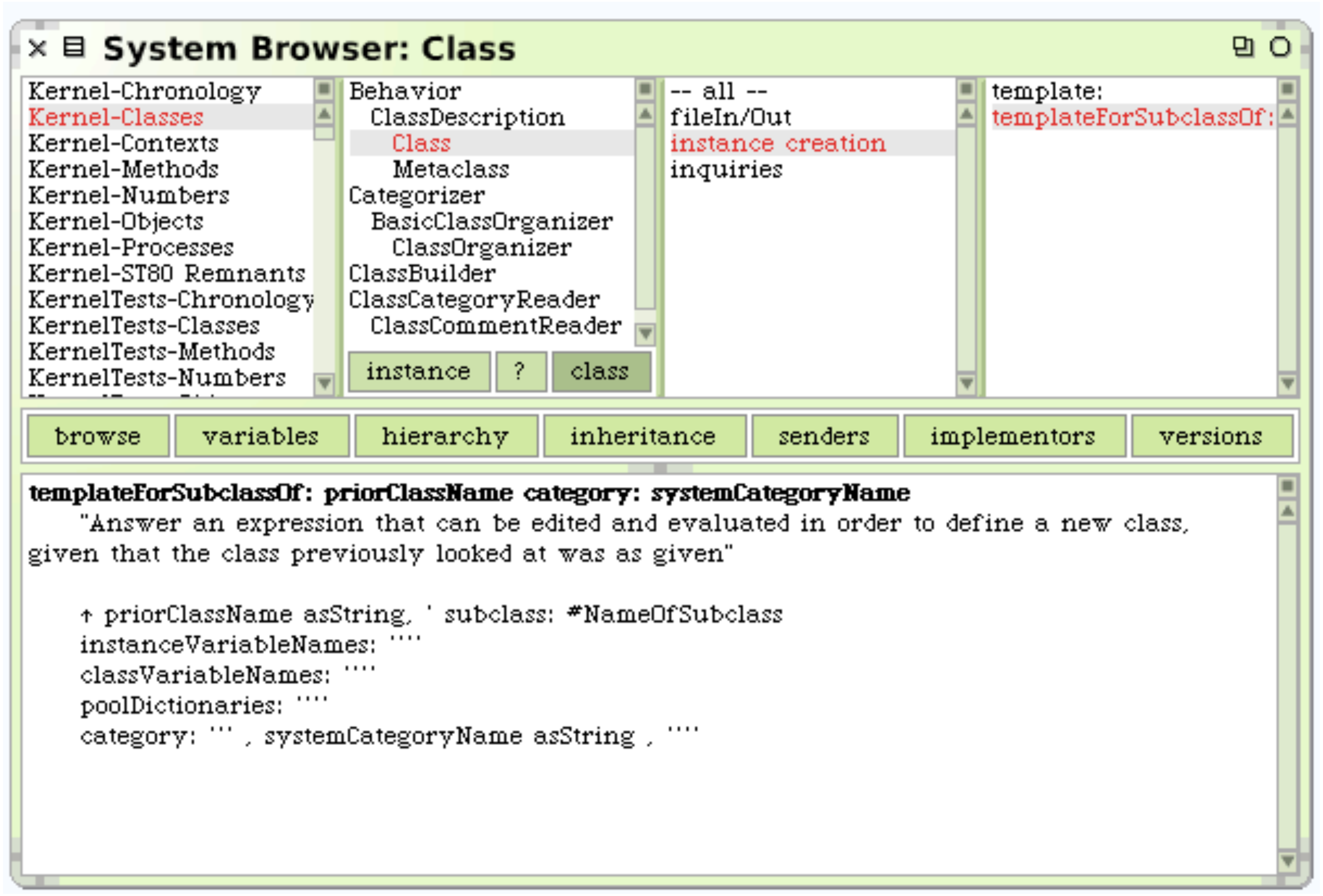
OmniBrowser - Meta-modeling Browsers

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 , ''''
```

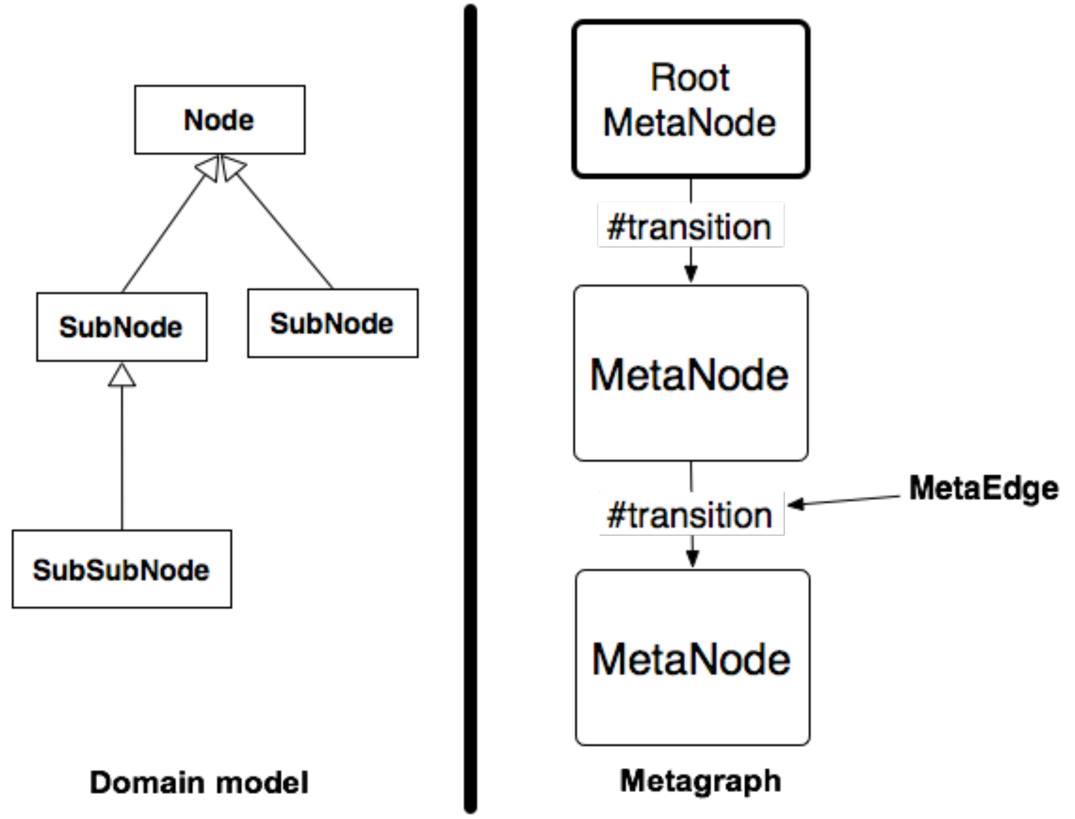
