Open-Source-Related

Synthesis of Component Properties
(Supported by National Science Foundation ITR grant CCR 0112654)

Development of software using off-the-shelf components seems to offer a chance for improving product quality and developer productivity. This research has developed an elementary theory of system synthesis from simple components, implemented tools that support the theory, and conducted experiments that reveal problems and issues in component composition.

Subdomain testing is the basis of the theory. Component developers describe their components by measuring approximations to functional and non-functional behavior on a finite collection of subdomains. Systems designers describe an application-system structure by the component connections that form it. From component specifications and a system structure, the theory predicts functional and non-functional behavior of the application. The calculations are made by a CAD tool that synthesizes the system properties. The system is not built, nor are any test executions performed. Neither the component sources nor executable code are needed by the systems designer. Furthermore, the CAD tool is much more efficient than it would be to assemble and execute an actual system. The theory interfaces with formal verification so that a mixture of tested and proved components can be utilized in system design.

The open-source connection is the method of tools development. Research prototypes have been developed by a diverse group of students on an open code base controlled by CVS. The research group includes graduate and undergraduate PSU students and high school students from the Apprenticeship in Science and Engineering program of Saturday Academy. Most of the work is done in intensive summer sessions, with different people involved each year. Everyone is encouraged to use and improve the code, under only very loose central control. Most of the code is in scripting languages running under Linux on a variety of machines at PSU and elsewhere.

CS 556 / OMSE 535, Implementation and Testing

In the usual offering of PSU’s CS 556 (and the similar OMSE 535) the class does a cooperative development project. Typically this is an ADT library to be used with a simple application system. Students code the library routines, unit test them (rather more than is done in practice, which is one point of the courses), and system test the application. Although “open source” as such is not explicitly named, that’s what is involved, since all the files are available under CVS, and people are encouraged to communicate to develop the best system possible. This past term we tried an additional experiment in 2-version programming: The class was divided into independent teams
which both did the implementation/test. At the end tools were provided that did extensive random testing at both unit and system level, using as test oracle a comparison of the two teams’ returned values.

Several aspects of open source are important to this course. First, since students may have a hard time meeting in person, they communicate by e-mail to discuss and improve the shared CVS code files. Second, the UNIX development environment is the best available: supporting tools are easy to write and share. Finally, and perhaps not so incidental as it appears, we had to switch the class project from the CS Sun servers to our Linux Lab machines because the system staff couldn’t make one of the support libraries work, while under Linux I could do it myself.