Discussion of HCI Design Patterns

The TC began the teleconference with a brief discussion of HCI design patterns and the difference between a pattern and a template. Dr. Abrams had recently returned from a business trip on which he learned of a book containing several hundred HCI design patterns. This opened the question in his mind as to whether UIML could/should be able to represent these design patterns.

The current template mechanism in UIML is not powerful enough to encompass an entire design pattern, but the TC decided that this was acceptable. The semantics of a template in the current UIML sense is “a pre-defined set of elements and their interactions” that can be re-used within the overall structure of a UIML document. A design pattern is an overarching principle that the designers use to construct the interface and may provide guidance on all aspects of the design.

Dr. Vanderdonckt and Mr. Limbourg explained that the ideas of patterns and templates are addressed separately in USIXML. Patterns come in two forms: Generative and Constructive. Constructive patterns provide building blocks that give structure to the interface. Generative patterns generate parts of the structure through transform rules.

The TC will investigate the book Dr. Abrams discussed to determine the structure of the language can be designed to support design patterns. The TC will discuss this in more
The TC also discussed the UIML-based design tools that Harmonia provided demonstrations of: the UIML Development Tool (UDT) and the UIML-CAD model-based design tool.

The UDT is a drag and drop authoring tool for creating UIML user interfaces. The tool is built on a flexible client-server architecture that centralizes all of the widget set specific code and palette information on the server. This allows the client to remain widget set generic while providing the editing capability to get the most out of the chosen widget set. The server returns image data for each widget whenever a property change is made. In this architecture, the HCI designer can choose between any number of active servers to find one that will support the vocabulary they wish to design for. Thus it is possible to have servers for C++ widget sets like GTK or QT and servers for Java based toolkits like Swing, AWT, or SWT. The common communication and standard networking protocols used allows the Toolkit Server to be implemented in a different language than the client, providing a true WYSIWYG experience.

Leveraging the flexibility of UIML, toolkit servers can be built to support abstract or generic vocabularies. The client can then use these servers to design interfaces in the abstract vocabulary. This would allow the UDT to create high-level, widget set independent models of the interface as well as low-level toolkit specific models.

The UDT also provides a mechanism for defining simple behaviors on widgets within the interface. Dr. Vanderdonckt remarked that this is an area in which many contemporary tools are lacking. He suggested the TC investigate a Spanish tool that he will reference on the e-mail list for ideas on defining behavior. It is the only tool that the TC knows of beside the UDT that provides decent behavior definition.

The TC also discussed the UIML-CAD tool, a model-based HCI design tool prototype. UIML-CAD is a forward engineering tool that starts with a high-level representation of the interface (such as a task model) and defines the model through several iterations down to a concrete final UI. The process consists of 4 separate models: a high-level conceptual model (task model), an abstract model that uses abstract widgets, a concrete model that uses platform specific widgets, and a final presentation model that incorporates style and device characteristics.

This prototype will provide a basis for new product development by Harmonia to create a UIML-based multi-path HCI design tool. This CAD-HCI tool will take the same model-based approach, but allow designer to infer the higher-level models from the lower-level (and often pre-existing) widget specific models.

### Action Items

- Jim Helms will update the Open Issues document to add the lessons learned from investigating DISL.
- Jim Helms will update the document recording the relationship of UIML to other languages/working groups to include XAML.
- Each member of the TC will send between 5 and 10 short questions about DISL to Robbie so that he may prepare some material for the DISL discussion at the next meeting. These questions should be sent no later than the Wednesday before the meeting.
<table>
<thead>
<tr>
<th>Adjournment</th>
</tr>
</thead>
<tbody>
<tr>
<td>The meeting ended at 1:00 PM EST to reconvene on November 15th, 2004.</td>
</tr>
</tbody>
</table>