Adaptive Interfaces for Declarative Presentation of Heterogeneous Content

Brian Lee, Scott R. Klemmer
Stanford University HCI Group
Computer Science Department
Stanford, CA 94305-9035
{balee, srk}@cs.stanford.edu

Ronen Brafman
Stanford University Multiagent Group
Computer Science Department
Stanford, CA 94305
brafman@cs.stanford.edu

ABSTRACT

Visibility of work practice is important because it enables peripheral participation of and facilitates coordination between colleagues. Moving activities from the physical world onto the digital desktop has diminished visibility by consigning the artifacts of work practice to the computer screen; the serendipity of stumbling across physical artifacts is lost. One method of reintroducing visibility is the proactive display of colleagues' digital work artifacts. This paper introduces an *adaptive content presentation technique* designed to improve the visibility of content for both ambient awareness and interactive browsing. In this work, we define the information presentation problem to be dynamically focusing user attention to a maximally useful subset of available information. Our technique takes a decision-theoretic approach to interface generation, using content metadata as inputs to our algorithm. The data view is generated dynamically, based on high-level attributes of the current state and a *declarative* relationship between the user's input and the resulting view. We have evaluated the technical efficacy of this algorithm by implementing it in the context of the ButterflyNet browser.

Author Keywords

Adaptive interfaces, decision theory, model-based UIs, awareness

ACM Classification Keywords

H.5.2. [Information Interfaces]: User Interfaces — *Graphical user interfaces (GUI); Interaction styles; User interface management systems (UIMS)*. D.2.2 [Software Engineering]: Design Tools and Techniques — *user interfaces*. H1.2. [Models and Principles]: User/Machine Systems.

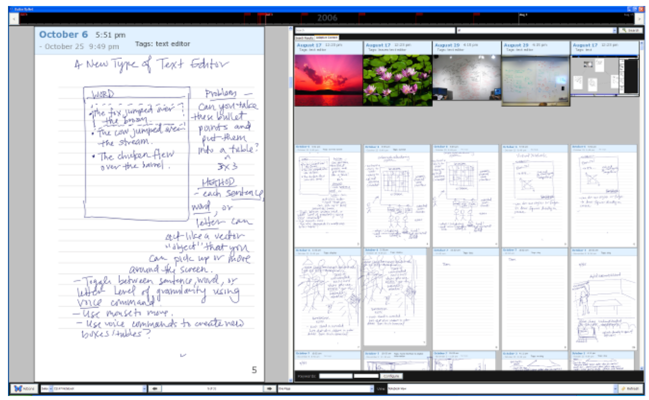

Figure 1. An adaptively generated view of designers' sketchbooks. Note the adaptive sidebar (*right*), which displays content related to the item currently in focus in the media browser (*left*).

INTRODUCTION

Traditionally, information browsing on PCs has had a pull model of information access: search. While search is successful, you can generally only find things that you know you're looking for. This is a result of the *imperative* query style: to see information, the user specifies the set and presentation style of information to view. For small information collections and pragmatic actions (*i.e.*, browsing to accomplish specific, well-defined tasks), imperatively specifying the set and presentation style of information to view is tractable and successful. This is the pragmatics of information browsers.

Increasingly, however, people use computers not only as tools for pragmatic action, but as tools for epistemic action [14]—to support thought processes and provide inspiration. This behavior is particularly prevalent in creative professions such as design, where the goal is often not to produce, but to learn. Some systems, *e.g.*, social networking and collaborative filtering web sites, have introduced interfaces designed to proactively inform users and encourage exploration. Still, most traditional computer tools include little, if any, support for epistemic activity.

To deal with increasingly large information collections—and, more importantly, for supporting epistemic tasks—we suggest that it can be more effective to *declaratively* specify

