

Employing Patterns and Layers for Early-Stage Design and Prototyping of Cross-Device User Interfaces

James Lin

USER Group

IBM Almaden Research Center

650 Harry Rd

San Jose, CA 95120-6099 USA

jameslin@us.ibm.com

James A. Landay

DUB Group

Computer Science and Engineering

University of Washington

Seattle, WA 98195-2350 USA

landay@cs.washington.edu

ABSTRACT

Designing UIs that run across multiple devices is increasingly important. To address this, we have created a prototyping tool called Damask, which targets web UIs that run on PCs and mobile phones, and prompt-and-response style voice UIs. In Damask, designers sketch out their design for one device while using *design patterns* to specify higher-level concepts within their design. Damask's patterns include pre-built UI fragments that are already optimized for each device. Designers also use *layers* to specify which UI parts are common across devices and which are specific to one device. Damask uses the sketches and patterns to generate designs for the other devices, which the designers can refine. A study performed with 12 professional UI designers found that, in the early stages, designers using patterns and layers in Damask created cross-device UIs that are rated at least as good as those created without patterns and layers, without more time.

ACM Classification: D.2.2 [Software Engineering]: Design Tools and Techniques – User Interfaces, H.5.2 [Interaction Interfaces and Presentation]: User Interfaces – Prototyping

General terms: Design

Keywords: cross-device user interfaces, mobile computing, voice user interfaces, web user interfaces, prototyping

INTRODUCTION

People often use a variety of computing devices, such as PCs, PDAs, and mobile phones, to access the same information and applications. The user interface needs to be different for each device, due to different input and output constraints. For example, the home page of a news web site typically has dozens of links, but it would not make sense to display that many links on a mobile phone, or recite all of them in a voice UI. Currently, designers designing cross-device UIs either have to design a UI separately for each device, which is time consuming and error prone, or use a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2008, April 5–10, 2008, Florence, Italy.

Copyright 2008 ACM 978-1-60558-011-1/08/04...\$5.00

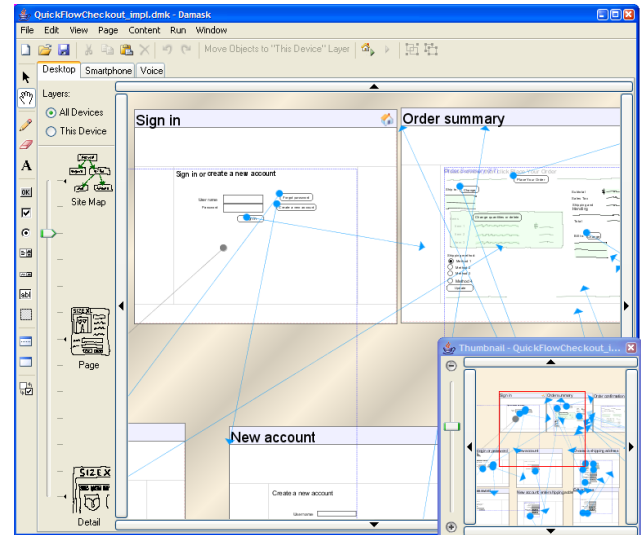


Figure 1. Damask's user interface.

program to automatically generate interfaces, which often leads to awkward interaction.

Current prototyping tools do not address the specific needs of cross-device UI design. A tool that would allow designers to quickly prototype UIs for multiple devices at the same time would allow them to explore a broader design space, potentially improving their final UIs.

Two concepts can aid in prototyping cross-device UIs: *design patterns* for representing common design solutions that have very different device-specific implementations, and *layers* for specifying which parts of a UI are common across all devices and which are specific to one device.

To evaluate these concepts within this domain, we incorporated them into an early-stage design and prototyping system called Damask (see Figure 1). To use Damask, a designer sketches out a UI design for one device, using patterns and layers as needed. At the same time, Damask constructs an abstract model from which it generates the UIs for other devices. The designer can then refine the generated UIs if necessary. Damask supports prototyping three types of interfaces: web-style interaction on a PC, web-style interaction on a mobile phone display, and a prompt-and-response style voice UI on a phone or PC.

The contributions of this work are 1) tool support for using design patterns and layers as a novel technique for cross-device UI design and prototyping, and 2) an evaluation that shows that designers using patterns and layers created cross-device UIs that were rated at least as good as, and in some ways better than, designing for each device from scratch, without taking more time.

In this paper, we first describe related work and the preliminary work that led to Damask. We then describe Damask itself, and report on our evaluation of design patterns and layers in cross-device UI design.

RELATED WORK

Our work is closely related to *model-based user interfaces*, which are used to design UIs based on an abstract model of the interface rather than visual appearance (see [29] for a summary and retrospective). This allows rendering the UI in multiple ways, such as using a drop-down list or presenting a voice menu instead of radio buttons for selecting an item. Several model-based design systems specifically address the issue of creating UIs for multiple devices [2, 7, 10, 20].

However, model-based UI tools often force designers to think at a high level of abstraction too early in the design process. Designers are accustomed to starting with concrete interfaces [22, 33]. In addition, specifying models often resembles programming, at which most designers are unskilled. Specifying models thus impedes their main task of designing UIs. UI Pilot [27] addresses this problem by presenting the model to the designer as a simple outline, optionally inferring it from a set of data.

PIMA [5], Hopi [3], and Microsoft's ASP.NET mobile controls [19] allow designers to design cross-device web applications, by laying out abstract widgets linearly in a constrained UI form designer. The representation is then converted into concrete device-specific UIs. In contrast, designers in Damask always work with concrete UIs.

Ding and Litz have created a prototype so that designers can annotate a desktop UI to describe how it should be transformed into a mobile UI [9]. Damask also focuses on concrete UIs instead of abstract ones, but the design patterns in Damask allow designers to reuse sets of transformations for multiple designs.

There also has been much work on automatically transforming interfaces meant for one device or modality to another. Much of it has focused on transforming existing desktop web UIs to handheld UIs at run-time (e.g., [6, 18]), which unfortunately often results in awkward interaction. Others have worked on converting GUIs to audio interfaces [21, 25], mostly to benefit the blind and visually impaired. With most of these tools, designers cannot modify the resulting transformation. By contrast, Damask is not meant for the implementation of the final UI, so designers are free to modify the generated UI.

GADGET [12] is an optimization toolkit, which can take an existing collection of UI controls and generate a layout for

them. It would be useful to Damask for creating, for example, a smartphone UI from a desktop UI, since many of the controls are the same, but less useful for generating voice UIs, where the controls are very different. SUPPLE [13] also uses optimization techniques to generate a UI, from an abstract UI model, a device model, and a trace of UI events created by a user. Damask could potentially use SUPPLE's algorithms to improve the generation of device-specific UIs.

There are several platforms that allow programmers to create a description of the UI of a remote control at a high level, which can then be realized on a variety of devices (e.g., [24, 26]). The target domain of universal remote controls is narrow and the UIs that are rendered from the abstract description must be appealing and useful immediately, without additional tweaking. Damask, on the other hand, allows designers to design a broader set of UIs, but the generated UIs will most likely be modified by the designers before being released.

Design patterns are increasingly popular for representing common UI design problems and their solutions, but few design tools directly support patterns (see [8] for a survey).

STUDY OF CURRENT PRACTICES

To get a better understanding of how designers currently design cross-device UIs, we interviewed nine UI designers across eight companies who worked on cross-device web UI projects. All of these projects targeted desktop PCs and mobile phones, and all but one also targeted PDAs. We focused our questions on how the designers handled design for multiple devices in their projects. These are some of the implications we drew from our interviews for cross-device UI design tools (for more detail, see [16 (Chapter 2)]).

Presenting retargeting results to one designer is useful. All of the designers designed the UI for a particular feature across multiple devices (as opposed to designing the entire UI for a particular device). Therefore, a tool that takes a UI for one device and presents UIs for other devices fits within current practice.

Designers need help maintaining consistency across devices. Consistency was identified as a major burden of cross-device design. The challenge is to keep the appropriate content consistent across devices, while letting the layout and navigation change to fit the device.

Support design patterns. Using design patterns as the foundation of a cross-device UI design tool is a sound idea (in fact, two of our participants had their own pattern collections), but allowing designers to create their own patterns is essential for the tool's long-term usefulness.

PROTOTYPE OF CROSS-DEVICE DESIGN TOOL

In addition to our interviews, we wanted to get more detailed feedback about how an early-stage cross-device design tool should behave and what features it should have. Since there were no such tools, we quickly designed and evaluated a prototype of one, called *HopiSketch*, which targeted PCs and Palm handheld devices (please see [16

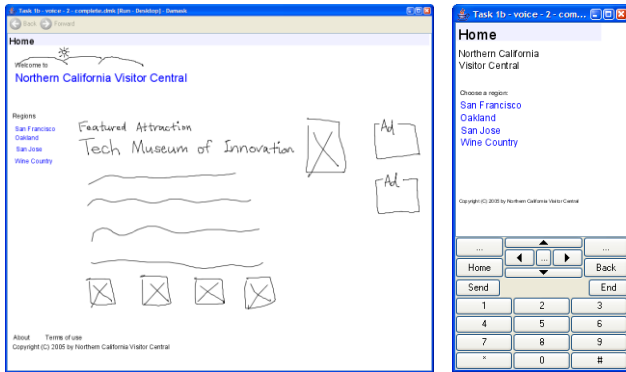


Figure 2. The Run window for desktop (left) and smartphone (right) UIs.

(Chapter 3)] for a detailed description of the tool and evaluation). From their experiences with HopiSketch, the six participants of a user study described how a tool for retargeting designs could be more useful, including:

Provide explicit control over retargeting. Four of the designers said they would like to be able to explicitly tag which sections of a page should be carried over to the retargeted design, and which should be omitted.

Support iterative design. Many designers would rather design a little for one device, retarget, look at the results, design a bit more for the first device, and so on. The tool also should be able to propagate changes made in a generated device-specific design back to the original.

Improve page splitting. All of the designers said that the algorithms for rearranging and splitting up content were important. One designer said that the tool should not split pages automatically, but instead create longer pages and allow designers to split the pages themselves.

Support different classes of devices. A cross-device design tool should be able to support the design of applications whose user interfaces have very different interaction flows depending on the device. HopiSketch and other model-based tools do not handle such design activities well because they only transform at the page and widget level. Higher levels of abstraction, such as design patterns, are needed.

DAMASK

Building on what we learned in our interviews and prototype evaluation, we created a tool for the early-stage design and prototyping of cross-device UIs called Damask. Using the approach that we outlined in [17], Damask includes a catalog of design patterns for use in designing desktop-based web sites. Some patterns are also useful for mobile phone and voice UIs, and these patterns include UI sketches that address the patterns' design problem—one for desktop web, one for mobile web, and one for prompt-and-response voice UIs.

With Damask, designers prototype their UI designs by sketching them and by adding design pattern sketches to their design for one device. While they do this, Damask

generates corresponding UI design sketches for the other two devices, which the designers can modify if desired. Finally, designers can use Damask to test and interact with their design sketches in a Run mode.

Damask's user interface is similar to other design tools, such as DENIM [23] and SUEDE [28] (see Figure 1). A pannable, zoomable canvas contains the designer's user interface design. The design includes which patterns it is using, as denoted by a blue outline and the name of the pattern. The tabs above the canvas let designers choose which device UI they want to view: desktop, smartphone, or voice. The designer can view different device-specific UIs at the same time in multiple windows.

DESIGNING DESKTOP AND MOBILE PHONE UIs

In Damask, a desktop or mobile web UI design consists of pages of content, linked with arrows defining behavior.

Pages, Controls, and Arrows

Pages represent a web page or a smartphone screen. The designer creates them by selecting the pencil tool and dragging out a rectangle in the canvas. Designers can resize pages, move them around by dragging them, mark a page as the home page, and split and merge pages.

To add content and interaction to a page, a designer adds *controls* to the page. The controls include labels, buttons, check boxes, radio buttons, list boxes, drop-down boxes, and text boxes. The designer can also sketch content inside a page by using the pencil tool.

An *arrow* between two pages represents a relationship between those pages. To create an arrow, the designer draws a stroke using the pencil tool between two pages. If a text or ink object is the origin of an arrow, that object becomes blue, like a hyperlink in a web page.

Run window

Damask allows designers to test their design sketches in a Run window (see Figure 2). Inside the Run window, the designer can navigate through the design as if it were running in a web browser. If the Run window is displaying a desktop page, back and forward buttons are provided to simulate a web environment. If it is displaying a smartphone UI, then a telephone keypad is shown beneath the displayed UI.

DESIGNING VOICE UIs

Designing a voice user interface in Damask is quite different than designing a web or smartphone UI. Damask's voice mode is similar to SUEDE's design graph area. A voice UI is represented by computer prompts, connected by possible human responses. Related prompts and responses are grouped together into forms.

Prompts and Responses

Prompts represent phrases that the computer speaks (see the yellow rounded rectangles in Figure 3). To create a prompt, the designer uses the pencil tool, taps within a form, and types the prompt.

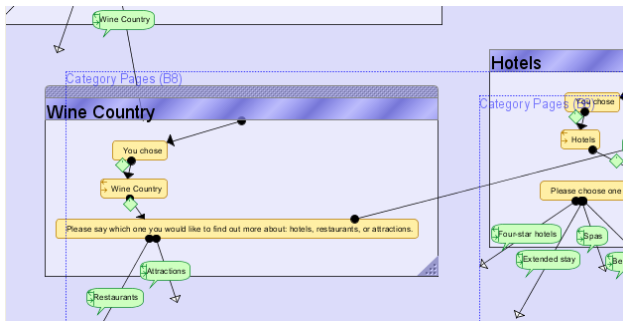


Figure 3. A voice UI design in Damask.

Responses represent the phrases that people say in response to prompts (see the green voice balloons in Figure 3). To create a response between two prompts, the designer uses the pencil tool to drag a line between them. The line becomes an arrow; a voice balloon with the response's text appears along the arrow. A response whose text is an asterisk (*) will match anything that is not matched by any other response.

Forms

Forms are similar in concept to forms in VoiceXML interfaces and analogous to pages in web sites (an example is the rectangle labeled Wine Country in Figure 3). They group together related prompts and responses into a single entity. Creating and manipulating forms is the same as interacting with pages, as described above.

VoiceXML Export and Run window

To verbally interact with the voice UI, Damask allows designers to export their UI to a VoiceXML file, which they can then upload to a web site such as Tellme Studio and interact with the voice UI over the phone. This method is not meant to be used for final deployment of a voice UI; it is solely for prototyping purposes.

Damask also includes a Run window equivalent of the VoiceXML file (see Figure 4). It is a dialog box with the text the computer would say, and a text box for users to type in their responses instead of speaking them.

VoiceXML export and the Run window are useful for quick tests of voice UIs, but if the user does not say or type one of the choices prescribed by the designer, it is not recognized and the user cannot proceed. A Wizard-of-Oz interface, like in SUEDE, would be more suitable for more extensive usability tests.

PATTERNS

Patterns are the primary mechanism in Damask for designers to create cross-device user interfaces that are *optimized* for each device. Damask includes a *Pattern Browser* (see Figure 5) with a catalog of 90 patterns, from

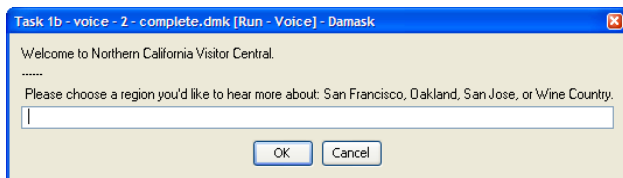


Figure 4. The Run window for voice UIs.

the book *The Design of Sites* [31]. Each pattern has five collapsible sections, which are similar to the pattern structures in books such as *A Pattern Language* [1] and *The Design of Sites*: name, background and image, problem, solution, and related patterns.

We extended eleven of these patterns for multiple devices, by adding UI design fragments for desktop, smartphone, and voice that designers can incorporate into their designs (see Table 1).¹ For example, suppose a designer wants to create a product page in an e-commerce UI design, and decides to use the CLEAN PRODUCT DETAILS pattern. To do this, the designer goes to the Pattern Browser and opens that pattern, then drags the pattern's design from the Pattern Browser and drops it onto his or her design in the main window.

Damask copies the pattern's designs and pastes them into the designer's design, creating *instances* of the pattern. The instances are surrounded by a blue dotted box plus the pattern's name. The designer can now change the pattern instance in any way they want, including adding, deleting, and moving around controls, arrows, or pages. Editing the pattern instance does not change the original pattern. This prebuilt functionality could potentially save a lot of time.

