



Cincom®

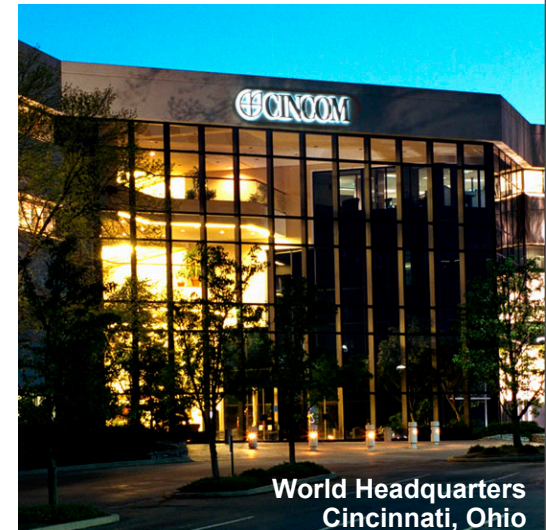
ESUG 2008

Cincom Smalltalk: Present, Future & Smalltalk Advocacy

- **What is *new & exciting* in Cincom Smalltalk?**

Arden Thomas

Cincom Smalltalk Product Manager



World Headquarters
Cincinnati, Ohio

SIMPLIFICATION THROUGH INNOVATION™

Welcome
August 24, 2008

Contents

- Where are we now?
- Where are we headed?
- Smalltalk Advocacy
- Web Velocity

Where are we Now?

New Products Released!

- VisualWork 7.6 - the craftsman developers choice
- ObjectStudio 8.1 - the business architects thinking tool
- ObjectStudio Classic 7.1.3

Where are we Now?

VisualWorks 7.6

- Revamped Refactoring Browsers
- Seaside
 - Supported
 - Opentalk for Seaside
 - Contributing to Seaside to move it forward
- O-R Mapping framework (GLORP) supported
- Revamped Hashing – greatly improved
- MySQL support (MySQL EXDI)

Where are we Now?

VisualWorks 7.6

- OSX Vm greatly improved
- New Hover help
- Flicker reduction
- WebToolkit refactored (easier headless config)
- Over 60 pages of enhancements in release notes!

Where are we Now?

VisualWorks 7.6 preview

- Grid Widget (from Widgetry project)
- Universal startup script (for Unix platforms)
- WriteBarriers
- Sparing Scrollbars (appear only when needed)
- Multithreaded COM (non blocking)
- Command line and Scripting support

Where are we Now?

ObjectStudio 8.1

- Only Vista Certified Smalltalk!
- New installer
- Fixes
- Seaside supported

Where are we Headed?

Goals

- Advance the product while retaining customers
 - No advancement = loss of customers
 - Changes to abrupt = loss of customers
- How?
 - Keep it compatible where reasonable
 - Change incrementally
 - Make moving to new technology easy
 - Porting guide or Porting tool
 - If you make them change or port:
MAKE IT WORTH IT!!!
 - Clear improvements they can sell or show

Where are we Headed?

Goals

- Grow the Smalltalk community
- How?
 - Improve the product
 - Smalltalk Advocacy
 - Embrace new technology like Seaside
 - Web Velocity
 - Get the product in developers hands (NC)

Where are we Headed?

Where does product input come from?

- Customers
- Internally
 - Engineering
 - Star Team
 - Sales
- Smalltalk Community
 - Active, engaged, and often opinionated users

What are we Asking?

9 for '09 The Next Product Cycle ideas list

- Fix, Refine or Replace
- Added or new functionality
- Removal
 - Debt reduction
- Innovation
 - A Smalltalk tradition
 - Lead the field

What did they tell us?

Make it look Modern

- It looks old – fix it
 - PM note: We need more than baby steps
- Common to both
 - VisualWorks
 - ObjectStudio8

What did they tell us?

Make Store work better in large installations

- Store
 - Performance
 - Merge
 - Configuration management
- Sometimes faster to load from scratch than update

What did they tell us?

Fonts

- Font system is dated, inflexible, and in some cases: broken
- Choices, sizing, specifying, matching
- More capable rendering

What did they tell us?

Internationalization

- Needs of several large customers
 - Need more than tiny steps
 - Researching CLDR based solution discussed
 - (CLDR - Common Locale Data Repository)

What did they tell us?

- Ongoing initiatives important still
 - VMasDLL - making CST a better player in multi-technology environments
 - C interface speed improvements
 - C interface tool improvements (parsing, generation)

What did they tell us?

