Spec

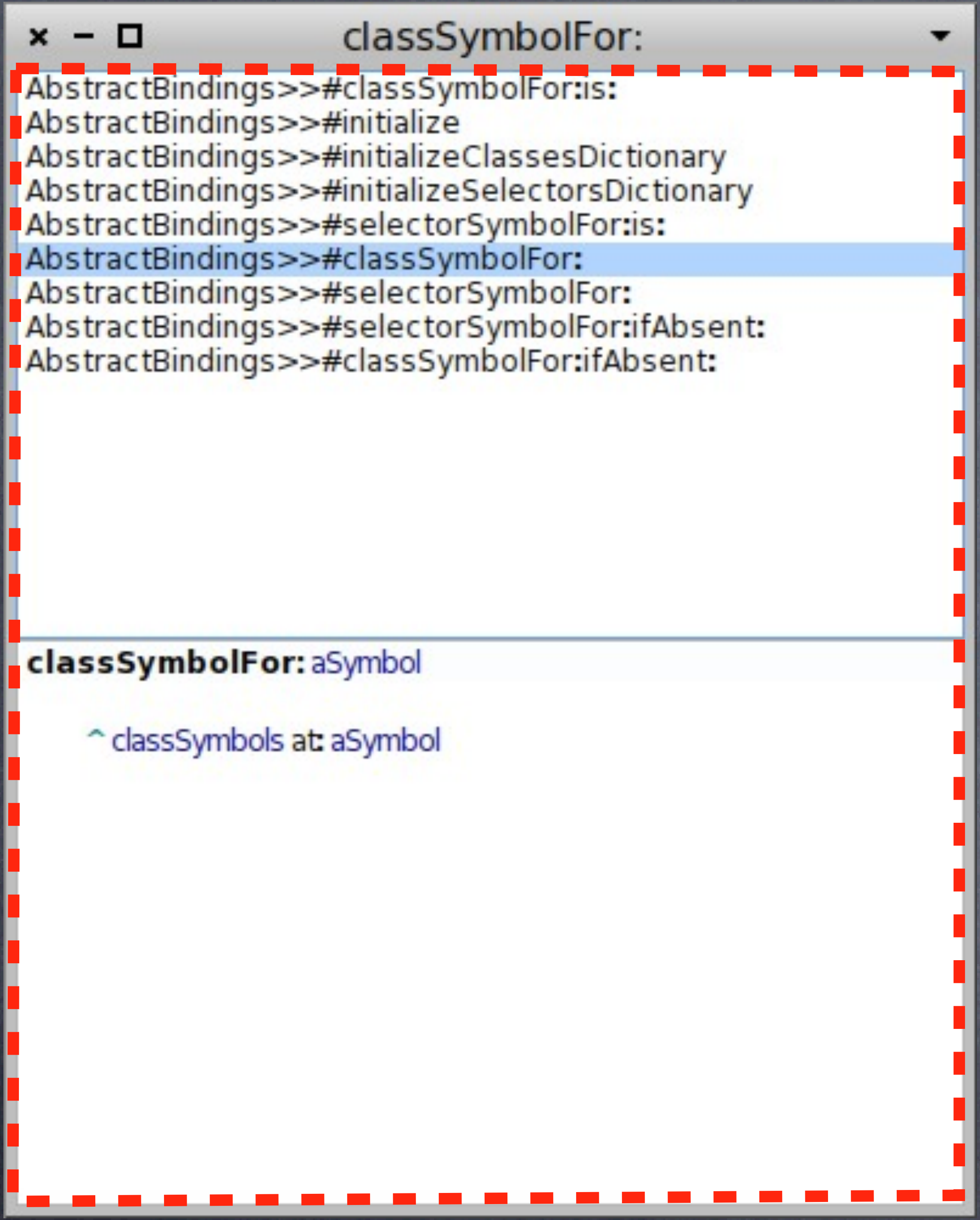
A Framework for the Specification and Reuse of UIs and their Models

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)

