
Computer Science Colloquium

Software Factories: Some Examples of Prototype Development

Prof. Robert Lechner
Computer Science Department, UMass Lowell

Wednesday, 1 March 2006
Olsen 311

Refreshments at 2:30, Talk from 3:00-4:00

This visionary text from Microsoft authors (Wiley 2004) [<http://www.softwarefactories.com/>] has the subtitle “The Industrialization of Software Development.” Its scope is far too broad to relate in one talk. It extends the work of the Object Management Group on Model-Driven Development in new directions using Domain-Specific Languages. Rather than cover it superficially I will try to motivate you to read it, by presenting concrete examples of using a prototype development framework, called COOL for Collaborative Object-Oriented Laboratory. [http://www.cs.uml.edu/~lechner/COOL-FAQ/COOL_FAQv6.PPT]. I believe such examples fit well within the Software Factory’s broader vision.

The goal of COOL is a testbed for ways to raise programming to higher levels of abstraction, as advocated in the Software Factories text. This includes the use of COOL for designing and bootstrapping itself. COOL consists of three components: an automatic code generator GENCPP for persistent object-relational data structures, a state model interpreter LCP to control the sequential execution of class methods to execute complex transactions, and a Block Diagram Editor BDE that can be used to visualize and to create various diagram types.

The input to GENCPP is a text file describing relational tables; the input to LCP is a database of state model definitions. The equivalent graphic models are called Extended Entity-Relationship Diagrams and State Transition Diagrams respectively. Filters convert between these text and graphic representations.

GEN provides both BDE and LCP with a persistent database. The persistent database format is flat relational tables containing ASCII text, which can be stored in a source code revision control system like CVS.

I believe this narrowly-scoped goal of realizing a framework for automating the repetitive aspects of software prototype development fits well within the Software Factory’s broader vision.

Bio: Dr. Lechner retired from the Computer Science Dept at UML in 2001, where he developed and continues to teach the 91.522/523/524 OOAD and Software Engineering course sequence. He has BS and MSEE degrees from Carnegie [Tech] and a PhD in Applied Math from Harvard [Computation Lab]. Prior to joining UML in 1982 he spent 20 years in computer-based system analysis at GTE/Sylvania, Honeywell IS and CSDraper Lab, and six years in the EE Dept at Northeastern U.