Leveraging Multi-Core

- Multi-Cores are become ubiquitous
- Leveraging this is really, really, *really* attractive
 - Just makes a compelling and exciting story
 - To developers, decision makers, technologists, experimenters, customers
- Multiple solutions possible – short term, long term
 - Multi images
 - Decision whether to multi thread any part of VM hard and controversial
 - Perhaps introduce frameworks like Map Reduce
 - limited perhaps, but big where useful, useful buzz for product

Roadmap forward

So What Made it?

- High Priority
- Priority
- Internal

Roadmap forward

High Priority

- Internationalization
- 64 bit VM work
- Improved Graphics (vet Cairo)
- GUI infrastructure work
- Graphics design work
- Modeling tools OS8
- OS8 GUI update (started)
- Shadow compilation & parcel generation
- DLLCC Speed
- High load threading safety
- Posix Style Delays

Roadmap forward

High Priority

Internationalization support

What: More extensive internationalization support to handle formatting, collation, display, and input for a very large set of languages and locales.

How: Unicode Common Locale Data Registry (CLDR).

“The Unicode CLDR provides key building blocks for software to support the world's languages”.

We will use the Unicode Common Locale Data Registry (CLDR) standard registry of locales, and ensure that we can handle display and input of Unicode characters on all platforms.

Why: Customers increasingly need to handle diverse and multiple locales to support internationally delivered applications

Roadmap forward

High Priority

64 bit VM support

What: Deliver our Smalltalk environments capable of running on 64 bit operating systems as 64 bit applications

Why: 64 bit addressability gives the ability to support many more objects in memory is the primary benefit. Other benefits include much larger SmallIntegers, and the introduction of SmallDouble.

Roadmap forward

High Priority

Modernization - Improved Graphics Capability

What: Greatly improve the graphics capabilities of our product, in rendering graphics

How: We are moving forward in vetting the experimental work integrating with the Cairo graphics framework. As long as it remains viable, we want to get this into preview in the next release.
Significant

Why: It improves the ability to deliver very attractive, modern interfaces, adding features like scaled vector graphics (SVG), and complex alpha channel transparencies and gradients.

Roadmap forward

High Priority

Modernization - GUI infrastructure work

What: Improvements and refinements to our gui infrastructure

How: Many improvements are well understood

Why: It makes the current gui simpler, faster, easier to use

Roadmap forward

High Priority

Modernization - Graphics Design Work

What: Improving the look of the IDE with modern icons and logos

How: Professionally designed graphics

Why: Modernize the attractiveness of the IDE

Roadmap forward

High Priority

Modernization – Object Studio 8 GUI update

What: Update the look of the IDE with modern widgets

Why: Modernize the attractiveness of the IDE

Roadmap forward

High Priority

Modeling Tools – Object Studio 8

What: Restore & enhance the OS classic modeling tool

Why: A core feature for new Object Studio users

Roadmap forward

High Priority

Shadow Loading & Parcel Generation

What: Load of packages either succeed or all the changes are rolled back (atomic load)

How: Changes are compiled into "shadow" namespaces, and installed into the real system once all code has successfully compiled

Why: More tolerant & capable loading.

Prevents errors when packages partially load, making it easier to load packages that modify the core system.

Roadmap forward

High Priority

DLLCC speed

What: Make calls out to external libraries faster

Why: Makes high frequency external calls perform

Roadmap forward

High Priority

High load threading safety

What: Improve concurrent operations robustness

Why: Improves heavy load robustness and for grid computing

Roadmap forward

High Priority

Better Delays

What: Make delays more robust

How: Posix compliance (Posix has standards for real time systems)

Why: Makes delays work correctly under broader conditions and stress

Roadmap forward

Priority

- Fonts
- Look Policies
- Store Improvements
- VMasDLL refinement
- DLLCC parsing/ gen
- Protocols enhancement
- Mapping tools for OS8
- Velocity on S3/EC2/SimpleDb
- External OSX work
- Research: Leverage Multi-Core
- Native Installers

Roadmap forward

Priority

Mapping Tools – Object Studio 8

What: Restore & enhance the OS classic mapping functionality

How: Use shared O-R mapping framework (glorp)

Why: A defining feature for Object Studio users

Roadmap forward

Priority

Web Velocity on EC2/S3/SimpleDb

What: Zero setup time to build web applications

How: Sign up for Amazon services from Web Velocity product page

Why: Remove barriers to entry for building and deploying web applications

Roadmap forward

Priority

Research: Leverage Multi Core

What: Make it easier to leverage multi-core computer

How: Research & consider different approaches

Why: Taking advantage of ubiquitous multi-core computers is attractive

Smalltalk Advocacy

Contents:

- Moving Smalltalk forward
- Legacy
- Pearls of Smalltalk
- The Second Surge

Moving Smalltalk Forward

Making the right information known/available

Why?

