OmniBrowser - Meta-modeling Browsers

David Röthlisberger
Software Composition Group
University of Berne
Squeak System Browser

```
templateForSubclassOf: priorClassName category: systemCategoryName

"Answer an expression that can be edited and evaluated in order to define a new class,
given that the class previously looked at was as given"

+ priorClassName asString, ' subclass: #NameOfSubclass
instanceVariableNames: '***
classVariableNames: '***
poolDictionaries: '***
category: '***, systemCategoryName asString, '***
```
Problems of Old Browsers

- Complex state management
- Guard code often spread over UI elements:
  - `selection notNil ifTrue: [self doAction]`
- Extensibility poor
OmniBrowser Approach

• Meta-modeling navigation in a metagraph
• Separate navigation model from domain model
OB: Meta-Modeling Navigation

Navigation modeled in metagraph, a state machine
Example: File Browser
File Browser: Metagraph
Metagraph in Code

OBFileBrowser class >> defaultMetaNode
  |directory file|
directory := OBMetaNode named: 'Directory'.
file := OBMetaNode named: 'File'.
directory
  childAt: #directories put: directory;
  childAt: #files put: file.
^directory
File Browser: Domain Graph

/ /home pic.jpg
todo.txt Squeak.app
Domain- and Meta-Model

Domain model

- OBNode
  - OBFileNode
    - path
  - OBDirectoryNode
    - directories
    - files

Metagraph

- Directory
  - #directories
- File
  - #files
Domain Model in Code I

**OBNode** subclass: **#OBFileNode**

instanceVariableNames: 'path'

[... ]

**OBFileNode** subclass: **#OBDirectoryNode**

instanceVariableNames: ''

[... ]
Domain Model in Code II

OBDirectoryNode >> directories
|dir|
dir := FileDirectory on: path
^dir directoryNames collect: [:each |
   OBDirectoryNode new path: (dir fullNameFor: each)]

OBDirectoryNode >> files
|dir|
^dir fileNames collect: [:each |
   OBFileNode new path: (dir fullNameFor: each)]
Root Node of Domain Graph

```
OBFileBrowser class >> defaultRootNode
  ^OBDirectoryName new path: '/'
```
File Browser in Action
Problems of Old Browsers

• Complex state management
• Guard code often spread over UI elements:
  `selection notNil ifTrue: [self doAction]`
• Extensibility poor
Navigation is not hard-coded, but modeled as a graph.
Problems Solved II

Metagraph easily extensible:
- different navigation
- more transitions
- more state properties
Changing Navigation Metagraph

Diagram:
- Directory
  - #directories
  - #files
- File
  - #filelink
  - #dirlink
- Sym Link
  - #symlink

OmniBrowser - Meta-modeling
Browsers
Adding new State Properties: Auto-selection

```plaintext
OBMetaNode >> autoSelect: aMetaNode
  autoSelect := edges detect: [:ea | ea metaNode == aMetaNode] ifNone: [nil]

OBFan >> autoSelection
  |auto|
  auto := parent metaNode autoSelect.
  ^auto ifNotNil: [children detect: [:ea | ea metaNode == auto] ifNone: [nil]]
```
Extending Metagraph: Icons

```

OBNode >> icon
  metaNode iconForNode: self

OBMetaNode >> iconForNode: aNode
  ^filters inject: nil into: [:icon :filter |
    filter icon: icon forNode: aNode]

OBMethodFilter >> icon: aSymbol forNode: aNode
  ^aNode isOverridden ifTrue: [#arrowDown]
    ifFalse: [#blank]
```
The OmniBrowser Framework

• Browser
• Node
• MetaNode
• Command - action manipulating nodes
• Filter - filtering and adapting nodes for display
• Definition - modifiable textual representation of a node, eg. method source code
Omnibrowser Core

OmniBrowser - Meta-modeling
Browsers
Widgets

• Lists / Columns
• Radio Buttons (modal filter)
• Menus
• Definition Panel
• Button Panel
• Mercury Panel
• Annotation Panel
• …
Realizing the System Browser

• More complex navigation
• But still just from left to right
• Modal filter for instance, comment, class, (traits)
• Numerous commands
System Browser: Metagraph
Metagraph in Code

OBSYSTEMBrowser class >> defaultMetaNode

| env classCategory |
env := OBMetaNode named: 'Environment'.
classCategory := OBMetaNode named:
                   'ClassCategory'.
env childAt: #categories put: classCategory.
classCategory ancestrySelector:
                        #isDescendantOfClassCat:.
self buildMetagraphOn: classCategory.
^env
System Br.: Domain Model
Root Node of Domain Graph

OBSYSTEMBROWSER >> defaultRootNode

^OBENVIRONMENTNODE forImage
System Browser in Action
System Browser on the Web
Several GUls

• GUls: Morphic, Web, GemStone
• Same metagraph
• Same domain model
• Widgets differ
Evaluation of OmniBrowser I

- **Strengths**

  - easy to use, extend, customize
Evaluation of OmniBrowser II

• Limitations
  - Navigation flow hard-coded (strict left-to-right approach)
  - Single-selection only, selection not modeled in metagraph
  - Widgets limited and fixed, difficult to extend
Summary

• OmniBrowser is a framework to create various browsers
• Extensible metagraph to model navigation and state
• Separated domain model (domain graph)
• Basis for various browsers (system browser, file browser, universe browser, package browser, inspector, debugger, etc.)
• GUIs available for Morphic, Web, GemStone