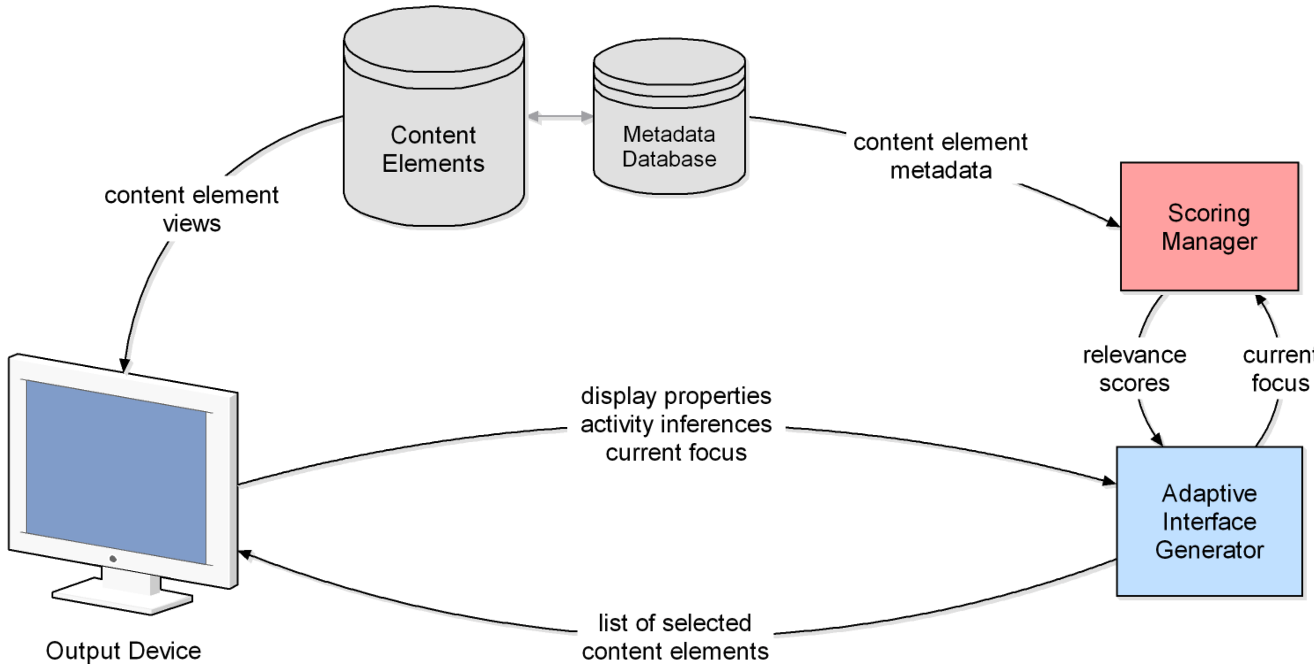

Figure 2. Architecture of a prototypical adaptive browsing system.

an information view. A declarative query style does not require that the user completely specify the results of the query, only an eventual goal, allowing queries to be made implicitly and for results similar to those directly requested to be returned. This opportunistic presentation of data facilitates epistemic activities, by presenting data of interest of which the user may not even be aware. Dynamically focusing user attention to a manageable subset of information that is "most relevant" at a given situation and time, and properly displaying this information, may have profound impacts on the quality and efficiency of the browsing user experience.

To address this problem, we introduce an adaptive interface [21] (see Figures 1, 2) that employs a decision-theoretic approach to selecting information. Unlike traditional static interfaces, adaptive interfaces are "aware" of both general and current user tasks, needs, and preferences. Adaptive interfaces attempt to optimize the presentation of information by emphasizing those contents which are most useful in a given context. Additionally, adaptive interfaces may be *proactive*: that is, they may display relevant content even when the user has not explicitly (imperatively) requested it. The hypothesis manifest in this work is that *proactively presenting information can increase awareness and serendipitous browsing*.

From a technical perspective, this work draws on prior work on model-based user interfaces [28] and automated layout [8, 20]. In particular, we draw on the idea of casting interface generation as constraint-based optimization [3].

This research has also been partially inspired by ButterflyNet [34], a mobile capture and access system targeted at user groups which make use of voluminous and varied sets of data, such as that of designers and field biologists. In this paper, we leverage the ButterflyNet system as a test bed for creating and manipulating heterogeneous content, and extend it to support adaptive interfaces.

This work offers four contributions: a precise definition of the information presentation problem we address, the various dimensions we use to analyze it, algorithms for calculating an appropriate rendering, and a technical evaluation via a manifestation of this adaptive technique in the ButterflyNet browser.

The rest of this paper is organized as follows. We begin by defining the adaptive interface problem and describing at a high level the dimensions that we use to analyze it. We then present a theoretical framework, with precise mathematical formulations of the various axes. We discuss the technical efficacy of our algorithm and outline implementation decisions. Finally, we present several scenarios of actual and envisioned use of adaptive interfaces.

ADAPTIVE INTERFACES

The goal of an adaptive interface may be stated as follows:

Given a set of content elements, a relevance measure, a set of layout constraints, and a user experience goal, show an interface layout that maximizes the utility of the display for the specified user experience.

