Eclipse 4's Modeled UI
less sweating over workbench APIs
more grinning over simple POJOs

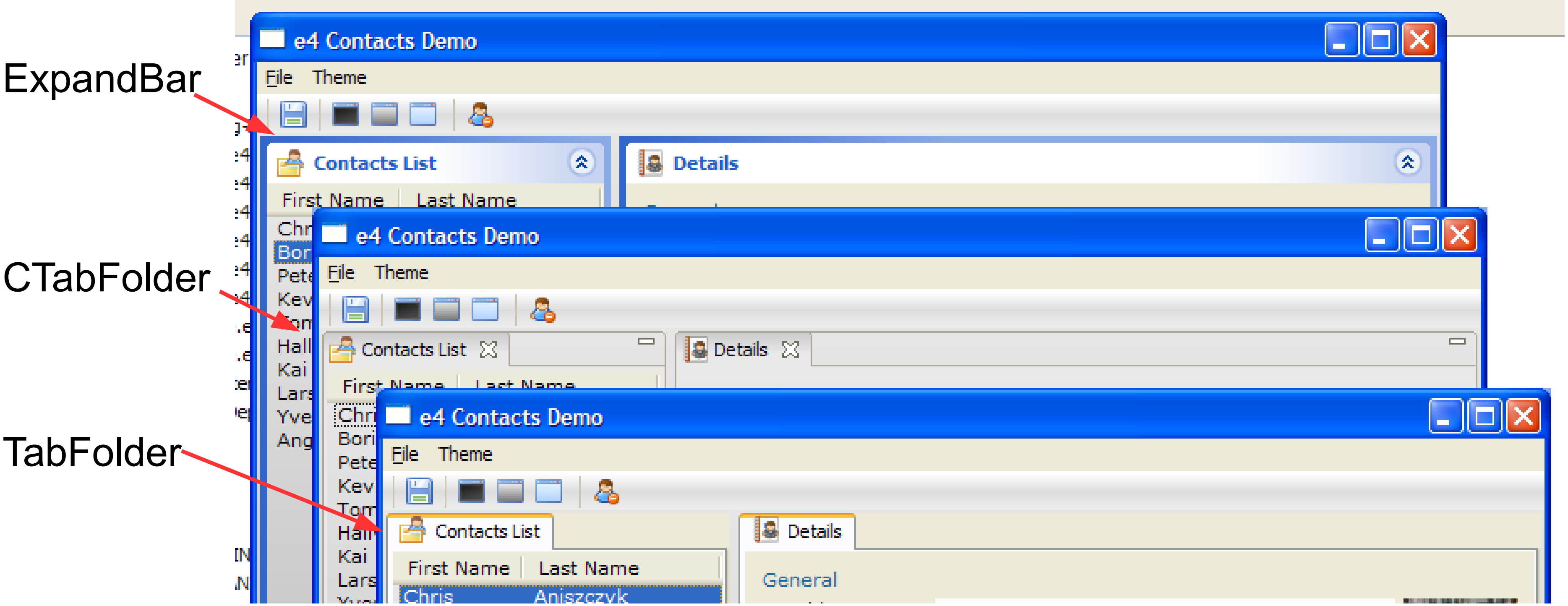
Remy Suen, IBM Rational

What is the modeled UI?

- An Eclipse 4 application is now backed by a model
- No more `IWorkbenchWindow` and `IWorkbenchPage`
  - ...which then suddenly became `IWorkbenchParts`...
  - ...not clear where the stacks or sashes came from...
- The model merely describes the structure of the user interface
  - This is similar to the `org.eclipse.ui.menus` extension point in a way

How is the model represented to the user?

- The renderers define the look
  - You can place the same XML snippet of an `org.eclipse.ui.menus` contribution into a menu or in a tool bar, the workbench would've sorted out the rest