Problems of Old Browsers

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

OmniBrowser Approach

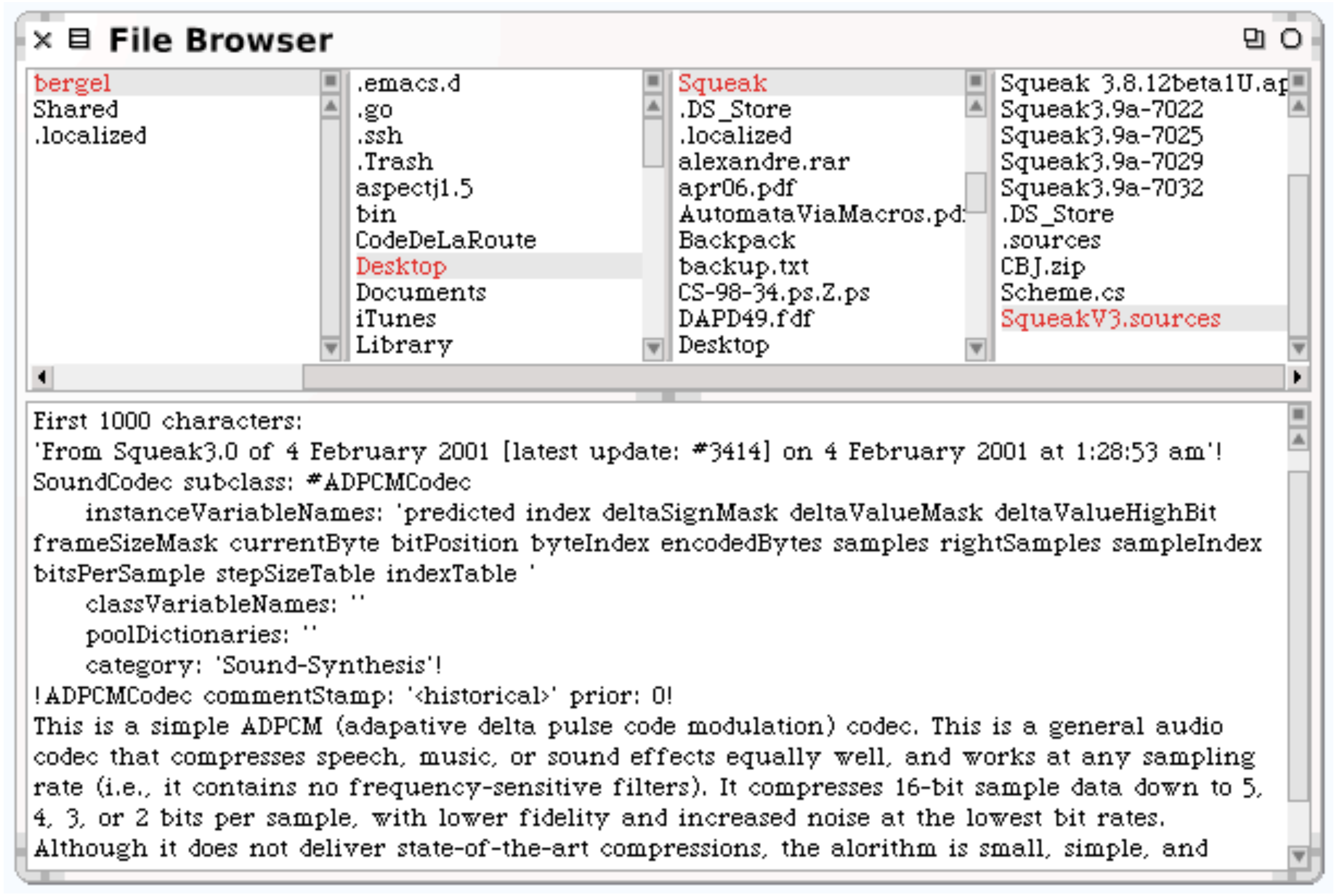
- Meta-modeling navigation in a metagraph
- Separate navigation model from domain model

OB: Meta-Modeling Navigation



Navigation modeled in metagraph, a state machine

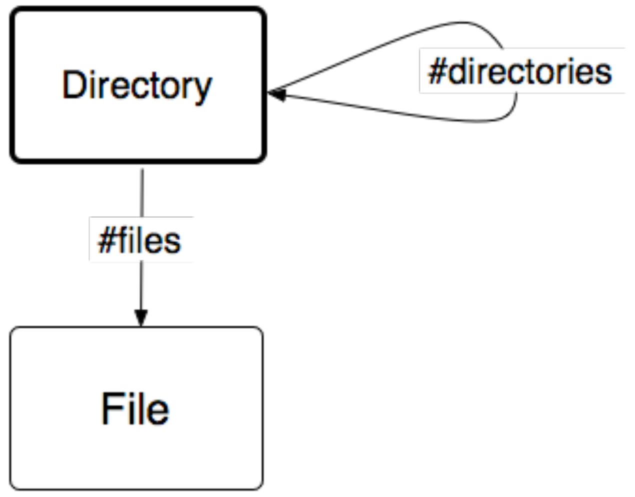
Example: File Browser



