Can Presenting Examples Improve Designers' Work? Creating Better Interfaces through Adaptive Display

Brian Lee, Savil Srivastava, Ronen Brafman, Scott R Klemmer
Stanford University HCI Group
Computer Science Department
Stanford, CA 94305-9035
[balee, savil, brafman, srk]@cs.stanford.edu

ABSTRACT
Designers frequently use examples for inspiration. Information technologies have facilitated designers' access to examples. This paper examines the hypothesis that designers can realize significant value from explicit mechanisms for creation by example modification. The work offers two contributions. First, we introduce techniques for dynamically creating interfaces for example-based design tools using decision-theoretic selection, designer specification, and end-user preference as inputs. We present a manifestation of these techniques in the Adaptive Ideas web editor, which leverages metadata to automatically generate displays of examples, semantically navigate these examples, and facilitate design by example modification. Second, we report on two experiments that explore if and how such presentation techniques impact the quality of resulting designs. These studies found that independent raters prefer designs created with the aid of examples, that the benefit examples provide is higher for novice designers than experts, and that designers prefer adaptively selected examples to random ones.

ACM Classification: H.5.2. [Information Interfaces]: User Interfaces—*Graphical user interfaces; Interaction styles*. D.2.2 [Software Engineering]: Design Tools and Techniques—*user interfaces*.

General terms: Design, Human Factors, Algorithms

Keywords: Adaptive interfaces, design by example

INTRODUCTION
Viewing examples of previous work is an established technique in many design disciplines: compendiums such as "The Big Book of Logos" [6] serve as highly regarded resources for inspiration. The advent of prolific, searchable web content has provided ready access to a broad array of work created by other designers. We believe that designers can realize significant value from explicit mechanisms for creation by example modification.

This research hypothesizes that structured corpus navigation aid designers in locating and working with inspirational examples. Building on earlier work (*e.g.*, [31]), we believe that design tools can leverage metadata to provide structure to a corpus of examples and aid serendipitous inquiry. We further submit that, once a designer has found a desirable example, explicitly supporting design by example modification can provide value. These hypotheses are motivated in part by practices in design and art disciplines, which encourage designing analogically [4, 24].

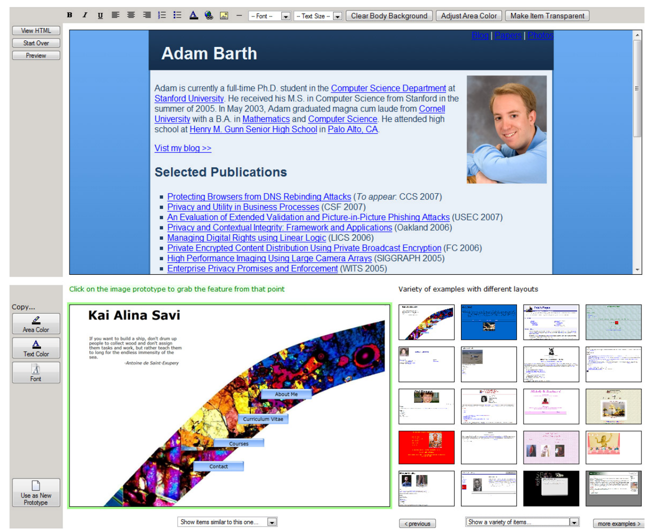


Figure 1. The Adaptive Ideas web editor. The top area shows the page being edited. The lower left section shows a focus example from which the user can borrow elements, while the lower right section shows an adaptively generated gallery of other examples.

This paper introduces a novel, decision-theoretic approach to adaptively laying out example designs and interface techniques for working with an example corpus. We have explored this approach by creating the Adaptive Ideas web design tool, which supports facet-based techniques for browsing a corpus of example designs, as well as borrowing of different features from the examples. This research draws on prior work on model-based user interfaces [26] and adaptive interfaces [10], and in particular on the idea of casting interface generation as a constraint-based optimization problem [3, 12, 32]. Though this work examines a specific instance of design (augmenting a visual design activity), the central roles that analogy and variation play in creative cognition (*e.g.*, [13, 28]) suggests that design patterns for analogical tools may have broader import.

We begin with a high-level description of the Adaptive Ideas system, illustrate its use with a scenario, and describe the algorithmic approach and implementation. Next, we present two experiments that test whether tools with explicit mechanisms for designing by example yield higher-quality results. We conclude with discussions of related work.

ADAPTIVE IDEAS OVERVIEW

Adaptive Ideas is a direct-manipulation web page editor augmented with an example browser that enables users to locate and copy elements from other pages. Our prototype is implemented by extending Firefox's built-in design mode (a graphical HTML editor). The prototype's example corpus comprises 250 web pages that we harvested and tagged with faceted metadata.

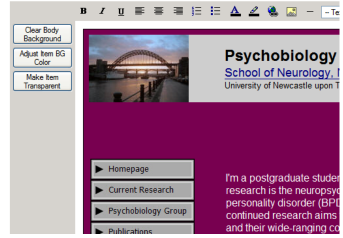
Adaptive Ideas introduces an optimization-based approach to selecting, presenting, and browsing design material, using decision-theoretic selection and end-user preference as inputs. The core of our approach is a *subset selection algorithm* that chooses examples from a corpus to display so as to maximize estimated design value. Using distance metrics calculated along facets of the available designs, we define two criteria of interest to the presentation of design materials: *similarity* and *variety*. Interface layouts are generated automatically by assigning utility values to visual elements and searching the space of valid interface layouts for an optimal solution, using dynamic programming and branch-and-bound methods to boost performance.