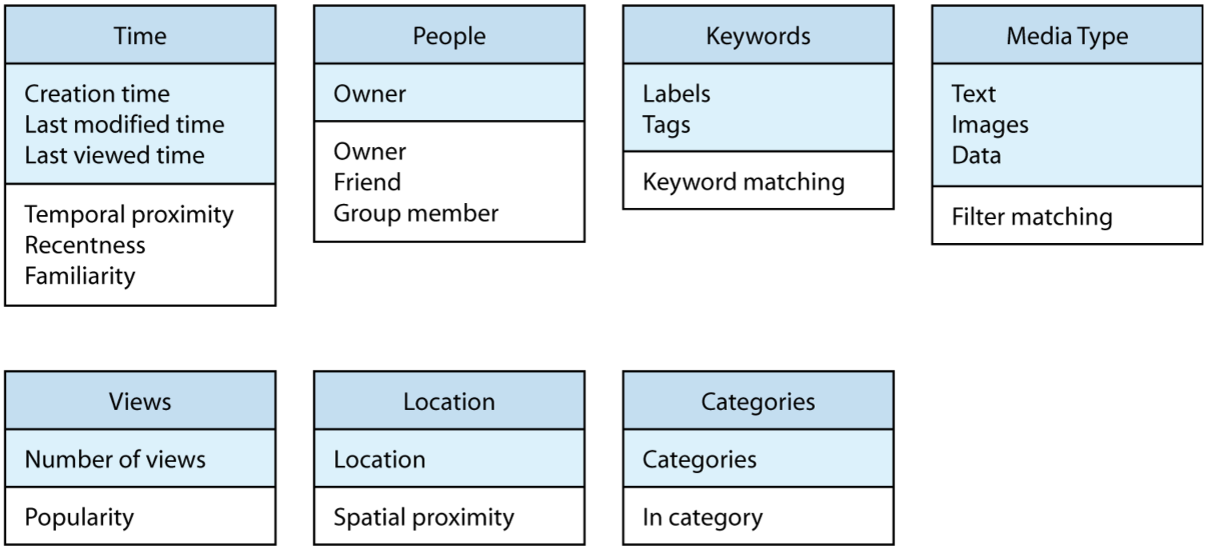

Figure 3. Example categories of content attributes (dark blue), attributes (light blue), and pivot axes derived from those attributes (white).

In this section, we define various parts of a high-level conceptual framework for adaptive interfaces.

Content

A *content element* is a single logical unit of information or data. Content is heterogeneous; we have identified three general categories of content:

Text: handwritten notes, text files, emails, articles

Images: photographs, whiteboard captures, other images

Data: tabular data, graphs, numerical data

The ButterflyNet system features three types of content: handwritten notes, pictures, and whiteboard captures.

Content has various *attributes*, or metadata (see Figure 3). Attributes may be flat or hierarchical. In the current version of the system, the attribute library comprises creation time, modification time, ownership, media type, and tags. Examples of other attributes that may prove valuable include location, last viewed time, and categories [33].

Relevance

At the heart of our decision-theoretic algorithm is the determination of *relevance*: figuring out what information is “most valuable” in a given situation. To assess relevance, we first model the user’s desired focus as a *pivot*: a collection of content attributes and values that serve as metrics for what the user wants to see. Pivots may be explicitly defined (*e.g.*, a keyword search) or implicitly defined (*e.g.*, inferred based on the content elements the user is currently viewing). Using pivots, we can define functions which estimate how useful or relevant a given content element is in a given situation.

Pivots may involve one or more axes along attribute dimensions. Some axes for pivoting based on attributes include keyword matching (matching target keywords based on textual annotations), temporal proximity (difference between creation time and focused item creation time), and spatial proximity (distance between creation location and focused item creation location); Figure 3 lists other possible pivot axes. In ButterflyNet, we have implemented pivots based on linear combinations of temporal proximity, recentness, keyword matching, and media type.

Presentation

The goal of the adaptive browser is to produce a *presentation*, or visual layout, based on display constraints. Presentation depends on the properties of both the output device and the content elements to be displayed.

Visual output devices have the following salient display characteristics:

- Physical width and height: length, *e.g.*, inches
- Display resolution: pixels per unit length
- Available screen space: number of pixels wide and high

Similarly, content elements have the following salient display characteristics:

- Aspect ratio: original shape of the content
- Presentation value: measure of the value of displaying this content at a given size

ButterflyNet does not currently consider physical properties of displays; only screen space in pixels and element aspect ratios are used to evaluate layouts.

Context

Lastly, we consider the role that the adaptive browser plays in the current activity. What is the user trying to do? What is the larger task, application, or user experience goal?