```
bergel        .emacs.d        Squeak                 Squeak 3.8.12beta1U.ap
Shared        .go             .DS_Store              Squeak3.9a-7022
.localized    .ssh            .localized             Squeak3.9a-7025
              .Trash          alexandre.rar          Squeak3.9a-7029
              aspectj1.5      apr06.pdf              Squeak3.9a-7032
              bin             AutomataViaMacros.pd   .DS_Store
              CodeDeLaRoute   Backpack               .sources
              Desktop         backup.txt             CBJ.zip
              Documents       CS-98-34.ps.Z.ps       Scheme.cs
              iTunes          DAPD49.fdf             SqueakV3.sources
              Library         Desktop
```

```
First 1000 characters:
'From Squeak3.0 of 4 February 2001 [latest update: #3414] on 4 February 2001 at 1:28:53 am'!
SoundCodec subclass: #ADPCMCodec
	instanceVariableNames: 'predicted index deltaSignMask deltaValueMask deltaValueHighBit
frameSizeMask currentByte bitPosition byteIndex encodedBytes samples rightSamples sampleIndex
bitsPerSample stepSizeTable indexTable '
	classVariableNames: ''
	poolDictionaries: ''
	category: 'Sound-Synthesis'!
!ADPCMCodec commentStamp: '<historical>' prior: 0!
This is a simple ADPCM (adapative delta pulse code modulation) codec. This is a general audio
codec that compresses speech, music, or sound effects equally well, and works at any sampling
rate (i.e., it contains no frequency-sensitive filters). It compresses 16-bit sample data down to 5,
4, 3, or 2 bits per sample, with lower fidelity and increased noise at the lowest bit rates.
Although it does not deliver state-of-the-art compressions, the alorithm is small, simple, and
```

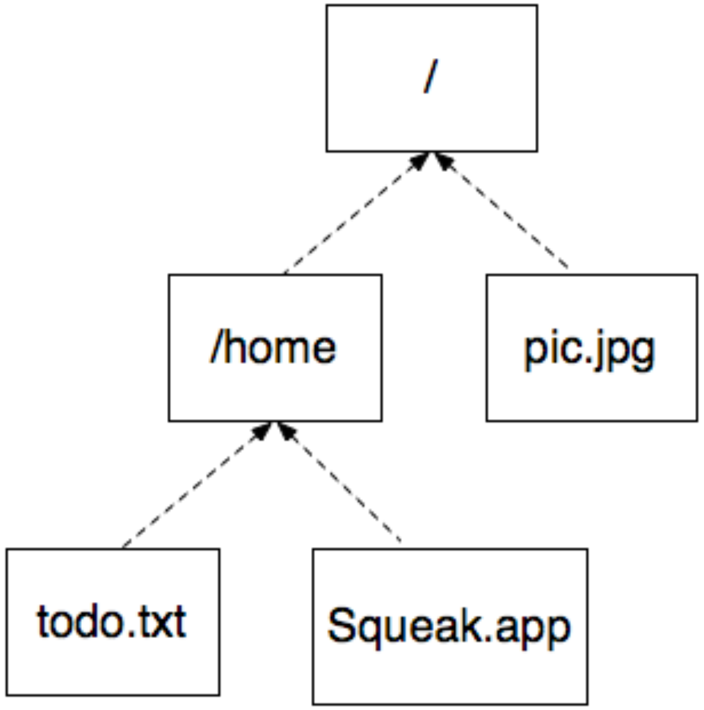
File Browser: Metagraph



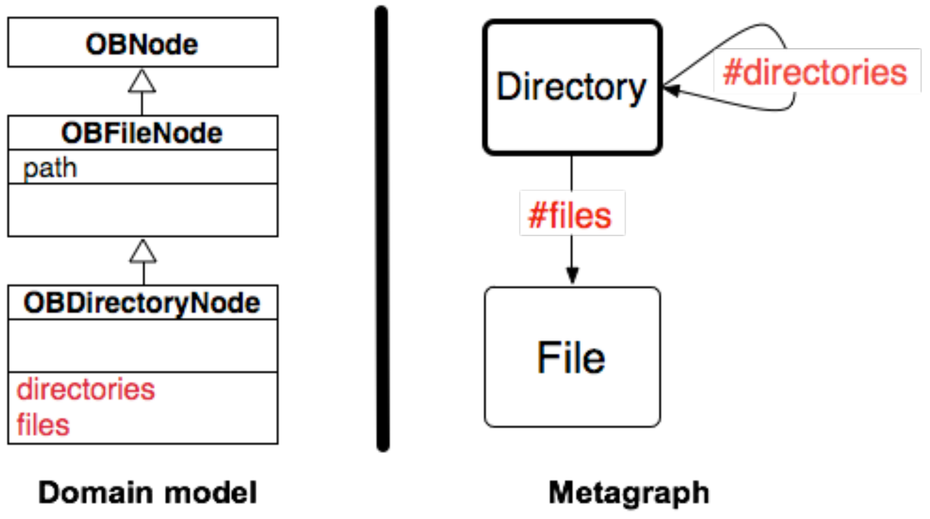
Metagraph in Code

