# Migration from VSE to VW with Pollock

Christian Haider

smalltalked Visuals

# Overview

- Motivation
- Migration Process
- The Baseline
- Migration details
- Conclusions

smalltalked Visuals

# Why this talk?

- Smalltalk market
  - VSE is dead
  - Pollock comes
- As a checklist for VSE projects
- Tell you that it is easy and worth it
- My experience
  - 2 private projects
  - 1 consulting project

smalltalked Visuals

# Why migrate?

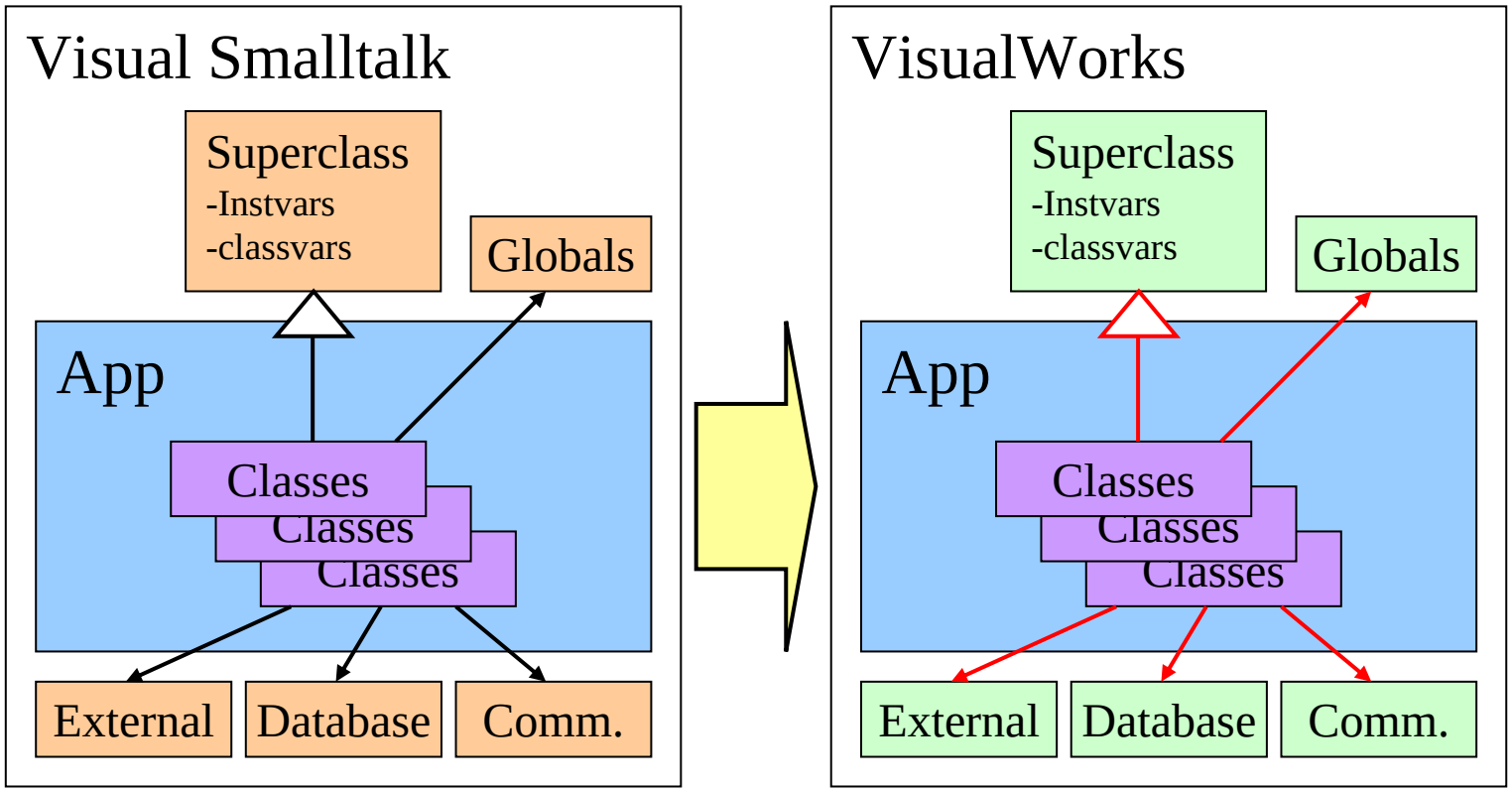
- Why migrate
  - Worry about support for new Windows versions
  - “System gets cracks”
  - New features
    - More Platforms
  - Increase Productivity
- Why not to migrate
  - Cost of VisualWorks
  - Cost of migration
    - GUI migration is now easier with Pollock
  - Cost of change
- Migrate to where?
  - VW, VA, Java, C#

smalltalked Visuals

# How to migrate

- Only migrate – don’t improve!
- The Project is migrating, not just the Code
  - Get a Consultant
  - Customer will do everything!
  - Consultant will explain, guide and help where appropriate (and keep the fingers off the keyboard :-)
  - Setting
    - One Table with one Keyboard and Mouse
    - Display with a Beamer
- Start with visual.im
  - Enhance the VisualWorks environment as needed
  - Consultant provides help and explanations when useful

