

Spec

A Framework for the Specification and Reuse of
UIs and their Models

Benjamin Van Ryseghem - INRIA

Stéphane Ducasse - INRIA

Johan Fabry - Universidad de Chile

- I - A new framework

- II - Specification

- III - Reuse

Goals of a UI Builder

- Managing events
- Describe user interfaces
- **Reuse previously described widgets**
- (Have a UI for placing sub elements)

classSymbolFor:

```

AbstractBindings>>#classSymbolFor:is:
AbstractBindings>>#initialize
AbstractBindings>>#initializeClassesDictionary
AbstractBindings>>#initializeSelectorsDictionary
AbstractBindings>>#selectorSymbolFor:is:
AbstractBindings>>#classSymbolFor:
AbstractBindings>>#selectorSymbolFor:
AbstractBindings>>#selectorSymbolFor:ifAbsent:
AbstractBindings>>#classSymbolFor:ifAbsent:

```

classSymbolFor: aSymbol

^ classSymbols at: aSymbol

AbstractBindings>> #classSymbolFor:

```

AColorSelectorMorph
AGroupHasBeenAdded
AGroupHasBeenCreated
AGroupHasBeenRegistered
AGroupHasBeenRemoved
AGroupHasBeenRenamed
AGroupHasBeenUnregistered
Abort
AboutDialogWindow
AbsolutePath
AbstractBindings
AbstractCategoryWidget
AbstractDescription

```

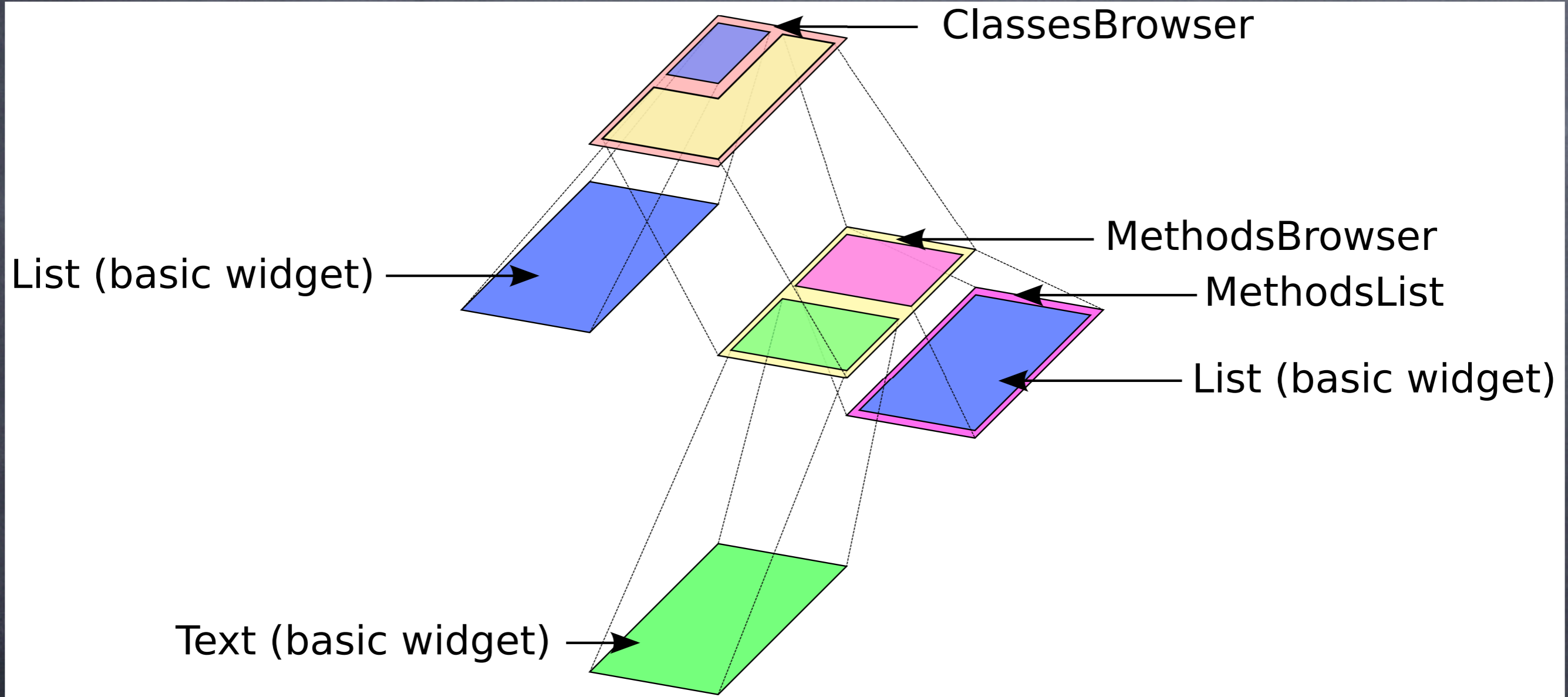
```

^ classSymbolFor:is:
initialize
initializeClassesDictionary
initializeSelectorsDictionary
selectorSymbolFor:is:
classSymbolFor:
selectorSymbolFor:
selectorSymbolFor:ifAbsent:
classSymbolFor:ifAbsent:

```

classSymbolFor: aSymbol

^ classSymbols at: aSymbol



I - A new framework

- On top of Morphic/PolyMorph
- (Pluggable on top of everything)

II – Specification

- UI Layout
- Widgets
- Presenter

UI Layout

- Declarative representation
- Multiple representations possible
- Simple layouts available

Widgets

- Declarative way to initialize sub widgets

```
self instantiateModels: #(
  userLabel      LabelModel
  userField      TextInputFieldModel
  gmail          LabelModel
  passwordLabel  LabelModel
  passwordField  TextInputFieldModel ).
```

- Precise changed: messages

Presenter

- The behavior of the UI itself
- A central point to link sub widgets
- Event based interaction

Widget interface

- Look at 'protocol*' protocols
- Explicitly documented API

Window title: ListComposableModel>>#displayBlock:

<ul style="list-style-type: none"> Spec-Layout Spec-Tests Spec-Tools Spec-Tools-ChangeSorter Spec-Tools-Editor Spec-Tools-Monticello Spec-Tools-Senders Spec-Tools-VersionBrowser Spec-Widgets Spec-Widgets-PolyWidgets Spec-Widgets-Utilities StartupPreferences System System-Announcements System-Applications System-Change Notification 	<ul style="list-style-type: none"> AbstractBasicWidget ButtonModel CheckBoxModel DropListModel LabelModel ListComposableModel IconListModel MultiColumnListModel RadioButtonModel SliderModel TextModel TextInputFieldModel TreeModel WindowModel DialogWindowModel 	<ul style="list-style-type: none"> --- all --- focus initialize private private-morphic protocol protocol-events 	<ul style="list-style-type: none"> allowToSelect allowToSelect: beMultipleSelection beSingleSelection displayBlock displayBlock: filteringBlock filteringBlock: items: listItems listSize menu menu: multiSelection multiSelection: resetFilteringBlock resetSelection resetSortingBlock selectedIndex
---	---	--	--

Navigation: Groups | Hierarchy | **Class** | Comments

Search: ListComposableModel >>#displayBlock:

displayBlock: aBlock
 <api: #block getter:#displayBlock registration: #whenDisplayBlockChanged:>
 "Set the one argument block used to wrap your domain specific items.
 The block should return something that can be displayed in a list - like a String or a Text"

wrapBlockHolder contents: aBlock

III – Reuse

- Widgets are generic
- Widgets are parametrizable
- Widgets can be reused

Tool	Pharo 1.4	Pharo 2.0	Percentage of reduction
MessageBrowser	MessageSet 488	MessageBrowser 329	33%
Senders/Implementers	FlatMessageListBrowser 463	MessageBrowser + 4	99%
ProtocolBrowser	ProtocolBrowser 49	MessageBrowser + 20	47%
VersionBrowser	VersionsBrowser 312	New VersionBrowser 57	82%
ChangeSorter	ChangeSorter (970) + DualChangeSorter (39) 1009	ChangeSorterApplication (410) + DualChangeSorterApplication (186) 596	41%
Total	2321	1006	57%

Reuse in Pharo 2.0

Next Steps

- Dynamic layouts
- More layouts
- Better documentation
- UI Builder
- Test framework ?

Conclusions

- Managing events: precise update mechanism
- Reuse previously described widgets
- Included in Pharo 2.0 as standard