```
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



Domain- and Meta-Model



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 := FileDirectory on: path.
^dir fileNames collect: [:each |
    OBFileNode new path: (dir fullNameFor: each)]
```

Root Node of Domain Graph

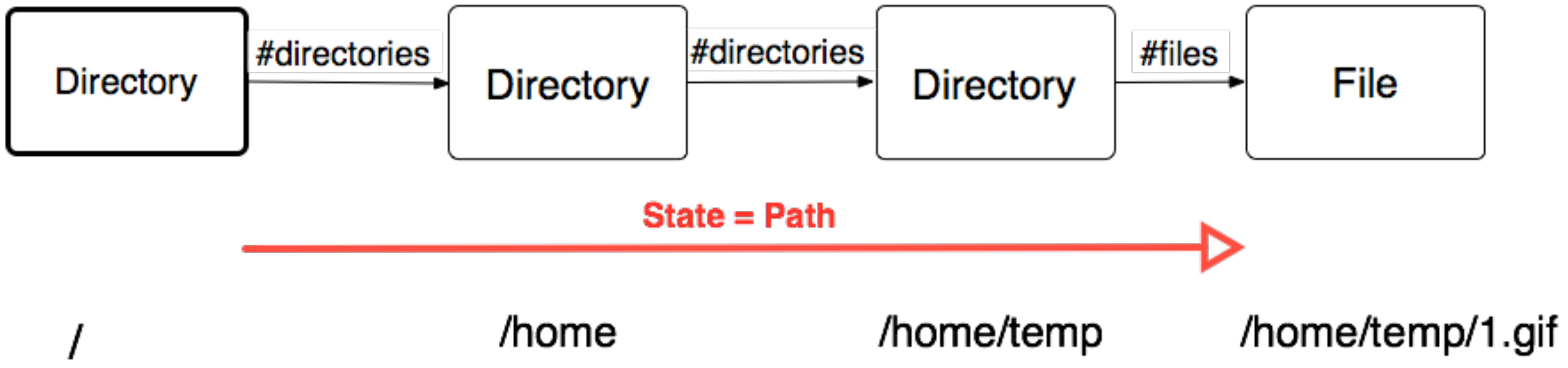
```
OBFileBrowser class >> defaultRootNode
  ^OBDirectoryNode new path: '/'
```

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

Problems Solved I



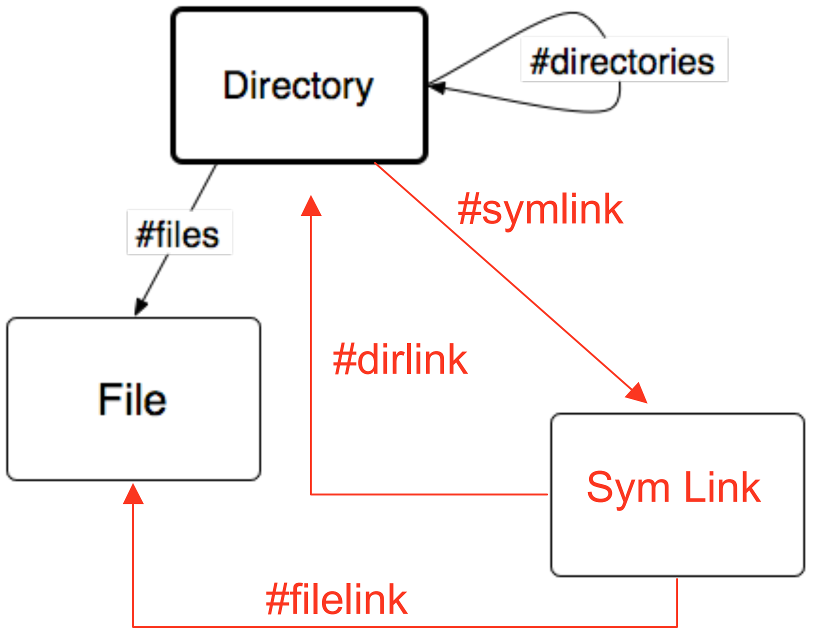
Navigation is not hard-coded, but modeled as a graph.

Problems Solved II

Metagraph easily extensible:

- different navigation
- more transitions
- more state properties

Changing Navigation Metagraph



Adding new State Properties: Auto-selection

```
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]]
```

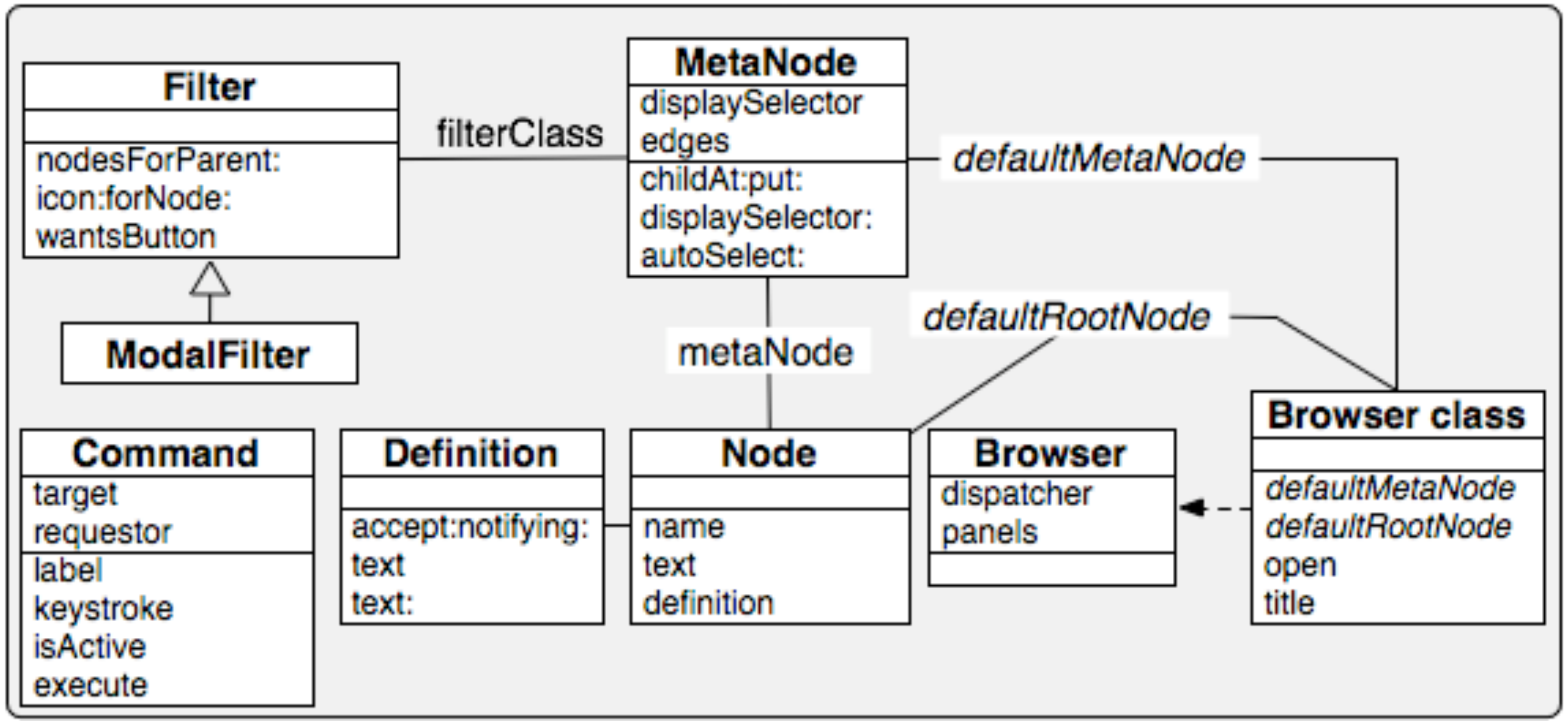
Extending Metagraph: Icons

```
methodMetaNode addFilter: OBMethodFilter new.