# Situation


Visual Smalltalk
Superclass
-Instvars
-classvars
Globals
App
Classes
Classes
Classes
External
Database
Comm.
VisualWorks
Superclass
-Instvars
-classvars
Globals
App
Classes
Classes
Classes
External
Database
Comm.

smalltalked Visuals

# Porting Steps - Overview

1. Create a Baseline (same source)
2. Translate Code (runnable code)
3. Test (same behavior)
4. Build Runtime (same product)
5. Make it Nice (new possibilities)

smalltalked Visuals

# Baseline

- Defined state
- Equivalent Source
- Same module structure and versions
- All sources and texts are integrated and published in Store
- Loads without serious warnings
- All work can be done in VisualWorks

smalltalked Visuals

# Towards the Baseline

Get all Sources into VisualWorks

- Module Structure…
- Namespace Name
- External Interfaces…
- FileIn Modules…
- Publish Baseline
- Load in fresh image

# Store Bundle Structure

- The Store structure should reflect the development process
  - If a Team/V repository exists, the same structure should be used
- Suggestion:
  - Application Development
    - **(VSE Porting)**
    - Application Runtime
      - Common
      - Module 1
      - …
      - Module n
    - Tools
    - Tests
- Define a version format (1.0.0.0 <major.minor.revision.build>)
- An Access DB can be used nicely as light weight Store

smalltalked Visuals

# External Interfaces

- DLL calls
  - <api: name args return>
- Sockets
  - Use VW
- Database
  - Use VW
- COM?
  - SmallCOM/X?
- Other?

# Environment

- Tool to fileOut XML VW sources
- VSE Porting package
  - VSE namespace
  - VSE.Class
  - VSE.DynamicLinkLibrary with api parser
- Tools package
  - Browser list icons

smalltalked Visuals

# FileIn

1. Open GHChangeList
   - Precondition: Undeclareds are empty
   - Set default namespace and encoding
2. Load all sources for one package
3. Replay all
   - Missing superclasses -> define as VSE.Class
   - External interfaces -> define as VSE.DynamicLinkLibrary
4. Resolve Undeclared…
5. Add text file comments to Pundles, Classes and Globals
6. Publish to Store

smalltalked Visuals

# Resolving Undeclareds

- Referenced class
  - Subclass VSE.Class
- Global, pool constant
  - Add Shared Variable in VSE (move to porting package)
- Instvar, class instvar, class var
  - add in VSE.Class subclass
- Global in system extension
  - Shared variable in referring Class (move to porting package)
- Instvar in system overwrite
  - Add as temp var and add a comment/marker

# The Migrant

- Missing references are resolved (empty porting package)
- Message send but not implemented issues are resolved
- The code loads without warnings
- Clean-up
  - The code critic reports no bugs or possible bugs.
  - There are no unnecessary system extensions or overrides
  - There is no unused code
  - The system and the applications run basically correct.
- Problem areas are identified, recorded and patched
- The behavior of the VSE system is basically preserved

smalltalked Visuals

# Towards the Migrant

- Eliminate the porting package
  - Resolve former Undeclareds
- Resolve ‘Methods send but not implemented’
- Options
  - **Rewrite Tool (!)**
  - Regex
  - Reimplement
  - Remove
- Publish and load in fresh image often

# Topics

- Translation by topics
  - Files
  - Registry
  - TCP/IP
  - Processes
  - Exceptions
  - Finalization
  - C calls
  - Dependency mechanism
  - GUI…

# Differences in the Base

- (Smalltalk at: #...)
  => (‘namespace.global’ asQualifiedReference value)
- TimeStamp => Timestamp
- Colors: Color, RGBColor, logical colors => ColorValue
- Pool constants (Clr*, Cr, Lf) => ColorValue, Character, …
- File access: File, Directory, Disk => Filename
- ObjectFiler => BOSS
- Application startup: SessionModel, SmalltalkLibrary, SmalltalkLibraryBinder => ObjectMemory, Parcel

# GUI Migration

- Emulate or replace?
- Port one existing GUI
  - Get the topics with the debugger
  - Change them generally with the rewrite tool