- Attentional level [11]: is the adaptive browser the focus of the interaction, when detail is important, or an ambient display, when glanceability [18] is the priority? These situations have very different implications for layout optimization.
- User preferences: preferences explicitly stated by the user (*e.g.*, relative weight of keywords versus temporal proximity)

In this work, we have explored two use modes for the adaptive browser—main browser and ambient/contextual sidebar—which feature correspondingly different user experience goals for the adaptive portion of the interaction.

THEORETICAL FRAMING

We now provide formal mathematical definitions for the key dimensions of the adaptive interface rendering problem.

Content elements (e) are the basic units of displayable information in our framework. Every content element has a media type T.

Input Variables

The adaptive interface receives the following inputs from the user and environment (see Figure 2):

Display (D): Target output display. For purposes of this algorithm, a display has a set of layout constraints C_D, *e.g.*, available screen space ($w_{max} \times h_{max}$).

Pivot (P): A set of zero or more (attribute, value) pairs representing some measure of interestingness in relation to the user's current desired focus.

User experience goal (G): This is an abstract value representing the desired user experience (*e.g.*, attentional level). User experience goals affect evaluation functions.

Evaluation Functions

Content is evaluated among two axes when calculating an adaptive rendering: presentation and relevance. Both of these functions are highly subjective—they are merely estimates of relative utility.

Content presentation value function (p): This function is an estimate of the value of presenting a given content element with media type T at a given size (w, h) for a given user experience goal G, and is written as $p_T(w,h,G)$. Generally, smaller sizes will receive lower presentation scores. However, different types of content may have different presentation values at the same size; see Figure 4 for an example of how presentation scores may vary by media type.

Content relevance function (r): This function is an estimate of the usefulness of seeing a content element given the current pivot and user experience goal, and is written as $r(e,G,P)$. Generally, content elements which are closer to the pivot (that is, whose attribute values are closer to the pivot's attribute values) will receive higher relevance scores.

Legal Rendering

A rendering, or presentation, is given by a set of tuples of the form:

$$\langle e, x, y, w, h \rangle$$

where e is a content element, x and y are the element's position in this rendering, and w and h are the width and height of the element in this rendering.

A *legal rendering* ϕ is a rendering in which all objects satisfy the display constraints C_D (*i.e.*, they do not fall

Figure 4. Example content presentation value functions (CPVF) for recognition of images versus text. *Left:* Graph of the CPVFs for pictures (red) and notes (blue). *Center:* A picture shown at small (1) and large (2) sizes. *Right:* A page of notes shown at the same relative dimensions (3 and 4). Note that, though both content elements occupy the same physical area, the picture is still recognizable at the small size, whereas the page of notes is not readable at all when small. Thus, these two types of content may have different utility values at the same scale (depending on the user experience goal).

outside the allocated screen area and do not overlap).

Estimated Value of a Content Element

Given a user experience goal G and pivot P, the estimated value of a content element is a function of its relevance value (relative to the current pivot) and its presentation value (at a given size). Our formulation uses a multiplicative function:

$$s(e, G, P, \phi) = p_T(w, h, G) \times r(e, G, P)$$

A more complex model would be one where an element's location also affects its value. In such a model, the same element would get a different score if it appeared in the center or the side, near the top or near the bottom. In our formulation, the value of an element does not depend on location.

Estimated Value of a Presentation

Given a user experience goal G and pivot P, the estimated value, or score, of a presentation with elements E is a function of the estimated values of all elements displayed in the presentation. We assume that the function is linear, specifically a sum of the individual values:

$$s(G, P, \phi) = \sum_{e \in \phi} s(e, G, P, \phi)$$

We recognize that there may be interactions between different content elements that may either increase (*e.g.*, due to synergies) or decrease (*e.g.*, due to clutter or overlap) the presentation score. In the current system, we assume that the contributions of a given content element are *independent* of the presence or absence of other content elements. Relaxing this restriction will be the subject of future work.

Tradeoffs

Intuitively, the information presentation problem is a tradeoff between showing a smaller number of more relevant items at larger sizes and showing a larger number of less relevant items at smaller sizes. The framework presented here quantifies this tradeoff neatly and succinctly. Though simplified in a number of ways, this framing identifies a quick and efficient way to evaluate potential adaptive renderings for quality.