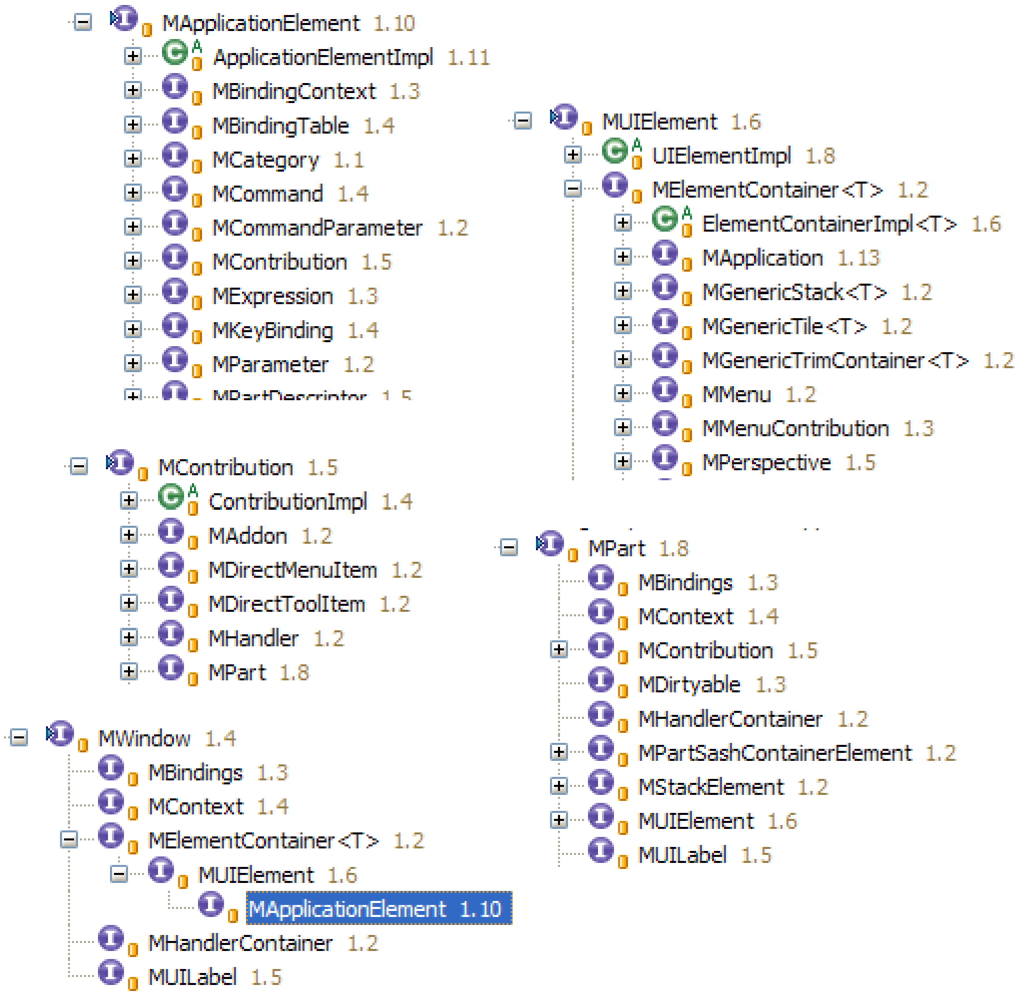
March 24, 2011

What's in? What's not?

- Pretty much anything you see right now in a regular Eclipse workbench window
  - Windows
  - Perspectives
  - Parts
  - Menu items
  - Tool bars
- Some things are not modeled though
  - Wizards
  - Dialogs
  - Preferences
    - There is a map for storing data though

How is the model constructed?

- Everything in the model is an `ApplicationElement`
- Data layer of abstract interfaces
  - –A `UILabel` has an icon, textual string, and a tooltip
  - –An `ElementContainer` has a list of children
  - –A `Contribution` has an implementing object and a map for storing data
- Physical layer of concrete implementations
  - –A `Part` extends `UILabel` and `Contribution`
  - –A `Window` extends `UILabel` and ElementContainer



```
⊟ MApplicationElement 1.10
   ⊞ ApplicationElementImpl 1.11
   ⊞ MBindingContext 1.3
   ⊞ MBindingTable 1.4
   ⊞ MCategory 1.1
   ⊞ MCommand 1.4
   ⊞ MCommandParameter 1.2
   ⊞ MContribution 1.5
   ⊞ MExpression 1.3
   ⊞ MKeyBinding 1.4
   ⊞ MParameter 1.2
   ⊞ MPartDescriptor 1.5
```

