TouchRAM
A Multi-Touch Enabled Software Design Tool
Supporting Concern-Oriented Reuse

Matthias Schöttle, Omar Alam, Franz-Philippe Garcia, Jörg Kienzle, Gunter Mussbacher
Software Engineering Laboratory
School of Computer Science, McGill University
Montreal, Canada
Email: Joerg.Kienzle@mcgill.ca
TouchRAM
A Multi-Touch Enabled Software Design Tool
Supporting Concern-Oriented Reuse

Matthias Schöttle, Omar Alam, Franz-Philippe Garcia, Jörg Kienzle, Gunter Mussbacher
Software Engineering Laboratory
School of Computer Science, McGill University
Montreal, Canada
Email: Joerg.Kienzle@mcgill.ca
• Tool of Agile Software Design Modelling
• Support for Class Diagrams, Sequence Diagrams, State Diagrams
• Reusable Concern Model Library
TouchRAM GUI

• **Multi-Touch**
  - Intuitive editing using multi-touch gestures
  - Significant speedup for
    - Navigating big models
    - Moving / rearranging classes
    - Establishing mappings between design concerns
  - Simultaneous support for multi-touch (TUIO) as well as mouse / keyboard input

• **Multi-User**
  - Every GUI Element can define its own gesture processors
**TouchRAM GUI**

- **Multi-Touch**
  - Intuitive editing using multi-touch gestures
  - Significant speedup for
    - Navigating big models
    - Moving / rearranging classes
    - Establishing mappings between design concerns
  - Simultaneous support for multi-touch (TUIO) as well as mouse / keyboard input

- **Multi-User**
  - Every GUI Element can define its own gesture processors
A concern groups software design models providing related functionality, and provides three interfaces to facilitate reuse.
A concern groups software design models providing related functionality, and provides three interfaces to facilitate reuse.
A concern groups software design models providing related functionality, and provides three interfaces to facilitate reuse.

- Variation Interface
- Customization Interface
- Usage Interface

Feature Model and Goal Model

Class Diagram
• **Variation Interface** of Observer
  • Exposes **possible design choices**
    • Optional, requires, excludes relationships also supported
  • Exposes impact on NFRs / qualities
**Observer Concern**

- **Customization Interface** of Observer
- **Exposes** general design classes and operations that need to be mapped to application-specific classes and operations

---

**Aspect IObserver**

**Structural View**

---

**ISubject**

<table>
<thead>
<tr>
<th>Modification</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ * Imodify(..)</td>
<td>add(IObserver a)</td>
</tr>
<tr>
<td>~ remove(IObserver a)</td>
<td>~ Set&lt;IObserver&gt; getObservers()</td>
</tr>
</tbody>
</table>

**Set**

<table>
<thead>
<tr>
<th>Modification</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>~ int size</td>
<td>~ Set&lt;IObserver&gt; create()</td>
</tr>
<tr>
<td>~ add(IObserver)</td>
<td>~ remove(IObserver)</td>
</tr>
<tr>
<td>~ destroy()</td>
<td></td>
</tr>
</tbody>
</table>

**mySet**

**mySubject**

---

**IObserver**

<table>
<thead>
<tr>
<th>Modification</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ startObserving(ISubject)</td>
<td>+ stopObserving()</td>
</tr>
<tr>
<td>~ Iupdate(ISubject)</td>
<td></td>
</tr>
</tbody>
</table>
**Observer Concern**

- **Customization Interface** of Observer
- **Exposes** general design classes and operations that need to be mapped to application-specific classes and operations

---

**Observer**

| Subject<|modify<|update>>
| Observer<|update>

---

**Mandatory Instantiation Parameters**

- **Customization Interface**
  - Exposes general design classes and operations that need to be mapped to application-specific classes and operations

---

**Observer**

| Subject<|modify<|update>>
| Observer<|update>
• Usage Interface of Observer
• Specifies classes that can be instantiated and operations that can be called
Observer Concern

- **Usage Interface of Observer**
- **Specifies classes that can be instantiated and operations that can be called**
Concern Reuse Process

1. **Use the variation interface** of the concern to select the most appropriate feature(s)
   - That provides the desired functionality
   - That maximizes positive impact on relevant non-functional application properties
   ⇒ TouchRAM generates the detailed generic design for the selected feature(s) of the concern

2. **Use the customization interface** of the generated design to adapt the generic design elements to the application-specific context
   ⇒ TouchRAM generates the application-specific design for the selected feature(s) of the concern

3. **Use the selected concern feature** within the application design according to the usage interface
TouchRAM: Demo

- Simple Video Slot Machine
  - SlotMachine has Reels with Symbols
  - Has a local GUI
  - Provides remote access
We Built This!

All other design concerns were reused.
Download TouchRAM:

Current features:
Multi-User, Multi-Touch
Concern-Oriented
Structural view (display, edit, weave)
Message view (display, edit, weave)
State view (display, basic edit, weave)
Feature view (simple display)
Simple consistency checks
Basic code generation

Planned features:
Feature view (display, edit, weave)
Impact model support
Elaborate consistency checks
UML import / export
Advanced code generation

Download TouchRAM: