SUPPLE: Automatically Generating User Interfaces

Krzysztof Gajos
and Daniel S. Weld
Motivation

• Current interfaces: complex & “One size fits all”
  ⇒ Adapt to **users** and **tasks**

• Variety of display devices & interaction contexts makes hand-designed interfaces expensive
  ⇒ Adapt to **device characteristics**

⇒ **Automatic interface generation is a scalable solution**
Approach

• Develop abstract representation for:
  • Interfaces
  • Display devices
  • Users

• Automatically generate interfaces from the abstractions
SUPPLE Architecture

- Interface Model
  Application or Appliance

- SUPPLE
  Device Model
  Target Device
  Display

- Target Device

- User Model
  User's Info Space
SUPPLE Architecture

- Interface Model
- Application or Appliance
- Device Model
- Target Device
- User Model
- User's Info Space
Examples of Applications

• A Classroom Controller
• An interactive FTP Client
• A distributed jukebox
• A stereo controller
Automatically Rendered Interfaces for Classroom Controller
Road Map

- Introduction
- Modeling
  - Interfaces
  - Devices
  - Users
- Inner workings of SUPPLE
- Results and evaluation
- Related Work
- Conclusions and future work
Modeling User Interfaces

- **simple types:** \( \text{int} | \text{float} | \text{string} | \text{bool} \)

- **derivative types:** \( \langle \tau, C_\tau \rangle \)

- **vectors:** \( \text{vector}(\tau) \)

- **containers:** \( \{\tau_i \mid i \in 1 \ldots n\} \)

- **actions:** \( \tau \to \text{nil} \)
Modeling User Interfaces

- simple types: \( \text{int}|\text{float}|\text{string}|\text{bool} \)
- derivative types:
- vectors:
- containers:
- actions:
Modeling User Interfaces

- simple types:
- derivative types: \( \tau, C_\tau \)
- vectors:
- containers:
- actions:

\(<\text{int}, [0,10]>\)

\(<\text{string}, \{\text{Computer 1, Computer 2, video}\}>\)
Modeling User Interfaces

- simple types:
- derivative types:
- vectors:
- containers: \( \{ T_i \mid i \in 1 \ldots n \} \)
- actions:

{bool, <string, \{Computer 1, Computer 2, Video\} >}
Modeling User Interfaces: Optional Attributes

- Label
- Set of likely values
- Exact value required
- ...

13
Road Map

- Introduction
- Modeling
  - Interfaces
  - Devices
  - Users
- Inner workings of SUPPLE
- Results and evaluation
- Related Work
- Conclusions and future work

Diagram:
- Interface Model
  - Application or Appliance
  - SUPPLE
  - Device Model
  - User Model
  - User's Info Space
  - Target Device
  - Display
Modeling Device Capabilities

- Device constraints
- Available widgets
  - Primitive widgets
  - Container widgets
- Match cost function for primitive widgets
- Navigation cost function for containers
Examples of Available Widgets

Pointer and Keyboard

Touch Screen
Match Cost Function
For Primitive Widgets

Match\((\langle \text{int}, [0,10]\rangle, \text{Level 7}, \text{change value from 7 to 8} \rangle = 3

Match\((\langle \text{int}, [0,10]\rangle, \text{Level 7}, \text{change value from 7 to 10} \rangle = 5

Match\((\langle \text{int}, [0,10]\rangle, \text{, change value from 7 to 10} \rangle = 1
Navigation Cost Function
For Container Widgets

- Inputs:
  - A transition type
  - A container widget
- Output: an estimate of user effort to navigate the interface
Example of Navigation Cost
Example of Navigation Cost
Road Map

- Introduction
- Modeling
  - Interfaces
  - Devices
- Users
- Inner workings of SUPPLE
- Results and evaluation
- Related Work
- Conclusions and future work
Modeling Users With Traces

- **Trace** as a model of usage pattern composed of **Trails** [Wexelblat and Maes, CHI 1999]