To see how tradeoffs might arise, imagine there is some content element c that has the highest content score, say 5. The fact that c has a high score motivates us to give it more screen space. To see this, imagine c' is a similar item that carries a lower content score, say 3. For any presentation choice, the total value of c will be higher than that of c' at the same size. Let us assume that c and c' can be displayed in small, medium, or large size, with relative values 1, 2, and 3. Furthermore, suppose that at most two medium but only one large item can fit on the user's screen. In that case, displaying c and c' in medium size has value of 16, which displaying c in large size only has value 15. However, if the content value of c was 6, we would prefer to display c in large size, and not display c'.

Abstract Algorithm

The best rendering is the maximum score over all legal renderings. We can compute the best rendering using the following abstract algorithm:

- Compute a value for each possible configuration. This will be the sum of the values of all elements displayed in the configuration. (Note that there is an additional constraint that the same item may only appear once in a rendering.)
- Return the presentation with the highest score.

Assuming no additional constraints beyond the requirement to fit all items on the screen, and using the current model of presentation scores, this problem can be viewed as a two-dimensional variant of the knapsack problem. This is a difficult problem, and an active area of research in operations research [15]. As we anticipate the existence of additional constraints, we believe that optimization algorithms for this problem will be an important topic of research.

IMPLEMENTATION

We have implemented adaptive interface techniques in the ButterflyNet system, which supports several different types of content, including handwritten notes, images, and whiteboard contents. ButterflyNet takes advantage of digital media and associated metadata to offer a rich interface for visualizing content. The normal method of accessing content in the ButterflyNet desktop application is through a *media browser*, in which a user browses through a logical collection of homogeneous content (*e.g.*, a notebook or photo album). Content elements currently in focus are displayed in the *content panel* on the left, while the *context panel* on the right presents menus, data, or other content related to the items in focus.

We apply adaptive interfaces in two modes of use: as a primary browser, and as a contextual sidebar.

Adaptive Interfaces

The *adaptive sidebar* (see Figure 1) is a contextual element that displays content related to whatever the user is browsing at the moment. As the user browses, the adaptive pivot changes automatically, reflecting content related to the elements in focus in the main browser. Users may double-click a content element in the sidebar to select that element: the media browser is changed to the appropriate media type (if necessary), and the selected element is brought into focus.

The *adaptive browser* (see Figure 5, left) allows users to browse all available content using an adaptive interface as the focal point. Users may pivot about individual elements by selecting them with a single-click; the adaptive browser then shows the content elements most closely related to the selected element. Users may also specify their own pivots by explicitly selecting attribute values such as keywords ("objects tagged with the words 'whiteboard design'"), creation or modification time ("items created on Tuesday, August 29, 2006 around 3:45 p.m."), ownership ("content from my group members"), media type ("all photographs"), or combinations of the above ("Erica's notes and photos from last Friday"). Users may also double-click an element to bring the element into focus in a media browser, switching the adaptive interface into sidebar mode.

Internally, two components drive the adaptive interfaces: an *adaptive interface generator* and a *scoring manager*. The scoring manager takes a content element, reads metadata for the element from the ButterflyNet database, and returns a score relative to the current pivot. The adaptive interface generator takes a scored set of content elements and returns a legal rendering (ordered list of elements and sizes) for the adaptive browser to display.

The ButterflyNet implementation of adaptive interfaces also offers an interface for modifying properties of the adaptive algorithms and renderings (see Figure 5, right). Relative weights of the various metadata facets are user-configurable via a direct manipulation UI. Users may filter based on media type: content may be grouped by media types (so that, for example, notes and pictures are displayed in different sections of the adaptive presentation), or one of more types of content may be hidden altogether.

Design Decisions

To narrow the presentation search space and keep the user interface responsive, we make three simplifying design decisions and assumptions in our implementation of the adaptive algorithm.

First, though content is heterogeneous and may have a number of different aspect ratios, we treat every content element as a *quantum unit* and allocate a *fixed aspect ratio and size*, similar to the PhotoMesa system [1]. This has the advantage of producing nicely aligned grids of elements, with the drawback that significant amounts of space may be wasted for elements that do not align well with the fixed aspect ratio (*e.g.*, portrait-oriented images in a landscape-shaped space).

Second, we only perform *discrete calculations* for layout. Rather than evaluating every possible element size that fits the fixed aspect ratio, we only evaluate sizes that result in an exact integer number of elements across (one across, two across, three across, *etc.*), in effect treating element sizes as discrete, rather than continuous.

Finally, we use a simple algorithm for showing relevance: a *row-major ordering* (left-to-right, top-to-bottom) where the most important items are at the top left. Other possibilities for displaying importance could include combinations of position, size (making important items larger), color (highlighting the closest matches), and other visual properties.

