Damask: A Tool for Early-Stage Design and Prototyping of Cross-Device User Interfaces

James Lin and James Landay
Group for User Interface Research, EECS Department
UC Berkeley
Berkeley, CA 94720-1776
{jimlin, landay}@cs.berkeley.edu

INTRODUCTION

People often use a variety of computing devices, such as PCs, PDAs, and cell phones, to access the same information. The user interface to this information needs to be different for each device, due to different input and output constraints. Currently, designers designing such cross-device user interfaces either have to design a UI separately for each device, which is time consuming, or use a program to automatically generate interfaces, which often results in interfaces that are awkward.

We are creating a system called Damask [3] to better support cross-device UI design. With Damask, the designer will design a UI for one device, by sketching the design and by specifying which *design patterns* the interface uses. The patterns will help Damask generate user interfaces optimized for the other target devices. The generated interfaces will be of sufficient quality so that it will be more convenient to use Damask than to design each of the other interfaces separately, and the ease with which designers will be able to create designs will encourage them to engage in iterative design.

Damask will be aimed at designers who want to design and prototype a UI targeted at three types of interfaces: the web accessed through a desktop, cell phone displays, and prompt-and-response style voice interfaces. We have picked these three because they represent the "extremes" of the range of devices that are widely used. For example, simply shrinking a screen designed for a desktop PC will not result in a good cell phone interface.

OVERVIEW OF DAMASK'S APPROACH

At a high level, Damask will include a catalog of design patterns that designers can use in their designs. Each design pattern will have specific *examples* of how the pattern has been used in other projects, and several generalized *solutions* capturing the essence of the examples. Each design pattern will have a separate solution for each device.

Designers will create their UI designs by sketching and by adding instances of patterns to their design for one device. Damask will take that design and generate UI design sketches for the other two devices, which the designers can go back and modify if desired. Finally, designers can use Damask (or SUEDE [2] for voice interfaces) to run their designs in a device simulator, so that they can interact with their design sketches.

DAMASK'S PROPOSED USER INTERFACE

Damask's proposed user interface consists of several regions (Figure 1). The canvas is where the designer will sketch the actual user interface design. The design will include which patterns it is using, as denoted by a red outline and the name of the pattern. There will be tabs above the canvas where designers will choose which target device they are viewing. To view the different device-specific UIs at the same time, the designer will be able to split the canvas or view the design in multiple windows.

Damask will also have a *Pattern Explorer* sidebar, where designers could browse for patterns to be instantiated in their designs, and the *Pattern* sidebar where designers could find the details about a particular pattern, instantiate a pattern, and create their own patterns. Each pattern will have eight parts, which correspond to the structure of patterns found in several pattern languages such as [1] and [5]:

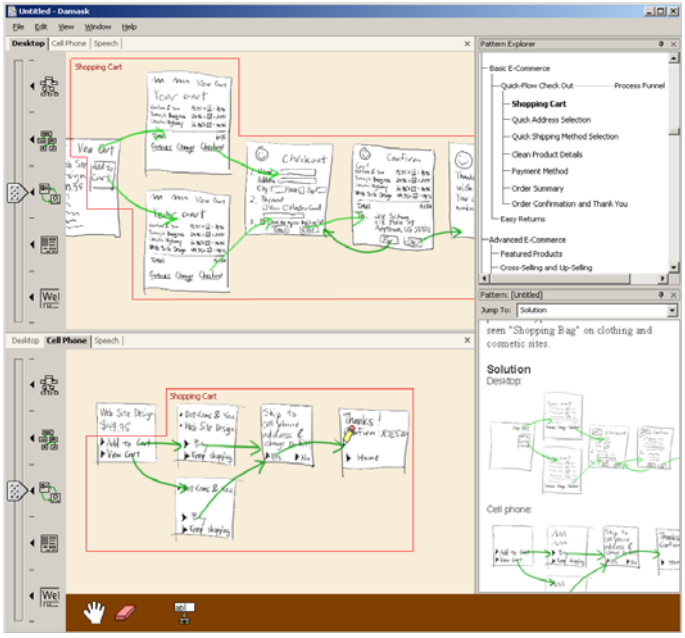

Figure 1. Damask's proposed user interface.

- name
- sensitizing image
- background
- problem
- forces
- examples
- solution
- references

Two of the sections warrant more elaboration. The Examples section will contain real examples of the pattern in use. It will also be constantly updated: whenever a pattern is instantiated, that instance will be added to the Examples section and will be continuously updated whenever the designer modified the instance.

The Solution section will contain generalized solutions for the pattern. Like the canvas, the Solution section will also be divided into three sections, with one solution for each device supported by Damask.

CREATING CROSS-DEVICE INTERFACES

Here is how we envision a designer using Damask to design a UI, for example, an e-commerce web site for the PC and cell phone. The designer decides to first target the PC, so he sketches out some web pages for the PC version of the web site. Here is one such page:

Instead of sketching out all of the pages from scratch, the designer takes advantage of the patterns built into Damask. He brings up the Pattern Explorer to browse through the patterns, and comes across the SHOPPING CART pattern (Figure 2). He sees that there are two generalized solutions for the SHOPPING CART, one for a PC and one for a cell phone (see Figure 2, right).

The structure of the designer's UI sketches and the pattern's solutions follow a visual language similar to

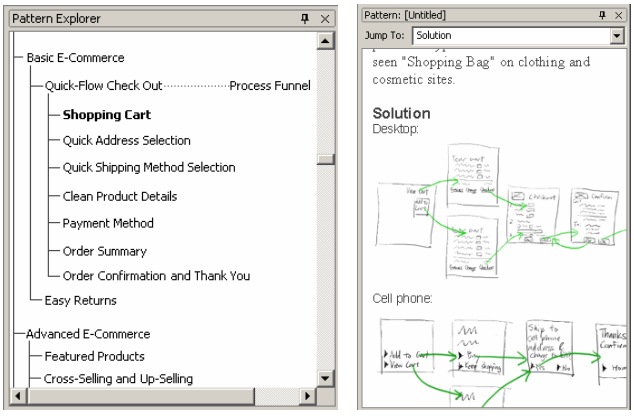

Figure 2. *Left:* The Pattern Explorer with the SHOPPING CART pattern highlighted. *Right:* The Pattern sidebar containing the SHOPPING CART pattern.

Figure 3. The PC version of the e-commerce web site, with shopping cart merged into it.

DENIM [4] and SUEDE [2]. A *page* represents a web page, cell phone screen, or voice prompt. A designer can sketch or type in a page. An *arrow* between two pages represents an action that the end-user performs to go from one page to the other. In a web page, the source of the arrow represents the hyperlink the end-user clicks on to go to the target page. In a cell phone screen, it represents a menu item that the end user selects. In a voice interface, the arrow is annotated with the response that the end-user says to go to the target voice prompt.

The designer picks the PC version of the SHOPPING CART solution and drags the leftmost page of the pattern into the canvas, bringing the rest of the pattern along. Then he drops it on top of the page that he first sketched. This merges the contents of that the pattern page with his sketched page and adds the rest of the pattern to his design. SHOPPING CART has now been *instantiated* in his design (Figure 3).

The pattern that the designer has just instantiated is very generic, for example, having mostly text placeholders instead of actual text. The designer now customizes the pattern instance to fit his own project. He replaces the text placeholders with actual text, moves widgets around, and adds his own images. He could even add pages and change the arrows if he decides that is appropriate. As the designer customizes the pattern instance, Damask keeps track of his customizations. The pattern is now fully integrated into his design (Figure 4).

At some point, the designer decides he is ready to work on the cell phone version of the web site. So he clicks on the Cell Phone tab just above the canvas. Damask first makes the cell phone-specific design by copying the PC-

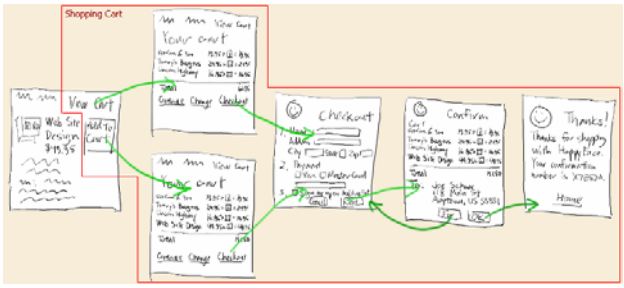

Figure 4. The PC version of the e-commerce web site, with shopping cart customized.

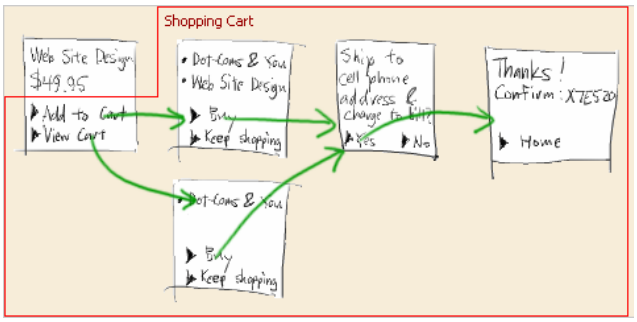

Figure 5. The cell phone version of the e-commerce web site generated by Damask.

specific design. Then it goes through the design and finds which parts of the design are pattern instances and which are not.

Damask modifies the parts of the design that are not pattern instances by applying traditional model-based UI techniques. For example, it will rearrange widgets to compensate for the smaller screen, and replace sets of radio buttons with drop-down boxes. Both perform the same abstract task, but drop-down boxes take up less space.

Damask replaces the pattern instances, which have been PC-specific up to now, with the corresponding cell-phone pattern solutions. It then takes the customizations that the designer applied to the PC-specific versions and applies them to the cell-phone versions. This results in a pattern instance specific to the cell phone but customized to the application that is being designed (Figure 5).

Not all of the customizations will necessarily be applied. For example, if the designer moves a widget in the PC version, Damask will not apply that customization to the cell-phone version, since the displays of cell phones are so limited that the designer would most likely have to move the widget again anyway. One of the biggest research challenges is deciding which customizations to take from one device-specific instance and apply it to the others.

The instances of SHOPPING CART within this project are automatically added to the Examples section of the SHOPPING CART pattern within Damask's pattern library. This encourages reuse of designs and could decrease the time and effort spent on future projects.

CREATING CUSTOM PATTERNS

Creating new design patterns in Damask consists of several steps:

- Choosing the fragments of a design from which to create a pattern.
- Generalizing those fragments to create generic pattern solutions.
- Showing how the device-specific solutions of the pattern relate to each other.

We will illustrate this with an example. Suppose Damask did not have a SHOPPING CART pattern, and the designer wanted to create one from his design. To create SHOPPING CART, the designer first opens up the Pattern Explorer sidebar, opens a context menu, and chooses New Pattern. An empty Pattern sidebar is created. The designer selects the part of the design he wants to become part of the SHOPPING CART pattern and drags it into the Pattern sidebar. Damask puts the fragment into the pattern's Solution and Examples sections and marks the design with the new pattern. (See Figure 6.)

The specific shopping cart that the designer dragged into the Pattern sidebar has actual text and other details that are not appropriate for a general solution of SHOPPING CART. To make the solution more general, the designer edits the solution, replacing actual text with placeholders, and so on.

Finally, the designer takes the device-specific pattern solutions and shows how they relate to each other. This is so that when Damask generates a UI for another device, it knows how to take the customizations the designer applied to the first device-specific pattern instance and apply them to the second device-specific pattern instance. If the designer does not specify these relationships in the pattern, then when Damask uses the pattern as the basis for automatically generating another interface for a second device, the solution specific to the second device will be used without applying any customizations to it.

