Design as Exploration
Creating Interface Alternatives through Parallel Authoring and Runtime Tuning

Björn Hartmann
Loren Yu, Abel Allison, Yeonsoo Yang
Scott R. Klemmer
Exploration of alternatives is central to design.
Prototypes for the Microsoft mouse
From Moggridge, Designing Interactions, Ch2
Tohidi et al, CHI 2006
How can tools support the creation of user interface alternatives?

Color picker #1

Color picker #2

Color picker #3
“[D]esigners frequently wanted to have multiple designs side-by-side [...]”

However [...] there is no built-in way in today’s implementation tools to have two versions of a behavior operating side-by-side.”

Myers et al., VL/HCC 2008
Interaction Designers Write Code

Adobe Flash

Arduino / Processing

Adobe Dreamweaver

Xcode (iPhone)
Programming requires working with two representations.

Editor: author behavior

Runtime: observe behavior
Outline

- Requirements for Alternatives of Programmed Interactions
- Juxtapose for Desktop Interactions: System & Evaluation
- Alternatives off the Desktop
- Related & Future Work
3 Requirements for Programmed Interactions
(based on 3 interviews & code inspection)

- Manage parameter variations
- Manage code alternatives
- Access variations & alternatives at runtime

```java
public int ATTENUATION = 20; // = 10; // = 50;
public int BASE_ALPHA = 50;
public int POSITIVE_RESPONSE = 6;
```
3 Requirements for Programmed Interactions

- Manage parameter variations
- Manage code alternatives
- Access variations & alternatives at runtime

```java
public static int calculateNextSize(int[][] currentSizes, int i, int j) {
    float denominator = 0;
    int sumOfNeighbors = 0;
    int maxOfNeighbors = 0;
    if (i != 0) {
        sumOfNeighbors += currentSizes[i - 1][j]; denominator += 1;
        maxOfNeighbors = currentSizes[i - 1][j];
        // if (j != 0) sumOfNeighbors += currentSizes[i - 1][j - 1]; denominator += .5;
        // if (j != currentSizes[0].length - 1) sumOfNeighbors += currentSizes[i - 1][j + 1]; denominator += .5;
    }
    if (i != currentSizes.length - 1) {
        sumOfNeighbors += currentSizes[i + 1][j]; denominator += 1;
        if (currentSizes[i + 1][j] > maxOfNeighbors) maxOfNeighbors = currentSizes[i + 1][j];
        // if (j != 0) sumOfNeighbors += currentSizes[i + 1][j - 1]; denominator += .5;
        // if (j != currentSizes[0].length - 1) sumOfNeighbors += currentSizes[i + 1][j + 1]; denominator += .5;
    }
    return 1;
}
```

Processing code brought in by an interviewee
3 Requirements for Programmed Interactions

- Manage parameter variations
- Manage code alternatives
- Access variations & alternatives at runtime
Juxtapose: Source alternatives...

A B C D

____________________

____________________

____________________

____________________

____________________

____________________
... are executed in parallel,
and tuned through an generated UI.
Parallel Editing

Juxtapose editor for ActionScript 2. Generates Flash files.

```actionscript
function FlashApplication() {
    _root["bar"].xscale = 0;
    _root["title_txt"].text="Slow Navy";
    _root["percentage_txt"].text="0%";
    //this line sets up the link to the source code editor
    var adapter:MultiplicIDEConnector3 = new MultiplicIDEConnector3;
    // set onEnterFrame handler to this object's function
    _root["control"] = this;
    _root.onEnterFrame = function() {
        this["control"].onEnterFrame();
    };
    listenerObject.click = function(eventObject:Object) {
        _root["control"].onClick();
    }
    _root["start_btn"].addEventListener("click", listenerObject);
}

public function map(x:Number):Number {
    return x+Math.sin(x*Math.PI)*5)/20;
}
```
Parallel Editing

[Image of a software interface with options for Alternative 1 and Alternative 2]
Parallel Editing

Linked Editing: Toomim 2004
Parallel Editing

```
FlashApplication {

    var showLocalStreets:Boolean = false;
    var showLocalNames:Boolean = true;
    var showMajorStreets:Boolean = false;
    var showMajorNames:Boolean = true;
    var showHighways:Boolean = true; // show highways and interstates?
    var showPOI:Boolean = true; // show points of interest?