RESULTS

We have assessed the technical efficacy of the techniques in two fashions: by measuring adaptive interface generation times in ButterflyNet, and by exploring how the handles we have provided enable the specification of different types of results.

Interface Generation

We tested our algorithms on a Pentium D 3.2 GHz running Windows XP with 2 GB of RAM. ButterflyNet and the adaptive browsers were implemented in Java and compiled using the Java SE 6 Beta 2 runtime. For the qualitative evaluations below, we used actual data sets from users of

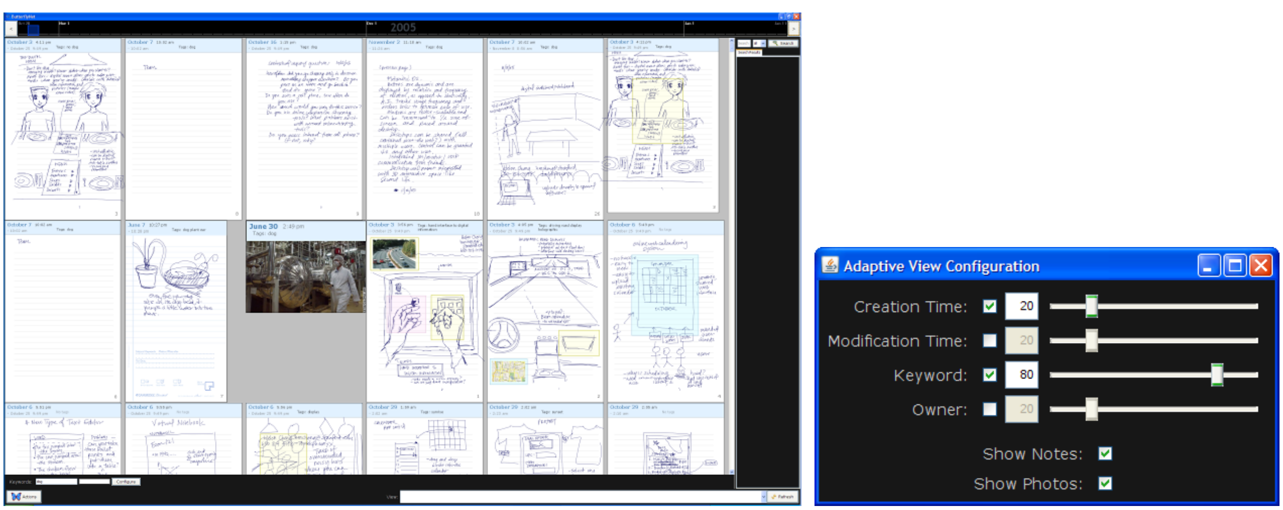

Figure 5. *Left:* An adaptive browser displaying heterogeneous content. *Right:* Direct manipulation interface for changing relative weights of metadata. Users can directly affect how the pivot scores content elements by favoring one metadata attribute or another.

the ButterflyNet system.

Database Access Time

ButterflyNet uses a database to maintain metadata, from which we draw inputs for our algorithm. On average, it took approximately four minutes to read 4,000 content elements (predominantly complex notes files with dozens of strokes) into an empty embedded database, or approximately 60 milliseconds per content element. This is a one-time cost, however, and is not incurred every time the rendering is requested, or even every time the program is run, only whenever a content element is created or modified.

Scoring and Layout Time

For a dataset of 270 content elements (notes and images from a group of three students over one quarter), the running time of the scoring algorithm was 1.21 milliseconds; the layout algorithm, 0.073 milliseconds.

For a dataset of 4,200 content elements (notes and images from a class of approximately 40 students over one quarter), the running time of the scoring algorithms was 12.3 milliseconds; the layout algorithm, 0.423 milliseconds.

We expected that scoring time would vary linearly as a function of the number of content elements, which is what we found. Similarly, we expected layout time to vary linearly as a function of screen space but be bounded by the number of content elements; the latter constraint was evidenced in our test cases.

Element Rendering Time

The above running times do not count the rendering time for content elements, which generally ranged in the hundreds of milliseconds for the most complex sets of objects in ButterflyNet. While loading and displaying several complex objects takes time, this is not a part of the adaptive algorithm.

In practice, overall time to display an adaptive presentation was dominated by the rendering time of the content elements, suggesting that the implementation efficacy problem in adaptive browsers is still predominantly one of element rendering and not of calculating layouts efficiently. This result demonstrates that we can effectively produce adaptive presentations in user-interactive timeframes.