The Adaptive Ideas interface provides an editing pane, an examples pane, and a focus example pane (Figure 1). The user can select an example to work from by clicking on one of the displayed examples; this action initializes the web editor with this example page. Subsequent examples, when selected, are displayed in the focus pane (Figure 1, bottom-left). The focus pane allows the user to see greater detail in the example page selected by displaying this example at a larger size than in the example pane (currently, 20 times larger). The user can select a feature of this example to borrow, and request to see examples in the example pane based on similarity of some parameter with the focus pane example.

Scenario

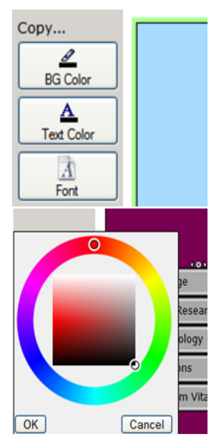
We illustrate the envisioned use of the Adaptive Ideas editor with an example scenario. Elaine Marsh is a 21-year-old economics student, starting her senior year. Studious and reserved, Elaine spends much of her time outside the classroom serving as vice president of the Alpha Beta Gamma honor society and volunteering as a tutor at a local high school. Elaine wants to make a homepage that details her undergraduate activities, including class projects, research papers, and leadership positions. Her vision for the page includes a mature, sophisticated design and a slightly conservative feel.

She opens the Adaptive Ideas web page builder and is presented with a variety of possible starting points for her website. She browses through them, looking for a design that she thinks is appropriate. Elaine chooses a two-column design with a purple background.



The interface displays her selection in the editing area. Links and buttons indicate features that the user can copy and where to paste them. A text note reminds Elaine that she can browse more examples and copy elements from each of them to her prototype, or she can edit things manually using the controls along the top of the interface.

Elaine decides that she would prefer a different background color, so she selects "Show a variety of background colors." The interface presents several examples spanning different hues, saturations, and brightness. Elaine selects one of the blue examples and then clicks "Show examples similar to this one"; a set of blue and purple examples is displayed. She sees an example with a tasteful light blue background that she fancies. She clicks on *background color*, clicks on the blue of the example, and clicks a third time on the prototype to replace the purple background with the new blue.



Elaine still isn't completely satisfied with the background color, so she clicks on the color widget at the top of the interface, and selects the background of the prototype. A color wheel pops up, and she lightens the blue. She then clicks on the text of the prototype web page, and enters the information and navigation she desires.

She continues to browse examples looking for inspiration. A page that uses the Georgia font catches Elaine's eye; after some consideration, she switches the prototype to Georgia, both for readability and style, and alters the font size using the manual controls. She adds a head-and-shoulders picture of herself to the top-left corner of the page. Next to the picture, she places a prominent link to her resume. Satisfied, she uploads the page to a server.

Selecting Examples to Display

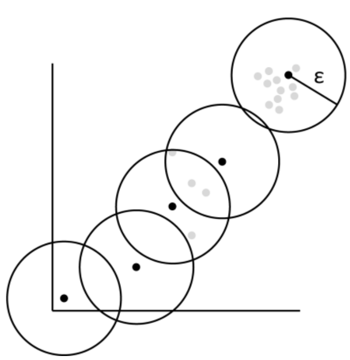
To decide what examples to display, Adaptive Ideas introduces a subset-selection algorithm that attempts to maximize estimated design value. It operationalizes design value through two proxy measures: the *usefulness* of example content to the user's current task or request, and the *value* associated with the size of display elements in the overall interface. This section describes how content is selected for display.

In the Adaptive Ideas prototype, the example content elements are homepages harvested from the web. These elements have faceted attributes [31], which can be flat or hierarchical. This paper focuses on using *visual properties* as attributes: background color, primary font, column layout, and visual density. Here, we manually assigned values for each page attribute; we believe that a production implementation could tractably assign them automatically.

For each pair of elements in the corpus, Adaptive Ideas computes distance metrics that model the likeness of each facet attribute pair. The distance between two attributes is a real number whose value depends on the properties of the attribute. For example, Adaptive Ideas defines the distance between two colors to be their Euclidean distance in the biconic HSB color space. In contrast, the distance for fonts is a simple ternary function: 0 if the fonts are the same, 1 if they are both serif or both sans serif fonts, 2 if the fonts do not share serif characteristics.

With Adaptive Ideas, users can request to see a set of examples that are either *similar* to a specified element, or that represent a *variety* of the options available along a particular dimension. To derive similarity subsets, Adaptive Ideas simply calculates the distance of all objects from the focus, sorts them in ascending order, and selects the first *n* items. Our intuition is that showing similar examples will be useful for designers that are looking for subtle design variations, and that showing a well-selected variety of examples will be more valuable early in design, when designers are looking for broad inspiration. This raises the question of what defines a "well-selected" variety: one that shows off all possible values of the given attribute, one that represents the distribution of the underlying dataset, or one following some other formula? For instance, the majority of websites in our prototype dataset have a background color of white. An algorithm that tries to represent the distribution of the dataset would contain mostly white web pages; in this case, an algorithm that focuses on displaying a variety of possible values may be more desirable, allowing users to see the full design space. On the other hand, such an algorithm may emphasize outliers or unusual points in the design space.

The Adaptive Ideas framework takes a spaced stochastic approach to selecting a representative variety. First, the system picks a random example from the dataset as a starting point. Next, a random example is selected from the remaining elements in the dataset which are at least ε distance away from all of the elements selected thus far, where ε is a spacing function defined on a per-attribute basis. Similar approaches are used in graphics for point sampling and in biochemical research for finding dissimilar subsets of compound databases.



The choice of ε significantly influences the behavior of the spaced stochastic algorithm. When ε is zero or small relative to the design space, this algorithm degenerates to the completely random case. As ε gets larger relative to the space, the algorithm has fewer elements from which to choose, and thus risks not filling up the space. We select a large ε, such that the theoretical maximum number of elements chosen is close to *n*.