```
Example: Rethinking the Progress Bar
Harrison et al., UIST 2007

Progress bar test application

Progress profiles
function FlashApplication() {

    _root["bar"].xscale = 0;
    _root["title_txt"].text="Linear";
    _root["percentage_txt"].text="0%";
    //this line sets up the link to the source code editor
    var adapter:MultilpicIIDEConnector3 = new MultilpicIIDEConnector3(t

    // set onEnterFrame handler to this object's function
    _root["control"] = this;
    _root.onEnterFrame = function() {
        this["control"].onEnterFrame();
    };

    listenerObject.click = function(eventObj:Object) {
        _root["control"].onClick();
    }

    _root["start_btn"].addEventListener("click", listenerObject);

    }

    public function map(x:Number):Number {
        return x;
    }

    public function onClick() {
        x=0;
    }
}
Implementing Parallel Editing: The Cursor Correspondence Problem

Straightforward solution:
Longest common subsequence algorithm
\text{diff}(A,B,\ldots,N)

Run during idle times
Limits of Parallel Editing & Execution

- Level of granularity for editing: Extension to multi-file projects is not trivial.

- Can the user make sense of behaviors running in parallel?
Limits of Event Echoing

Delete this file?

Yes
No

Delete this file?

Yes
No
Ok
Parameter Tuning

```javascript
var angmax: Number = 270;
var dimin: Number = 61;
var maxchild: Number = 3;
var maxrec: Number = 4;
```
Example: Tuning Phosphor

Baudisch, UIST 2006
Example: Tuning Phosphor
Baudisch, UIST 2006
Generating Tuning Interfaces

Juxtapose User Library

User’s Application:
// …
Number A = 10;
Boolean B = true;

Send variable info

Inspect

Juxtapose User Process

User Process
Generating Tuning Interfaces

Juxtapose User Library

User’s Application:
//...
Number A = 10;
Boolean B = true;

Juxtapose User Library

Send tuning message

Update

Juxtapose Process

User Process
What parameter ranges should be chosen?

0-0?  
0-10?  
0-255?  
Number.minValue – Number.maxValue?

Any nontrivial program property is undecidable.
Take a guess, then give control to the user.

```javascript
var angmax:Number = 270;  // @RANGE 0..360
var dimin:Number = 61;    // @RANGE 0..100
var maxchild:Number = 3;  // @RANGE 0..8
var maxrec:Number = 4;    // @RANGE 0..8
```

At authoring time: source annotations
Limits of Tuning

- Which variables matter?
  Juxtapose extracts only globals
Limits of Tuning

- Which variables matter?
  Juxtapose extracts only globals

- Are variables read again?
  Juxtapose uses callbacks

```javascript
01 var tunable = 5; //@RANGE 0..100
03 var callback = function(varName, oldVal, newVal) {
04   redraw();
05   return newVal;
06 }
07 this.watch('tunable', callback);
```
Understanding User Strategies & Measuring Benefits

- Questions:
  - How does Juxtapose fit into interaction prototyping practice?
  - Can we quantify the benefits?

- Method:
  - Lab evaluation, N=18, students
  - 75-90 minute sessions with design tasks & survey
Mapping Task

Given ActionScript code that loads a multilayered map, develop navigation options for a handheld GPS prototype for pedestrians and bicyclists.
Alternatives for Map Navigation & Rendering
Participants’ Strategies

- Linked vs. unlinked editing with multiple alternatives applied by all participants.
- Alternatives used both for different scenarios, as well as “scratch space”
- Snapshots important to save state
Suggested improvements

- Automate code restructuring for callbacks
- Introduce direct manipulation for parameters in user’s application (when possible)
Tree Matching Task

Search in 4D parameter space. Given recursive drawing code for this:

Produce these:
Mean Parameter Changes Tested per Minute

- Changes/minute
- Control
- Juxtapose

- Control: 2.60
- Juxtapose: 64.26
Tree Matching Task: Mean Completion Times by Tree

- Tree 1: not significant
- Tree 2: not significant
- Tree 3: p<0.01
- Tree 4: p~0.01

The graph shows the mean completion times for each tree under control and Juxtapose conditions, with error bars indicating the variability.
Outline

- Requirements for Alternatives of Programmed Interactions
- Juxtapose for Desktop Interactions: System & Evaluation
- Alternatives off the Desktop
- Related & Future Work
Juxtapose Mobile

Alt.1  Alt.2  Alt.3

_________________________
_________________________
_________________________
_________________________
_________________________

Alternative 1  Alternative 2  Alternative 3
Two menu navigation alternatives
Juxtapose Mobile