Exploring the Content Space

An important part of our adaptive browsing approach is the inclusion of direct-manipulation handles that enable users to retrieve different types of information depending on the desired goal. Broadly speaking, the ability to change the relative weights of the five metadata types included in the current library produced results that we observed to be relevant and informative. For example, as one author browsed his notebook in the media browser, we observed the presented notes and pictures, which came from both his personal collection and those of other users chosen by the adaptive sidebar. Changing the relevant facet weights had intuitive effects on the rendering produced:

Time correlation: An increase in the weight of timestamps produced a collection of notes and photos from the same event (*e.g.*, a lecture or field outing).

Content correlation: An increase in the weight of keyword matching (relative to timestamps) returned a collection of notes and photos related to the subject of the focused items (*e.g.*, items labeled with the name of a company project).

Awareness: A decrease in the weight of content belonging to the user exposed the user to more content belonging to other users, thereby raising awareness of others' activities. Conversely, an increase in the weight of content belonging to the user's project, or just the user's own content, narrowed the scope of the awareness "feed" to more familiar documents.

SCENARIOS

Drawing on results from longitudinal studies of design education and practice [17] and observations of use of our implementation, we have constructed three scenarios that reflect envisioned uses for adaptive interfaces in designers' work practice.

In the following scenarios, Ada, Erica, Justin, and Leland are designing a new interactive web site for players of a popular online fantasy video game. The website will enable visitors to read the latest game tips, post messages in online forums, and learn about upcoming events. Young, ambitious, and technologically savvy, the four designers use a company tool for supporting design (ButterflyNet) to organize, retrieve, and share their project-related information, including handwritten notes, freehand sketches, whiteboard captures, photographs, diagrams, and text documents.

Enhancing Group Design Practice

Erica, Justin, and Leland head down to the studio meeting room for their weekly brainstorm. Upon entering the studio meeting room, they notice that the digital whiteboard is in screensaver mode, displaying a series of pictures and notes related to the upcoming meeting. The screensaver is actually an adaptive screensaver; based on knowledge of users' calendars, the screensaver has inferred the purpose of the meeting and is proactively cycling through salient content from the previous week's meeting, plus a sprinkling of related material (based on keyword and category attributes) from other designers and teams at the company. Leland walks up to the whiteboard and selects some of last week's notes for review; the three of them usually walk through the previous week's notes together to establish context for this week's meeting. While at the board, Leland notices an interesting whiteboard discussion from the game interaction design team on spell documentation, a hot topic of debate in his group as well. He moves the capture to the foreground so that it fills the whiteboard, and sits down to begin the meeting with the other group's notes on the board as a starter for discussion.

Finding the Rationale behind a Decision

Ada missed the design session; she was consulting on another project at her company, dealing with user forums. When she returns to her office, she opens her design browser and requests content from Wednesday at noon, the time of her group's weekly design session. As she scans whiteboard captures and notes from the meeting, she notices that the group decided to remove certain privacy options from users' online profiles. Curious, she does some searches on keywords she finds on items related to the new topic of discussion ("privacy," "opt-out"), looking for the rationale behind the decision to make this alteration. After browsing for a bit, she comprehends the reason for the change but disagrees with it, and prints out a few salient notes for debate material at the next staff meeting.

Writing a Project Summary

Erica is writing a summary of the work that their group has done on the web site project over the past year. She begins by opening her design browser and perusing her own design notebooks. As she browses her notebooks, related material comes up in the sidebar, including other team members' notebook pages, whiteboard captures from group design sessions, and text documents and emails generated by the team. The contextual aspect of the adaptive interface allows her to browse more flexibly: rather than having to seek out individual documents with explicit searches, she browses paths of "relatedness," reviewing associated material, bringing context elements into focus, looking for important pieces of information in their collective design repository. Eventually, she flags ten documents for closer inspection.

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.*, [22, 24]) 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 [28] provides a retrospective overview of this field. After the initial string of successes that Szekely identified, this field slowed down in the early 1990s, primarily because the desktop PC did not provide sufficient diversity to mandate a higher-level representation: the value of abstraction is derived from the lower margin costs of repurposing—with one platform, there was no amortization to be had.

As ubiquitous computing has edged towards reality, the playing field has changed. We now have Weiser's "computing by the inch, foot, and yard" [31], and model-based interfaces offer significant promise in managing the diversity of computing platforms. An example of this success is Pierce's work on divisible user interfaces [10], which provides a unified representation for applications whose interface is partitioned across multiple devices. This re-emergence of model-based abstractions comes very much from the same spirit as the current paper. The place that this current work fits into this larger picture is that it introduces decision-theoretic techniques for specifying the display portion of these applications.