The algorithm continues picking elements until either *n* elements have been selected or there are no legal elements remaining, *i.e.*, every remaining unselected element is less than ε distance away from an element in the selected subset. If more elements are needed, the system selects elements at random from the full set of remaining elements until *n* have been chosen. On balance, ε guarantees that distinctly different values for the given attribute will be represented in the variety set, while filling out remaining elements randomly implies that some of the underlying distribution of values will be reflected. A variant would be to iterate over successively smaller values of ε until enough legal elements are found; this would further emphasize the breadth of the design space.

Laying out the Overall Interface

The Adaptive Ideas framework handles the layout for both static content elements and dynamic interactive elements. It uses a combination of developer specifications and adaptive techniques. Developers may create XML-based *templates* to partially specify the appearance and behavior of an example display. Similar to Damask [16], templates allow interface designers to specify grouping and layout of content and interactive elements in a device-independent fashion. Using templates, designers can also embed interactive web components which are not part of the Adaptive Ideas system but which provide other services, such as email and calendaring systems. This ability to include any HTML snippet that represents interactive content introduces a mash-up approach to creating adaptive displays.

The Adaptive Ideas algorithm decides how to visually present the display elements by selecting a *layout style*. The layout style is a function of the output display *D* and template *T*, and is specified as a set of tuples:

$$<e, x, y, w, h>$$

Here, *e* is an element, *x* and *y* are the element's position in this layout, and *w* and *h* are the width and height in pixels of the element. In the Adaptive Ideas implementation, all elements are allocated rectangular regions.

Presentation Value

A key consideration when choosing a layout is deciding the size to render display elements. Larger items are generally easier to read and select, and therefore correspond to higher value. However, increased space for one element necessarily implies less space for another. We encode this tradeoff in a *presentation value function*: $p(e,w,h,D)$. This function estimates the utility of presenting a display element e at a given size (w, h) in a given display D.

In general, larger sizes receive higher presentation scores and smaller sizes receive lower scores. However, the relationship between size and value is not necessarily linear. (One could extend the Adaptive Ideas architecture so that this relationship is defined on a per-instance basis, perhaps using metrics similar to those of Suh *et al.* [25].) Our prototype defines the value of an image as a monotonically increasing function that starts at zero (attributing no value to seeing an invisible image), scales up quickly at small sizes (attributing some value to see a small image, *e.g.*, a thumbnail), and approaches an upper limit at larger sizes (once large enough, an image provides little additional value in growing larger).

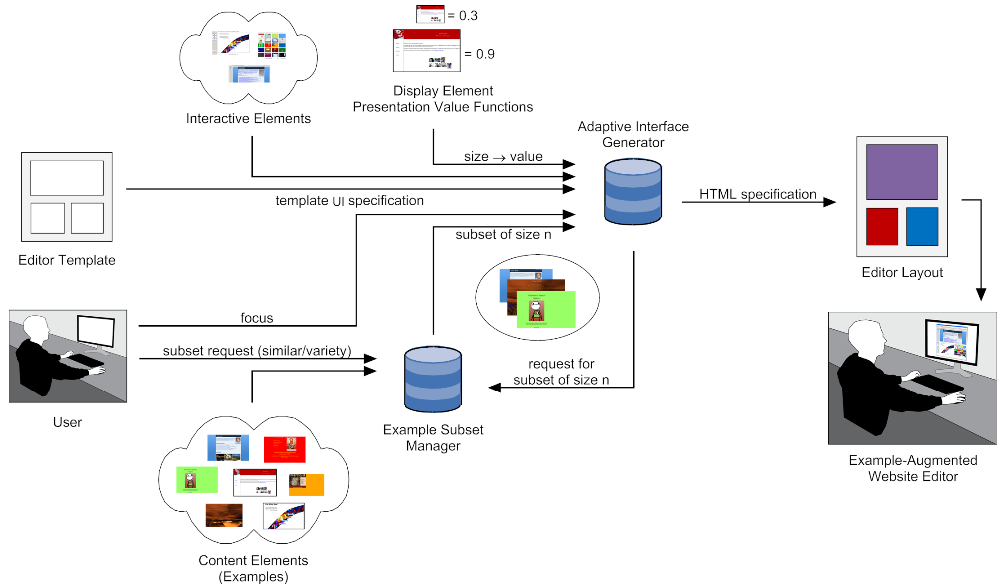


Figure 2. The Adaptive Ideas system uses subset selection algorithms and layout functions to automatically generate example-augmented interfaces.

We derived the presentation value functions for the current Adaptive Ideas system by assessing utility of the various element types (text, images, interactive widgets) at different sizes, then hand-tuning the functions and their parameters. The presentation value of containers is the sum of the presentation values of the content elements it contains.

To estimate the usefulness of seeing an interactive element d given the current foci F, Adaptive Ideas employs a *display relevance function*: $r(d,F)$. Generally, r rates elements that the user has explicitly requested higher than those being shown peripherally, and assigns elements which are not needed an r score of 0.

Adaptive Calculations

To summarize, Adaptive Ideas receives the following inputs from the system and the environment: content elements, interactive elements, output display, and a design template (Figure 2). The algorithm searches the space of possible layouts and selects the layout with the maximum estimated utility for the given output display.

Estimated Display Element Value

For displays of content elements in information grids, we use *row-major ordering* (left-to-right, top-to-bottom). Given a focus F, the estimated value of an interactive element at a given size is a multiplicative function of its relevance to the given focus and its presentation value at the given size:

$$s(e,w,h,F) = p(e,w,h) \times r(e,F)$$

A low *presentation* or *relevance* score yields a low value, even when the other input score is high: a highly relevant item is of little value if it is unrecognizable, and a prominently displayed item is not valuable if it is not relevant. In the current prototype, location does not affect estimated value; the framework could certainly be extended to provide location-dependent value estimation.

