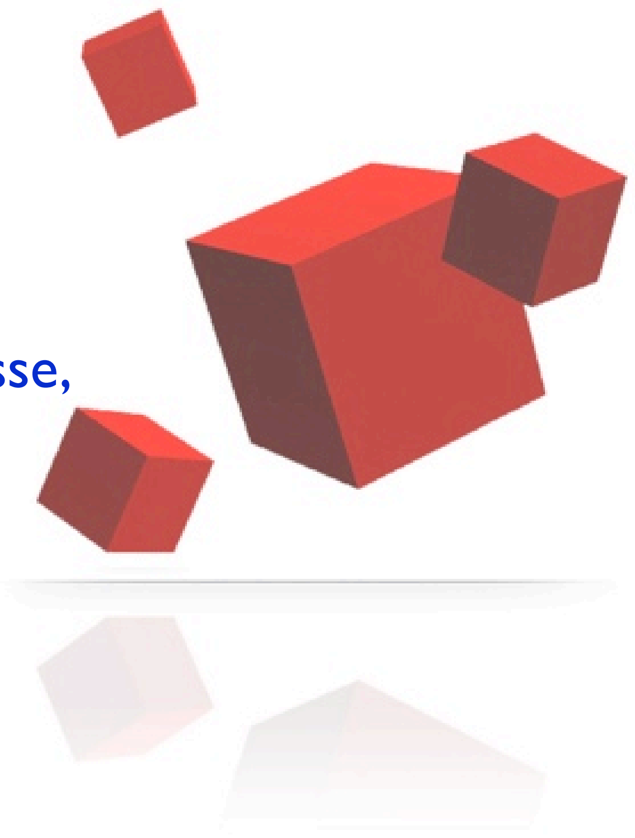
Meta-Driven Browsers



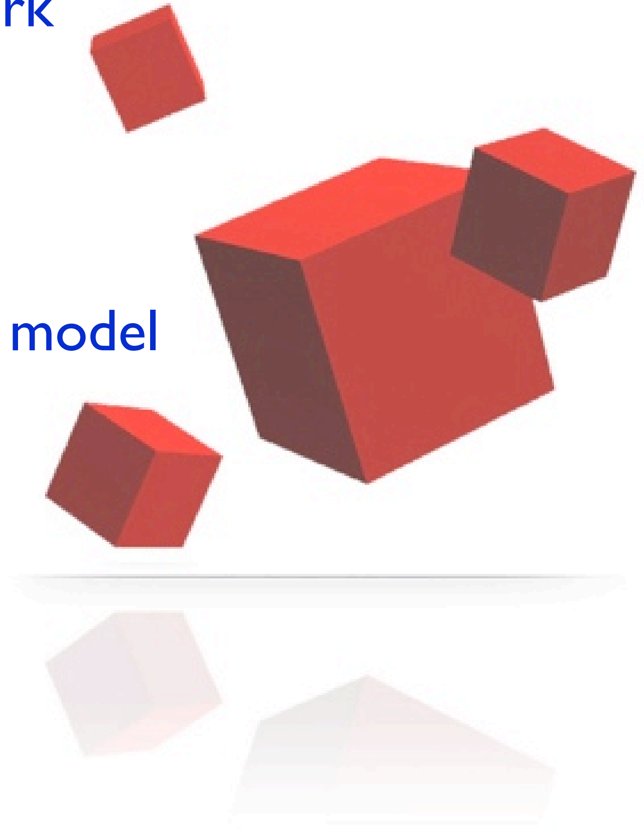
Alexandre Bergel, Stéphane Ducasse,
Colin Putney, Roel Wuyts

Text

ESUG 2006
Prague, Czech Republic

Outline

1. The OmniBrowser framework
2. Graph and metagraph
3. Interaction with the domain model
4. The System browser
5. Conclusion



The OmniBrowser Framework

- A sophisticated framework to define new browsers
- It is structured around:
  - an explicit **domain model**
  - a **metagraph** (a state machine) that specify navigation with the domain model
  - a list of **actors** that define interactions

Domain Model: Files

```
OBNode subclass: #FileNode
  instanceVariableNames: 'path'
  ...

FileNode>>name
  ^ (FileDirectory directoryEntryFor: path) name

FileNode>>text
  ^ 'File named: ', self name
```

Domain Model: Directory

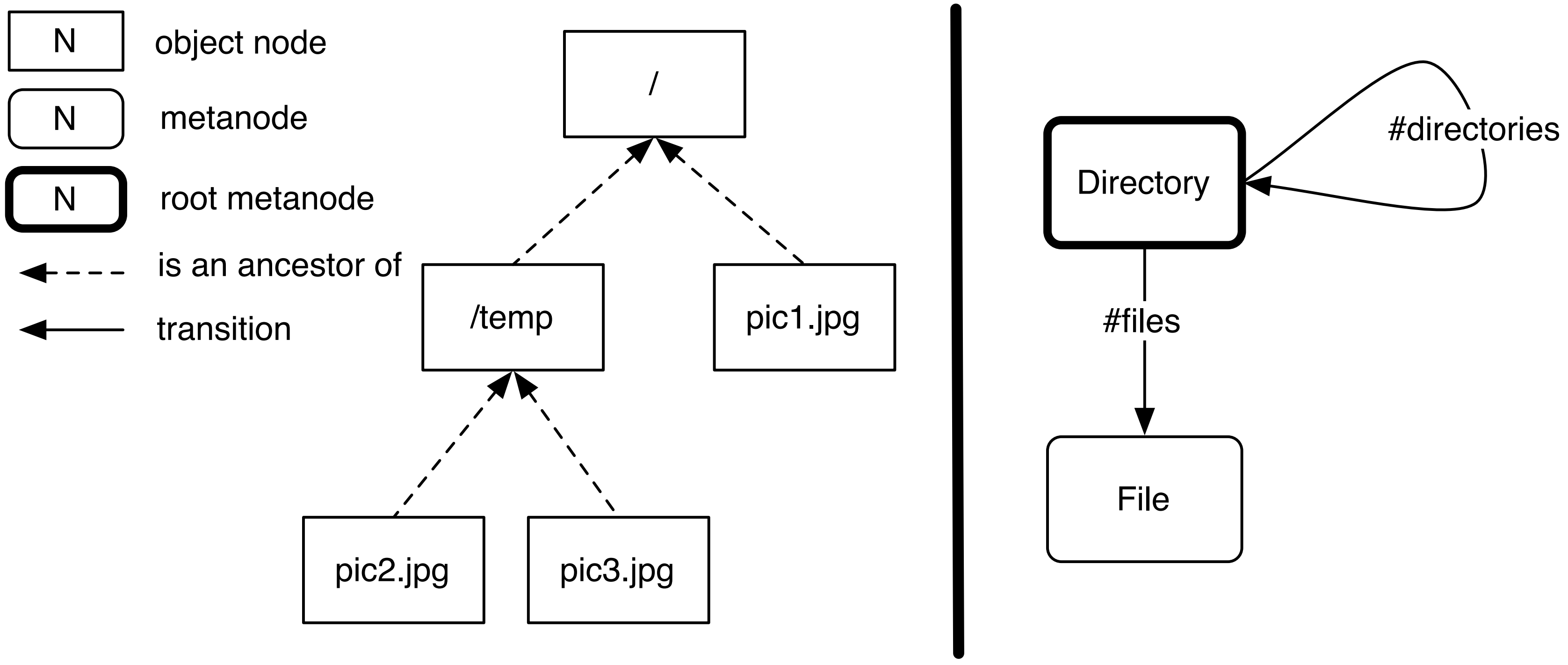
```
FileNode subclass: #DirectoryNode

DirectoryNode>>directories
  | dir |
  dir := FileDirectory on: self path.
  ^ dir directoryNames collect:
     [:each |
       DirectoryNode on: (dir fullNameFor: each)]

DirectoryNode>>files
  | dir |
  dir := FileDirectory on: self path.
  ^ dir fileNames collect: [:each |
     FileNode on: (dir fullNameFor: each)]

DirectoryNode>>text   ^ path
```

