CS 535 Object-Oriented Programming & Design
Fall Semester, 2008
Doc 16 VW GUI 2
Nov 6 2008

References

VisualWorks GUI Developer's Guide, GUIDevGuide.pdf in the docs directory of the VW distribution

Issues with ButtonExample

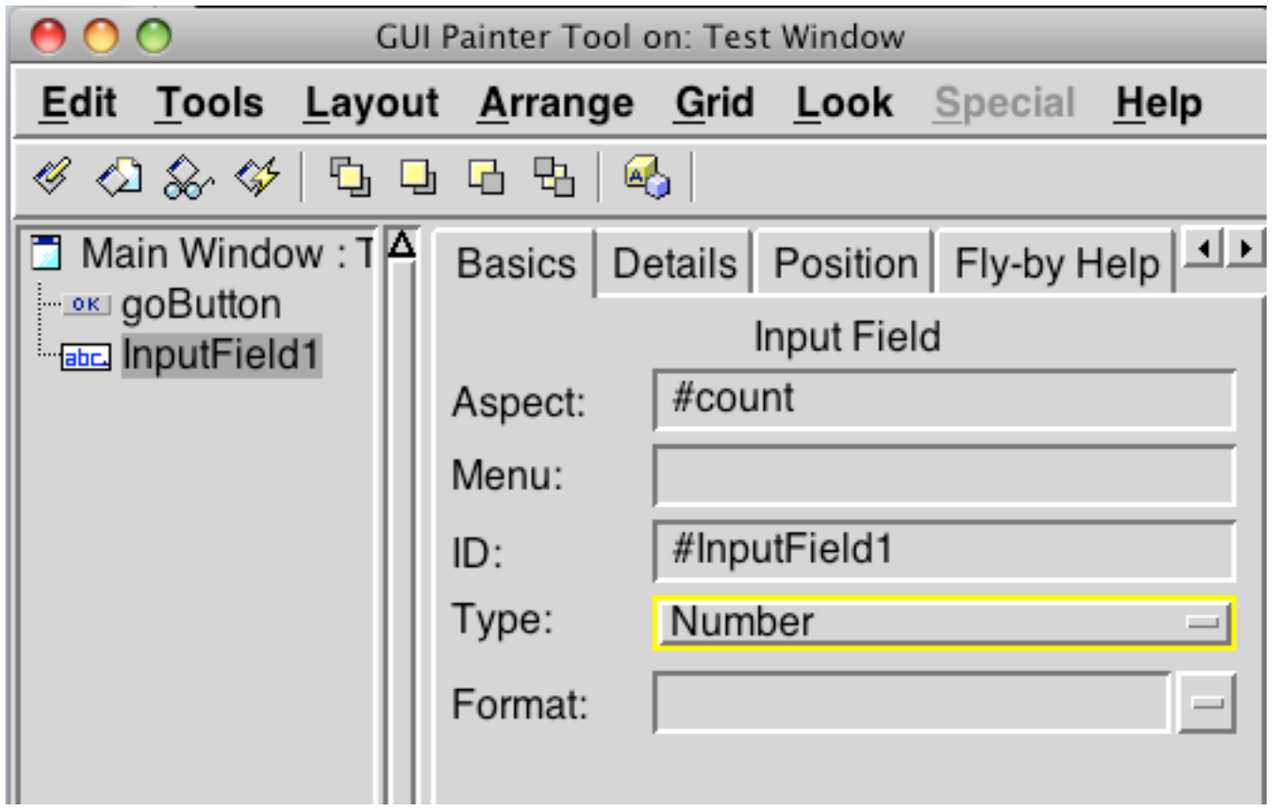
Strings rather than numbers

Dealing with ValueHolders rather than with values

View code/logic in domain code

Well there really is no domain objects as the example is so simple, but the issue needs to be addressed.