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



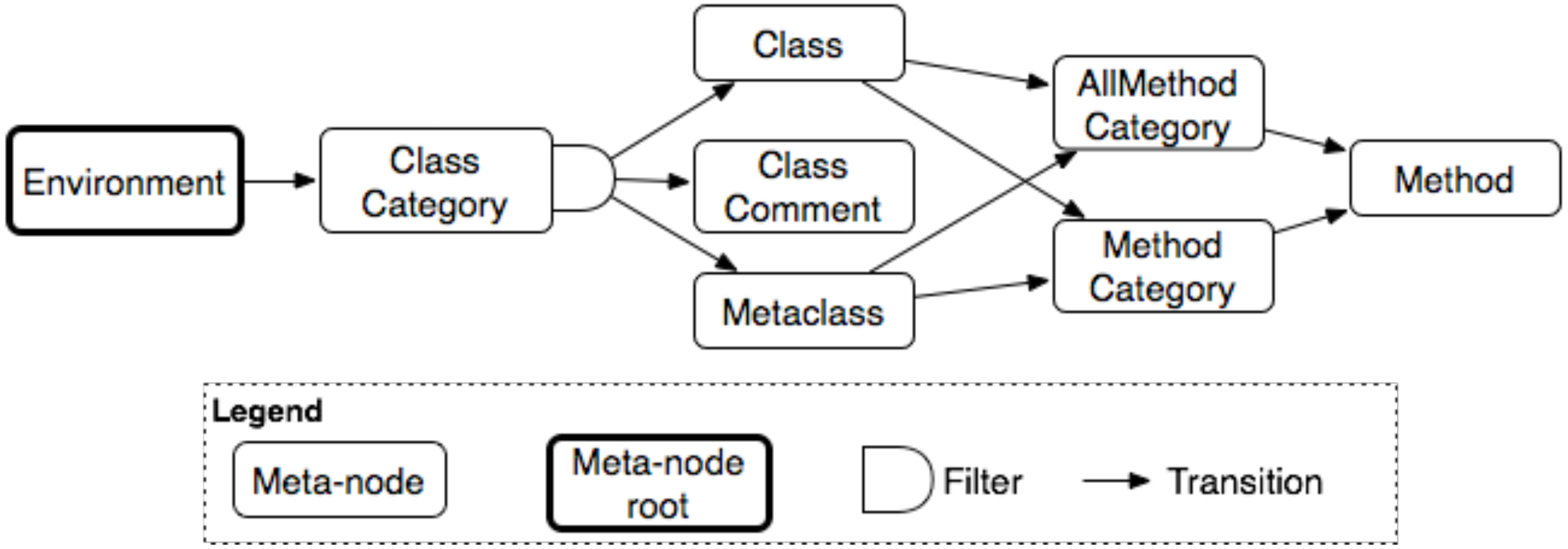
Widgets

- 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

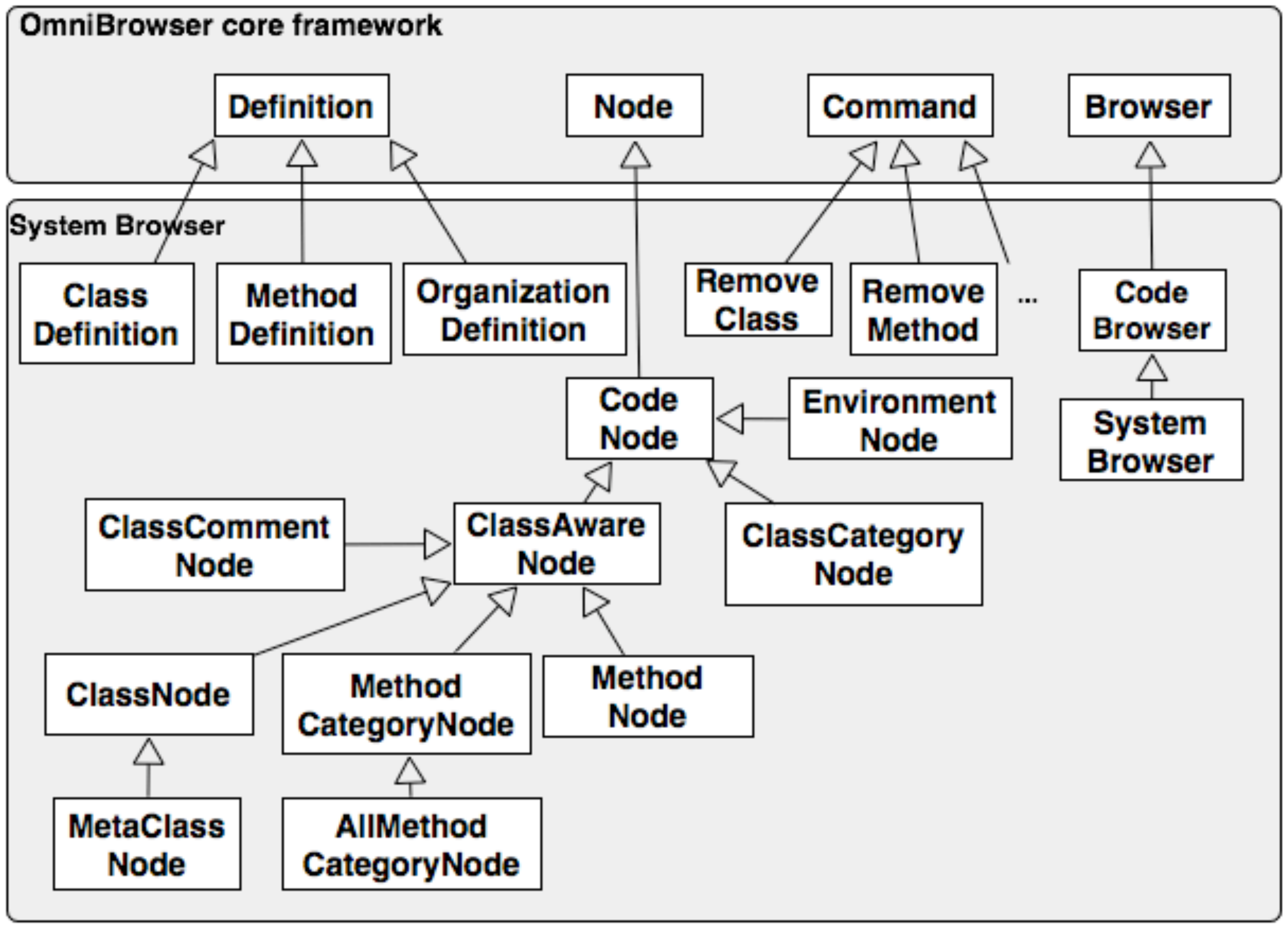