One proposal for showing these relationships is to draw

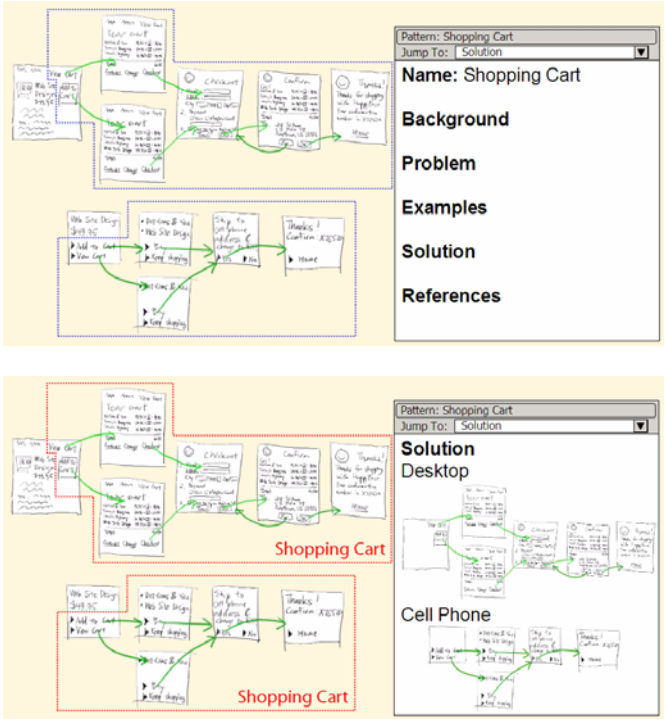

Figure 6. *Top:* Highlighting the portion of a design from which to create a pattern. *Bottom:* The results of dragging the highlighted sections to the pattern sidebar.

lines between the related parts. In this case, the designer views both the PC and cell phone SHOPPING CART solutions and draws lines between them to show how they are related. For example, he draws a line from the shopping list on the first page of the PC solution, to the shopping list on the first page of the cell phone solution. This way, when the designer fills in the list in the shopping cart in a PC design, and then asks Damask to generate a cell phone design, Damask knows to take the contents of the shopping list in the PC version, and put them into the corresponding list in the cell phone version.

There are many research questions to be answered here, such as what happens if the designer relates two parts that are not exactly the same (such as a list with three elements with a list with one element), whether such a simple mechanism is sufficiently powerful in enough cases, and how to design such a mechanism that does not overwhelmingly clutter the UI sketches.

REFERENCES

1. Alexander, C., S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel, *A Pattern Language*. New York: Oxford University Press, 1977.
2. Klemmer, S.R., A.K. Sinha, J. Chen, J.A. Landay, N. Aboobaker, and A. Wang, SUEDE: A Wizard of Oz Prototyping Tool for Speech User Interfaces. *CHI Letters: Proceedings of User Interfaces and Software Technology: UIST 2000*, 2000. **2**(2): pp. 1-10.
3. Lin, J. and J.A. Landay. Damask: A Tool for Early-Stage Design and Prototyping of Multi-Device User Interfaces. In Proceedings of *The 8th International Conference of Distributed Multimedia Systems (2002 International Workshop on Visual Computing)*. San Francisco, CA. pp. 573-580, Sept. 26-28, 2002.
4. Lin, J., M. Thomsen, and J.A. Landay, A Visual Language for Sketching Large and Complex Interactive Designs. *CHI Letters: Proceedings of Human Factors in Computing Systems: CHI 2002*, 2002. **4**(1): pp. 307-314.
5. van Duyne, D.K., J.A. Landay, and J.I. Hong, *The Design of Sites*: Addison-Wesley, 2002.