We carefully created the patterns' designs so that the structure of the design is optimized specifically for each device, and that changes in one device are propagated appropriately to the other devices. For example, in the CLEAN PRODUCT DETAILS pattern (Figure 6), the desktop and voice UIs have very different structures, which would be hard to achieve through automatic generation. At the same time, a label in a web page titled "Product name" is linked to a voice

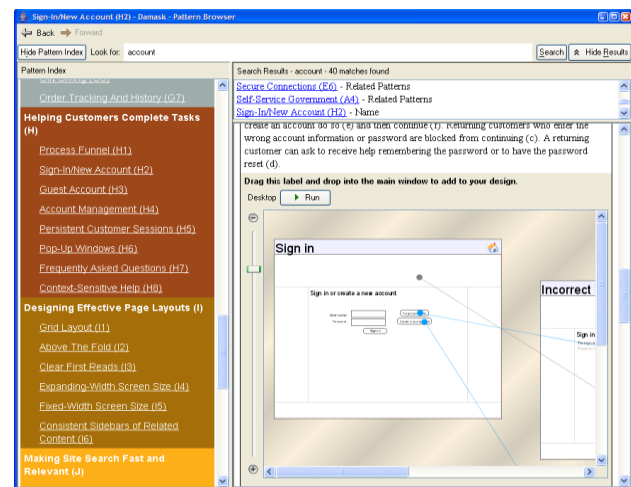


Figure 5. Damask's Pattern Browser consists of three parts: a search area along the top, an index of patterns on the left, and the selected pattern in the main area.

¹ We added UI designs to only the patterns in Table 1 primarily because they formed a cohesive group around which we could create an experiment for evaluation. Damask-based solutions are suitable for more patterns, but were not implemented due to time constraints.

B8 Category Pages	F5 Quick Shipping Method Selection
C1 Homepage Portal	F6 Payment Method
F1 Quick Flow Checkout	F7 Order Summary
F2 Clean Product Details	F8 Order Confirmation and Thank You
F3 Shopping Cart	H2 Sign In/New Account
F4 Quick Address Selection	

Table 1. The patterns that include Damask UI designs.

prompt that says the same thing, so that if the designer changes the label to the name of a specific product, the voice prompt would also change. To achieve this linkage, a pattern's designs uses *layers* (described in the next section), which designers can also use directly.

We chose to use design patterns because they represent an increasingly popular method of discussing design solutions, as evidenced by the number of UI design pattern books and online patterns collections recently published (e.g., [14, 30-32, 34]). Also, as we stated earlier, two of our interview participants had their own pattern collections, indicating that designers are becoming comfortable with the design pattern concept.

LAYERS

Borrowing a well-known concept from design tools like Adobe Photoshop, layers are used for determining if a UI element is on every device or only one device. For each device, there are two layers: *This Device* and *All Devices*. Radio buttons near the top left-hand corner of the canvas control the current layer (see Figure 1). When a designer adds or removes an object to the This Device layer in one device, the UI designs for the other devices are not affected.

When a designer adds an object to the All Devices layer in one device, Damask generates a corresponding object and adds it to the UI designs for the other two devices. Removing an object or changing its textual content in the All Devices layer also removes or changes it on the other devices. However, moving an object, in either layer, does not affect the corresponding object in the other devices, since the layout of a UI is usually device dependent.

Mapping UI elements across devices

When a designer adds a UI element to the All Devices layer in one device UI, Damask needs to determine what UI element to add to the other device UIs and where. For pages and forms, it is straightforward: creating a page in the desktop or smartphone UI also creates a form in the voice UI, and vice versa. If a designer splits a page in, say, the smartphone UI, both pages still correspond to one desktop page and one voice form. This is because splitting a page is done typically for display purposes only; the logical grouping of the elements is still the same.

For UI controls, when the designer adds a control to the desktop design, Damask adds the same control to the smartphone design, and vice-versa. However, the mapping between desktop/smartphone controls and voice controls is more complicated (see Table 2).

In addition, suppose in a voice UI the designer creates a response that points to another form, and Damask decides to generate a control other than a button or hyperlink for the desktop and smartphone UIs (as specified in Table 2). Damask will also add OK buttons next to those generated controls. This is because a voice response encapsulates both receiving user input and acting on that input, whereas a web page has controls for the user to input data and a separate button for the user to submit the data.

As for the location of the generated controls, Damask adds them below existing content in the corresponding pages. While this is a simple and predictable policy, it is not particularly smart. A commercial version of Damask would try to infer which controls are grouped together using a more robust layout algorithm, such as that used in SUPPLE [13] or GADGET [12].

Synchronizing text between visual and voice UIs

Often, the text between a desktop or smartphone UI and a voice UI is different, even if the meaning is the same. For example, a desktop UI might have a check box labeled "Send Me a Copy," whereas a voice UI would ask, "Would you like me to send a copy?" expecting a yes or no. A designer could move the voice control to the This Device layer, but this would remove the check box from the desktop and smartphone views. The designer then would have to create new check boxes in those views, which

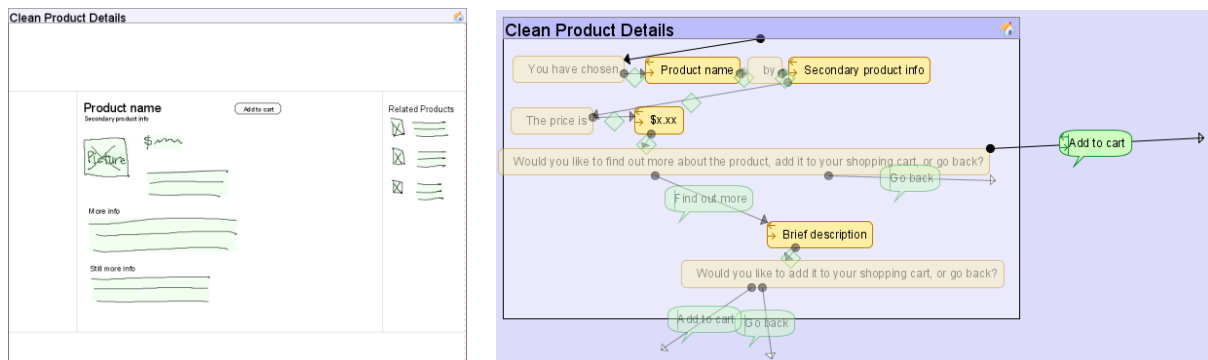


Figure 6. The desktop (left) and voice (right) versions of the CLEAN PRODUCT DETAILS pattern.





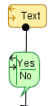


Desktop/smartphone control		Voice control
Label	Text	Prompt 
Button		Response 
Hyperlink	Text	
Radio buttons	<input checked="" type="radio"/> Item 1 <input type="radio"/> Item 2	Response 
List box	<div>Item 1</div> <div>Item 2</div>	
Drop-down box	<div>Item 1</div> <div>Item 1</div> <div>Item 2</div>	
Check box	<input checked="" type="checkbox"/> Text	Prompt + Response 
Text box		Response 

Table 2. Desktop/smartphone controls and their corresponding voice controls.

would not be synchronized with each other.

Instead, we have another mechanism for controlling the synchronization of text between voice and visual UI designs. On the left side of each prompt and response, there is a two-arrow icon. The designer clicks on the icon to toggle between keeping the text for that prompt synchronized and unsynchronized. When the text is not synchronized, the icon is dimmed.

IMPLEMENTATION

Damask is written in Java 2 Standard Edition version 1.4. Besides the standard Java libraries, it uses two other libraries. SATIN [15] is a library for pen-based applications. Damask uses it for recognizing how ink strokes should be grouped. Piccolo [4] is a 2D graphics library that Damask uses for the canvas area. The patterns are stored in external files in an XML-based format called the Pattern Language Markup Language (PLML) [11].

Damask's architecture is based on the model-view-controller (MVC) software design pattern. MVC decouples the data model from the UI, allowing multiple views and types of user interaction to manipulate the same data model. In Damask's case, the view is Damask's UI, while the model is the abstract model of the UI being designed.

UI designs in Damask are represented internally as *scenegraps*. A scenegraps is a tree of elements, where the hierarchy models the containment relationship. In Damask, there are four scenegraps. One represents the abstract model, and the other three represent desktop, smartphone, and voice UIs.

EVALUATION OF DAMASK

In evaluating Damask, we wanted to find out:

- Were designers willing to learn and understand the collection of design patterns? Did the time that it took to do this offset any increase in speed in using the patterns?
- Were the designs created with patterns and layers of higher quality than those created without?
- Did designers find the patterns useful, or did they feel too constrained by them?
- Did the designers think layers were an appropriate concept for describing which parts of a UI design gets retargeted to other devices?

The evaluation consisted of two phases with two separate pools of designers: designing desktop and smartphone UIs, and designing desktop and voice UIs. We did not address designing for all three types of devices at once, as that would have made the experimental session with each participant prohibitively long. The participants used a Fujitsu T Series Lifebook (1.4 GHz Pentium M with 1 GB RAM) running Windows XP Tablet PC Edition 2005 and Java SE 5.0, at a screen resolution of 1024×768.

Phase 1: Desktop and smartphone UI design

In Phase 1, eight designers were divided evenly by random selection into two groups. One group used Damask without layers and patterns first, and the other used it with layers and patterns first. In both groups, the evaluation was spread out over two sessions on two days. The sessions were separated by a mean of 16 days, depending on the participant's schedule.

On the first day, the participants in the first group were first given a tutorial on Damask, without layers or patterns. Then, after warm-up tasks on using the Tablet PC and Damask, they used Damask without layers and patterns to design an online music store for the desktop and smartphone, based on a set of requirements we gave them. There was no specified time limit, but it was designed to take about 1½ to 2 hours. They finished with a questionnaire on using Damask without layers or patterns.

On the second day, the first group's participants were given a demonstration of layers and patterns in Damask. They then used the Pattern Browser for 15 minutes to learn the patterns in preparation for a brief quiz. After a warm-up task on using layers and patterns, the participants then used Damask *with* layers and patterns to design an online bookstore for the desktop and smartphone. The requirements were similar to those for the music store. They finished with a questionnaire on using Damask with layers and patterns.

The second group's participants had the days reversed, and all of the warm-ups and tutorials were on the first day.

Phase 2: Desktop and voice UI design

Four designers participated in Phase 2. They were also

divided into two groups. However, there were three sessions instead of two, to give them more time to learn the UIs for designing desktop and voice designs, which are very different. The average time between sessions 1 and 2 was 7 days and between sessions 2 and 3 was 6 days. The first session, a Damask warm-up, was the same for both conditions, while the second and third sessions were swapped between the two groups. The tasks were essentially the same as in Phase 1, except that voice UIs were created instead of smartphone UIs. We also de-emphasized sketching, so that text typed into the desktop design would easily be transferred into the voice design.

Participants

The eight participants in Phase 1 had at least five years of experience designing web sites, but less than two years of mobile phone UI experience. Each participant was promised a US\$250 Amazon.com gift certificate. The four participants in Phase 2 had at least three years of experience designing voice UIs, but at most one year of web experience. The Amazon.com gift certificate in this phase was worth US\$125 more than in Phase 1, since there was one more session per person.

Results

We looked at several aspects of how patterns and layers affected the design process.

Time spent by participants. On average, the participants spent the same amount of time designing the desktop design with and without patterns and layers (see Tables 3 and 4).² However, in the patterns and layers condition, over 16% of that time was spent exploring the pattern browser. Over extended use, we would expect the desktop time in this condition to go down, as the time needed to browse through the patterns would likely decrease.

Among the desktop/smartphone participants, designers spent statistically significantly less time designing the smartphone UI using patterns and layers (on average, 20 minutes) than without (42 minutes; see Table 3). Among the desktop/voice participants, there is a difference between the two conditions in the time overall and in voice design time (29 minutes vs. 16 minutes), but it is not statistically significant (see Table 4). The time savings for the second device reflects the fact that, using patterns and layers, the designers did not have to start from scratch for the second device, and that the generated designs were good enough that the designers were able to start designing using those designs as a base.