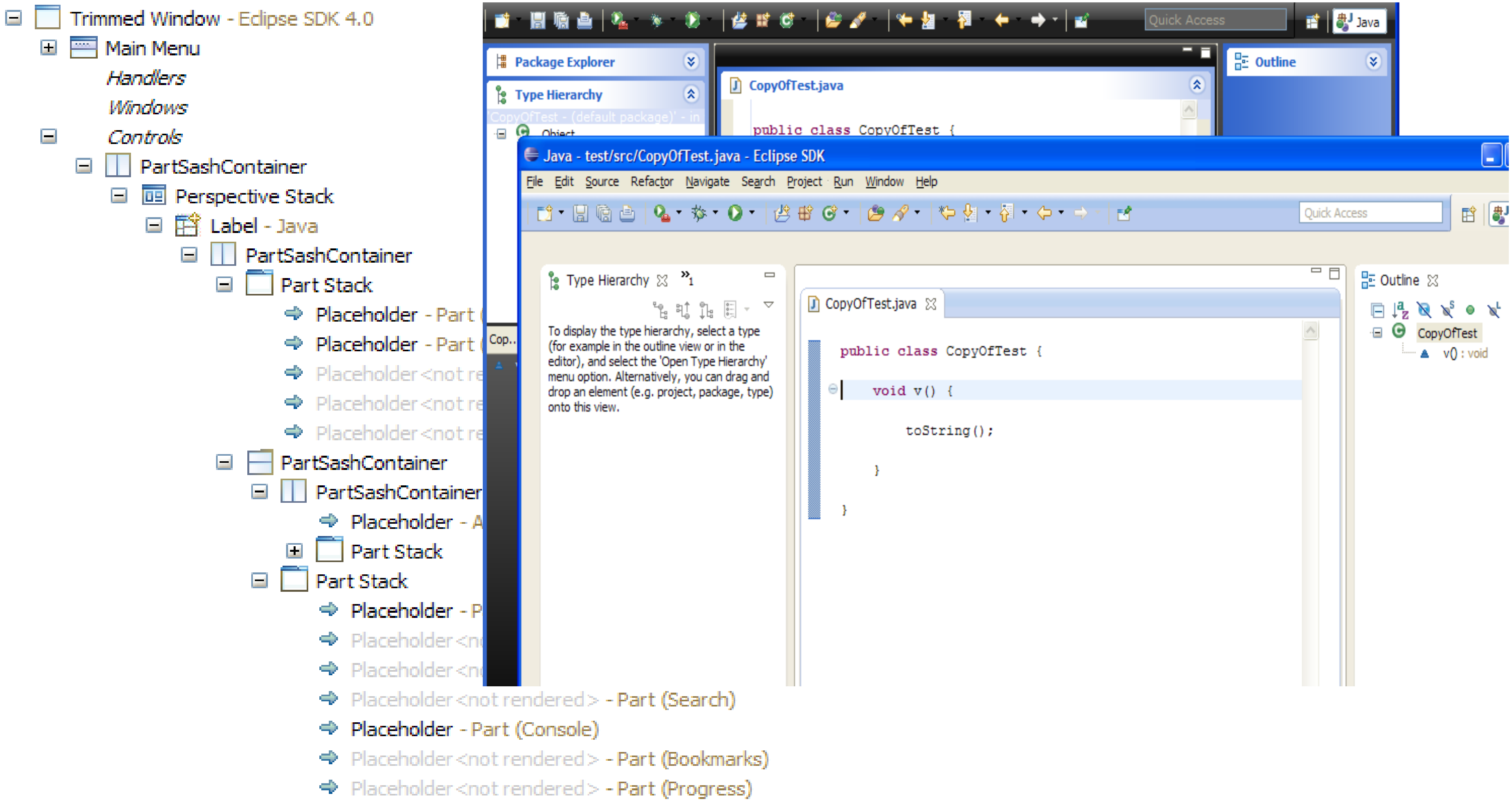
```
⊟ MUIElement 1.6
   ⊞ UIElementImpl 1.8
   ⊟ MElementContainer<T> 1.2
      ⊞ ElementContainerImpl<T> 1.6
      ⊞ MApplication 1.13
      ⊞ MGenericStack<T> 1.2
      ⊞ MGenericTile<T> 1.2
      ⊞ MGenericTrimContainer<T> 1.2
      ⊞ MMenu 1.2
      ⊞ MMenuContribution 1.3
      ⊞ MPerspective 1.5
```

```
⊟ MContribution 1.5
   ⊞ ContributionImpl 1.4
   ⊞ MAddon 1.2
   ⊞ MDirectMenuItem 1.2
   ⊞ MDirectToolItem 1.2
   ⊞ MHandler 1.2
   ⊞ MPart 1.8
```

```
⊟ MPart 1.8
     MBindings 1.3
     MContext 1.4
   ⊞ MContribution 1.5
     MDirtyable 1.3
     MHandlerContainer 1.2
   ⊞ MPartSashContainerElement 1.2
   ⊞ MStackElement 1.2
   ⊞ MUIElement 1.6
     MUILabel 1.5
```

```
⊟ MWindow 1.4
     MBindings 1.3
     MContext 1.4
   ⊟ MElementContainer<T> 1.2
      ⊟ MUIElement 1.6
           MApplicationElement 1.10
     MHandlerContainer 1.2
     MUILabel 1.5
```

What does the model for the SDK look like?



