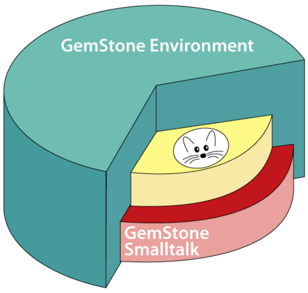
GS/Squeak:
Smalltalk as a Language Implementation Platform

Bachelor Project Proposal, WiSe 2017/2018–SoSo 2018
Software Architecture Group, Prof. Dr. Robert Hirschfeld

GS/Squeak – Languages and Dialects on GemStone/S

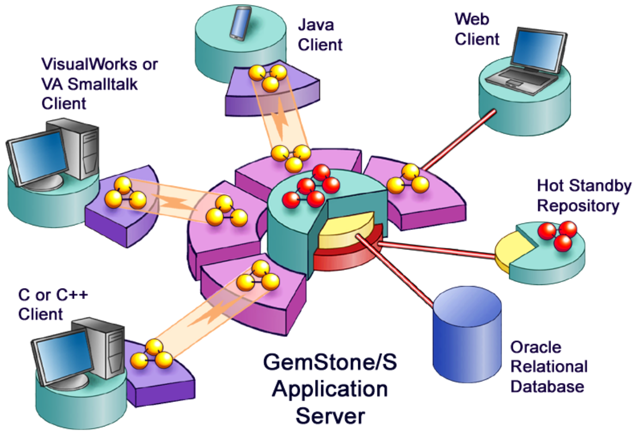
The goal of this project is to design and implement the Squeak/Smalltalk dialect into the GemStone environment and thus allow developers to take advantage of the Squeak tools and environment to deploy on GemStone easily. The participants will also investigate tools that are required in Squeak when developers plan to interact with vast amounts of objects as in the GemStone system. The Vivide development environment and framework, which supports low-effort, script-based tool construction in Squeak, will be available as a basis for implementing these tools.



Working with GemStone/S

GemStone/S is a unique application environment: on the one hand, an object-oriented database system (OODBMS) with persistence-by-reachability that supports applications involving terabytes of objects, on the other hand a full-featured Smalltalk implementation and live programming environment. GemStone/S has been used in the production of large-scale finance, logistics, and telecommunications applications for more than 30 years. Besides Smalltalk, GemStone/S interfaces with or implements SQL, Ruby, JavaScript, Java, C, and also other Smalltalk dialects. Applications written in these languages can interact with the data store. GemStone/S customers use these interfaces and implementations to leverage the high availability OODBMS from a variety of different code bases.

The GemStone persistence method is similar to image-based persistence common for other Smalltalks. However, under the hood GemStone is a client-server system that allows multiple clients to connect to one large object store and work on the "image" concurrently through ACID transactions. Applications running on GemStone would traditionally use the GemStone/S language, a dialect of Smalltalk. Over time, interfaces to other languages emerged.


Java Client
Web Client
VisualWorks or VA Smalltalk Client
Hot Standby Repository
C or C++ Client
GemStone/S Application Server
Oracle Relational Database

GemStone/S Smalltalk applications are typically developed in other Smalltalk environments independent of GemStone, based on the preferences of the developers for tools and development experience of different Smalltalk systems. However, at deployment time, when these applications are moved to run on GemStone, developers have to deal with incompatibilities between the

GemStone/S dialect of Smalltalk and the implementation dialect. Thus, incompatibilities become a major factor to consider and developers are not always able to use their preferred Smalltalk implementation during development, if incompatibilities with GemStone are an issue.

GemStone Language Development

GemStone offers separate execution contexts provided by the VM (environments) in which different languages can run. Different contexts can also interact, for example, to implement primitive behavior of one language in another. The technology for this interaction between different languages has been proven in the MagLev project, a Ruby implementation on top of GemStone/S. With these environments, GemStone users can deploy their code without modifications, provided the dialect that was used during development is also available as a language environment on GemStone.

Implementation

The implementation of the project will involve programming with GemStone/S, the GemStone/S VM interface written in C, Squeak/Morphic, the Vivide framework (https://github.com/hpi-swa/vivide), and Smalltalk's meta-programming facilities. It will also involve learning about virtual machines, their object representations, and execution semantics. An agile, iterative process will be employed for software development.

Organization

A group of about five to eight (5–8) 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 Institute in Potsdam. Project participants are expected to communicate with our partner via e-mail, chat, or voice on a regular basis. In WiSe 2017/2018 participants will familiarize themselves with the systems, main steps in design and implementation of a solution are to be executed in SoSe 2018. Expected results include a working software accompanied by appropriate documentation.

Partner & Contact

Dale Henrichs
GemTalk Systems, Beaverton, OR, USA
http://www.gemtalksystems.com/, dale.henrichs@gemtalksystems.com

Prof. Dr. Robert Hirschfeld,
Tobias Pape, Fabio Niephaus, Patrick Rein, Johannes Henning, Marcel Taeumel
Software Architecture Group, Hasso Plattner Institute, Potsdam
http://www.hpi.uni-potsdam.de/swa, hirschfeld@hpi.uni-potsdam.de