All of the participants designed for the desktop first, and then the other device second. There were almost no instances where a designer designed for the desktop, then the second device, and then went back to the desktop. While designers did not work iteratively between devices,

² We considered designers who used one or fewer patterns in the pattern condition to be outliers and removed them before analysis. This ended up eliminating 2 out of the 12 designers.

	Desktop	Smart-phone	Total
No patterns or layers	1:08	0:42	1:50
Patterns and layers	1:08	0:20	1:28
<i>In Pattern Browser</i>	0:11	< 0:01	
<i>p</i> (2-tailed t-test)	>0.50	<0.01	0.17

Table 3. Mean time spent by desktop/smartphone participants creating UI design, in hours and minutes (gray background = $p < 0.05$, 2-tailed t -test)

this may be because they had only two hours to work. We expect more iteration over a period of days, which would be more realistic for a design project.

Quality analysis of desktop designs. We wanted to find out whether using patterns and layers affected the quality of the desktop designs, compared to the designs without patterns or layers. To do this, we recruited 18 professional designers online to judge the 24 desktop designs. We randomly assigned two designs (with and without patterns and layers) from two designers to each judge, and presented the four designs in random order. We made the assignments such that each designer was evaluated by three judges. We did not tell the judges anything about patterns or layers, or that only two people created the four designs under different conditions. The judges had backgrounds similar to that of the designers, but they had more experience designing e-commerce web sites.

For each design, the judges looked at “sketchy”-looking HTML versions that we created instead of the original, to eliminate the participants’ sketching and handwriting as a confounding factor. They then filled out a questionnaire evaluating, on a scale of 1 (low) to 5 (high), the layout of the product pages and checkout pages, how well the product pages were linked to each other, how well the checkout pages were linked to each other, and an overall rating. We also asked how skilled they thought the designer was and how complete the design was.

For our analysis, we first wanted to find out if the availability of patterns and layers had an overall effect on all of the aspects listed in the questionnaire. We calculated the mean of the responses from the three evaluators, for each condition, designer, and question. We then conducted a multivariate analysis of variance (MANOVA). The availability of patterns and layers was the independent variable, and each question was a dependent variable. We found no statistically significant effect of the availability of patterns and layers on all ratings as a whole (Wilks’ $\lambda = .53$,

	Desktop	Voice	Total
No patterns or layers	0:48	0:29	1:17
Patterns and layers	0:50	0:16	1:06
<i>In Pattern Browser</i>	0:08	0:00	
<i>p</i> (2-tailed t-test)	>0.50	0.13	0.09

Table 4. Mean time spent by desktop/voice participants creating UI design, in hours and minutes

Question	Desktop		Smartphone	
	No patterns	Patterns	No patterns	Patterns
Links among product pages	2.87	3.12	3.33	2.81
Layout/design of product pages	2.63	3.13	2.90	2.43
Links among shopping cart/checkout pages	3.03	3.40	3.05	3.43
Layout/design of shopping cart/checkout	2.43	3.30	2.48	3.33
Overall rating	2.73	3.37	2.76	3.29
Completeness	2.33	3.17	2.14	2.95
Perceived skill of designer	2.63	3.13	2.71	3.00

Table 5. Average ratings for the desktop and smartphone designs (1 = low, 5 = high; gray background = $p < 0.05$, ANOVA)

$F_{7,12} = 1.54$, $p = 0.24$). We then wanted to see which specific ratings had significant differences. We conducted analyses of variances (ANOVA) on each question. Using Fisher's PLSD method, each ANOVA was tested at the 0.05 level.

The ANOVAs showed that there were statistically significant differences for three ratings. Ratings were higher for designs with patterns and layers in:

- completeness of the design
- page layout and design of the shopping cart and checkout process
- overall rating

See Table 5. Using patterns and layers did not make a significant difference on the ratings for the other questions. This may be because those aspects are not as well covered by the eleven patterns that included Damask designs, or that it was easy to address them without patterns and layers.

Quality analysis of smartphone designs. We also performed an evaluation of the quality of the smartphone designs, using the same questionnaire and method as for the desktop designs. There are 16 smartphone designs over the two conditions. We recruited 12 professional designers online and randomly assigned four designs to each judge, so that each design was evaluated by three people. Compared to the designers, the judges had more experience designing for mobile phones.

Using the same analysis technique as for the desktop designs, we found no overall statistically significant effect from the availability of patterns (MANOVA, Wilks' $\lambda = .25$, $F_{7,12} = 2.61$, $p = 0.13$). However, after running ANOVAs on each question, we found statistically significant differences for two of them. Designs with patterns rated higher in the page layout and design of the shopping cart and checkout process, and overall completeness of the design (see Table

5). We believe this is due to similar reasons as the desktop designs.

Quality analysis of voice designs. To evaluate the voice designs, we first modified the voice designs to fill in placeholders with actual items to give the illusion of finished designs. We then asked four judges, all HCI experts, to listen to each of the eight designs. The order in which each judge listened to the designs was random. For each design, they were given a specific book or CD to "buy," along with information to complete the purchase. They then answered a questionnaire, including rating of the design from 1 (did not like) to 7 (liked very much).

We found no statistically significant difference in the ratings between the voice designs with and without patterns (5.08 without patterns, 4.17 with patterns, 2-tailed t -test, $p = 0.38$). The designs with and without patterns each had their strengths and weaknesses.

Of the designs that include patterns, many of the evaluations described them as being efficient and fast to use. However, there were also many complaints about the excessive wordiness and the lack of a "goodbye," which stem directly from the patterns. Since the patterns in *The Design of Sites*, on which the Damask patterns are based, do not contain voice-specific solutions, we created the solutions ourselves. Unfortunately, we did not have the opportunity to consult voice designers at the time, which led to some awkward voice solutions. Fixing them would likely improve these designs.