- **Trail** as a “coherent” sequence of elements the user interacted with

- Trail format independent of rendering

\[
T = \{ <\text{root}, -, -> \\
<"\text{Left light:Power}", \text{off}, \text{on}> \\
<"\text{Vent}", 1, 3> \\
<"\text{Projector:Input}", \text{video}, \text{computer 1}> \\
.... \}
\]
Road Map

- Introduction
- Modeling
  - Interfaces
  - Devices
  - Users
- Inner workings of SUPPLE
- Results and evaluation
- Related Work
- Conclusions and future work
Rendering As Optimization

• **Cost function** ($) -- estimated user effort to manipulate a rendering of the interface

• Inputs:
  • A rendering
  • A user trace

• Cost function derived from
  • Match cost function
Rendering As Optimization

\[ \$(\phi, \mathcal{T}) \equiv \sum_{T \in \mathcal{T}} \sum_{i=1}^{\vert T \vert - 1} N(\phi, e_{i-1}, e_i) + M(\phi(e_i), v_{old_i}, v_{new_i}) \]

Iterate over all entries in all trails
Cost of navigating between successive elements
Cost of manipulating the value of the current element
Rendering Algorithm: Properties

• A constrained branch-and-bound search algorithm (like A*) -- at each step assign a widget to an abstract interface element

• Guaranteed to find a rendering with the lowest cost
Design Choices:

- **Constraint propagation methods:**
  - None, Forward Checking (FC), Full

- **Variable ordering:**
  - Bottom-up, Top-down, MRV

- **Admissible Heuristic:**
  - Estimate of the total cost of the entire interface
Performance

Time in seconds

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>none/bottom-up</td>
<td>89.1</td>
</tr>
<tr>
<td>FC/bottom-up</td>
<td>351.3</td>
</tr>
<tr>
<td>full/bottom-up</td>
<td>70.8</td>
</tr>
<tr>
<td>full/MRV</td>
<td></td>
</tr>
<tr>
<td>full/top-down</td>
<td></td>
</tr>
</tbody>
</table>
Road Map

- Introduction
- Modeling
  - Interfaces
  - Devices
  - Users
- Inner workings of SUPPLE
- Results and evaluation
- Related Work
- Conclusions and future work
Results And Evaluation

- Adapting to device characteristics
- Building an interactive interface
- Conceptual study
  - Adapting to usage patterns
Adapting to Device Characteristics
An Interactive Interface

![FTP Client Login Interface](image)
An Interactive Interface
An Interactive Interface
Preliminary Study

• Four “experts”

• Same widget library and conditions as for SUPPLE
Preliminary Study

Human Designer A

SUPPLE
Preliminary Study

Human Designer B
Preliminary Study

Human Designer C
Preliminary Study

Human Designer C

SUPPLE with a trace
Road Map

- Introduction
- Modeling
  - Interfaces
  - Devices
  - Users
- Inner workings of SUPPLE
- Results and evaluation
- Related Work
- Conclusions and future work
Related Work

- Personal Universal Controller (PUC) [Nichols, et al, UIST’02]
- iCrafter [Ponnecanti, et al, UbiComp’01]
- Xweb [Olsen, et al, UIST’00]
- GADGET [Fogarty, et al, UIST’03]
- XIML [Puerta & Eisenstein, IUI’02]
Contributions

- Formal definition of the problem
- User-specific rendering
- An efficient algorithm
- Evaluation of speed and quality
Future Work

• Cross-device consistency (with Anthony Wu)
• Learning of match and navigation cost functions
• Transforming interfaces
• Explicit customization
• Incorporate design heuristics
• More complex applications
• Wider range of devices
Acknowledgments

• The anonymous subjects

• Comments from: Mark Adler, Alan Borning, Gaetano Borriello, Tessa Lau, Jeffrey Nichols, Steven Wolfman, Alexander Yates
More Information

• Demo tonight!

• SUPPLE:
  http://www.cs.washington.edu/ai/supple

• Krzysztof Gajos:
  kgajos@cs.washington.edu

• Daniel S. Weld:
  weld@cs.washington.edu