An Architectural Approach Maximizing Platform Portability and User Interface Component Reuse

Richard N. Taylor
Department of Information and Computer Science
University of California, Irvine
Irvine, California 92697-3425
taylor@ics.uci.edu

http://www.ics.uci.edu/~taylor/
http://www.ics.uci.edu/pub/arch/
Overview of C2

- A component- and message-based architectural style
- C2 architectures are networks of concurrent components hooked together by connectors

- no component-to-component links
- “one up, one down” rule for components
- connector-to-connector links are allowed
- “many up, many down” rule for connectors
- all communication by exchanging messages
- substrate independence
KLAX: Example Application in the C2 Style

Clock Logic  Status Logic  Chute Logic  Well Logic  Palette Logic

Next Tile Placing Logic  Tile Match Logic  Relative Pos Logic

Status Logic

Status Artist  Well Artist  Chute Artist  Palette Artist

Tile Artist

Layout Manager

Graphics Binding
An Implementation Architecture for KLAX

- Clock Logic
- Status ADT
- Chute ADT
- Well ADT
- Palette ADT
- Next Tile Placing Logic
- Tile Match Logic
- Relative Pos Logic
- Status Logic
- Status Artist
- Well Artist
- Chute Artist
- Palette Artist
- Tile Artist
- Layout Manager
- Graphics Binding

GSW February 27, 1997
Richard N. Taylor
Implementing C2 Architectures

- Extensible framework of abstract classes for C2 concepts
- Implemented in Java, C++, and Ada (partially)
Runtime Manipulation of C2 Architectures

- *ArchShell* - tool for interactive construction, execution, and runtime modification of C2 architectures
  - current implementation supports the Java class framework
- Provides functionality similar to a UNIX shell
  - e.g., construction of pipe-and-filter architectures in *csh*
- Goes beyond support found in UNIX shells
  - dynamic loading and linking new architectural elements
  - sending C2 messages to running components interactively
  - testing runtime behavior of individual components
  - experimentation with partial architectures
Example of Runtime Manipulation of C2 Architectures

**Environment Architecture**

- JOP Repository
  - JFox WWW Browser
  - IDN Parser
  - Component Incorporation Tool

**Application Architecture**

- Stack ADT
  - Stack Artist 1
    - Graphics Binding
  - Stack Artist 2
Support for Reuse in C2

- Component heterogeneity
- Substrate independence
- Internal component architecture
- Asynchronous message passing via connectors
- No assumption of shared address space
- No assumption of single thread of control
- Separation of architecture from implementation
Example of OTS Reuse in C2