Estimated Layout Value

Given a focus F, the estimated value of a layout is the sum of the estimated values of all elements displayed:

$$s(F,L) = \sum_{e \in L} s(e,w,h,F)$$

Here, w and h are the width and height of element e in layout L. This function presumes that the contributions of a given element are *independent* of the presence or absence of other elements. There are, of course, cases where the total value may be supermodular (the whole is greater than the sum of the parts) or submodular (diminishing returns).

Finding the Optimal Interface Layout

The task of optimally laying out a set of elements (e.g., how large each should be) can be viewed as a two-dimensional variant of the knapsack problem [17], where the constraining resource is display real estate. As knapsack exhibits the optimal-substructure property, we use dynamic programming; here, partial interface layouts are the subproblem result being cached. To boost performance, we take a branch-and-bound approach, wherein subproblem solutions that are estimated to be worse than the current best are immediately discarded. We also iterate through possible dimension values in five-pixel increments instead of evaluating every possible integer width and height.

Optimizing layout is simplified when laying out content elements in an information grid. As fractional displays of content elements are useless, the algorithm needs only to search through a small range of discrete size settings, specifically sizes that result in an exact integer number of elements either across or down for a given size. Finding the best set of content items to display at a given size then becomes a greedy search, linear in the number of elements.

Intuitively, the information presentation problem trades off showing a smaller number of items at larger sizes and showing a larger number of items at smaller sizes. The Adaptive Ideas framework succinctly quantifies this tradeoff, enabling efficient evaluation of candidate interfaces.

EVALUATION

The Adaptive Ideas editor enables designers to parametrically browse through a set of examples pages and select attributes from them for use in the pages they create. To understand whether designers produce higher-quality pages when provided with explicit mechanisms for viewing and borrowing example of other designs, we conducted two user studies. The first experiment tested whether designers produced better pages when designing *with* an example-augmented editor than without one. The second tested whether *adaptively* selecting examples for display yielded a different user experience than *randomly* selecting them.

Experiment 1: Designing With vs. Without Examples

Experiment 1 comprised two parts. In the first part, we asked participants to design web pages for the same two personas. In the second part, we asked a separate set of participants to rate the pages designed in the first part.

Method, First Part: Designing

Twenty-six students from our university participated in the first part of the study for course credit. All of the participants were frequent web users; half had little-to-no web design experience (here, we refer to these participants as "novices"), half had previously made more than one web page (here, we refer to these participants as "experts"). In both conditions, users were presented with 12 templates to choose from as the initial layout; they were free to modify these initial designs as they saw fit.

This study used a simplified version of the Adaptive Ideas editor. In the *Examples* case, the subset manager was set to always display a random subset of the examples; our rationale for this was to first test whether a straightforward example augmentation helped. (Experiment 2 looks at the benefits of adaptive display.) Also, we disabled the mechanism for automatically copying attributes of examples; users had to manually specify the desired attributes. Our rationale here was similar: that automatic copying provides a performance enhancement, and so a positive result for manual example-borrowing would be a stronger one. The control case did not display the example pane and focus pane to the user at all.

Every participant received the *Elaine* persona first and the *Bob* persona second (see Appendix). Half the participants used the Examples editor first; the other half used the Control editor first. We evenly distributed the experts and novices across the two groups. Figure 3 shows a sampling of web pages produced during the study.

Users were presented with a questionnaire before and after each task. Some questions—aimed at tracking self-confidence in web design and perceived need for external assistance—were repeated before and after. In the post-task questionnaire, users were also asked to report on their satisfaction with the web page they had made.

Additional questions were presented to users after the Examples condition. These questions asked about whether viewing examples made the design process more engaging; the effects of examples on a user's evaluation of his or her own web page; the influence of examples on the participant's resulting design, and the types, number, and size of desired examples.

Figure 3. Left: "Elaine" pages created in the Examples condition. Right: "Elaine" pages created in the Control condition

Rate on a scale of 1 to 7, where 1 = not at all, 4 = moderately), and 7 = a lot

1. How suitable is the color scheme of the web page for the persona depicted above?
2. How suitable is the layout of the web page for the persona depicted above?
3. How suitable is the overall design of the web page for the persona depicted above?

Below, the scale is 1 (strongly disagree) to 7 (strongly agree)

4. This is a sophisticated web page.
5. This is a visually attractive web page

Figure 4. The five questions raters were asked about each of the pages created in Experiment 1.

Method, Second Part: Rating

Forty-six students from our university participated in the second part of the study for course credit; none of them participated in the first part. This second part was conducted over the web, and asked participants to rate a subset of the 50 pages created in the first part. (Two pages for each of 25 participants; one participant did not successfully complete the study.) Participants were shown 14 randomly selected pages: 7 each of Elaine and Bob. To eliminate an effect of persona order on rating, we counter-balanced the presentation order; *i.e.*, half the participants saw the Elaine set first and half saw the Bob set first. At the beginning of each "block" (Elaine or Bob), participants were shown the persona description, and the site explained they would see several pages that people had created for that persona. In total, this yielded 644 page ratings: most pages were rated by 13 participants; some by 12. For each page, the survey asked five questions about the quality of the page for that persona. Raters responded to each question using a 7-point Likert scale; Figure 4 lists the questions and summarizes the scale used.

Results

To avoid priming confounds, the following analysis of the ratings is solely for the first task. As predicted by the hypothesis that seeing examples aids design creation, pages created in the Examples condition were rated more highly ($\mu=4.0$, $\tilde{x}=4.0$, $\sigma=0.4$) than those created in the Control condition ($\mu=3.3$, $\tilde{x}=3.1$, $\sigma=0.7$); this difference is significant ($p < 0.0146$). Aggregating both conditions, pages created by experts were rated more highly ($\mu=4.01$, $\sigma=0.97$) than those created by novices ($\mu=3.4$, $\sigma=0.68$); this was also true for each condition independently. Figure 5 presents a graphical summary of the data.

For the first part of the experiment, the questionnaire used a 7-point Likert scale, ranging from 1 (strongly disagree) to 7 (strongly agree). In their responses, novices disagreed with the statement, "the display of examples did not help me as I already knew what I wanted the web page to look like" ($\mu=2.9$, $\sigma=1.55$) while experts were neutral ($\mu=3.9$, $\sigma=1.44$). Novices also felt that the examples allowed them to gauge how good or bad the web page they were making was ($\mu=5.0$, $\sigma=1.76$), while experts were once again neutral ($\mu=4.1$, $\sigma=1.688$). In addition, novices disagreed ($\mu=2.4$, $\sigma=1.2$) more than experts ($\mu=3.2$, $\sigma=1.8$) on whether the examples restricted their own creative instincts.

Results were mixed as to whether the participants who used the Examples-display editor gained more confidence in their own design capabilities than those who used the sans-Examples-display editor, or the Control. An independent samples t-test was conducted to compare the results of the question asked in the second task: *During the process of making the web page, to what extent did you feel the need to surf the internet for better design ideas?* There was a significant difference in the score for people who had used the Examples-based editor ($\mu=3.46$, $\sigma=1.46$) compared to those who used the Control ($\mu=2.3$, $\sigma=0.751$) $t(18) = 2.528$, $p=0.021$ (equal variances were not assumed). However, results were not significant in response to the other similar self-confidence reporting questions.

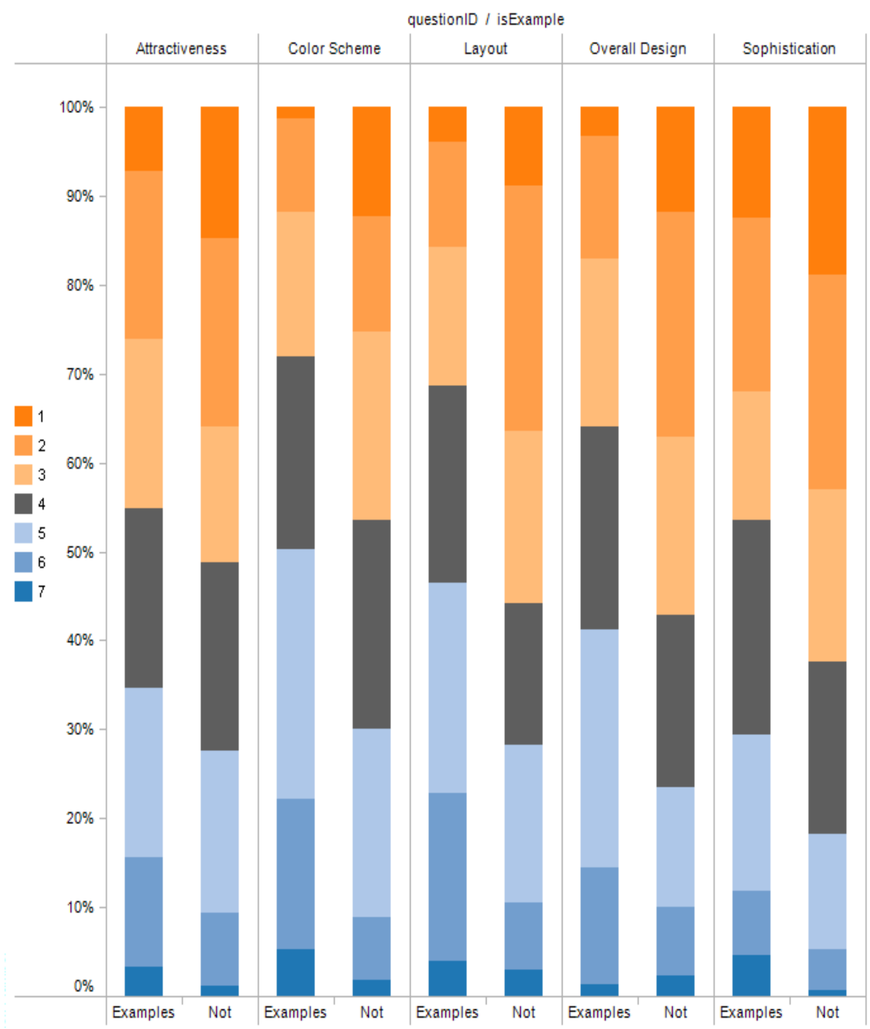


Figure 5. Ratings for the first task, according to question asked. The median for pages created in Examples condition is greater than those created in the control condition.

Responses to practical aspects of examples display

Finally, the participants indicated that the display of examples would have been more helpful if a lesser number of example web pages were shown at one time. The current implementation displays 20 example pages in the example pane when both the web editor and the focus pane were both initialized, laid out in five rows of four examples each. The current display size of each example was 103 pixels by 64 pixels. Participants felt that the number of examples displayed were excessive (μ=5, σ=1.7), and felt that the number of examples displayed should have been fewer with each example displayed at a larger size (μ=4.9, σ=1.6). Participants were told that there were 20 example web pages shown to them at a time, and then asked what they felt the ideal number of pages shown is. The responses indicated that half the number of pages would have been ideal (μ=10, σ = 3.2).

Experiment 2: Adaptive vs. Random Example Selection

Nine subjects participated in this study; they were compensated with a US$15 gift certificate. Participants' ages ranged from 24 to 30; six were male. All were frequent web users; two self-rated as experienced or expert web designers; the others had little-to-no web design experience.

Method

Participants were seated at a workstation with the Adaptive Ideas editor. Sessions began with a demonstration of its functionality. Participants were then asked to create web sites for two different personas (see Appendix), using a different variation of the interface for each. The *standard features* variant disabled the similarity and variety features and sorted the examples randomly: users could view all examples, but could only browse them using the next and previous page controls. The *adaptive features* variant offered the full set of controls described in the Implementation section. Personas and interfaces were varied across participants using a Latin square ordering. After the two tasks, participants completed an eighteen-question survey with no time restrictions. This questionnaire asked about their background and about the user experience of the two variants of the system they used. The two variants of the system were referred to as "circle" and "square" (*not* by whether it was the adaptive version or not).

Results

Figure 6 summarizes the questionnaire results. Participants found the general presentation of examples highly useful (mean=4.5, median=4.5, σ=0.53, on a 5-point Likert scale), and appreciated the ability to borrow features directly from example web pages (μ=3.9, $\tilde{x}$=4, σ=0.83). Participants found the adaptive browsing features to be helpful in finding examples, indicating the variety tool to be most useful while exploring the design space (μ=4.4, $\tilde{x}$=5, σ=1.01), although the similarity tool was also welcomed (μ=3.9, $\tilde{x}$=4, σ=0.78). In general, participants did not find the examples to be distracting (μ=2.2, $\tilde{x}$=2, σ=0.83). Several users expressed a desire for the ability to browse along aesthetic or social attributes, such as formality.

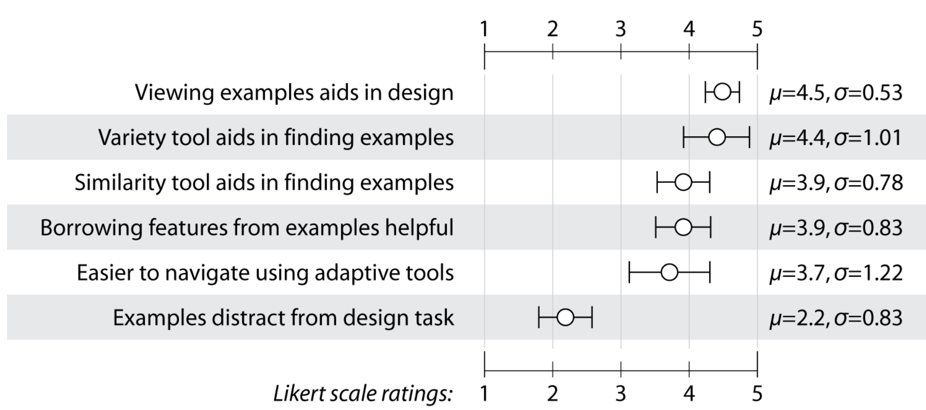


Figure 6. Questionnaire results from Experiment 2.

Participants responded mildly positively as to whether it was easier to navigate examples using the adaptive features interface (μ=3.7, $\tilde{x}$=4, σ=1.22). During the studies, however, we observed that several participants resorted to long stretches of "linear" browsing while using the standard features interface, during which they clicked on many examples in a row in order to examine them.

Interaction logs supported these observations. In the adaptive interface, users selected approximately 50% fewer items for larger viewing (μ=195.8 in the standard features interface versus μ=96 in the adaptive case, $p<0.05$). We hypothesize that the variety and similarity tools lent themselves to more efficient exploration, allowing designers to quickly form a mental model of the design space.

In aggregate, novices responded more positively about the value of examples (μ=4.7, $\tilde{x}$=5, σ=0.49) than experts. In particular, novices responded more positively that, "the ability to see a variety of items along a given dimension in the Square [adaptive] system was helpful in finding useful examples" (μ=4.8, $\tilde{x}$=5, σ=0.38).

The two self-rated experts differed strongly in their opinions about the use of examples for website design. One found the limitations of the example-borrowing interface "annoying" and thought the examples wasted screen real estate and distracted from the task. The other experienced participant commented that the browsing of examples worked well with her personal strategy for this type of design task: "That's my philosophy of designing web sites: I like to find a template or exemplar that I think is good and then tweak it by hand." This suggests that example-augmented design tools should allow users to disable example display.

RELATED WORK

This research draws on three areas of prior work: model-based user interfaces, automatic layout systems, and document scoring systems. We discuss each in turn.

Model-Based User Interfaces

The area of model-based user interfaces (*e.g.*, [21]) began with the interest of creating tools for specifying interfaces declaratively through high-level semantics, rather than imperatively by the pixel-level details of the implementation. Szekely [26] provides a retrospective overview of this field. In this work, we use model-based techniques to produce interfaces that do not depend directly on output specifics, but are instead generated dynamically and optimized along high-level, task-oriented values and constraints.

Automatic Layout

Several projects have explored the automatic layout of interfaces and/or information. Perhaps the two most closely related systems are SUPPLE [12] and RIA [32], which examined constraint-based optimization approaches to interface adaptation. We apply a decision-theoretic strategy similar to that of SUPPLE and RIA, but with significantly different constraints. Adaptive Ideas addresses both user interface elements and visual information sources, and supports a mix of adaptive generation and designer specification. It also has the additional burden of rendering layouts interactively, potentially introducing interesting tradeoffs between optimality and performance. Finally, the RIA system dealt with highly-structured, heavily-faceted metadata [31]; its algorithms depended on an intricate understanding of the dimensions and their relationships. Adaptive Ideas is designed for less-structured, more loosely-related data.

Many techniques have been introduced for laying out and browsing large image collections. PhotoMesa [1], a zoomable image browser which encouraged serendipity using a 2D space-filling layout, inspired several design decisions in our implementation (*e.g.*, quantum elements). Saliency-based cropping methods [25] are another innovation that could be applied to later versions of our adaptive interface, posing interesting questions regarding presentation value functions for content. Our adaptive interface research extends this body of work by applying novel techniques in the context of large heterogeneous data sets.

