GLASS 2.0

Dale Henrichs
VMware, Inc.

ESUG 2011
Edinburgh, Scotland
August 22, 2011

GLASS 2.0

- Overview of GLASS
- VMware Cloud Foundry
- tODE demo

What is GLASS?

- A platform for deploying web-based Smalltalk applications in GemStone/S
  - Develop in Squeak/Pharo, Deploy in GemStone
    - OmniBrowser-based development tools
    - Compatibility layer for the Squeak and Pharo code base
    - Many popular projects ported to GemStone/S

Why use GLASS?

- GemStone/S is a high performance, high reliability alternative when your web application outgrows a single vm
  - With the “transparent persistence” and “transparent concurrency” features of GLASS
    - Add multi-vm persistence to web app without changes to your source code
  - **Free for commercial use**

Transparent Persistence

- GemStone is based on same persistence model used in Squeak/Pharo
  - Objects rooted in “Smalltalk” are considered persistent
- GemStone uses transactions instead of “save image” in Squeak/Pharo
  - commit
    - Changed objects in vm saved to repository
  - abort
    - Entire object graph in vm updated to current view
      - Dropping any changes on the floor

Transparent Persistence (cont'd)

- GemStone transaction logic is embedded in the web framework
  - Abort/Commit on HTPP request boundaries
    - Fresh view at start, changes saved to disk before user sees response
- No additional application-level code needed

Transparent Concurrency

- On commit, GemStone merges changes to object graph with latest view in repository
  - if two vms concurrently update same object
    - Commit conflict thrown
- Conflict management code embedded web framework
  - On conflict HTTP request is retried
    - As if request had been delayed in arriving
- No additional application-level code needed

VMware Cloud Foundry

- Cloud Foundry is a platform for building, deploying, and running cloud-based apps
  - CloudFoundry.com is a complete, hosted environment offered by VMware
  - CloudFoundry.org is an open source project

Cloud Foundry and GLASS

- We are actively integrating GemStone/S and Maglev into Cloud Foundry framework
- Plans to include support for Seaside/Aida/Iliad/Pier
- When finished we'll have cloud-based hosting service for GLASS
  - Will make deploying to GemStone/S even easier