The designs without patterns had their own pluses and minuses. In general they were praised for their efficiency, but some judges noticed that three of the four designs without patterns also did not have a "goodbye" message. One design did not include a confirmation number, and another did not say the final total cost, problems that would have been easily avoided by using patterns.

Pattern usage and ratings. Most of the design participants thought that Damask's patterns were useful for the given task; the average rating was 6.0 out of 7. For general usefulness, the average was also positive, 5.3 out of 7.

The participants who used design patterns extensively said that patterns saved them time because they would not have to "reinvent the wheel." Participant 9 said, "They let you skip the step of creating them and just pick up stuff that's already been proven to work." Participant 10 noted, "They help enforce consistency and they also save a bunch of time when designing something that doesn't really require a whole bunch of innovation. Nice that the patterns are flexible, too. I can delete portions if I don't want to use them."

But two participants said the patterns' solutions seemed to assume too much. Participant 4 said, "Sometimes they assumed I was using them in a particular way." Participant 15 said, "Even with 80 [sic] patterns, sometimes the pattern does not match what you want."

Out of the 11 patterns with Damask solutions, the participants on average explicitly used 4.3 patterns and in total used 8 patterns (2 patterns contain other patterns). Also, 9 patterns or more were used (explicitly and implicitly) by 8 of the 12 participants. This indicates that most designers were able to find and use the patterns in Damask that were relevant to the design task, even after only 15 minutes to familiarize themselves with the 90-pattern collection.

There were several patterns that the participants tried to use but did not have any Damask-based solutions that they could directly incorporate into their designs. On average, the designers tried to use 3.5 patterns that were unimplemented. Out of the three patterns most frequently tried, two of them would likely have shortened the participants' design time even more.

Layers usage and ratings. All of the participants understood layers well enough to perform the task. Out of the 12 designers, 8 rated the ease of understanding layers at least 5 out of 7 (7 being easy to understand), while the other 4 rated it 2 or 3 (average = 4.92, std dev = 1.73).

The participants' ratings for how much they liked layers were mixed. 6 out of 12 designers rated how much they like layers 5 to 7 and the other 6 rated it 2 to 4. The overall average was 4.75 out of 7 (std dev = 1.66).

The most frequent mistake that participants made was forgetting which layer they were currently in, and therefore making changes that did not propagate the way they expected. Another usability problem that most of the participants ran into was moving objects between layers. For example, suppose there were a button in all devices. If designers wanted to remove the button from the smartphone UI only, their natural inclination was to try to use the eraser and erase the button in smartphone view, rather than going back to the desktop view and moving the object from All Devices to This Device. One way to address this issue is having tools that are only active for a specific layer.

Usability of Damask. We found that designers had little problem understanding the basic concepts of Damask's UI. Although they all ran into the quirks commonly found in a research prototype, by the second session, they had all gotten much more comfortable using the interface.

After each session, the designers were asked to write down what they liked and did not like about Damask. The most frequently mentioned positive aspects were the design patterns (mentioned by 7 out of 12 designers), Run mode (7), layers or the ability to create UIs for multiple devices simultaneously (5), and sketching (5).

The top three negative aspects, each mentioned by four designers, were layers, the awkwardness of Damask's sketching interface, and the difficulty of panning and zooming around the canvas. Note that layers and the sketching UI were mentioned both positively and negatively, indicating that the concepts may be good but the execution needs to be improved.

Discussion. We can now answer the questions we posed at the beginning of this section:

- Designers were willing to learn and understand Damask's collection of design patterns. Designers were at least as fast using patterns and layers as not, even including the time for browsing through the patterns.
- The desktop and smartphone designs created with patterns and layers were rated higher than those created without, with regards to the completeness of the design, the shopping cart and checkout process, and in the desktop case, the overall rating.
- There was no overall difference in ratings between voice designs created with or without patterns and layers, likely due to the maturity of the voice patterns.
- Designers found the patterns useful, as evidenced by their positive reactions and that on average, 8 out of 11 patterns with Damask solutions were used.
- Layers are a useful concept for describing which parts of a UI design are retargeted to other devices, although the execution can be improved.

FUTURE WORK AND CONCLUSION

While we have shown patterns to have promise for designing cross-device UIs, large patterns were sometimes hard to understand. To help with this, we would like to make pattern solutions collapsible and add annotations directly to pattern solutions. Also, allowing designers to create and share their own patterns would greatly enhance the usefulness and customizability of Damask.

Damask addresses the design of cross-device user interfaces in a novel way, with design patterns and layers. Design patterns facilitate the creation of cross-device user interface designs that are optimized for each target device, by including pre-built UI design fragments that can have very different interaction flows depending on the device. Layers let designers control which parts of their design are retargeted to other devices and which are not. Our evaluation shows that designers can use layers and patterns to create designs at least as good and as quickly as those not using layers and patterns.

ACKNOWLEDGMENTS

We would like to thank the designers and evaluators who participated in our experiments. HopiSketch was done in collaboration with Lawrence Bergman, Guruduth Banavar, Danny Soroker, Richard Cardone, and others at IBM Watson Research Center. Thanks also to the following for their help with Damask: Wei (Michelle) Xue, Qing Li, Madhu Prabaker, Elizabeth Goodman, Beverly Harrison, Wendy Ju, Leila Takayama, Shumin Zhai, Jason Hong and Scott Klemmer. Much of this research was conducted at UC Berkeley and was funded in part by Hewlett-Packard and the National Science Foundation under Grant No. 9985111.