The selection of what information is visible and its arrangement for the user has significant implications for the cognitive activities that are ready-at-hand [15], and the effective presentation of personal information has been the subject of considerable activity. Furnas's fisheye calendar [11] first introduced the idea of a *focus+context* visualization: the calendar item in focus was displayed larger and with local detail; non-focus items would correspondingly shrink. More generally, this example demonstrated how constraints can be effectively used to manage screen layout globally, and this present research is a continuation in that vein. Other research has explored book-like metaphors for information collections [5], and facet-based approaches to search [9]. Our approach draws strongly on faceted search; it distinguishes itself in that display elements are not constrained to be *only* those requested—elements with similarities to those requested may be displayed as a means of providing for serendipity in search and browsing.

Ambient displays have explored the use of spaces and surfaces for proactive presentation of information [7, 29, 30]. Our research follows up on this work by applying adaptive techniques to contextual displays. In particular, we are exploring the peripheral presentation of examples and other epistemic artifacts to encourage exploration.

Document Scoring

We turn to the question of the underlying algorithms and information model. As with prior work on information foraging [20], we seek to improve the *information scent* of interfaces. More precisely, the goal of this paper is to provide scents of potentially valuable information in addition to the specific information has requested. The use of *small steps* observed by Teevan *et al.* in their study of orienteering behavior [27] points to the value of providing scent via contextual information.

As the quantity of information we work with increases [18], and metadata becomes ever more prevalent [2], improved techniques for sorting this information are required. Adaptive user interfaces have proven particularly useful in managing personal information. Rhodes' Remembrance Agent demonstrated the use of richer types of metadata—most notably location—as a means for retrieving information [22]. Perhaps most similar to Adaptive Ideas is Horvitz *et al.*'s email ranking system [23], which employs decision-theoretic techniques to prioritize and rank emails that are likely to contain higher value information or be more urgent; this work was very inspirational in framing our approach. Haystack[14] takes a highly flexible approach to data presentation and user interaction that could easily integrate adaptive techniques to increase visibility.

The information model in this work draws on the idea of faceted metadata [31], the conceptually distinct dimensions of the metadata. Of particular value has been the recent research on lightweight techniques for labeling photographs with rich metadata [8, 19], and the use of those in information retrieval. Again, the difference with this work is that while we employ the same ontological mechanisms, the contribution lies in the use of this schema to enable proactive and adaptive display.

CONCLUSION

This paper contributes an algorithm for dynamically selecting content for and generating layouts of example artifacts, using a combination of decision-theoretic selection, designer specification, and end-user preference. Two user studies were conducted which indicated that novice web page makers found the Adaptive Ideas web page maker to be useful. These studies found that independent raters prefer designs created with the aid of examples, that the benefit examples provide is higher for novice designers than experts, and that designers prefer adaptively selected examples to random ones. In addition to the technical and empirical contributions, we hope that researchers will find this approach to evaluation useful. The experimental paradigm of having one set of subjects performing a design task and a second, independent set of subjects rating task results may be useful for evaluating certain classes of systems and interactions that are difficult to assess using traditional methods.

Future work includes integrating other attributes (*e.g.*, page metadata such as creation time, title, and keywords; aesthetic properties such as genre and formality), and deriving design attributes and values programmatically. We also plan to investigate alternative representations for examples (*e.g.*, representative exemplars of example subsets), and examine example-based interactions that use more implicit cues from task activities to proactively display content.

ACKNOWLEDGEMENTS

We thank Heidy Maldonado, Ron Yeh, Wendy Ju, and Yoav Shoham for their insightful discussions, and the National Science Foundation and the Wallenberg Global Learning Network for sponsoring this research (NSF IIS-0534662, KAW 2004.0184). We thank Intel for technology donations.