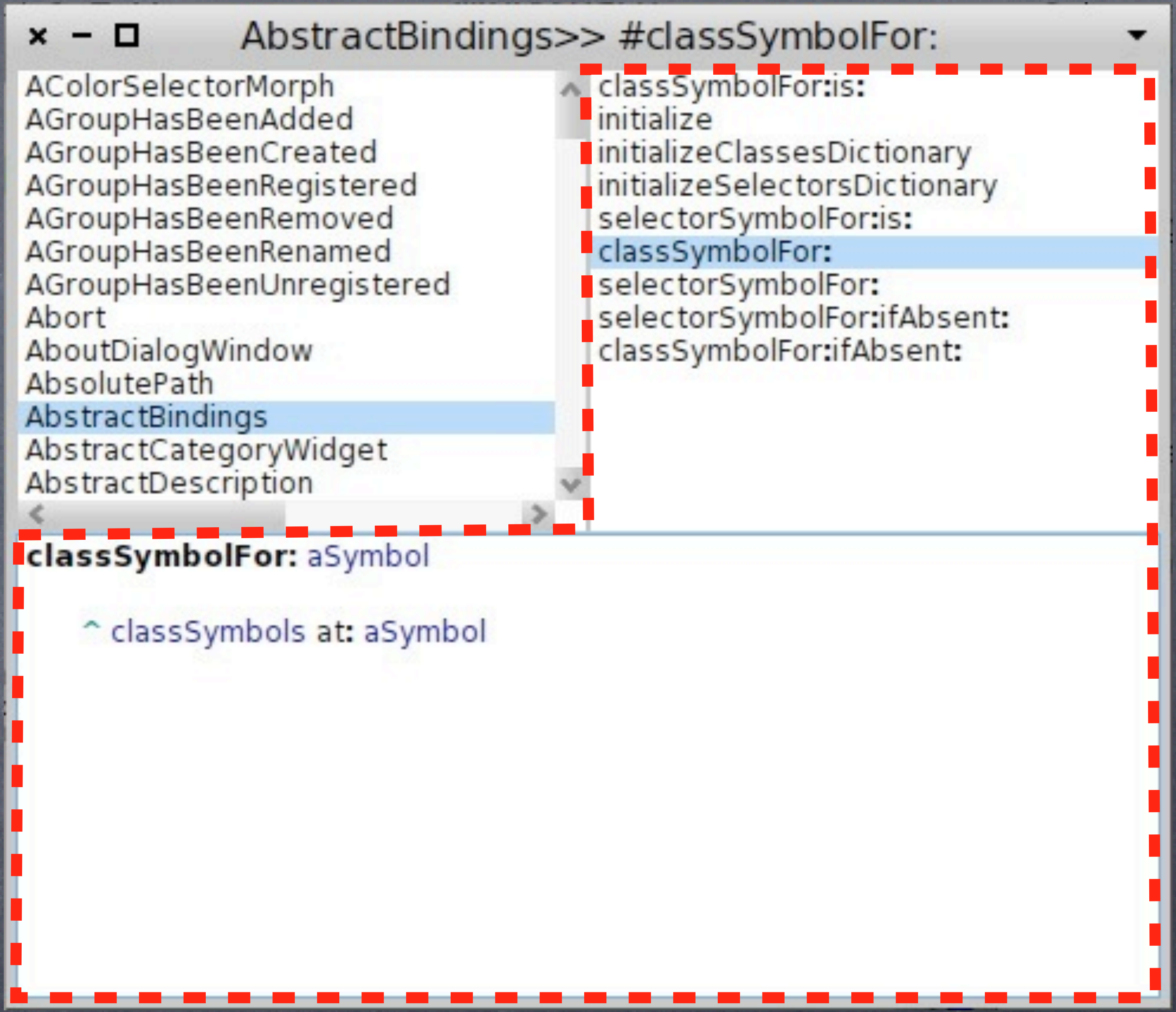


classSymbolFor:

```
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
```



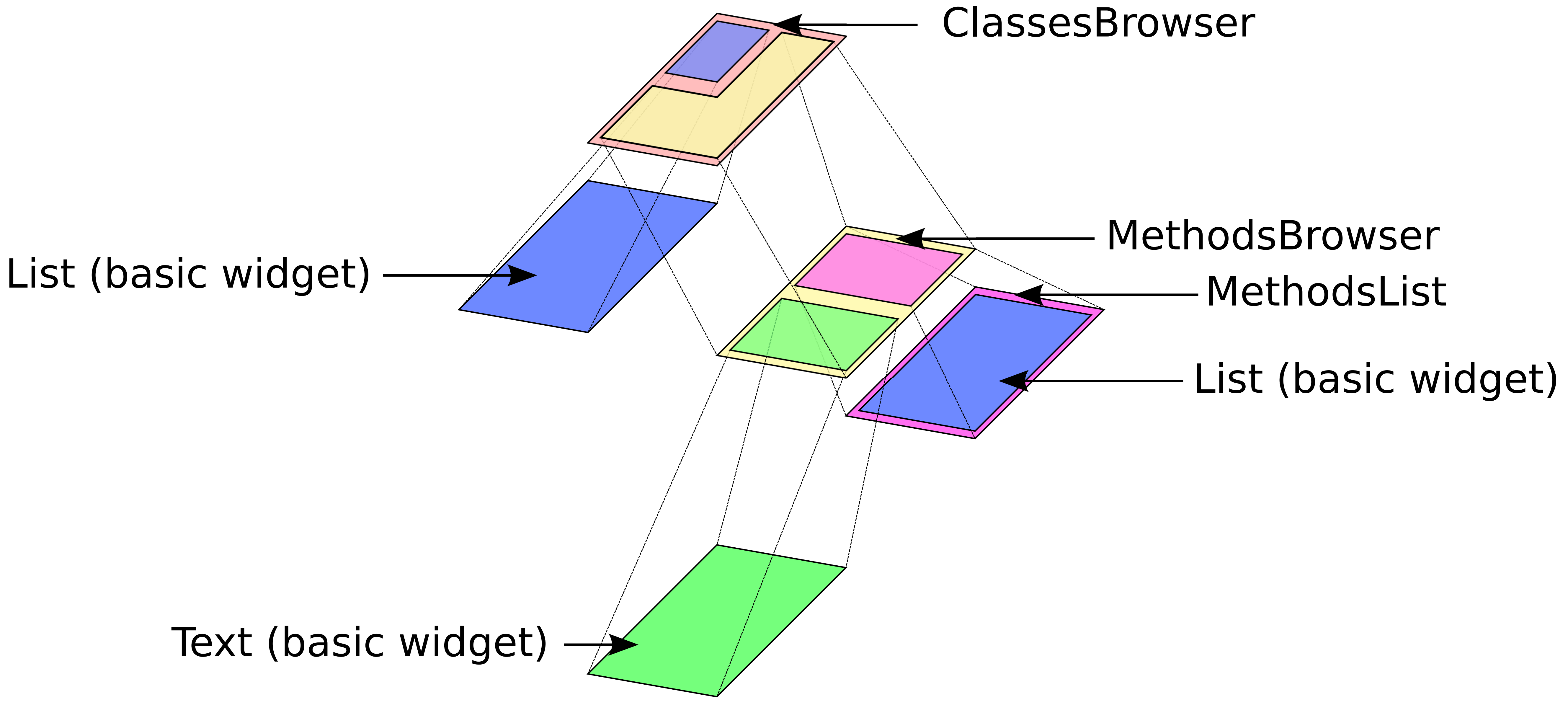
AbstractBindings>> #classSymbolFor:

```
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
```


ClassesBrowser
MethodsBrowser
List (basic widget)
MethodsList
List (basic widget)
Text (basic widget)

