SUPPLE: Automatically Generating Personalizable User Interfaces

Krzysztof Gajos,
Raphael Hoffmann
and Daniel S. Weld

University of Washington
Computer Science & Engineering
SUPPLE: Automatically Generating Personalizable User Interfaces

Krzysztof Gajos,
Raphael Hoffmann
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
• Also: custom built applications
  ⇒ Adapt to data structures and processes

⇒ Automatic interface generation is a scalable solution
Approach

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

• Automatically generate interfaces from the abstractions
  • Generated “optimal” interfaces w.r.t. a usage difficulty estimate
SUPPLE Architecture

Interface Model

Application or Appliance

SUPPLE

Device Model

Target Device

User Model

User's Info Space

Application or Appliance

Interface Model

Device Model

Target Device

User Model

User's Info Space
SUPPLE Architecture

- Interface Model
- Application or Appliance
- SUPPLE
- Device Model
- Target Device
- User Model
- User's Info Space
Automatically Rendered Interfaces for a Classroom Controller
Automatically Rendered Interfaces for a Classroom Controller
Email Client

30.06 06:30 PM supple@softhome.net Test
28.06 11:24 PM Isabel Raikar <raikar@unterreuth.de> AW:
28.06 11:24 PM Isabel Raikar <raikar@unterreuth.de> AW:
25.06 08:18 AM supple@web.de Hey!
[ 25.06 01:02 AM supple@web.de Re: TEST]
18.06 07:51 PM MailDeliverySubsystem <MAILER-DAEMON>

Details

Senders: Isabel Raikar <raikar@unterreuth.de>
Date: 28.06 11:24 PM
Recipients: supple@web.de Raphi <Raphael.hoffmann@unterreuth...
Subject: AW: modified Templates

Content

Sorry for replying late, we may be starting a call center in a month so I
am bit busy, but we have worked on your modifications and are sending the
templates, please suggest us some colours if you dont like what we are using.

I will be available tomorrow evening our time by around 4-5.

Have a Great Weekend
Best Regards
Prasad

Click!
Rendering Algorithm:

Intuition:

- **Light Level**: \( \text{int}, [0,10] \)
- **Power**: bool
- **Light**: \( \text{int}, [0,3] \)
- **Light Bank**: \( \text{int}, [0,3] \)
- **A/V**: bool
- **Screen**: bool
- **Input**: \( \text{string}, \{\text{data1, data2, video}\} \)
- **Vent**: \( \text{int}, [0,3] \)
- **Projector**: bool
- **Classroom**: \( \text{int}, [0,3] \)
Rendering Algorithm: Intuition

Light Level: \( \text{!}: <\text{int}, [0,10]> \)
Power: \( \text{!}: \text{bool} \)
Light: \( \text{!}: \{ , \} \)
Light Bank: \( \text{!}: \{ , , \} \)
Light ... Light ... {  ,  ,  }
Input:
!: <string, \{data1, data2, video\}>
Vent:
!: <int, [0,3]>
Power: 
!: bool
Screen:
!: bool

Rendering Algorithm:

Intuition
Usage Patterns and Automatic Interface Generation

• Cast interface generation as an optimization problem

• Find “best” interface for the recorded usage pattern
Adapting To Usage Patterns

SUPPLE with an empty trace

SUPPLE with a “lights-heavy” trace
Adaptation Strategies

• Complete Makeover
• Content Eliding
• Altered Prominence
• Split Interfaces
Complete Makeover

SUPPLE with an empty trace

SUPPLE with a “lights-heavy” trace
Content Eliding
Altered Prominence
Split Interfaces
Adaptation Strategies

- Complete Makeover
- Content Eliding
- Altered Prominence
- Split Interfaces
Example
Example
Example

Click!
Example
Adaptation
Customization
Customization
Customization

- Orientation: Portrait, Landscape
- Duplex: Enable, Disable
- Pages per sheet: 1, 2, 3, 4, 5
- Poster Printing: none, 2x2, 3x3, 4x4
- Copies: 1, 2, 3, 4, 5
- Specific Pages: From 1 To 1
- Number of copies: 1
- Print to file
Customization
Customization
Adaptation: Problem Statement

• **Inputs**: UI description, Trace, Device Model, *fraction of space for shortcuts*

• **Outputs**:
  - Optimal *set of shortcuts* given rendering & Trace
  - Optimal “static” interface *given shortcuts*, Trace, etc.

• **Hard**: shortcuts depend on rendering; rendering depends on shortcuts
Adaptation: Problem Statement

- **Inputs**: UI description, Trace, Device Model, fraction of space for shortcuts

- **Outputs**:
  - Optimal *set of shortcuts* given rendering & Trace
  - Optimal “static” interface *given shortcuts*, Trace, etc.

- **Hard**: shortcuts depend on rendering; rendering depends on shortcuts

- **Not So Hard**: in practice we do not need to worry about rendering depending on shortcuts
Customization
Supported Customizations

• Setting default values
• Copying functionality
• Removing functionality
• Constraining rendering
Customization: Novel Contributions

- Out of order undo
- Generalization
Why Should We Care

• Different users use the same software to accomplish different tasks
• Users differ in their cognitive, perceptual and motor abilities
• Users differ in their sense of aesthetics
• Users own very different display devices
• Organizations use individual processes that require custom-built software and thus custom-built interfaces
Our Contributions

• User Interface generation as optimization
• Efficient algorithm
• Built-in adaptation
• Built-in customization (with out-of-order undo and generalization)
More Information

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

Krzysztof Gajos: kgajos@cs.washington.edu
Raphael Hoffmann: raphaeh@cs.washington.edu
Daniel S. Weld: weld@cs.washington.edu

DEMO

Today, 1:30 - 4pm
in CSE 603 (HCI Lab)