Anthony Anjorin, Enes Yigitbas, Hermann Kaindl

CONSISTENT RUNTIME ADAPTATION OF USER INTERFACES
Why is UI Development Challenging?

How to retain a constant level of usability for a wide range of varying contexts?
Design-Time (Static) Adaptation

Concrete UI

- iOS
  - m2t
- WWW
  - m2t

Abstract UI

- Choose
  - m2m
- Edit
  - m2m
- Compose
  - m2m

Task and Domain

- Book Flight
  - Choose Departure Airport
  - Choose Destination Airport
  - Choose Flights

- Compose Email
  - Confirm and Send
  - Compose Contents

Not unproblematic, see [1], but relatively well-understood …

Runtime Adaptation is a Necessity

<table>
<thead>
<tr>
<th>Task and Domain</th>
<th>Abstract UI</th>
<th>Concrete UI</th>
<th>Final UI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compose Email</td>
<td>Choose</td>
<td>Choose</td>
<td>.swift</td>
</tr>
<tr>
<td></td>
<td>Confirm</td>
<td>Edit</td>
<td>.html5</td>
</tr>
<tr>
<td></td>
<td>and Send</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Environment**
  - Gets dark, loud, very bright, shaky, ...

- **User**
  - Gets tired, irritable, drunk, ...

- **Platform**
  - Loses connectivity, battery level drops, ...

---

*Choose Departure Airport*

*Choose Destination Airport*

*Choose Flights*
An Adaptive E-Mail Client

- Context model
- Context delta
- Context model space
An Adaptive E-Mail Client

UI model space

consistent UI model

UI model space
An Adaptive E-Mail Client
(Pragmatic) Solution Architecture

Context Model Space  →  Correspondence Models  →  UI Model Space

- **Context** represents Sensor Services used to generate an initial structure.

- **Sensor Services** [Android]

- **UI Model Space** used to generate initial structure

**An “initial” UI is typically generated that cannot be arbitrarily manipulated at runtime.**
(Pragmatic) Solution Architecture

Context Model Space

Correspondence Models

UI Model Space

Modelling Layer

System Layer

Context

represents

used to generate

used to generate

used to generate initial structure

Sensor Services

[Android]

Adaptation Logic

Synchroniser

[Android]

UI@Runtime

notify

modifies

notifies

Sensor Services

observe

modifies

modifies
Challenge 1: System Level Correctness

Mapping of adaptation logic to synchroniser can be buggy (or simply wrong)
Challenge 1: System Level Correctness

The synchronisation engine itself can of course be buggy.
Bx: Use Consistency Checking (CC)
The adaptation logic must make some assumptions about the starting state of the UI. These might be violated as the system evolves...
Challenge 2: Initial UI ≠ Adaptation Logic?

If this UI model were chosen as the initial UI, no single adaptation “rule” would ever apply!
If this fails, then the required "starting conditions" are violated (the initial UI is not in the domain of the consistency restorer)
Challenge 3: Desirable Properties of Adaptation Logic

UI cannot be further adapted if it reaches this dead-end state.
Bx: Laws, Static Analyses, Model Checking?

Termination, Deterministic Behaviour, Totality, Least Surprise, ...

SYNC

Inbox
Drafts
Sent
Spam
Trash
Settings

Log Out
Bx: What should we be improving?

Focus on **other** consistency restorers (not just fwd, bwd). Achieving reasonable **scalability** for CC is currently hard.
Bx: What should we be improving?

(Yet) another argument for **tolerance**: Users can (and often) choose to reject an adaptation (and expect to be able to continue working with the adaptive UI)
It is much too difficult to develop systems that use bx technology. We need architectural templates, standard plug-n-play interfaces, adapters, converters, patterns, …
Consistent Runtime Adaptation of User Interfaces

Context Model Space       Correspondence Models       UI Model Space

<table>
<thead>
<tr>
<th>Context</th>
<th>Sensor Services</th>
<th>Adaptation Logic</th>
<th>UI@Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>observe</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>represents</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

System Layer

Modelling Layer

used to generate
used to generate
used to generate initial structure

used to
used to
used to

[ Android ]
[ Android]
[ Nools ]
[Android]

plug to sync

consistency