Graph and Metagraph to define browsers



(a) Instantiated domain (b) Metagraph

Metagraph and browser definition

Creation of a browser:

```
OBBrowser subclass: #FileBrowser
```

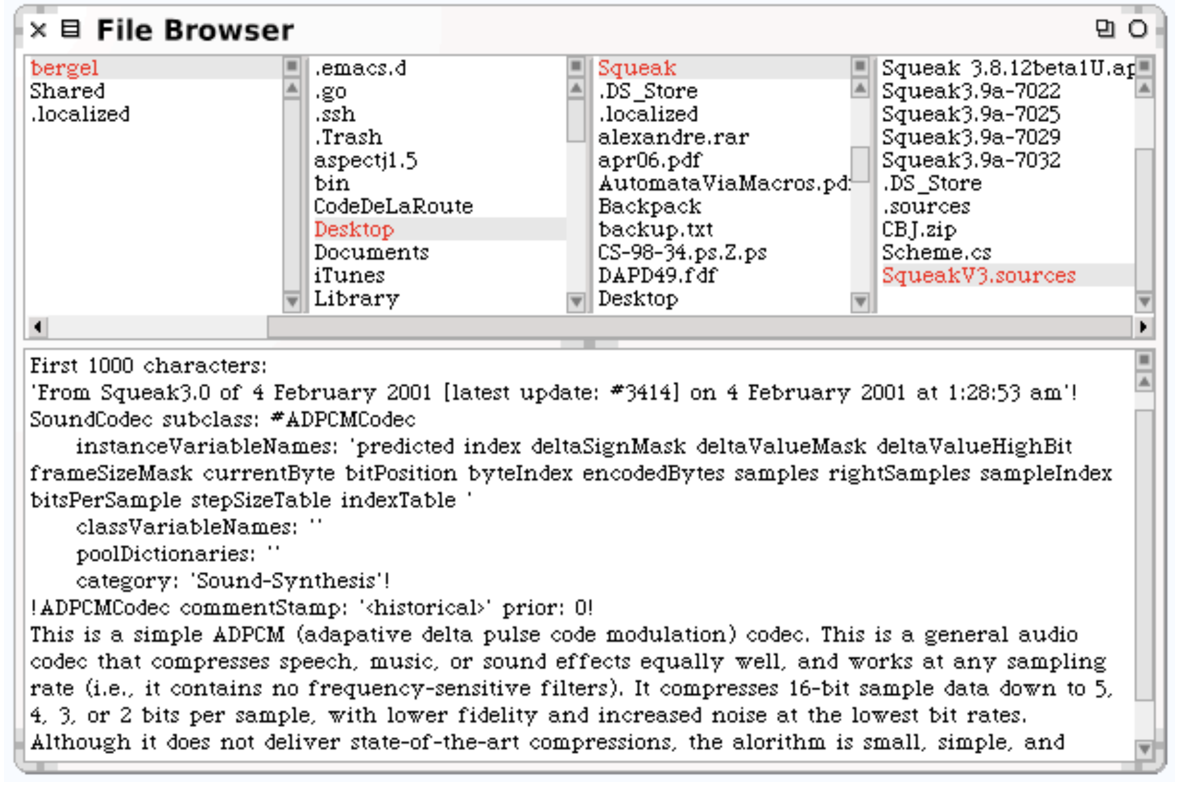
Root nodes:

```
FileBrowser>>defaultRootNode
  ^ DirectoryNode on: '/'

FileBrowser>>defaultMetaNode
  |directory file |
  directory := OBMetaNode named: 'Directory'.
  file := OBMetaNode named: 'File'.
  directory
     childAt: #directories put: directory;
     childAt: #files put: file;
     addActor: FileActor new.
  ^ directory
```

Automatic layout with columns and a pane

- The GUI is built by the framework
- It uses a layout similar to the Smalltalk System 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
```

Interacting with the domain model with actors

An actor defines a column menu:

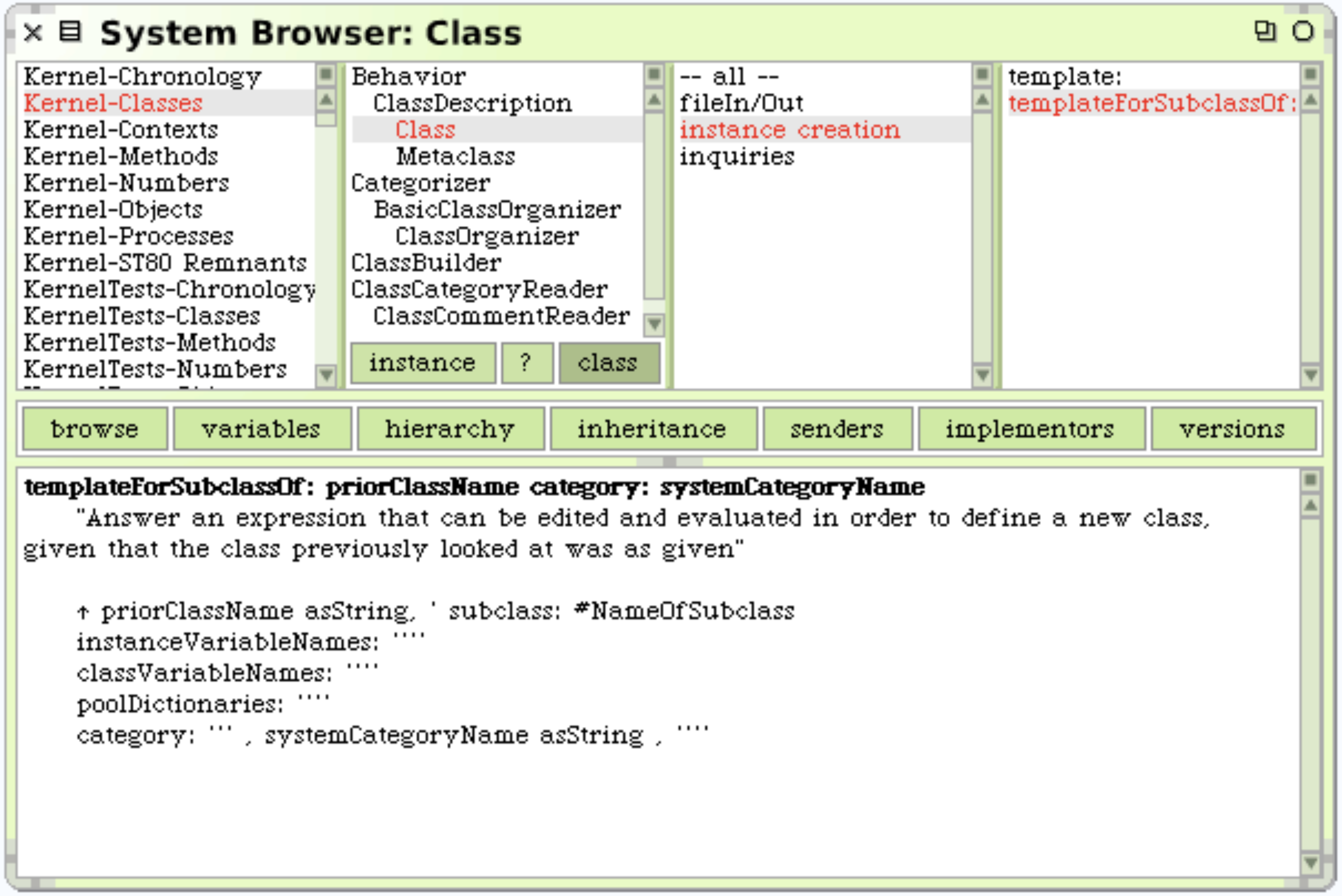
```
OBActor subclass: #FileActor

FileActor>>actionsForNode: aNode
  ^ {OBAction
       label: 'remove'
       receiver: self
       selector: #remove:
       arguments: {aNode}
       keystroke: $x
       icon: MenuIcons smallCancelIcon.
     ...}
```

Important notions of OmniBrowser

- Core notions:
  - **Nodes**: what my domain is made of?
  - **Metagraph**: how do I navigate in my domain?
  - **Actors**: how do I interact with my domain?
- Filter: filtering domain nodes
- Definition: accepting new definitions of nodes

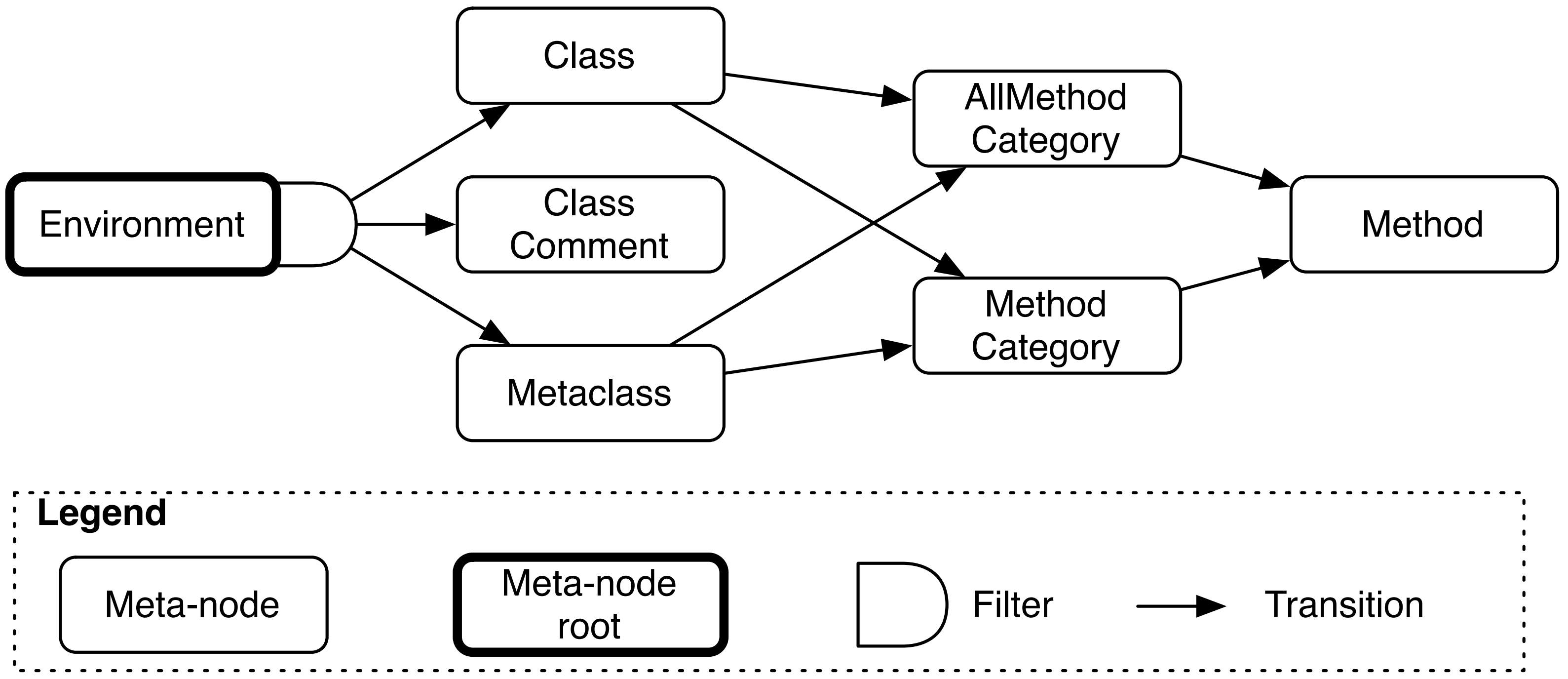
The new system browser...



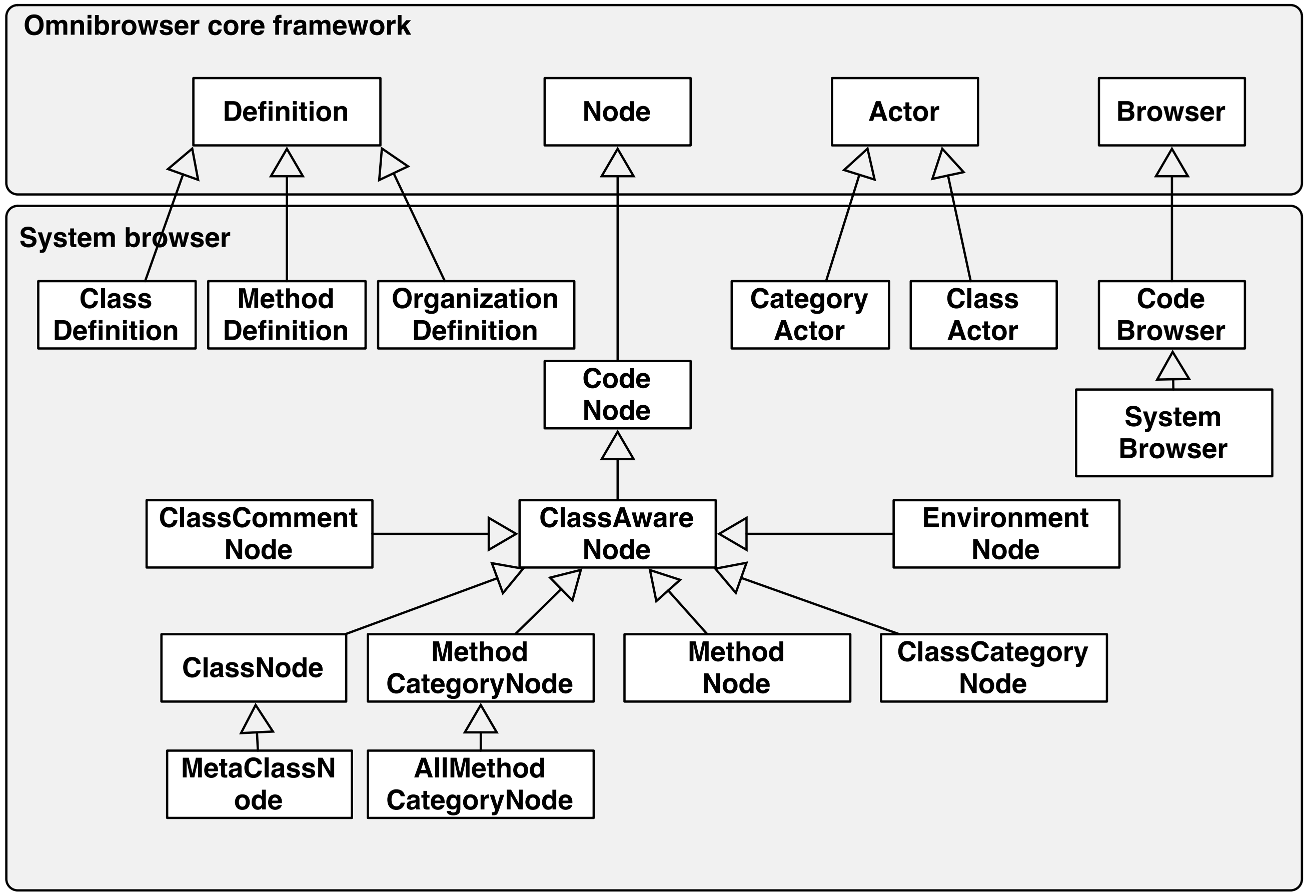
```
templateForSubclassOf: priorClassName category: systemCategoryName
    "Answer an expression that can be edited and evaluated in order to define a new class,
given that the class previously looked at was as given"

    ↑ priorClassName asString, ' subclass: #NameOfSubclass
    instanceVariableNames: ''''
    classVariableNames: ''''
    poolDictionaries: ''''
    category: ''' , systemCategoryName asString , ''''
```

... its Metagraph ...



… and its implementation



Limitations of OmniBrowser ...

- Hardcoded flow
  - Navigation has to follow the left-to-right list construction
  - Would be difficult to implement Whiskers
- Currently selected item
  - Difficulty to implement advanced browsing facilities like in VisualWorks

… and its strenghts

- Ease of use
  - do not need to deal with graphical objects

- Explicit state transition
  - graphical objects are automatically updated.

- Separation of domain and navigation
  - better readability of the code

Conclusion

- Framework to build easily new browser
- Based on notion of nodes, metagraph, actors, definition and filters
- Included per default in Squeak 3.9
- Already existing browsers:
  - changes, implementors, senders, variables, version, ...
  - coverage browser
  - dual browser
  - Traits browser
  - Pier browser