Configuring the Widget for Numbers



Using Numbers in Button Example

ButtonExample>>count

```
^count isNil
    ifTrue:
        [count := 0 asValue]
    ifFalse:
        [count]
```

ButtonExample>>go

```
self count value: (self count value  + 1) .
Dialog warn: 'Hi'.
^self
```

Adapters

ButtonExample Adapter

```
ButtonExample>>countAdapter

	| countAdapter |
	countAdapter := AspectAdaptor subject: self.
	countAdapter
		forAspect: #count;
		subjectSendsUpdates: true.
	^countAdapter
```

```
ButtonExample>>initialize

	count := 0
```

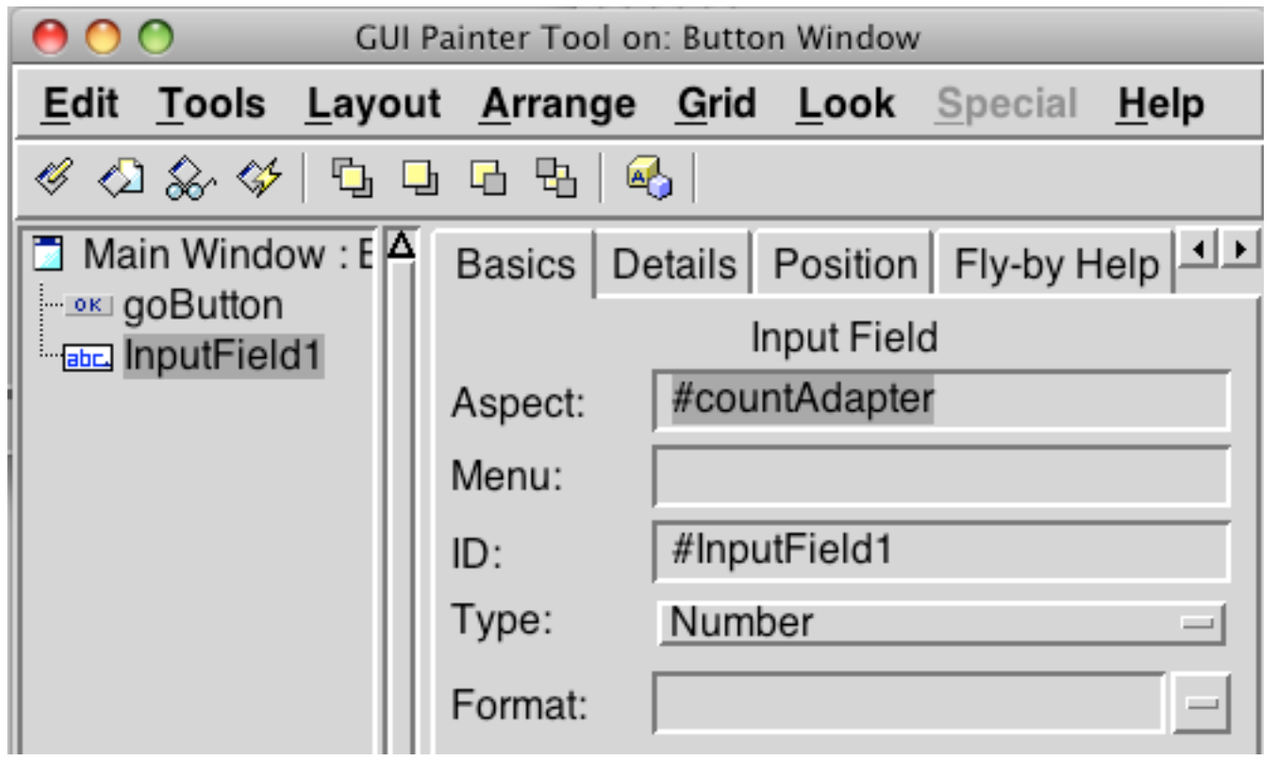
```
ButtonExample>>count

	^count
```

```
ButtonExample>>go

	count := count + 1.
	self changed: #count.
	Dialog warn: 'Time to go'.
	^self
```

```
ButtonExample>>count: anInteger

	count := anInteger
```


The text widget wants a ValueHolder, we want to have an integer. So we use an adapter that looks like a ValueHolder to the text widget. However when it needs to access the value it uses the count and count: methods.

Using a Number rather than a String



View & Domain Logic Mixed

ButtonExample

- Simple example
- Designed to show how to use a widget

It handles both view logic and domain logic

Simple Domain Class

```
Smalltalk defineClass: #Counter
	superclass: #{Core.Object}
	instanceVariableNames: 'count '

Counter class>>new
	^super new initialize

Counter>>count
	^count

Counter>>count: anInteger
	count := anInteger

Counter>>increment
	self count: count + 1

Counter>>initialize
	count := 0
```

