Hybrid Solutions for Feature Interaction Detection and Resolution

Mario Kolberg
Evan Magill
Stephan Reiff-Marganiec

University of Stirling

Muffy Calder

University of Glasgow

Dave Marples

Global Inventures Inc
Context of Research

► Interaction Handling Techniques
  ◆ Offline:
    ◆ not suitable in context of legacy systems, deregulated market
  ◆ Online:
    ◆ information available at runtime too limited for resolution

► HFIG Project
  ◆ 1998-2001: funded by EPSRC, Mitel, Citel
  ◆ joint between Glasgow and Strathclyde (later Stirling) Universities
  ◆ investigate combination of offline & online techniques
Aims and Objectives

Detect and resolve feature interactions
- in the presence of legacy systems
  - (fragile code, no reliable documentation)
- in a deregulated market
  - (third party features, short development periods)

Approach shall
- be embeddable in legacy and new architectures
- not require changes to features or legacy code
- not require design time information
- automatically detect and resolve interactions at runtime
Outline

- Types of Interactions
- Detecting Interactions at Run-time
  - Dave Marples PhD thesis
- Message-Centric Approach
  - Stephan Reiff-Marganiec PhD thesis
- User-Centric Approach
  - Mario Kolberg’s PhD thesis
- Results
- Conclusions
Types of Interactions

- **STI**: Shared Trigger Interactions
  - more than one feature reacts to a trigger
  -> *Message-Centric Approach*

- **SAI**: Sequential Action Interactions
  - one feature’s actions trigger another feature

- **LI**: Looping Interactions
  - special case of SAI’s
  -> *User-Centric Approach*

- **MTI**: Missed Trigger Interactions
  - one feature’s actions prevent triggering another feature
Detecting Interactions at Runtime

- Features are embedded in a cocoon

- Transactional approach:
  1. Commit and rollback
  2. Copies of features
Message-Centric Approach

- Automatically selects good (if not best) resolutions
- Concentrates on handling STI’s

- FM constructs solution space as before
- Pruning and extraction allow to find resolutions
  - Guided by general rules
- Iterative improvement
  - Analyse solution space, define rules, analyse again, refine rules, ...
What are Solutions?

- **Solution**
  - a trace from one or more features running concurrently

- **Solution space**
  - the set of all solutions

- **Resolution**
  - a trace from the solution space that does not violate resolution rules

- **Resolution space**
  - the set of all resolutions
FM with Rule Based Resolution
Resolution

Message Independent Rules
- Duplicate subtrees sharing the same parent
- Largest number of features
- Highest priority
- Choose one

Message Dependent Rules
- Classes of messages (announcements, tones, ...)
- Regular expressions describing undesired behaviours
Example Resolution Rules

Some rules in DESK

- connecting a user to two different resources
- routing to two different locations
- routing a call away and changing user’s state
- routing a call away and connecting to resource
- changing a user’s state and connecting to a resource
User-Centric Approach

- Filtering approach
- Qualification of Sequential Action Interactions
- High-level view on connections
- Detects that certain features change behaviour as perceived by the user
- Simple algorithm
- Good run-time performance
Describing Features

TP: B; (A, B) → (A, C)

- Triggering party
- Connection type
  - Source, destination
  - Original connection
  - Connection after feature activation
  - Parties & Treatment
Interaction Analysis

- Analysis pairs of features
- Compare two feature descriptions according to four rules
  - Single User Dual Feature Control
  - Connection Looping
  - Redirection and Treatment
  - Diversion and Reversing
An Example

Initial Call Attempt, RtC is armed

RtC initiates Callback, OCS blocks

RtC, OCS(POT 1)
The Approach in Action

- Explore behaviour with on-line technique
- Cocoons
- If SAI detected → get connection equation
- Apply 4 rules

RtC: TP: 2; (1, 2) → (2, 1)
OCS: TP: 2; (2, 1) → (2, Treatment)
Single User Dual Feature Control

**CFB:** TP: B; (A, B) $\rightarrow$ (A, C)

**CFU:** TP: B; (A, B) $\rightarrow$ (A, C)

**AR:** TP: A; (B, A) $\rightarrow$ (A, B)

**HL:** TP: A; (A, B) $\rightarrow$ (A, B)
Connection Looping

**CFB**: TP: B; \[(A, B) \rightarrow (A, C)\]

**CFU**: TP: C; \[(A, C) \rightarrow (A, B)\]
### Redirection and Treatment

<table>
<thead>
<tr>
<th></th>
<th>TP:</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CFB:</strong></td>
<td><strong>TP:</strong></td>
<td>C;</td>
<td>(A, C)</td>
<td>→</td>
<td>(A, B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OCS:</strong></td>
<td><strong>TP:</strong></td>
<td>A;</td>
<td>(A, B)</td>
<td>→</td>
<td>(A, Treat)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AR:</strong></td>
<td><strong>TP:</strong></td>
<td>B;</td>
<td>(A, B)</td>
<td>→</td>
<td>(B, A)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OCS:</strong></td>
<td><strong>TP:</strong></td>
<td>B;</td>
<td>(B, A)</td>
<td>→</td>
<td>(B, Treat)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Diversion and Reversing

**CFB**: TP: C; (A, C) $\rightarrow$ (A, B)

**AR**: TP: B; (A, B) $\rightarrow$ (B, A)

**CFB**: TP: A; (B, A) $\rightarrow$ (B, C)

**AR**: TP: B; (A, B) $\rightarrow$ (B, A)
## Results

<table>
<thead>
<tr>
<th></th>
<th>CFU</th>
<th>CW</th>
<th>CFB</th>
<th>OCS</th>
<th>TCS</th>
<th>VMS</th>
<th>RtC</th>
<th>ACB</th>
<th>DND</th>
<th>HL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFU</td>
<td></td>
<td>M</td>
<td>M, U</td>
<td>U</td>
<td>M, U</td>
<td>M, U</td>
<td>M, U, U</td>
<td>M</td>
<td>M, U</td>
<td></td>
</tr>
<tr>
<td>CW</td>
<td></td>
<td></td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>OCS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>U</td>
<td></td>
<td></td>
<td>U</td>
</tr>
<tr>
<td>TCS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M, U</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>VMS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M, U</td>
<td>M</td>
</tr>
<tr>
<td>RtC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M</td>
</tr>
<tr>
<td>ACB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DND</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **10 features**
- **49 interaction scenarios**
- **M → Message-Centric approach (STI, 28 cases)**
  - Found “best” solution for all cases
- **U → User-Centric approach (SAI, 21 cases)**
  - Sometimes subjective decision
Conclusions

Presented approaches
- improve detection mechanism
  - qualification of interactions
- add automated resolution
- are complementary
  - each handles different class of interactions

Future work
- qualification of interactions into desired and undesired as perceived by user
- application in other areas:
  - home networking, component based systems, IP telephony
Any Questions?