Calypsoo browser
What’s wrong with Nautilus?

- Works only on local “live” environment
  - could not browse remote image
  - could not browse not loaded code (MCSnapshotBrowser)
- Very difficult to improve
  - Full of spaghetti
    - No model behind Nautilus
    - Everything inside single class NautilusUI
Calypso navigation model

- First class scopes
  - PackageScope
  - ClassScope
  - MethodScope
- Select scope from environment

  ```small
  scope := ClySystemNavigationEnvironment currentImage
  selectScope: ClyClassScope of: {Collection. Set}.
  ```

- SystemScope represents global environment space

  ```small
  ClySystemNavigationEnvironment currentImageScope
  ```
Calypso navigation model

• First class queries
  • MessageSenders
  • ClassReferences

• Evaluate query in scope

  scope query: (ClyMessageSenders of: #(do:))

  scope query: (ClyMessageImplementors of: #(do:))
Calypso navigation model

• First class query results
  • SortedClasses
  • HierarchicallySortedClasses

• Specify what kind of result should be built by query

  scope query: (ClyMessageSenders of: #(do:) as: ClyHierarchicallySortedMethods)

• Retrieve all available items with simple query

  scope query: ClySortedMethods

  scope query: ClyHierarchicallySortedMethods
Any query returns cursor

- Cursor provides stream access to underlying query result
  - caches portion of result items
  - loads new portion on demand
  - only cached items compute properties
- In remote scenario only cursor and cached items are transferred to client
Cursor provides subsequent query API

packageCursor := ClySystemNavigationEnvironment

queryCurrentImageFor: ClySortedPackages.

classCursor := packageCursor query: ClySortedClasses from: [#Kernel asPackage].

classCursor := classesCursor query: ClyHierarchicallySortedClasses.

classCursor := classesCursor queryContentInScopeWith: 'AST-Core' asPackage.

classCursor := classesCursor queryContentInScopeWithout: 'Kernel' asPackage.

methodCursor := classCursor query: ClySortedMethods from: {Point. Array}.

methodCursor := methodCursor queryContentInNewScope: ClyClassSideScope.
Actual query result is collection of environment items

- Environment item wraps actual object and extends it with properties:
  - position in query result
  - depth in query result
  - domain specific properties
    - ClyAbstractClassTag
    - ClyOverridenMethodTag
- Environment plugins can extend properties of particular items
  - ClyInheritanceAnalyzerEnvironmentPlugin marks abstract classes with ClyAbstractClassTag
- Properties are computed lazily for portion of requested items
  - it can be slow to retrieve them
Browser navigation views

• All navigation views are based on fast table
• Special data source based on cursor
• Only visible part of table are requested from cursor
  • which is covered by cursor cache
• Data source describes tree structure
  • what query to use to expand each kind of item
Browser navigation views

cursor := ClySystemNavigationEnvironment
    queryCurrentImageFor: ClySortedPackages.
dataSource := ClyCollapsedDataSource on: cursor.
dataSource childrenStructure: {
    ClyPackageScope -> ClySortedClassGroups.
    ClyClassGroupScope -> ClyHierarchicallySortedClasses}.
table := FTTableMorph new.
table
    extent: 200 @ 400;
dataSource: dataSource;
openInWindow.
Browser navigation state

- Most state incapsulated inside navigation data sources:
  - what kind of query it represents
    - method group mode or variable mode
  - in what scope result are shown
    - instance side or class side
  - what kind of result are used
    - flat methods or hierarchical methods
Browser navigation state

• First class selection represent selected objects
  • ClyDataSourceSelection represents manually selected items
  • ClyDataSourceHighlighting represents highlighted items
    • to highlight groups of selected methods
  • ClyDesiredSelection responsible to restore selected items when data source is changed
    • automatic selection of same method when new class selected
• Selections maintain correct visible state after any changes
Browser navigation state

• No special flags like in old browser
How extend browser

- Commands
- Table decoration
- Browser tabs
- Method groups
- Class groups
Commander

• Calypso commands are based on Commander

• Every command is first class object subclass of CmdCommand with execute method

• Multiple activators can be attached to command class to declare specific way to access command from particular application context
  
  • CmdContextMenuCommandActivator
  
  • CmdShortcutCommandActivator
  
  • CmdDragAndDropCommandActivator
  
  • ClyToolbarCommandActivator (for elements from browser toolbar)
  
  • ClyTableIconCommandActivator (to support method icons of Nautilus)

Command example

```
RenamePackageCommand>>execute
    package renameTo: newName

RenamePackageCommand class>>systemBrowserShortcutActivator
    <commandActivator>
    ^CmdShortcutCommandActivator by: $r meta for: ClyPackageSystemBrowserContext

RenamePackageCommand class>>systemBrowserMenuActivator
    <commandActivator>
    ^CmdContextMenuCommandActivator byRootGroupItemFor: ClyPackageSystemBrowserContext

RenamePackageCommand>>defaultMenuItemName
    ^'Rename'
```
Command example

RenamePackageCommand>>prepareFullExecutionInContext: aToolContext
    super prepareFullExecutionInContext: aToolContext.
    package := aToolContext lastSelectedPackage.
    newName := UIManager default
        request: 'New name of the package'
        initialAnswer: package name
        title: 'Rename a package'.
    newName isEmptyOrNil | (newName = package name)
        ifTrue: [ ^ CmdCommandAborted signal ]

