case Studies: Microwave Oven, Distributed Factory Automation, Electronic Commerce.

Materials Provided

A copy of the instructor's viewgraphs.

Prior Courses and Tutorials

Hassan Gomaa has considerable experience in teaching courses and tutorials on software design methods. This tutorial has been presented at the Software Product Line Engineering Conference in Boston, August 2004, and at the Symposium on Applied Computing in Santa Fe, March 2005, and will be presented at the UML Conference in October, 2005. He has also taught short industrial product line design courses at Omron Inc. and at Japanese Software Engineering Center, both in June 2005. The tutorial material is based on his book “Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures", Addison Wesley Object-Oriented Technology Series, 2005.

Gomaa has made conference presentations and tutorial presentations at many international conferences. In December 2005, he was a keynote speaker at the Asia-Pacific Software Engineering Conference. In July 1994, he was an invited speaker at CASE Japan. He has presented tutorials and seminars on various aspects of object-oriented, client/server, distributed, real-time software design, and software product line design, at the IEEE International Conference on Software Engineering (1995, 1996, 2000, 2001), IEEE International Conference on the Engineering of Complex Computer Systems (1995, 1996), European Software Engineering Conference (1993), IEEE International Conference on Software Reuse (1996), IEEE ASSET (1998), OOPSLA (1995), UML Conference (2000, 2005), Workshop on Software Performance (2000), Washington Ada Symposium, Tri-Ada, and the Washington chapter of the ACM (several times).

He has taught several in-depth industrial courses on software design in North America, Europe, Japan and Korea, at the Software Productivity Consortium (for SPC’s member companies including Lockheed Martin, Northrop Grumman, Boeing, Rockwell, General Dynamics), Mitre Corporation, NASA, LCC, Pragmatics, Nokia, Philips Research Labs, BAE Systems, Raytheon, Boeing, Hitachi Institute of Technology, and Korean Information Processing Association. He has also co-presented a two-day video course for the National Technical University on "Software Design Methods and CASE Tools for Concurrent, Real-Time and Distributed Systems". He has also prepared several videotaped lectures on software design for the Software Engineering Institute, Carnegie Mellon University and an introductory videotaped course on real-time software design.

Instructor Biography

Hassan Gomaa is Chair and Full Professor in the Department of Information and Software Engineering at George Mason University, Fairfax, Virginia. He received a B.Sc.(Eng.) in Electrical Engineering from University College, London University, and the DIC and Ph.D. in Computer Science from Imperial College of Science and Technology, London University.

He has worked in both industry and academia, and has published over 130 technical papers and three textbooks. His book, "Software Design Methods for Concurrent and Real-Time Systems", was published by Addison Wesley in 1993 and was translated into Chinese in 2003. His second book, entitled “Designing Concurrent, Distributed, and Real-Time Applications with UML”, was published by Addison Wesley in 2000 and was translated into Chinese in 2004. His latest textbook entitled “Designing Software Product Lines with UML” was published by Addison Wesley in July 2004.

He has considerable experience in teaching courses and tutorials on software design. He has made conference and tutorial presentations at many international software engineering conferences. He was a keynote speaker at the Asia-Pacific Software Engineering Conference in December 2004. He has taught several in-depth industrial courses on software design in North America, Europe, Japan, and Korea. He also consults in both the technical and management aspects of software engineering.

His current research interests include object-oriented analysis and design for concurrent, real-time, and distributed systems, software product lines, component-based software architecture, software reuse, software performance engineering, intelligent software agents, software engineering environments, and software process models. His research has been funded by several organizations including the National Science Foundation, NASA and DARPA.