REFERENCES

1 Bederson, B. B. PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps. *UIST 2001: ACM Symposium on User Interface Software and Technology*. pp. 71–80.
2 Berners-Lee, T., J. Hendler, and O. Lassila. The Semantic Web, *Scientific American*, May, 2001.
3 Brafman, R. I. and D. Friedman. Presentation Adaptation for Rich Media Messages. *STRIMM Consortium Working Paper 2003*.
4 Buxton, B., *Sketching User Experiences*: Morgan Kaufmann, San Francisco, Californiapp. 2007.
5 Card, S. K., L. Hong, J. D. Mackinlay, and E. H. Chi. 3Book: a scalable 3D virtual book. *CHI 2004: ACM Conference on Human Factors in Computing Systems*. pp. 1095–98.
6 Carter, D. E., *The Big Book of Logos*: Watson-Guptill.384 pp. 2001.
7 Churchill, E. F., L. Denoue, J. Helfman, and P. Murphy. Sharing multimedia content with interactive public displays: a case study. *DIS 2004: ACM Conference on Designing Interactive Systems*. pp. 7–16.
8 Davis, M., S. King, N. Good, and R. Sarvas. From context to content: leveraging context to infer media metadata. *MM 2004: ACM International Conference on Multimedia*. pp. 188–95.
9 Dumais, S., E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff I've seen: a system for personal information retrieval and re-use. *SIGIR 2003: ACM Conference on Research and Development in Information Retrieval*. pp. 72–79.
10 Fischer, G. User Modeling in Human–Computer Interaction. *User Modeling and User-Adapted Interaction* 11(1). pp. 65-86, 2001.
11 Furnas, G. W. Generalized fisheye views. *CHI 1986: ACM Conference on Human Factors in Computing Systems*. pp. 16–23.
12 Gajos, K. and D. S. Weld. SUPPLE: automatically generating user interfaces. *IUI 2004: Proceedings of the 9th international conference on Intelligent user interface*. pp. 93–100.
13 Gentner, D., K. J. Holyoak, and B. N. Kokinov, *The Analogical Mind: Perspectives from Cognitive Science*: M.I.T. Press.520 pp. 2001.
14 Karger, D. R., K. Bakshi, D. Huynh, D. Quan, and V. Sinha. Haystack: A general-purpose information management tool for end users based on semistructured data. *Proc. CIDR*. pp. 13–26, 2005.
15 Kirsh, D. The Intelligent Use of Space. *Artificial Intelligence* 73(1-2). pp. 31–68, 1995.
16 Lin, J. and J. A. Landay. Employing Patterns and Layers for Early-Stage Design and Prototyping of Cross-Device User Interfaces. *To be published in Proceedings of CHI Letters: Human Factors in Computing Systems, CHI 2008*.
17 Lodi, A. and M. Monaci. Integer linear programming models for 2-staged two-dimensional Knapsack problems. *Mathematical Programming* 94(2-3). pp. 257–78, 2003.
18 Lyman, P. and H. R. Varian, *How Much Information?* 2003. http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/
19 Naaman, M., S. Harada, Q. Wang, H. Garcia-Molina, and A. Paepcke. Context data in geo-referenced digital photo collections. *MM2004: ACM International Conference on Multimedia*. pp. 196–203.
20 Pirolli, P. and S. K. Card. Information foraging. *Psychological Review* 106(4). pp. 643–75, 1999.
21 Puerta, A. R., E. Cheng, T. Ou, and J. Min. MOBILE: user-centered interface building. *CHI 1999: ACM Conference on Human Factors in Computing Systems*. pp. 426–33.
22 Rhodes, B. J. The wearable remembrance agent: A system for augmented memory. *Personal Technologies* 1(4). pp. 218–24, 1997.
23 Sahami, M., S. Dumais, D. Heckerman, and E. Horvitz. A Bayesian approach to filtering junk e-mail. *AAAI 1998: Workshop on Learning for Text Categorization*.
24 Schön, D. A., *The Design Studio: An Exploration of its Traditions & Potential*. London: RIBA Publications Limitedpp. 1985.
25 Suh, B., H. Ling, B. B. Bederson, and D. W. Jacobs. Automatic thumbnail cropping and its effectiveness. *UIST 2003: ACM Symposium on User Interface Software and Technology*. pp. 95–104.
26 Szekely, P. Retrospective and Challenges for Model-Based Interface Development. *DSV 1996: Design, Specification, and Verification of Interactive Systems*. pp. 1–27.

27 Teevan, J., C. Alvarado, M. S. Ackerman, and D. R. Karger. The perfect search engine is not enough: a study of orienteering behavior in directed search. *CHI 2004: ACM Conference on Human Factors in Computing Systems*. pp. 415–22.
28 Terry, M. and E. D. Mynatt. Recognizing creative needs in user interface design*.* *Proceedings of the 4th conference on Creativity & cognition*. pp. 38-44, 2002.
29 Vogel, D. and R. Balakrishnan. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. *UIST 2004: ACM Symposium on User Interface Software and Technology*. pp. 137–46.
30 Wisneski, C., H. Ishii, A. Dahley, M. Gorbet, S. Brave, B. Ullmer, and P. Yarin. Ambient Displays: Turning Architectural Space into an Interface between People and Digital Information. *COBUILD 1998: International Workshop on Cooperative Buildings*. pp. 22–32.
31 Yee, K.-P., K. Swearingen, K. Li, and M. Hearst. Faceted Metadata for Image Search and Browsing. *CHI 2003: ACM Conference on Human Factors in Computing Systems*. pp. 401–08.
32 Zhou, M. X. and V. Aggarwal. An optimization-based approach to dynamic data content selection in intelligent multimedia interfaces. *UIST 2004: ACM Symposium on User Interface Software and Technology*. pp. 227-36.

APPENDIX: PERSONAS AND TASK

Users were presented with the two personas of first Elaine and then Bob. A task was also specified for each persona, as detailed below. Users were allowed 20 minutes to complete each task.

Elaine's persona was described as follows:

Elaine Marsh is a 21-year-old economics student, starting her senior year at Stanford University. Studious and reserved by nature, Elaine spends much of her time outside the classroom serving as vice president of Bases – the student business association. She also volunteers as a tutor at a local high school. Elaine wants to make a homepage that details her undergraduate activities, including class projects, research papers, and leadership positions.

In her personal life, Elaine enjoys making origami and traveling to new places. During her time at Stanford, she has traveled to New York, Boston, Paris and London. She lives in the arts-themed dorm, Kimball in Manzanita Park.

Her vision for the page includes a mature, sophisticated design and a somewhat professional feel.

The users were asked to make a web page for Elaine with the following minimum requirements:

- Choose a layout suitable for Elaine's vision of her page.
- Place an image that is relevant to Elaine's activities. This image may be from an example page, a web page on the internet or somewhere on the computer.
- Customize a vertical or horizontal (or any other fancier variation) menu bar.
- Set the Color of the page such that the color scheme is true to Elaine's vision of her page.
- Set the fonts of the page.
- Enter textual information about Elaine.

Beyond these requirements, the users were free to customize the page as much as desired.

Similar to the first task, for the second task, the users were presented with the persona of Bob:

Bob Jones is a 20-year-old junior at Stanford University, majoring in Human Biology. His interest in human biology have led him to do research at the Stanford Medical School on the physiological effects of being in love. Being an extrovert, he enjoys being part of Stanford's social events. He is part of the organizing committees for Dance Marathon and the Viennese Ball, where he in charge of marketing and publicity.

In his spare time, Bob practices his violin skills and goes on long cycle rides. The previous summer, he joined a group which biked across the United States from San Francisco to New York. Bob lives in the Roble dorm.

Bob is interested in having a web page for himself, and wants it to be reflective of his personality. He wants it to be a resource for anyone interested in his activities, research and leadership positions.

The task details were identical to those presented for Elaine, with the emphasis on making the visual design of the web page suitable to Bob's personality.