REFERENCES

1. Alexander, C., S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel, *A Pattern Language*. Oxford University Press: New York, 1977.
2. Ali, M.F., M.A. Pérez-Quñones, M. Abrams, and E. Shell. Building Multi-Platform User Interfaces With UIML. Proc. *CADUI'2002*, 225-236.
3. Banavar, G., L.D. Bergman, Y. Gaeremynck, D. Soroker, and J. Sussman, Tooling and System Support for Authoring Multi-Device Applications. *Journal of Systems and Software*, 2004. **69**(3): 227-242.
4. Bederson, B.B., J. Grosjean, and J. Meyer, Toolkit Design for Interactive Structured Graphics. *IEEE Transactions on Software Engineering*, 2004. **30**(8): 1-12.
5. Bergman, L.D., G. Banavar, D. Soroker, and J. Sussman. Combining Handcrafting and Automatic Generation of User-Interfaces for Pervasive Devices. Proc. *CADUI'2002*, 155-166.
6. Buyukkokten, O., H. Garcia-Molina, A. Paepcke, and T. Winograd, Power Browser: Efficient Web Browsing for PDAs. Proc. *CHI 2000*, 430-437.
7. Calvary, G., J. Coutaz, and D. Thevenin. A Unifying Reference Framework for the Development of Plastic User Interfaces. Proc. *EHCI 2001*, 173-192.
8. Deng, J., E. Kemp, and E.G. Todd. Managing UI Pattern Collections. Proc. *CHINZ '05*, 31-38.
9. Ding, Y. and H. Litz. Creating Multiplatform User Interfaces by Annotation and Adaptation. Proc. *IUI 2006*, 270-272.
10. Eisenstein, J., J. Vanderdonckt, and A. Puerta. Applying Model-Based Techniques to the Development of UIs for Mobile Computers. Proc. *IUI 2001*, 69-76.
11. Fincher, S., Perspectives on HCI Patterns: Concepts and Tools (Introducing PLML). *Interfaces*, 2003(56): 26-28.
12. Fogarty, J. and S.E. Hudson, GADGET: A Toolkit for Optimization-Based Approaches to Interface and Display Generation. Proc. *UIST 2003*, 125-134.
13. Gajos, K. and D.S. Weld. SUPPLE: Automatically Generating User Interfaces. Proc. *IUI '04*, 93-100.
14. Graham, I., *A Pattern Language for Web Usability*. Addison-Wesley: London, 2003.
15. Hong, J.I. and J.A. Landay, SATIN: A Toolkit for Informal Ink-based Applications. Proc. *UIST 2000*, 63-72.
16. Lin, J., *Using Design Patterns and Layers to Support the Early-Stage Design and Prototyping of Cross-Device User Interfaces*, Unpublished Ph.D. Dissertation, University of California, Berkeley, 2005. http://jameslin.name/research/damask/dissertation/jlin_dissertation.pdf
17. Lin, J. and J.A. Landay. Damask: A Tool for Early-Stage Design and Prototyping of Multi-Device User Interfaces. Proc. *DMS 2002*, 573-580.
18. Lopez, J.F. and P. Szekely, Web Page Adaptation for Universal Access, in Proc. *UAHCI 2001*, 690-694.
19. Microsoft, *ASP.NET Mobile Controls*, 2002. Microsoft Corp.: Redmond, WA. <http://asp.net/mobile>
20. Mori, G., F. Paternò, and C. Santoro, Design and Development of Multidevice User Interfaces through Multiple Logical Descriptions. *IEEE Transactions on Software Engineering*, 2004. **30**(8): 507-520.
21. Mynatt, E.D. and W.K. Edwards. An Architecture for Transforming Graphical Interfaces. Proc. *UIST '94*, 39-47.
22. Newman, M.W. and J.A. Landay. Sitemaps, Storyboards, and Specifications: A Sketch of Web Site Design Practice. Proc. *DIS 2000*, 263-274.
23. Newman, M.W., J. Lin, J.I. Hong, and J.A. Landay, DENIM: An Informal Web Site Design Tool Inspired by Observations of Practice. *Human-Computer Interaction*, 2003. **18**(3): 259-324.
24. Nichols, J., et al., Generating Remote Control Interfaces for Complex Appliances. Proc. *UIST 2002*, 161-170.
25. Olsen, D.R., S.E. Hudson, R.C.-M. Tam, G. Conaty, M. Phelps, and J.M. Heiner. Speech Interaction with Graphical User Interfaces. Proc. *INTERACT2001*.
26. Ponnekanti, S.R., B. Lee, A. Fox, P. Hanrahan, and T. Winograd. ICrafter: A Service Framework for Ubiquitous Computing Environments. Proc. *Ubicomp 2001*, 56-75.
27. Puerta, A., M. Micheletti, and A. Mak. The UI Pilot: A Model-Based Tool to Guide Early Interface Design. Proc. *IUI 2005*, 215-222.
28. Sinha, A.K., S.R. Klemmer, and J.A. Landay, Embarking on Spoken-Language NL Interface Design. *International Journal of Speech Technology*, 2002. **5**(2): 159-169.
29. Szekely, P. Retrospective and Challenges for Model-Based Interface Development. Proc. *DSV-IS '96*, 1-27.
30. Tidwell, J., *Designing Interfaces: Patterns for Effective Interaction Design*. O'Reilly Media: Sebastopol, CA. 384pp., 2005. <http://time-tripper.com/uipatterns/>
31. van Duyne, D.K., J.A. Landay, and J.I. Hong, *The Design of Sites*. Addison-Wesley: Boston, 2002
32. van Welie, M. and H. Trätteberg. Interaction Patterns in User Interfaces. Proc. *PLoP 2000*. <http://jerry.cs.uiuc.edu/~plop/plop2k/proceedings/Welie/Welie.pdf>
33. Wagner, A., Prototyping: A Day in the Life of an Interface Designer, in *The Art of Human-Computer Interface Design*, B. Laurel, Editor. Addison-Wesley: Reading, MA. pp. 79-84, 1990.
34. Yahoo, *Yahoo! Design Pattern Library*, 2006. <http://developer.yahoo.com/ypatterns/>

