GemStone/S

@

IT Services

Vienna University of Technology

georg.gollmann@tuwien.ac.at
• Application
• Implementation
• Remarks
• Q&A
Application: Swiss Army Knife

• All administrative tasks of a support organisation with 25,000 customers.
• Not a lot of data but a lot of dependencies.
• Therefore business rules must be stored with the data and must be applied consistently.
• Single part time developer.
Some Services

• Campus Software
• Student Software
• Systems Support
• Account Management
• Authentication Service
• White Pages & Mail Router Feed
Implementation

- GemStone/S
  - Persistence
  - Concurrent Transactions
  - Constraints
  - Class Versions
  - Access Control
Implementation

• GemStone/S (cont)
  • Indexes
  • Namespaces
• GeODE (1993-97)
Web Interface

• Since early 1995
• Implemented inside GemStone/S
• Message passing paradigm
  /receiver.message?param=value
• Method must be in category „HTML Reply“
Web Interface (cont.)

• One server process for the general public, separate ones for each internal user on different ports.
• Mechanisms to avoid session hijacking.
• Was ported to become one of the first web servers for Squeak.
Secure Web Interface

• Since 1998
• Front end linking to SSLeay/OpenSSL
• Uses RPC Gem to keep OpenSSL and GemStone in separate address spaces.
Development

• Completely via the Secure Web Interface
• Class Browser adapted to the Web:
  • System: classes by class category
    • Class: methods by method category
      • Category: methods including comments
        • Methodeditor
Development (cont.)

- Method finder:
  - Implementors
  - Senders
  - References to
  - Substring
Development (cont.)

• Only stack dumps for debugging
  • Mitigated by design for testability
• Workspace
Display Framework

• asLink (Object, Collection, String, Dictionary,...)
• asHTML
  • title:
  • itemize:
  • tabulate:
Edit Framework

- editForm
  - formHeaderFor:
  - label:id:value:size:
  - label:id:list:selected:size:
  - formFooter
Edit Framework (cont.)

- Update Setup
  - field id
  - instance variable (constraintOn:)
- getter method
- setter method
Edit Framework (cont.)

• doUpdate
  • authorization check
  • error handling
• update:
  • updateList:
• fromString:
Delayed Actions

• GemStone/S is a transactional system
• Some actions can’t be undone by an abort (e.g. sending mail)
• Therefore delay them until after a successful commit
• Implemented as a queue in the session state
Exception Handling

• Catch all handler to keep the server alive
• Only generic message for the user (hacker?)
• Specific information by eMail for the maintainer
Code

• 95 classes
• 2505 methods
• 26023 lines
• 1012862 characters

„On my most productive days, the number of lines of code goes down“.
Trygve Reenskaug
Hardware

- HP L1000
  - dual 360 MHz CPU
  - 786 MB memory
- Linux box for testing
Repository

- ~1 million objects
- ~100 MB net data
- 200 MB SPC
Performance

• up to 370,000 hits/month so far
• mostly via SSL
• machine is mostly idle :-)
• some statistics could be faster...
Documentation

- Documentation inside
- Notes may reference any object
- Thus always up to date
Remarks

• Methodologies?
• Extreme Programming as a guideline
• Small is efficient
• Homogenous is efficient
  • Smalltalk throughout
  • Everything inside GemStone
Remarks (cont.)

• Better development environment (change sets)
• More SUnit Tests
  • Access rights errors are hardest to test for
Assessment

• Stable
• Efficient
• Rapid Application Development
• Extensive Change Support
Q & A