Using the Domain Object

```
Smalltalk defineClass: #ButtonExample
        superclass: #{UI.ApplicationModel}
        instanceVariableNames: 'count '
```

```
initialize
        count := Counter new
```

```
countAdapter
        | countAdapter |
        countAdapter := AspectAdaptor subject: count.
        countAdapter
                forAspect: #count;
                subjectSendsUpdates: true.
    ^countAdapter
```

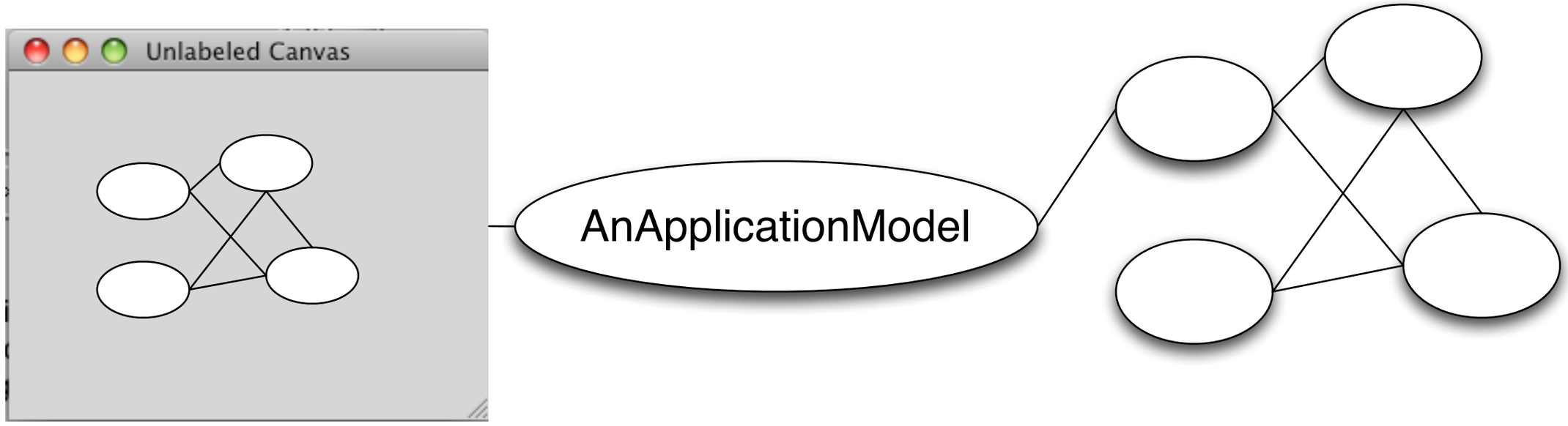
```
go
        count increment.
        count changed: #count.
        Dialog warn: 'Time to go'.
        ^self
```

Issue - Who changes count?

ButtonExample class controls when count changes

ButtonExample can then inform window of changes
Keeps Counter class independent of GUI

Issue - Who changes count?



What if other objects can change count?

ButtonExample will not be able to inform window of changes

When we have multiple domain objects a message to one may result in changes to one or more objects. If these objects contribute to the display it may not be possible for the application model to notify the view how to update itself correctly.

Domain Objects Updated

```
Smalltalk defineClass: #Counter
        superclass: #{Core.Object}
        instanceVariableNames: 'count '

Counter class>>new
        ^super new initialize

Counter>>count
        ^count

Counter>>count: anInteger
        count := anInteger

Counter>>increment
        self count: count + 1.
     self changed: #count

Counter>>initialize
        count := 0
```