RenamePackageCommand>>canBeExecutedInContext: aToolContext
    ^aToolContext isPackageSelected
Drag and drop example

**MoveClassToPackageCommand>>execute**

classes do: [:each | package addClass: each]

**MoveClassToPackageCommand class>>systemBrowserDragAndDropActivator**

<commandActivator>

^CmdDragAndDropCommandActivator

  for: ClyClassSystemBrowserContext toDropIn: ClyPackageSystemBrowserContext

**MoveClassToPackageCommand>>prepareExecutionInDragContext: aToolContext**

super prepareExecutionInDragContext: aToolContext.

classes := aToolContext selectedClasses

**MoveClassToPackageCommand>>prepareExecutionInDropContext: aToolContext**

super prepareExecutionInDropContext: aToolContext.

package := aToolContext lastSelectedPackage
Table decoration

- ClyTableDecorator subclasses with class side methods

```smalltalk
ClyAbstractClassTableDecorator class>>browserContextClass
    ^ClyClassSystemBrowserContext

ClyAbstractClassTableDecorator class>>wantsDecorateTableCellOf: aDataSourceItem
    ^aDataSourceItem isMarkedWith: ClyAbstractClassTag

ClyAbstractClassTableDecorator class>>decorateTableCell: anItemCellMorph of: aDataSourceItem
    | nameMorph |
    nameMorph := anItemCellMorph nameMorph.
    nameMorph emphasis: TextEmphasis italic emphasisCode.
    nameMorph color: nameMorph color contrastingColorAdjustment contrastingColorAdjustment
```
Table decoration

• Commands with ClyTableIconCommandActivator

```smalltalk
RunTestMethodCommand class >> systemBrowserTableIconActivator
  <commandActivator>
  ^ ClyTableIconCommandActivator for: ClyMethodSystemBrowserContext

RunTestMethodCommand >> buildTableCellIconFor: anItemCellMorph
  testMethod isPassedTest ifTrue: [^ anItemCellMorph iconNamed: # testGreenIcon].
  testMethod isErrorTest ifTrue: [^ anItemCellMorph iconNamed: # testRedIcon].
  testMethod isFailedTest ifTrue: [^ anItemCellMorph iconNamed: # testYellowIcon].
  ^ anItemCellMorph iconNamed: # testNotRunIcon
```
Browser tabs

- Calypso provides multiple tools instead of single source code view
  - ClyClassCreationTool
  - ClyClassCommentEditorTool
  - ClyClassDefinitionEditorTool
  - ClyMethodCreationTool
  - ClyMethodCodeEditorTool
- Tools are managed by tabs
- Tools are built in background due to nice TabsManager package
Browser tabs

- Tools are subclasses of ClyBrowserTool with `#build` method and activation context where they should be applied

```smalltalk
ClyMethodDiffTool>>build
diffMorph
  contextClass: self rightMethod sourceCode;
hResizing: #spaceFill;
vResizing: #spaceFill;
showOptions: false.
self addMorph: diffMorph fullFrame: LayoutFrame identity

ClyMethodDiffTool class>>activationContextClass
  ^ClyMethodSystemBrowserContext

ClyMethodDiffTool class>>shouldBeActivatedInContext: aBrowserContext
  ^aBrowserContext selectedMethods size > 1
```
Demo for more tools
Method groups

- Third panel in browser shows method groups
  - TaggedMethodGroups based on method tags (protocols)
  - NoTagMethodGroup for unclassified methods
  - ExtendedMethodGroup for all class extensions
  - InheritedMethodGroup to toggle all inherited methods visibility
  - SuperclassMethodGroup to toggle concrete superclass methods visibility
  - ExternalPackageGroup for concrete package extending class
  - VariableMethodGroup for variables mode of browser
  - more
Method groups

- Method groups are built by method group providers
- Group providers are extended by environment plugins

```smalltalk
ClyInheritanceAnalyzerEnvironmentPlugin>>collectMethodGroupProvidersFor: classes
      ClyOverriddenMethodGroupProvider. ClyRequiredMethodGroupProvider]
    collect: [ :each | each classes: classes]

ClyAbstractMethodGroupProvider>>buildGroupItemsOn: items
    (classes anySatisfy: [ :each | each hasAbstractMethods ])
    ifFalse: [ ^self ].
    items add: (ClyAbstractMethodGroup classes: classes) asEnvironmentItem

ClyAbstractMethodGroup>>includesMethod: aMethod
    ^aMethod sendsSelector: #subclassResponsibility
```
Demo for method groups
Class groups

- First panel in browser shows packages expanded to class groups
  - TaggedClassGroup based of class tags (RPackageTag)
  - NoTagClassGroup for unclassified classes
  - ExtendedClassGroup for classes extended by current package
Class groups

• Class groups are built by class group providers

• Group providers are extended by environment plugins

```smalltalk
ClySystemEnvironmentPlugin>>collectClassGroupProvidersFor: aPackage
  ^{ClyExtendedClassGroupProvider. ClyTaggedClassGroupProvider}
  collect: [ :each | each package: aPackage ]

ClyExtendedClassGroupProvider>>buildGroupItemsOn: items
  package extendedClassNames ifEmpty: [ ^self ].
  items add: (ClyExtendedClassGroup in: package) asEnvironmentItem
```
What is missing

- Navigation history
  - go back and forward in browser
  - should works for tabs selection too
- Some commands and refactoring
  - maybe we do not everything from Nautilus
- report at https://github.com/dionisiydk/Calypso/issues
The end