Hierarchical Inter-Object Traces for Specification Mining

David Lo†, Shahar Maoz‡
†Singapore Management University ‡ The Weizmann Institute of Science, Rehovot, Israel

Introduction

- Scalability and comprehensibility of results are challenges of specification mining methods
- We mine modal scenarios from execution traces [1]
- We propose the use of object hierarchies as an abstraction mechanism over event traces
- Traces at higher abstraction level are shorter and hence can be mined more efficiently
- Result: Improved scalability & comprehensibility

Mining Steps

i. Abstract the trace to a desired abstraction level.
ii. Mine LSCs from the high-level abstracted traces following the method in [1]
iii. Let user choose a set of mined LSCs to be refined.
iv. The miner refines the LSCs by adding low-level details to the mined LSCs. Order of corresponding events is preserved in the refinement.

Example Mined Live Sequence Charts (LSCs)

LSC Msg Flash: Abstract

<table>
<thead>
<tr>
<th>backend</th>
<th>jabber</th>
<th>ui</th>
<th>title scroller</th>
<th>title flash</th>
</tr>
</thead>
<tbody>
<tr>
<td>getMsgListener(...)</td>
<td>msg(...)</td>
<td>msg(...)</td>
<td>msg(...)</td>
<td>start(...)</td>
</tr>
</tbody>
</table>

LSC Msg Flash: Refined

<table>
<thead>
<tr>
<th>Jabber</th>
<th>Backend</th>
<th>Cht Wndws</th>
<th>Cht Wndw</th>
<th>JID</th>
<th>ChtSplit Pane</th>
<th>Title Scroller, Plugin</th>
<th>Title Flash, Plugin</th>
<th>Flash</th>
</tr>
</thead>
<tbody>
<tr>
<td>getMsgListener(...)</td>
<td>msg(...)</td>
<td>apndMsg(...)</td>
<td>equals(...)</td>
<td>apndMsg(...)</td>
<td>start(...)</td>
<td>start(...)</td>
<td>start(...)</td>
<td></td>
</tr>
</tbody>
</table>

Mined LSCs Description

- Whenever a message is received, the chat window's title bar starts flashing and the message's text is scrolled in the title bar.
- Abstract LSC: inter-package communication
- Refined LSC: inter-class communication
- Additional events
- More concrete participants

Trace Snippet – JETI Messaging App.

...<backend.jabber.jabber.Backend,getMessageListener(...)>
...<backend.jabber.ui.ChatWindows,getMessage(...)>
...<ui.ChatWindows.ui.ChatWindow,appendMessage(...)>
...<ui.ChatWindow.jabber.JID,equals(...)>
...<ui.ChatWindow.ui.ChatSplitPane,appendMessage(...)>
...<ui.ChatSplitPane.plugins.titlescroller.Plugin, start(...)>
...<ui.ChatSplitPane.plugins.titleflash.Plugin, start(...)>
...<titleflash.Plugin,titleflash.Plugin,Flash, start(...)>

[i.e. variant of the above repeat many times during execution]
[object identities are implicit in the trace]

Focus on inter-object behavior
During abstraction: some inter-object events become intra-object and are dropped

Hierarchy of Packages

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>jeti.plugins.titlescroller.Plugin</td>
<td>jeti.plugins.titleflash.Plugin</td>
<td>jeti.plugins.titleflash.Plugin,Flash</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Future Work & Refs

i. Case studies using different types of hierarchies (e.g., packages, component compositions)
ii. Tool support
iii. Integration with past work in [1] and [2]