- Trimmed Window - Eclipse SDK 4.0
  - Main Menu
  - *Handlers*
  - *Windows*
  - *Controls*
    - PartSashContainer
      - Perspective Stack
        - Label - Java
          - PartSashContainer
            - Part Stack
              - Placeholder - Part (
              - Placeholder - Part (
              - Placeholder <not re
              - Placeholder <not re
              - Placeholder <not re
            - PartSashContainer
              - PartSashContainer
                - Placeholder - A
                - Part Stack
              - Part Stack
                - Placeholder - P
                - Placeholder <n
                - Placeholder <n
                - Placeholder <not rendered> - Part (Search)
                - Placeholder - Part (Console)
                - Placeholder <not rendered> - Part (Bookmarks)
                - Placeholder <not rendered> - Part (Progress)

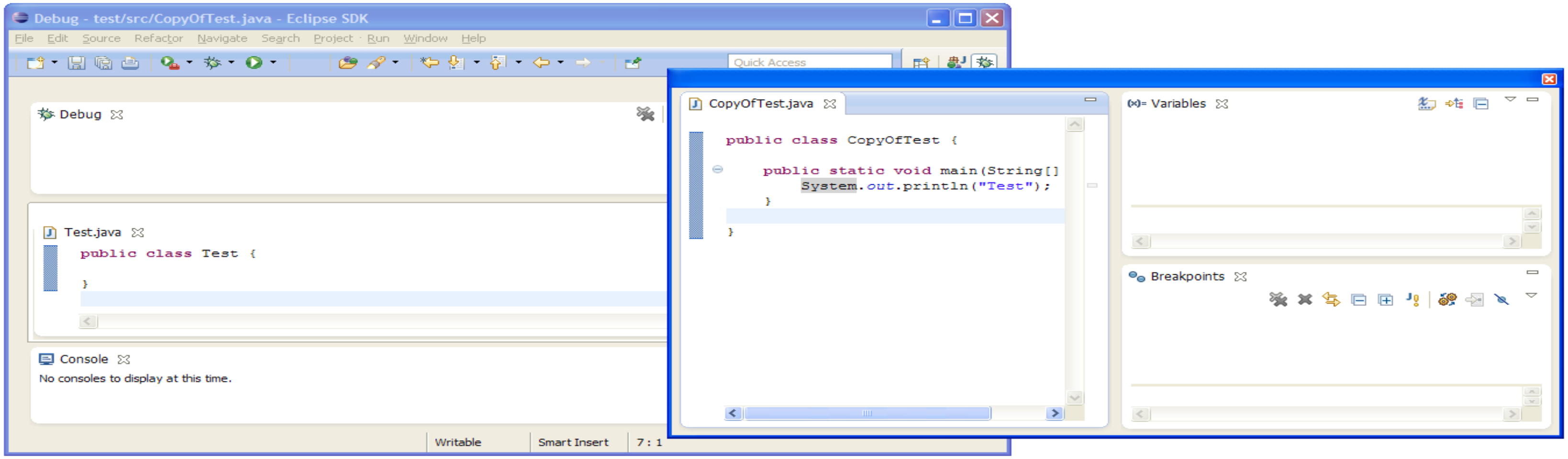
To display the type hierarchy, select a type (for example in the outline view or in the editor), and select the 'Open Type Hierarchy' menu option. Alternatively, you can drag and drop an element (e.g. project, package, type) onto this view.

```
public class CopyOfTest {

    void v() {

        toString();

    }

}
```

March 24, 2011

What's in it for me here?