Juxtapose Runtime on PC

TCP/IP

Juxtapose Wrapper
User’s Flash Application

Alt. 1
var1
Alt. 2
var2
Juxtapose for Microcontrollers

Arduino:
8-bit Atmel AVR RISC chip programmed in C with gcc
How might one implement variable tuning in a language without reflection?
Parse code and build symbol table

```java
int dly = 25;
boolean blink=false;
void setup() {...}
//...
```

<table>
<thead>
<tr>
<th>Var. Name</th>
<th>Type</th>
<th>Pointer</th>
</tr>
</thead>
<tbody>
<tr>
<td>“dly”</td>
<td>int</td>
<td>&amp;dly</td>
</tr>
<tr>
<td>“blink”</td>
<td>boolean</td>
<td>&amp;blink</td>
</tr>
</tbody>
</table>
void initVarTable(void) {
    varTable[1].varName = PSTR("dly");
    varTable[1].varType = VAR_TYPE_INT;
    varTable[1].varPtr = &dly;

    varTable[2].varName = PSTR("blink");
    varTable[2].varType = VAR_TYPE_BOOLEAN;
    varTable[2].varPtr = &blink;
}

// plus a bunch of communication code...
Serial message:

```
tune "dly" value 100
```

```
*(varTable["dly"].varPtr) = 100;
```
## So What’s Different?

<table>
<thead>
<tr>
<th>Target Platform</th>
<th>How are alternatives executed? Why?</th>
<th>How do users interact with alternatives?</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Laptop" /></td>
<td>In parallel</td>
<td>Sequentially or in parallel</td>
</tr>
<tr>
<td><img src="image" alt="Phone" /></td>
<td>In parallel, because commodity hardware is (relatively) cheap</td>
<td>Sequentially (parallelism is possible, but less appealing)</td>
</tr>
<tr>
<td><img src="image" alt="Custom Hardware" /></td>
<td>Sequentially, because custom hardware is expensive to build</td>
<td>Sequentially</td>
</tr>
</tbody>
</table>
What’s Important?

A structured approach to alternatives eliminates cruft and enables more exploration.
What’s Important?

A structured approach to alternatives eliminates cruft and enables more exploration.

Tool support requires connecting authoring and execution environments.

3 Techniques:

- Linked editing to manage code alternatives
- Execute set of programs side-by-side
- Auto-generate tuning interface
Related Work

Augmented Development Tools

*Partials*, Terry, PhD Thesis 05
*Amulet Inspector*, Myers, IEEE ToSE 97
*JPie*, Goldman, Sci. of Comp. Prog. 98
*ChucK*, Wang, NIME 04
*Adobe Image Foundation Tookit*, 08

Alternatives in Other Domains

*Design Galleries*, Marks, SIGGRAPH 97
*Spreadsheets for Images*, Levoy, SIGGRAPH 94
*Spreadsheets for Visualization*, Chi, IEEE CGA 98
*Parameter Spectrums & Parallel Pies*, Terry, UIST 02 & CHI 04
*Subjunctive Interfaces*, Lunzer, TOCHI 08
*TEAM STORM*, Hailpern, CHI 07

(only first authors listed)
Related Work

Augmented Development Tools

**Partials**, Terry, PhD Thesis 05

**Amulet Inspector**, Myers, IEEE ToSE 97

**JPie**, Goldman, Sci. of Comp. Prog. 98

**ChucK**, Wang, NIME 04

**Adobe Image Foundation Tookit**, 08

```java
System.out.println("Hello " + partial(String: "World", "Mundo", "Mom"));
```

**Language-level support**
Related Work

Alternatives in Other Domains

*Design Galleries*, Marks, SIGGRAPH 97

*Spreadsheets for Images*,
Levoy, SIGGRAPH 94

*Spreadsheets for Visualization*,
Chi, IEEE CGA 98

*Parameter Spectrums & Parallel Pies*,
Terry, UIST 02 & CHI 04

*Subjunctive Interfaces*, Lunzer, TOCHI 08

*TEAM STORM*, Hailpern, CHI 07

Alternatives are generated automatically from formal specification
Related Work

Alternatives in Other Domains

**Design Galleries**, Marks, SIGGRAPH 97

**Spreadsheets for Images**, Levoy, SIGGRAPH 94

**Spreadsheets for Visualization**, Chi, IEEE CGA 98

**Parameter Spectrums & Parallel Pies**, Terry, UIST 02 & CHI 04

**Subjunctive Interfaces**, Lunzer, TOCHI 08

**TEAM STORM**, Hailpern, CHI 07

*Exploit spatial syntax*
Current Work: Revisions for Interaction Design

Revisions & comments for text in Microsoft Word

Revisions & comments for interactions in d.tools