- Give current customers information for justification
- Let past users know it is safe to return, a safe choice
- Attract new people to Smalltalk

Perception is changing:

Table 1. Four Phases of Software Aging

Definition	Adult	Mature	Aging	Elderly
	Language or tool is actively evolving. Thriving and expanding community of users and third parties complement primary vendor. User base expands.	Language or tool is in general use. Community is active and includes some third parties. Use in existing deployments may continue to expand. User base may be growing or declining, but relatively stable.	Language or tool use primarily in existing deployments. Community activity has declined. Vendor may assume community sponsorship. Niche applications may continue to thrive.	Skills base and community support are shrinking and small. Little or no third-party support.
Criteria				
Frequency of Releases	Regular, functionality expanding	Regular, focus on platform and patch support	Irregular to seldom	Never
Availability of Third-Party Training	High	Declining	Low	None
Ability to Source Trained Skills (internal or external)	High	Stable	Declining, regional shortage	Declining, low
Use in New Projects (regionally)	Frequent	Stable	Seldom	Never
Growth Rate of Application Population	Rapid growth	Targeted growth	Stable	Negative (declining)
Programming Constructs	Stable, frequent extensions	Stable	Stable, lagging current standards	Stable, deficient
Third-Party and Community Tool Support	Growing	Stable	Declining, incomplete	Minimal
Hardware Support	Adequate	Adequate, may be delayed	Defined, incomplete	Shrinking
Management Policies				
Content of Yearly Viability Review	Versions in use, skills plan, vendor support and release road map. Event review: Mergers, acquisition, new replacement product launched	Versions in use, skills plan, vendor support and release road map. Event review: Mergers, acquisition, new replacement product launched	Assess skills and technology risks and operational risks. Vendor-support commitments. Event review: Mergers, acquisition, replacement product launched	Review migration, retirement plans. Event review: Mergers, acquisition, replacement product launched
Short-Term Actions	Assess architectural suitability	Assess architectural suitability	Actively discourage; closely monitor vendor commitment and road map	Forbid expansion
Long-Term Strategy	Encourage design-driven processes for repeatability	Migrate to most-current form; increase tool use	Avoid new use	High priority to replace

Source: Gartner (December 2007)

Moving Smalltalk Forward

Making the right information known/available

- What Information?
 - Legacy of Xerox PARC – Great story!
 - Pearls of Smalltalk – differentiation
 - Smalltalk's Second Surge
 - Technology ahead of its time now hitting the mark

Moving Smalltalk Forward

It is a *great* time to be in Smalltalk

- Our current clients are expanding usage & licenses
- We are picking up new Smalltalk customers
- There continues to be interest in Smalltalk. New versions of Smalltalk are being developed or started, such as Syx, and IronSmalltalk, IO & obvious influence on languages like Ruby, Python
- New, experimental languages continue to borrow (but now at deeper levels) from Smalltalk, such as Newspeak, ...

Legacy of Xerox PARC

“Take a look at the computer in front of you”

- PARC pioneered the basics of everything we take for granted today!
- Office metaphor
- A whole new generation does not know the rich heritage and deep significance of Smalltalk in modern computing
- “The technology that invented the future”

Pearls of Smalltalk

How is Smalltalk different?

- Built in a research Lab by a select group
- Built over a decade, with no product pressures
- Started from scratch every two years
- Goal of making computing easy for people
- Learning focus
- These conditions are rare

Pearls of Smalltalk

How is Smalltalk different?

- Productivity
- Elegance
- Pioneering
- Portability
- Full Source code
- Enjoyment
- Broad Applicability

Pearls of Smalltalk

Get it in their hands!

Just try it anyway- there are benefits!

- even if you have no intention of using it currently. Why?
- We are frequently told by developers that they never really, fully “got” OO, until they learned Smalltalk
- It is still a great language for proof of concept work, and experimental work.

Smalltalk's Second Surge

Smalltalk was (is?) ahead of its time

- The market is starting to recognize specific features that Smalltalk has that were overlooked the first time
 - Dynamic Languages are becoming hot
 - Many languages “borrowed” a few obvious features from ST
 - They missed some key features that are being recognized now
- Other languages may come and go
..... Smalltalk will still be around in 50 years
 - It was simply too well designed

Smalltalk's Second Surge

Steve Job's Parc visit story

Quoted from "Triumph of the Nerds, part III"

“And they showed me really three things. But I was so blinded by the first one I didn't even really see the other two.

.... they showed me .. object orienting programming
but I didn't even see that.

.....they showed me was a networked computer
system I didn't even see that.

.....I was so blinded by the graphical user interface.
I thought it was the best thing I'd ever seen in my life”

Smalltalk's Second Surge

The Grand influencer of many languages

- What did they borrow?
- More importantly – What did they *miss*
 - *Three things they missed*

Smalltalk

1) The Closures !!!

- Closures (blocks) in Smalltalk and their clean, clear, consistent integration and usage, are responsible for much of the *magic*
- Q: “Smalltalk hasn’t really changed much. Isn't it old? Other languages add things like new control structures”
- Closures make it possible to create new control structures.
- The ability to create new control structures allow refinement and evolution without changing the definition of the language

Smalltalk

2) Reflection

- Reflection (dynamic) is the ability of a program to observe and modify itself during execution
- Many think the **future in programming** lies in ***meta-programming*** (programs that program).
- Rails and Seaside are examples of Meta-programming, relying on reflection in their respective languages
- Consistency - being fully, consistently OO is an advantage in reflection - “Turtles all the way down”

Example : Whatever

Smalltalk

3) The Syntax !

- One of the “jewels” of Smalltalk
- Simple, consistent, expressive, robust
- Perhaps (by design) the most natural for people to learn
- Simply:
 - object message
 - object message parameter
 - object message: parameter

Smalltalk

A memo in English

Memo

to: ESUG attendees

from: Arden Thomas

re: Seaside & Smalltalk

date: 25 Aug 2008

Thanks Niall !

Smalltalk

A memo in Smalltalk

Memo

to: 'ESUG attendees'

from: 'Arden Thomas'

re: 'Seaside & Smalltalk'

date: '25 Aug 2008' asDate

Smalltalk

Some Cultural Differences - educate

- Debugging
 - The Debugger in Smalltalk is a Live environment that is desirable to spend time in ... you can even code and test in it.
- Tools
 - It has them!
 - You don't use your own editor
- “Snapshot” (Image based) Environment
 - Like putting your notebook to sleep
 - Can still rebuild from files/database/parcels

Web Velocity

What, How & Why?

- What is it?
- How is it different?
- Why should I be interested?

Web Velocity

What is it?

“Web development for the web, in the web, on the web”

Web Velocity is a tool for developing attractive & capable Web applications

In-browser tool assistance makes it fast and easy to create dynamic, Ajax enabled, web applications. Build, test and debug your applications *in* the browser.

Web Velocity

What is it ... made of?

- VisualWorks Smalltalk
- Seaside framework
- O-R Mapping (Glorp)
- Infrastructure and browser tools
 - (scaffolding, ActiveRecord, Web tools)

Web Velocity

Why should I be interested?

Surprise: “It’s not for you”!

WHAT?!

- It is software for your sister

WHAT???

Web Velocity

Sister?

Web Velocity is not targeted *at* Smalltalkers

(If you are on CST license you get most of it)

If you are a Smalltalk user & advocate, Web Velocity might be something you can recommend to a non-Smalltalk using "sister" group in your organization

Web Velocity

So, Why should I be interested?

- For the community, the interest of growing the Smalltalk community
 - Bypasses the “Language Wars” obstacle
 - Gives customers powerful current frameworks for development
 - Stealth approach to the Smalltalk experience

Web Velocity

Why should I be interested?

- It can be recommended to groups wanting to build robust web applications
- It can be recommended to groups wanting to tackle complex or difficult web applications
- It can be recommended to non-Smalltalk groups in a Smalltalk using organization

Web Velocity

Stay tuned this week for the

Demo!

James will be giving a presentation and demonstration for Web Velocity on Thursday morning

Thanks!

- Star Team (Smalltalk Strategic Resources)
 - athomas@cincom.com Smalltalk Product Manager
 - jrobertson@cincom.com Smalltalk Evangelist
 - sfortman@cincom.com Smalltalk Director
- <http://www.cincomsmalltalk.com>
- <http://www.cincomsmalltalk.com/userblogs>



©

**2008 Cincom Systems, Inc.
All Rights Reserved
Developed in the U.S.A.**

CINCOM and the Quadrant Logo are registered trademarks of Cincom Systems, Inc.

All other trademarks belong to their respective companies.