ButtonExample Updated

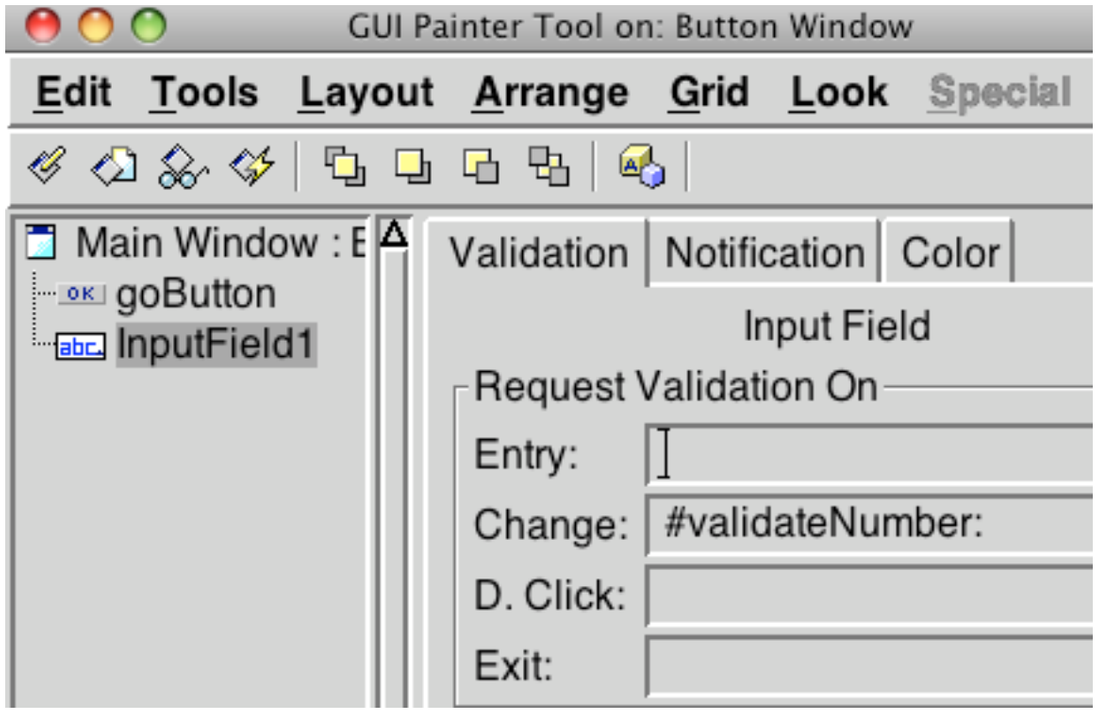
```
Smalltalk defineClass: #ButtonExample
        superclass: #{UI.ApplicationModel}
        instanceVariableNames: 'count '
```

```
initialize
        count := Counter new
```

```
countAdapter
        | countAdapter |
        countAdapter := AspectAdaptor subject: count.
        countAdapter
                forAspect: #count;
                subjectSendsUpdates: true.
    ^countAdapter
```

```
go
        count increment.
        Dialog warn: 'Time to go'.
        ^self
```