- The previous diagram seemed to look like the same old, same old
- Users can now detach things that couldn't be detached
  - And likewise create shapes inside those detached windows that were not possible before


Debug - test/src/CopyOfTest.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Debug
Test.java
Console
CopyOfTest.java
Variables
Breakpoints
Writable
Smart Insert
7 : 1

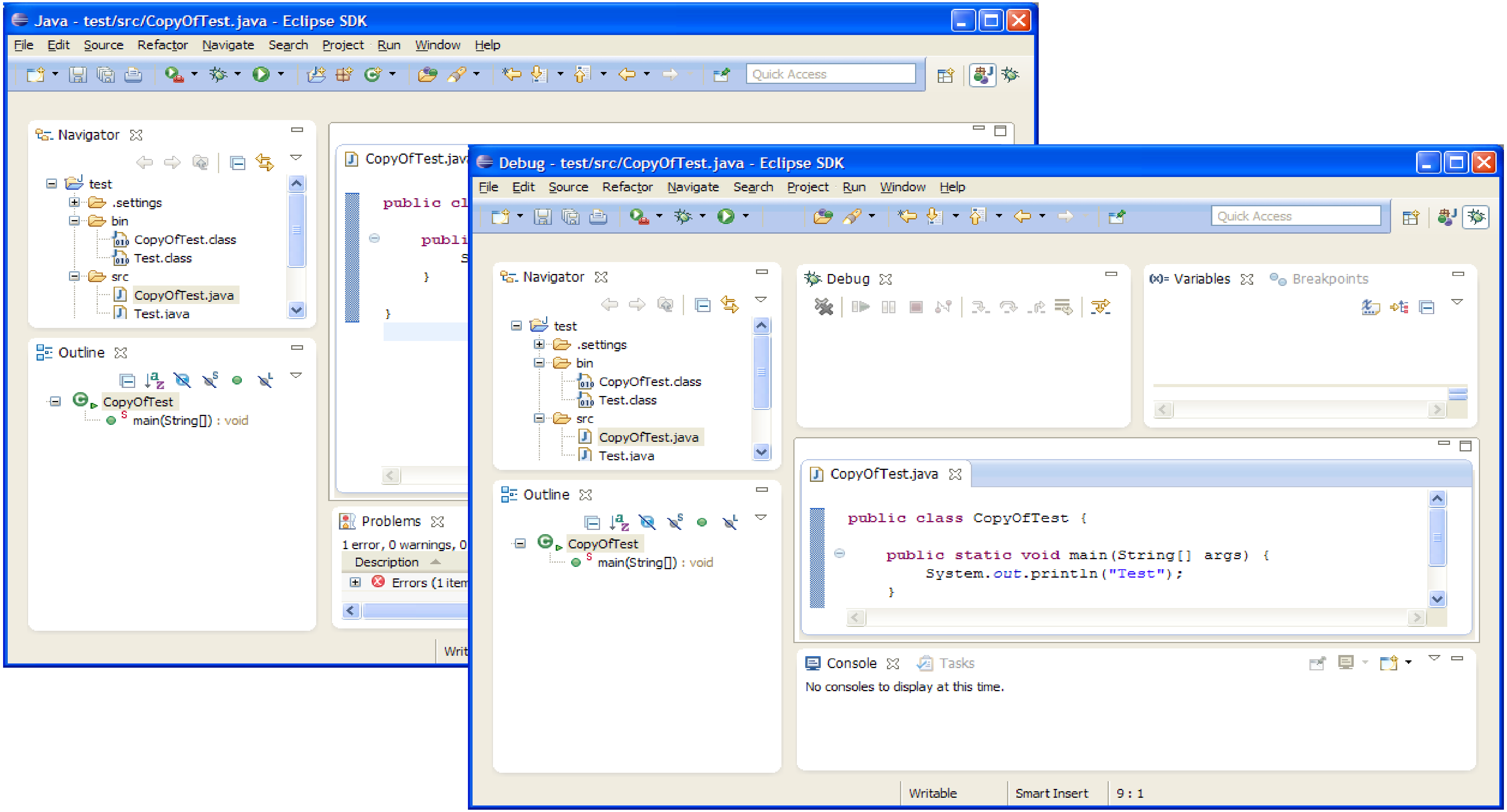
```
public class Test {

}
```

No consoles to display at this time.

```
public class CopyOfTest {

    public static void main(String[]
        System.out.println("Test");
    }

}
```

- Don't feel bound by perspectives
  - Perspectives have traditionally defined a containment model for parts and stacks
  - Elements can now physically reside outside of the perspective

What do you mean by having elements outside of the perspective?



Java - test/src/CopyOfTest.java - Eclipse SDK

Debug - test/src/CopyOfTest.java - Eclipse SDK

```
public class CopyOfTest {

    public static void main(String[] args) {
        System.out.println("Test");
    }
```

1 error, 0 warnings, 0

No consoles to display at this time.

March 24, 2011

How clean is the code?

- How would I have a stack show the 'Console' view?

```
bottomStack.setSelectedElement(consolePart);
```

- How do I move a part into another stack?