tODE
the Object-centric Development Environment

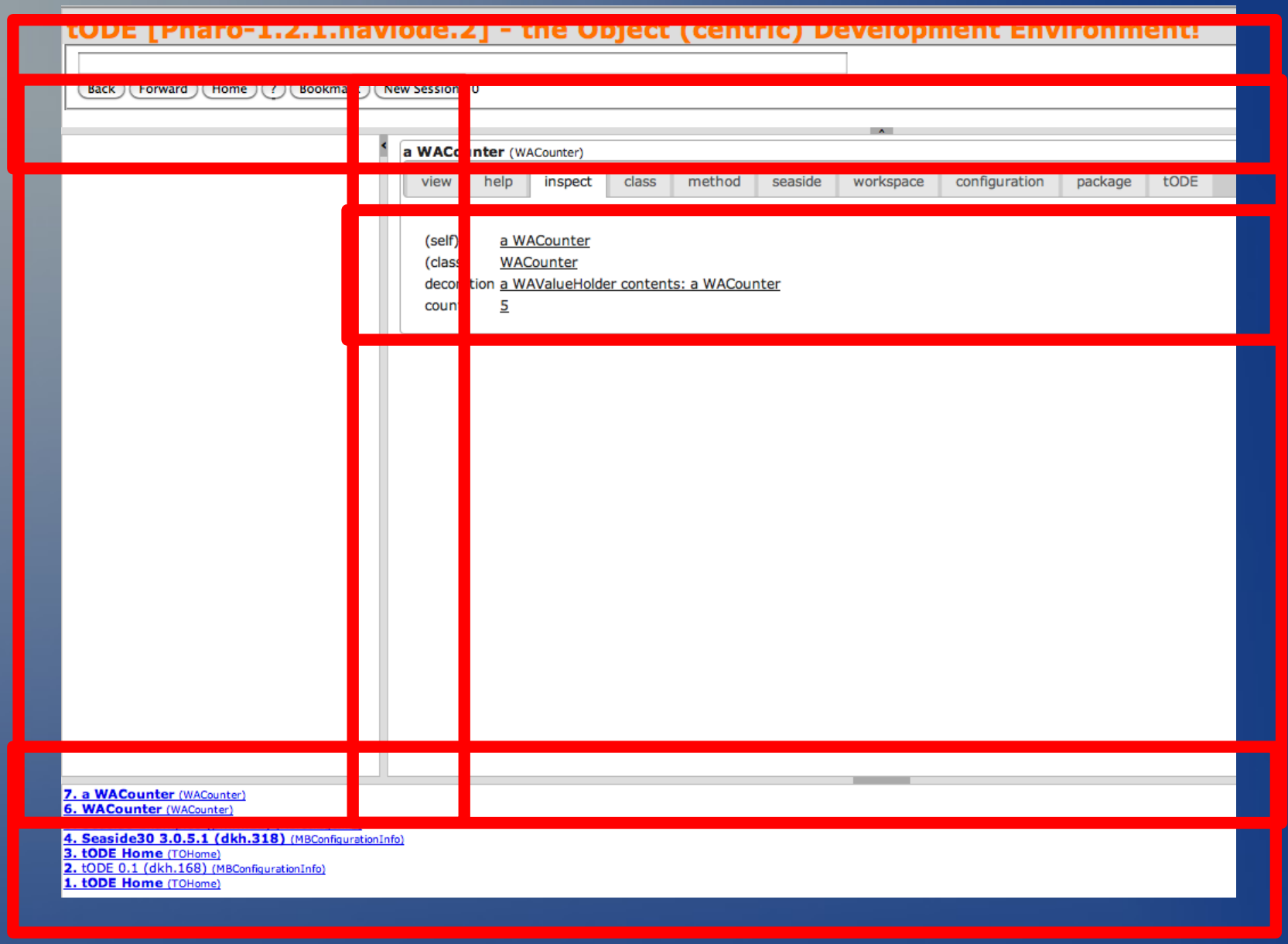
- A Proof of Concept for a Smalltalk IDE that runs in a web browser
  - Written in Seaside
  - Ported to Pharo and GemStone
  - MIT license
- A small, powerful framework that is easy to customize

Why tODE?

- tODE was created to provide the Smalltalk IDE for deployed web-applications
  - The traditional Smalltalk GUI is too slow when used across the WAN
  - Limited access to machines running in cloud
    - HTTP access only in Cloud Foundry
- Smalltalk without an IDE?
  - No way!

tODE paradigm

- Modeled on a web browser – tODE is an "Image Browser"
  - URL is a Smalltalk expression
  - LINKs follow object references
  - HISTORY is a stack of object references
  - PAGE is a set of tabbed PANES for an object
  - PANE contains a rendered aspect of the object


Back
Forward
Home
a WACounter (WACounter)
view
help
inspect
class
method
seaside
workspace
configuration
package
tODE
(self) a WACounter
(class) WACounter
decoration a WAValueHolder contents: a WACounter
count 5
7. a WACounter (WACounter)
6. WACounter (WACounter)
4. Seaside30 3.0.5.1 (dkh.318) (MBConfigurationInfo)
3. tODE Home (TOHome)
2. tODE 0.1 (dkh.168) (MBConfigurationInfo)
1. tODE Home (TOHome)

- Search
- etc.

tODE Demo

Conclusion

- Successful Proof of Concept
  - Functional ...
  - Simple, extensible framework
  - Need a (web framework) neutral rendering language
  - Need a bit of CSS and javascript polish

Future work

- tODE
  - Head towards alpha release
- GemStone/S for Cloud Foundry
  - Keep an eye out for further info
- For more info
  - http://seaside.gemstone.com/
  - http://cloudfoundry.org/
  - http://code.google.com/p/tode/
  - http://gemstonesoup.wordpress.com