COLLABNET
Where Subversion™ Meets the Enterprise.

HPI

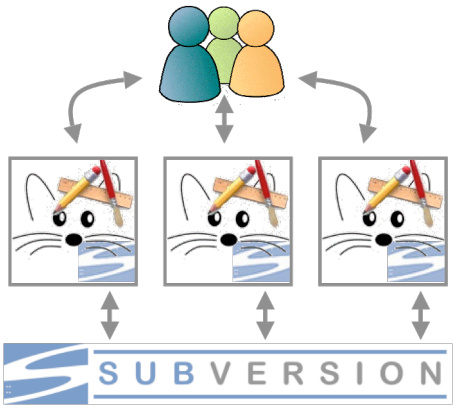
Bachelor Project Proposal, WS 2007/2008 – SS 2008
Software Architecture Group, Prof. Dr. Robert Hirschfeld

SqueakSVN

Squeak

Smalltalk is a purely object-oriented programming environment. Developing applications in Smalltalk is notably different from developing in other programming languages. Instead of working on text *files* containing the source code, programmers operate directly with *classes* and *methods* as the *semantic units* of the object-oriented program. The central tool for software development in Smalltalk is the *browser*, which makes the handling of these program elements very easy.

The browser and all other development tools are an integral part of the Smalltalk environment that programmers can adapt and extend to better support their working habits and the development process they are following. Combined with agile development practices, the Smalltalk environment is an ideal tool for explorative programming and rapid prototyping.



Squeak is an open-source implementation of the Smalltalk programming language. Created by Alan Kay and members of the original Smalltalk team, Squeak is maintained by a very active and steadily growing community of researchers, educators, and engineers. Besides its use in industry and academia, Squeak and its Etoys system are part of the OLPC XO, also known as the $100 laptop.

Subversion

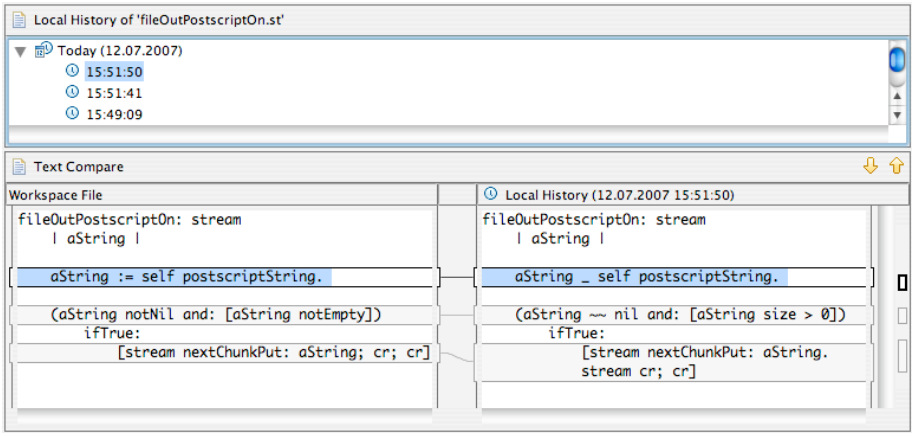
Subversion (SVN) is an open-source source code version control system developed by a CollabNet-led community of developers that was introduced as a successor to the popular CVS. SVN addresses many of the problems of CVS such as issues related to the handling of binary files, the atomicity of commits, and branching/tagging.

SqueakSVN

The ability to develop with fine-grained semantic units like classes and methods is one of the many advantages of most Smalltalk systems. Integrating these semantic units with widely used and supported source code control systems such as SVN, however, illustrates a drawback of this approach: The semantic units need to be mapped onto a coarser level of abstraction, usually to the level of individual files that the source code control system manages.

Several approaches to bringing source code version control to Smalltalk address this problem by implementing their own infrastructures in proper support of the units they need to deal

with. Some of them, most notably Monticello for Squeak, only allow for the versioning of complete snapshots of units, containing entire classes instead of fine-grained differences/deltas.



The purpose of SqueakSVN is to take advantage of the technical properties of Subversion as well as its growing popularity and support. By interfacing Squeak with Subversion, Squeak developers will be able to stay in their productive environment and also benefit from the team support and version control offered by Subversion. SqueakSVN will not only interface Squeak to SVN, but will also provide appropriate tool support and extensions to integrate version control and team programming facilities into the standard tooling of Squeak.

Implementation

The project will analyze common workflows and existing approaches to team programming, code sharing, and version control in the Squeak/Smalltalk environments such as Fileouts, Changesets, Envy Developer, Store, Monticello, Projects, or SAR. A mapping of Squeak's semantic units to structures to be maintained within SVN is to be designed. An extensive set of tools is to be developed that integrate SqueakSVN into the Squeak development environment. Examples include diff/change viewers, browsers and repository managers. The implementation of SqueakSVN will be carried out in the Squeak (http://squeak.org) environment, a modern, open source, highly portable, fast, and full-featured implementation of Smalltalk. It will provide an interface to Subversion (http://subversion.tigris.org), the open-source source code management system by CollabNet (http://www.collab.net/subversion). Squeak's development tools are to be extended to seamlessly integrate with SVN's versioning control facilities. Extreme Programming (XP) and other agile methodologies will be employed for software development.

Organization

A group of about six to eight (6-8) students may participate in the project. Organization and tasks will be determined by the project participants. The project will be carried out at the Hasso-Plattner-Institut in Potsdam. Project participants are expected to communicate with our partner via email, chat, or voice on a regular basis. In WS 2007/2008, participants will work on initial design sketches and prototypes. Main steps in design and implementation are to be executed in SS 2008. Expected results include a working software system accompanied by appropriate documentation.

Partner & Contact

Dr. Jeff Eastman, CollabNet, Brisbane, California, USA (www.collab.net, jeastman@collab.net)
Prof. Dr. Robert Hirschfeld, Chair of Software Architecture, Hasso-Plattner-Institut, Potsdam (www.hpi.uni-potsdam.de, hirschfeld@hpi.uni-potsdam.de)