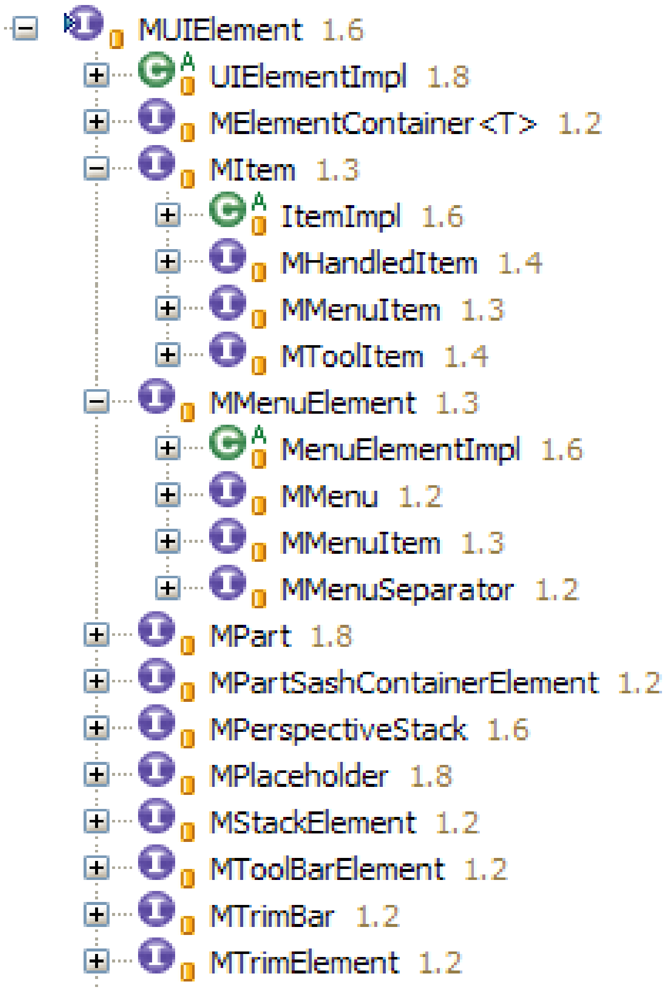
```
otherStack.getChildren().add(consolePart);
```

- How do I close all my parts in a stack?

```
bottomStack.setToBeRendered(false);
```

Wouldn't these new shapes get me tied up in a knot?

- An `ElementContainer` can contain a list of `UIElements`...
  - ...that sounds rather open-ended and a recipe for disaster...
- Generics and modified code helps prevent invalid elements from being added to a container's list of children
  - Can't add a `Window` into a `PartStack`
  - Can't add a `MenuItem` into a `ToolBar`
- Can't set the selected element of a container to an element that hasn't been rendered



- MUIElement 1.6
  - UIElementImpl 1.8
  - MElementContainer<T> 1.2
  - MItem 1.3
    - ItemImpl 1.6
    - MHandledItem 1.4
    - MMenuItem 1.3
    - MToolItem 1.4
  - MMenuElement 1.3
    - MenuElementImpl 1.6
    - MMenu 1.2
    - MMenuItem 1.3
    - MMenuSeparator 1.2
  - MPart 1.8
  - MPartSashContainerElement 1.2
  - MPerspectiveStack 1.6
  - MPlaceholder 1.8
  - MStackElement 1.2
  - MToolBarElement 1.2
  - MTrimBar 1.2
  - MTrimElement 1.2

Q&A

Where do I go for more information?

- Ask questions on the forums or the mailing list
  - http://www.eclipse.org/forums/index.php?t=thread&frm_id=12
  - https://dev.eclipse.org/mailman/listinfo/e4-dev (e4-dev@eclipse.org)
- Find us on IRC
  - irc.freenode.net at #eclipse-e4

Legal Notice

-
- Java and all Java-based trademarks are trademarks of Oracle Corporation in the United States, other countries, or both.
- Eclipse and the Eclipse logo are trademarks of Eclipse Foundation, Inc.
- IBM and the IBM logo are trademarks or registered trademarks of IBM Corporation, in the United States, other countries, or both.
- Rational and the Rational logo are trademarks or registered trademarks of IBM Corporation in the United States, other countries, or both.
- Other company, product, or service names may be trademarks or service marks of others.
- THE INFORMATION DISCUSSED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, AND IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, SUCH INFORMATION. ANY INFORMATION CONCERNING IBM'S PRODUCT PLANS OR STRATEGY IS SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.