I - A new framework

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

II - Specification

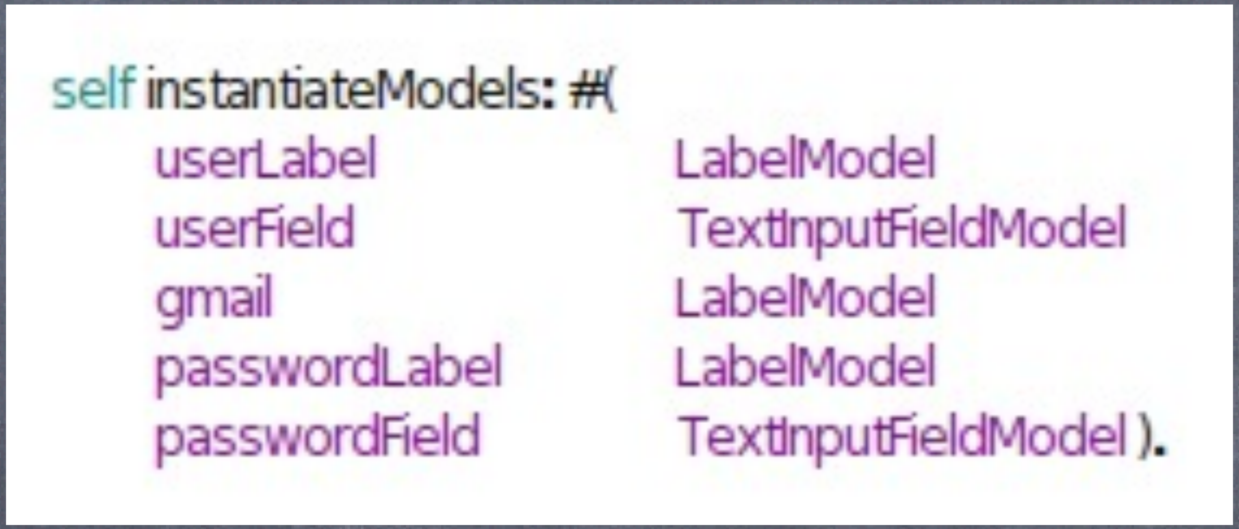
- UI Layout
- Widgets
- Presenter

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 ).
```

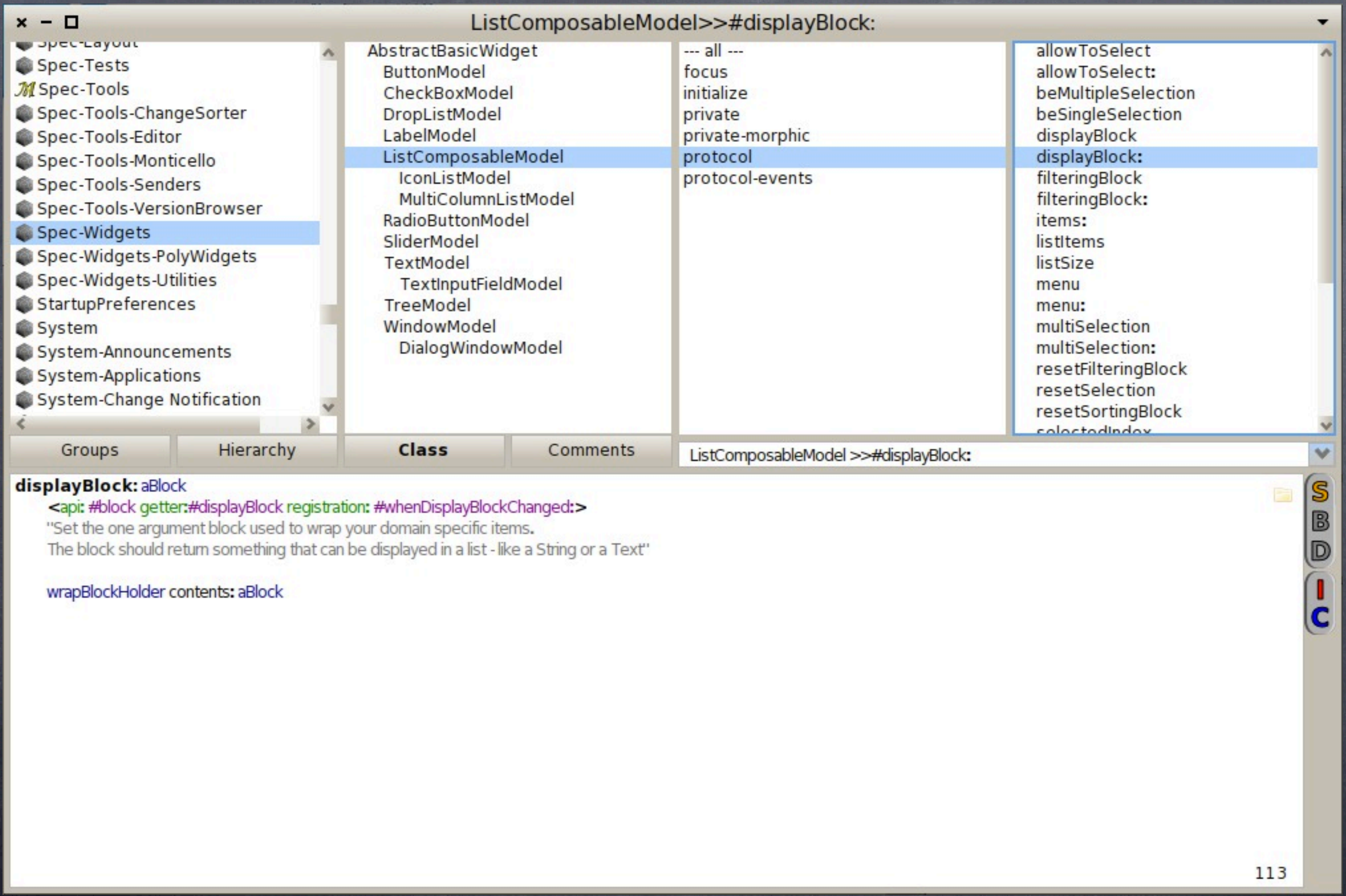
- Precise changed: messages

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

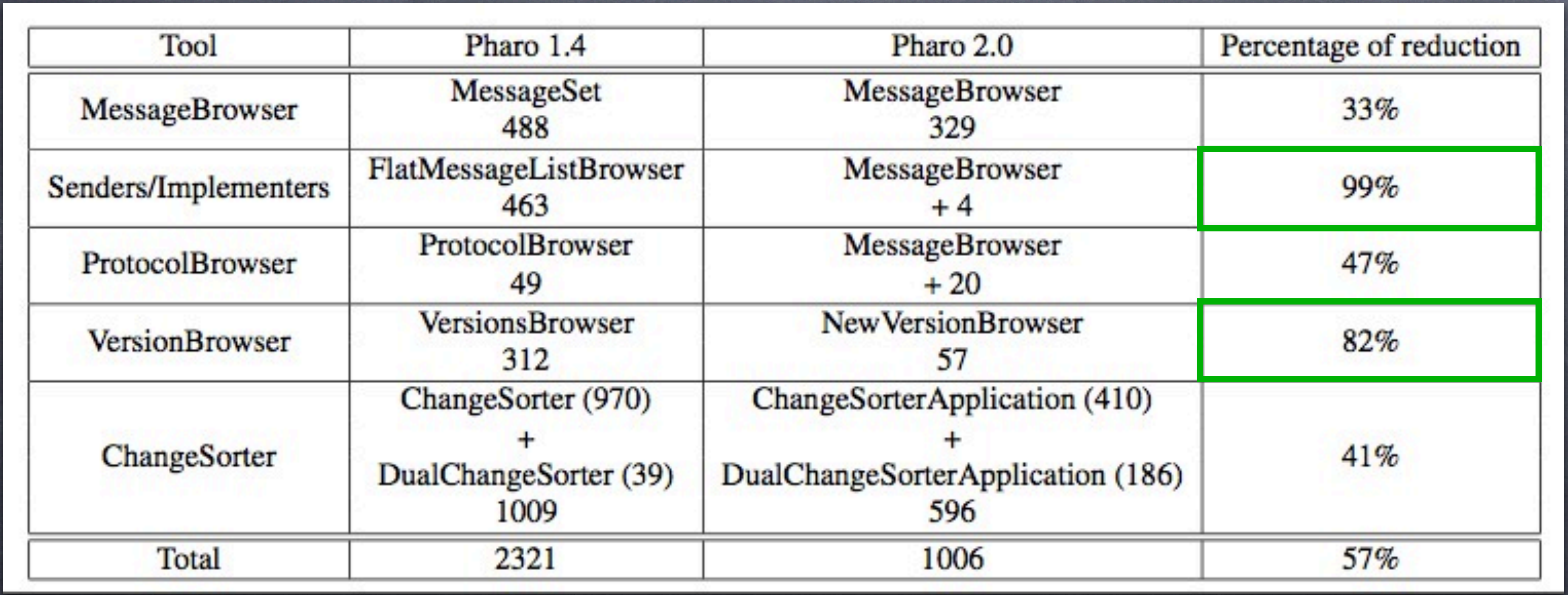


```
displayBlock: aBlock
    <api: #block getter:#displayBlock registration: #whenDisplayBlockChanged:>
    "Set the one argument block used to wrap your domain specific items.
    The block should return something that can be displayed in a list - like a String or a Text"

    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	NewVersionBrowser 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