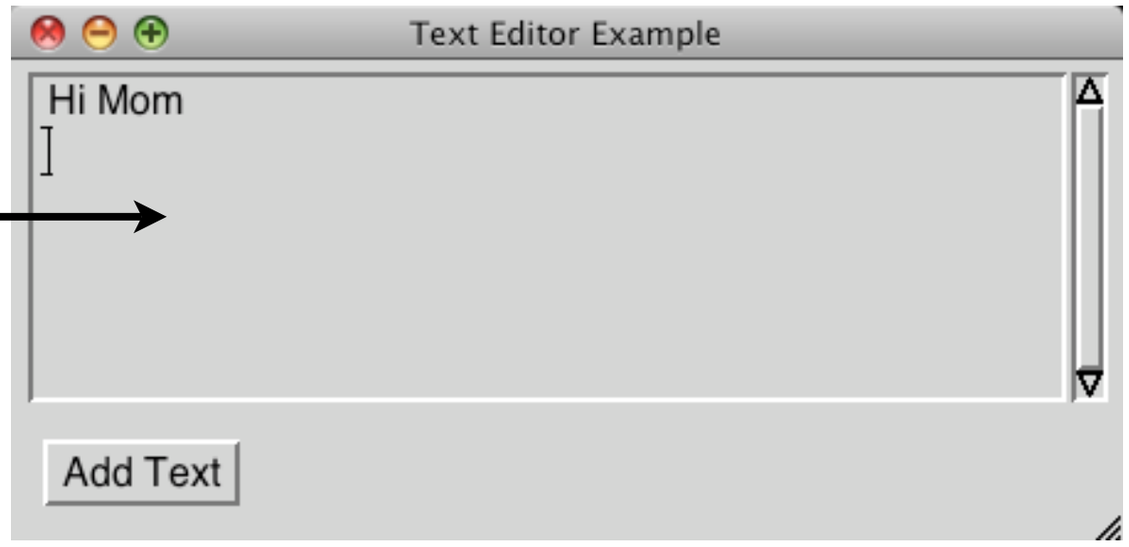
Validating Input



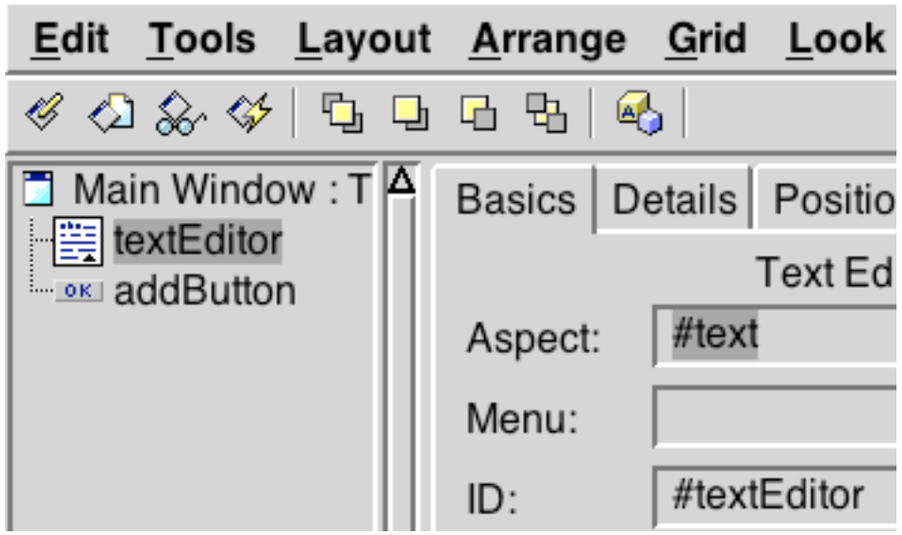

```
ButtonExample>>validateNumber: aController
    | entry |
    entry := aController editValue.
    ^entry >= 0
```

Text Editor Example

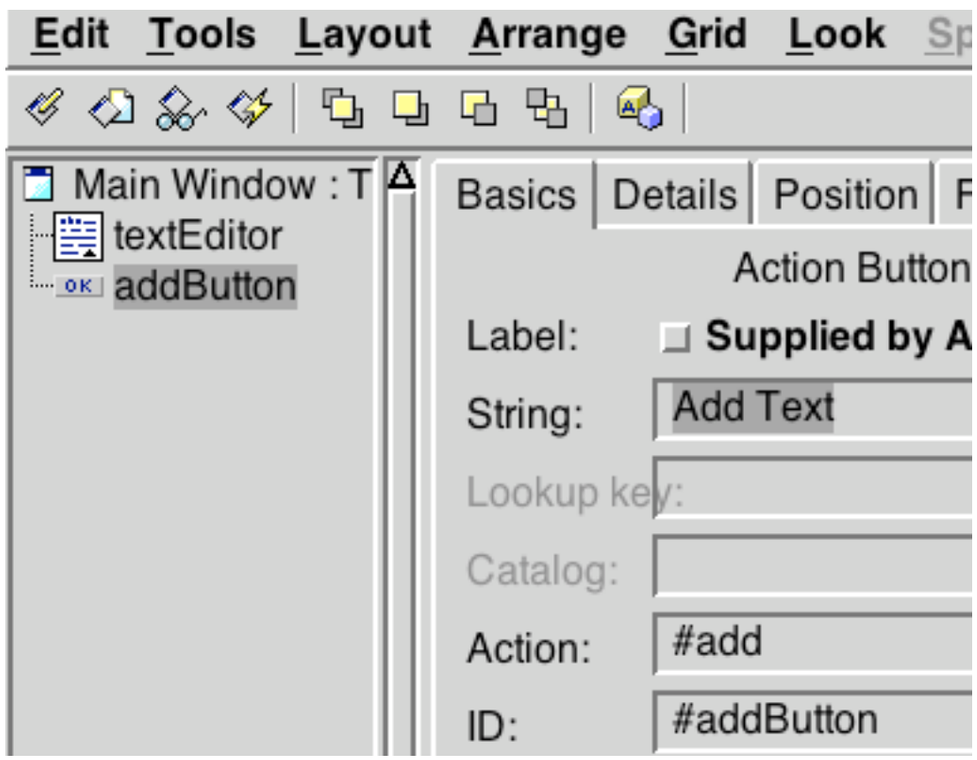
Text Editor
widget



Text Editor Example




```
TextExample>>text
	^text isNil
		ifTrue:
			[text := 'Hi Mom' asValue]
		ifFalse:
			[text]
```




```
TextExample>>add
	self text
	value: self text value , '\Add more text\' withCRs
```