XXIV. Interface Objects

Three-Tier Architectures
The Presentation layer
Sequence Diagrams for User Interface Classes
Prototyping the User Interface
User Interface Class and Package Diagrams
Model-View-Controller Architecture Revisited
Statechart Diagrams for Dialogue Dynamics

The Three-Tier Architecture, Revisited

- User interfaces for an information system are part of the presentation layer in a three-tier architecture.
- The three-tier architecture separates cleanly user interfaces from application logic/business classes and from data storage components of the system.
- Business classes “know nothing” about how their (business) objects will be presented to the users.
Check Campaign Budget

Add a Dialog Box Object
Prototyping the Dialogue

- Prototyping can be used to determine what the interface will look like.

![Check Campaign Budget dialog box](image)

Class Diagram for Interface Classes

Composition specifies that a dialogue box is made up of other components.
Another Class Diagram

CBWindow can also be represented as a class with the graphical components that make it up as attributes.

<table>
<thead>
<tr>
<th>CBWindow</th>
</tr>
</thead>
<tbody>
<tr>
<td>clientLabel</td>
</tr>
<tr>
<td>campaignLabel</td>
</tr>
<tr>
<td>budgetLabel</td>
</tr>
<tr>
<td>checkButton</td>
</tr>
<tr>
<td>closeButton</td>
</tr>
<tr>
<td>budgetTextField</td>
</tr>
<tr>
<td>clientChoice</td>
</tr>
<tr>
<td>campaignChoice</td>
</tr>
</tbody>
</table>

Packages for Interface Classes

- Package diagrams show the dependencies among interface classes in different packages.
Revised Class Diagram

- Composition shows that a dialogue box is made up of other components from the AWT package.

Prototyping the Dialogue

- There are several ways for entering the Client and Campaign name:
  - Use a separate look-up window for each class;
  - Allow the user to enter a part of a name, then have the system return a list of close matches;
  - Use a tree data structure to show clients and campaigns in a tree-like hierarchy.
Alternative Dialogue Prototypes

- Separate window for look-up

Alternative Dialogue Prototypes: Three View Control
Updating the Sequence Diagram

- Choice:
  - Client (CL)
  - Campaign (CA)
  - Lookup

Updating the Class Diagram
Model-View-Controller

1: Broadcast
2: Update self
3: Notify Change
4: Update Presentation
5: Ask What Has Changed
6: Notify Change
7: Request Model data

The Java ActionListener Approach

1: ActionPerformed (ActionEvent e)
2: Inspect Event
3: [Event of Interest] Notify Class of Event
4: Update Self
Modelling the Dynamic Behaviour of the Interface

- The sequence diagrams show the sequential view of the user working through the fields on the screen from top to bottom.
- But in GUI interfaces the user can click on the interface object out of sequence.
- What happens if the user clicks on the Check button before a client and a campaign have been selected?
- To specify what happens, we can use Statechart diagrams!

![Statechart Diagrams](image)

Client listbox, Campaign Listbox, Cursor
**CheckButton, BudgetTextbox**

```
Load Dialog  

<table>
<thead>
<tr>
<th>Disabled</th>
<th>enabled( )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>disable( )</td>
</tr>
<tr>
<td></td>
<td>*cursor.set(NORMAL)</td>
</tr>
<tr>
<td></td>
<td>onclick/</td>
</tr>
<tr>
<td></td>
<td>*cursor.set(WAIT)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Checking</td>
<td></td>
</tr>
</tbody>
</table>
|            | Do/ *budgetTextbox.setText 
|            | (campaignListbox.getCampaign( ) 
|            | .checkCampaignBudget( )) |

Load Dialog  

<table>
<thead>
<tr>
<th>Clear</th>
<th>setText(surplus)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Entry/ setText(&quot;&quot;&quot;)</td>
</tr>
<tr>
<td></td>
<td>clear( )</td>
</tr>
<tr>
<td></td>
<td>Displayed</td>
</tr>
</tbody>
</table>
```

**Additional Readings**