System Browser: Metagraph



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

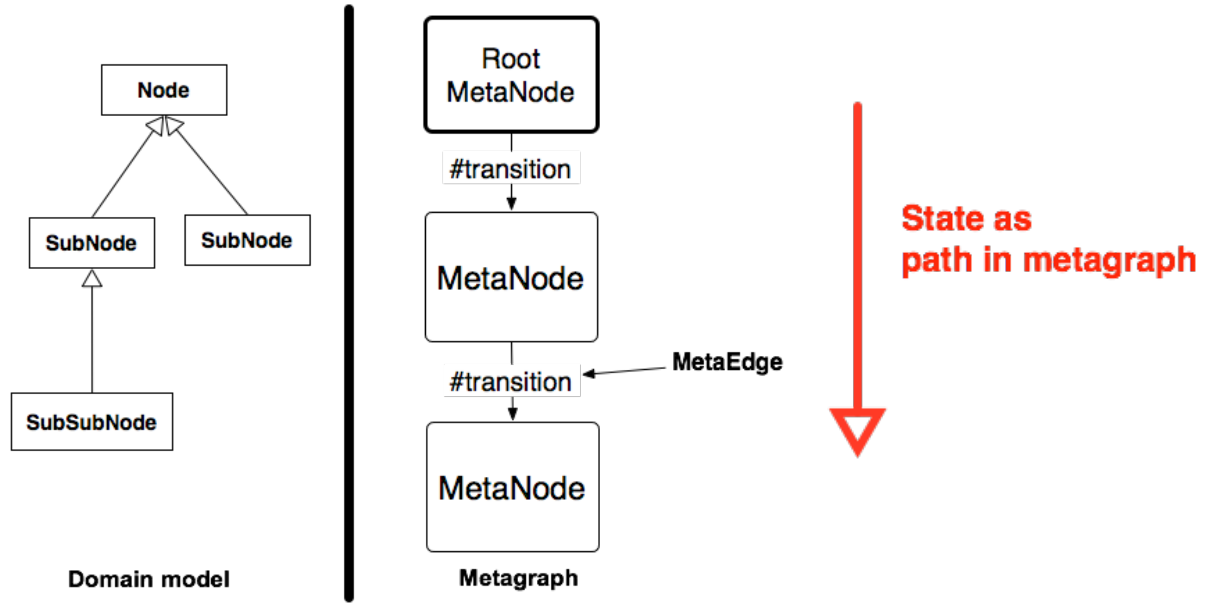
System Browser on the Web

Several GUIs

- GUIs: 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