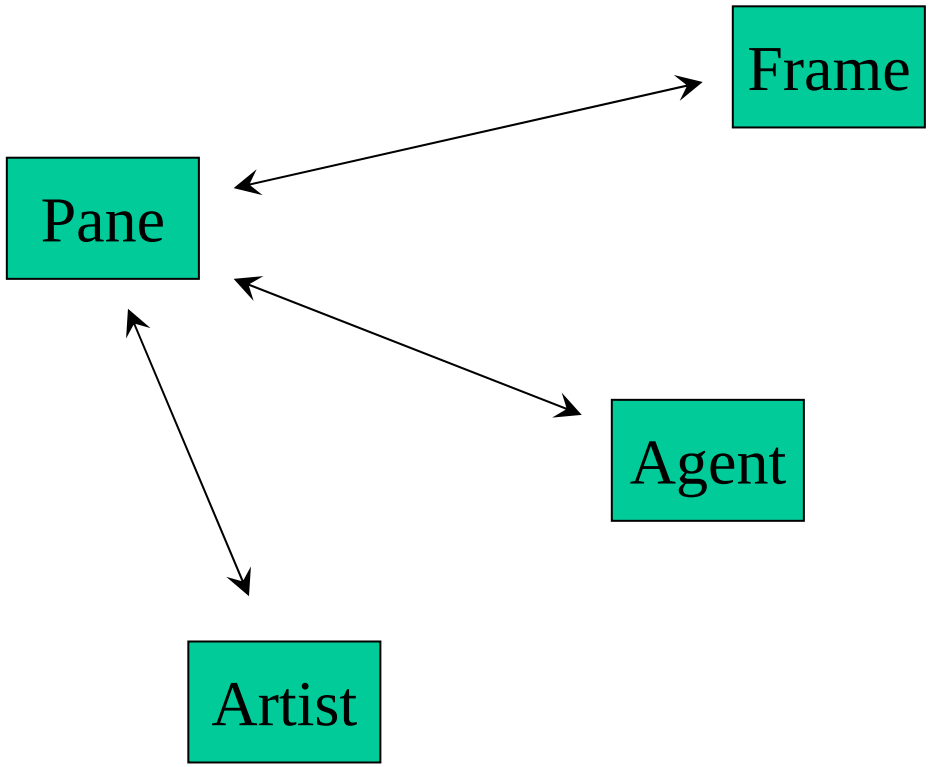
# Differences in the GUI

- CursorManager, MessageBox , Prompter => Dialog
- PrinterDialog [Windows Goodies], FontDialog
- Display, Bitmap, Images, Icons
- Text, Fonts
- Window opening
- Windows messages: wm*

# GUI differences

| Digitalk | Wrapper | Pollock |
|---|---|---|
| ViewManager | ApplicationModel | UserInterface |
| SubPane | VisualPart | AbstractPane |
| Nesting: own pane | SubCanvas | Form |
| #needsContents | (model) | Window #opened |
| Windows events | Controller | Controller |

# Pollock Model



# Widgets

| | |
|---|---|
| GroupBox | GroupBox |
| Button | Button |
| CheckBox | CheckBox |
| RadioButton | RadioButton |
| ListBox | ListBox |
| TabControl | TabControl |
| Toolbar | Toolbar |
| ToolbarSeparator | ToolItemSeparator |
| ToolbarButton | ToolbarButton |
| ToolbarToggleButton | ToolbarButton |

| | |
|---|---|
| StaticText | DisplayText |
| EntryField | InputField |
| TextPane | TextEdit |
| RichEdit | TextEdit |
| ComboBox | DropDownList |
| TreeView | TreeView |
| TabControlPage | Page |
| StatusField | -- |
| StatusWindow | -- |
| ImageList | -- |

# Adding a Widget

- VSE

```
listBox := ListBox new.
listBox font: self defaultFont.
listBox printSelector: #printItem.
listBox when: #changed:
        send: #setSelection: to: self.
listBox when: #needsContents
        send: #setList: to: self.
frame := LayoutFrame
        topLeftRatio: 0 @ 0
        bottomRightRatio: 1 @ 1.
listBox layoutFrame: frame.
view addPane: listBox.
```

- Pollock

```
listBox := ListBox new.
listBox textStyle: self artist listStyle.
listBox displaySelector: #printString.
listBox when: #selectionChanged
        send: #setSelection: to: self.


frame := FractionalFrame
        fractionLeft: 0 top: 0
        right: 1 bottom: 1.
listBox frame: frame.
window addComponent: listBox.
```

# Tests

Resolve the remaining differences

- Module tests
  - SUnit test
    - First in VSE
    - Then migrate to VW
- Application tests

# Runtime and Performance

- Build Runtime
  - RTP bug: exit on last window close
- Test Runtime
- Test Performance (WTS)

smalltalked Visuals

# Make it Nice

- Separate development from runtime
- Structure namespaces
- Define protocols (if not Team/V)
- Comment Pundles, Classes, Globals (if not Team/V)
- Write your formatter
- Make use of VW Features
  - MultiProc-UI
  - Opentalk, CORBA
  - Net and Web Support
  - .net and CE Support (in Beta)
  - Pollock

smalltalked Visuals

# Conclusion

- It is easy and can be done in a few weeks
- All code is rewritten and looked at
  - Duplications
  - Nil disease
  - Coding Style
  - Forgotten code
  - Direct variable access
  - Branches
  - Long methods

# Recommended Tools

- StoreForAccess (lightweight on disk)
- GHChangeList (adds Namespace and package support)
- Coding
  - Refactoring Browser
  - Rewrite Tool
  - Regex Tool
  - Code Critic
- SUnit (standard for module tests)
- Environment
  - RBCodeHighlighter (highlights while typing)
  - RBTabs (avoids too many browsers)
  - AutoComplete
  - RBSUnitExtensions (adds SUnit buttons to the browser)
  - RBStoreExtensions (shows versions graphically)
  - ExtraIcons (nice list icons, essential for testing)
  - WinIcons (icons for VW tools)