Clock Logic  Status ADT  Chute ADT  Well ADT  Palette ADT

Next Tile Placing Logic  Tile Match Logic  Relative Pos Logic

Status Logic

Status Artist  Well Artist  Chute Artist  Palette Artist

Tile Artist

Layout Manager

Graphics Binding

Dialog  OTS Constraint Solver
### A Library of C2 Components

<table>
<thead>
<tr>
<th>Constraint Maintained</th>
<th>Manager Version Number</th>
<th>Constraint Manager Version</th>
<th>Constraint Maintained</th>
<th>Manager Version Number</th>
<th>Constraint Manager Version</th>
<th>Constraint Maintained</th>
<th>Manager Version Number</th>
<th>Constraint Manager Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>SkyBlue &amp; Amulet</td>
<td>6</td>
<td>All</td>
<td>SkyBlue &amp; Amulet</td>
<td>6</td>
<td>All</td>
<td>SkyBlue &amp; Amulet</td>
<td>6</td>
<td>All</td>
</tr>
<tr>
<td>Amulet</td>
<td>5</td>
<td>Resizing</td>
<td>Amulet</td>
<td>5</td>
<td>Resizing</td>
<td>Amulet</td>
<td>5</td>
<td>Resizing</td>
</tr>
<tr>
<td>SkyBlue</td>
<td>4</td>
<td>Resizing</td>
<td>SkyBlue</td>
<td>4</td>
<td>Resizing</td>
<td>SkyBlue</td>
<td>4</td>
<td>Resizing</td>
</tr>
<tr>
<td>None</td>
<td>3</td>
<td>All</td>
<td>None</td>
<td>3</td>
<td>All</td>
<td>None</td>
<td>3</td>
<td>All</td>
</tr>
<tr>
<td>None</td>
<td>2</td>
<td>None</td>
<td>None</td>
<td>2</td>
<td>None</td>
<td>None</td>
<td>2</td>
<td>None</td>
</tr>
<tr>
<td>None</td>
<td>2</td>
<td>None</td>
<td>None</td>
<td>2</td>
<td>None</td>
<td>None</td>
<td>2</td>
<td>None</td>
</tr>
<tr>
<td>None</td>
<td>2</td>
<td>None</td>
<td>None</td>
<td>2</td>
<td>None</td>
<td>None</td>
<td>2</td>
<td>None</td>
</tr>
<tr>
<td>SkyBlue &amp; Amulet</td>
<td>2</td>
<td>Resizing</td>
<td>SkyBlue &amp; Amulet</td>
<td>2</td>
<td>Resizing</td>
<td>SkyBlue &amp; Amulet</td>
<td>2</td>
<td>Resizing</td>
</tr>
<tr>
<td>None</td>
<td>2</td>
<td>SkyBlue &amp; Amulet</td>
<td>None</td>
<td>2</td>
<td>None</td>
<td>None</td>
<td>2</td>
<td>None</td>
</tr>
<tr>
<td>None</td>
<td>1</td>
<td>SkyBlue &amp; Amulet</td>
<td>None</td>
<td>1</td>
<td>None</td>
<td>None</td>
<td>1</td>
<td>None</td>
</tr>
<tr>
<td>None</td>
<td>1</td>
<td>SkyBlue &amp; Amulet</td>
<td>None</td>
<td>1</td>
<td>None</td>
<td>None</td>
<td>1</td>
<td>None</td>
</tr>
<tr>
<td>None</td>
<td>1</td>
<td>SkyBlue &amp; Amulet</td>
<td>None</td>
<td>1</td>
<td>None</td>
<td>None</td>
<td>1</td>
<td>None</td>
</tr>
<tr>
<td>None</td>
<td>1</td>
<td>SkyBlue &amp; Amulet</td>
<td>None</td>
<td>1</td>
<td>None</td>
<td>None</td>
<td>1</td>
<td>None</td>
</tr>
<tr>
<td>None</td>
<td>1</td>
<td>SkyBlue &amp; Amulet</td>
<td>None</td>
<td>1</td>
<td>None</td>
<td>None</td>
<td>1</td>
<td>None</td>
</tr>
<tr>
<td>None</td>
<td>1</td>
<td>SkyBlue &amp; Amulet</td>
<td>None</td>
<td>1</td>
<td>None</td>
<td>None</td>
<td>1</td>
<td>None</td>
</tr>
<tr>
<td>None</td>
<td>1</td>
<td>SkyBlue &amp; Amulet</td>
<td>None</td>
<td>1</td>
<td>None</td>
<td>None</td>
<td>1</td>
<td>None</td>
</tr>
<tr>
<td>None</td>
<td>1</td>
<td>SkyBlue &amp; Amulet</td>
<td>None</td>
<td>1</td>
<td>None</td>
<td>None</td>
<td>1</td>
<td>None</td>
</tr>
<tr>
<td>None</td>
<td>1</td>
<td>SkyBlue &amp; Amulet</td>
<td>None</td>
<td>1</td>
<td>None</td>
<td>None</td>
<td>1</td>
<td>None</td>
</tr>
</tbody>
</table>

### Component-Based GUI Architecture

- **Palette Artist**: 1. Palette Location, Tile Location
- **Palette ADT**: 1. Palette Location, Tile Location
- **Well Chute Artist**: 1. Palette Location, Tile Location
- **Layout Manager**: 1. None, All SkyBlue, All Amulet, Resizing SkyBlue, Resizing Amulet, All SkyBlue & Amulet
Conclusions

- Runtime manipulation of architectures and OTS component reuse facilitated by style features
  - isolation of changes inside components
  - component substitutability
  - partial communication and service utilization
- Component library and application family created easily
- Heuristics for OTS component integration

<table>
<thead>
<tr>
<th>Problem with OTS Component</th>
<th>Integration Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inadequate Functionality</td>
<td>Source Code Modification</td>
</tr>
<tr>
<td>No Message-Based Communication</td>
<td>Wrapper</td>
</tr>
<tr>
<td>Different Programming Language</td>
<td>IPC Connector</td>
</tr>
<tr>
<td>Different OS Process</td>
<td>IPC Connector</td>
</tr>
<tr>
<td>Message Interface Mismatch</td>
<td>Domain Translator</td>
</tr>
</tbody>
</table>