Automatic Layout

Several projects have explored the automatic layout of interfaces and/or information. The most closely related system in the literature is SUPPLE [8], which examined a constraint-based optimization approach to interface adaptation. Another system, the Personal Universal Controller [20], performed automatic layout of complex service interfaces on different devices using a different theoretical model. We apply a decision-theoretic strategy similar to that of SUPPLE to the area of information presentation, but with significantly different constraints. Rather than addressing user widgets, we deal with information sources. This work has the additional burden of needing to render layouts in user-interactive timeframes (< 100 milliseconds) in order to keep interactions fluid, potentially introducing interesting tradeoffs between optimality and performance.

Image browsing research has proposed many novel methods of dealing with the problem of laying out large sets of data. 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 [27] are another innovation that could be applied to later versions of our adaptive browser, posing interesting questions regarding content presentation value functions. Our adaptive interface research extends this body of work by applying novel techniques in the context of large heterogeneous data sets. In general, image browsers deal with a homogenous set of data: pictures.

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 [13], and the effective presentation of personal information has been the subject of considerable activity. Furnas's fisheye calendar [7], an early system in this area, introduced the idea of a *focus+context* visualization: the calendar item in focus was displayed largely and with local detail; non-focus items would correspondingly shrink. More generally, through this example, it 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 [4], and facet-based approaches to search [6]. In this work, we make no particular ideological commitment to maintaining the navigation affordances of prior technologies, though certainly the existing "user base" of paper books would make a compelling case for doing so. Our approach is more similar to that of faceted search, with the exception that the displays elements are not constrained to be *only* those requested—elements with similarities to those requested may also 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 [30, 32]. Prior work has attempted to facilitate serendipitous generation of ideas by peripherally displaying notes [9]. Our research follows up on this work by applying adaptive techniques to ambient and contextual displays. In particular, we are exploring the peripheral presentation of notes and other epistemic artifacts to encourage exploration and increase visibility of work practice.

Document Scoring

With software architecture and information presentation addressed, we now turn to the question of the underlying algorithms and information model. Similar to prior work on information foraging [23], we seek to improve the information scent of interfaces. Or, 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 [29] points to the value of providing scent via contextual information.

As the quantity of information we work with increases [16], and metadata becomes ever more prevalent [2], improved techniques for sorting this information are required. Adaptive user interfaces have proven particularly useful in managing our personal information. Rhodes' Remembrance Agent demonstrated the use of richer types of metadata—most notably location—as a means for retrieving information [25]. Perhaps most similar to this project is Horvitz *et al.*'s email ranking system [26], 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 [12] 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 [33], the conceptually distinct dimensions of the metadata. Of particular value has been the recent research on lightweight techniques for labeling photographs with rich metadata [5, 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 AND FUTURE WORK

This work offers four contributions: a precise definition of the information presentation problem we address, the various dimensions we use to analyze it, algorithms for calculating an appropriate rendering, and a technical evaluation via a manifestation of this adaptive technique in the ButterflyNet browser.

The contribution of this paper is largely an existence proof of the tractability of the approach. We are currently in the process of planning a study of the benefits of these adaptive display techniques with design teams as the population.

ACKNOWLEDGEMENTS

We thank Ron Yeh 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 are also grateful to Intel for technology donations.

All human subjects research was conducted under Stanford University IRB approved protocol 3392.

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 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.
5 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.
6 Dumais, S., E. Cutrell, J. 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.
7 Furnas, G. W. Generalized fisheye views. *CHI 1986: ACM Conference on Human Factors in Computing Systems*. pp. 16–23.
8 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.
9 Hsieh, G., K. Wood, and A. Sellen. Peripheral display of digital handwritten notes. *CHI 2006: ACM Conference on Human Factors in Computing Systems*. pp. 285–88.
10 Hutchings, H. M. and J. S. Pierce. Understanding the whethers, hows, and whys of divisible interfaces. *AVI 2006: Proceedings of the Working Conference on Advanced Visual Interfaces*. pp. 274–77.

11 Ju, W., B. Lee, and S. R. Klemmer. Range: Exploring Implicit Interaction through Electronic Whiteboard Design. *In submission*.
12 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.
13 Kirsh, D. The intelligent use of space. *Artificial Intelligence* **73**(1-2). pp. 31–68, 1995.
14 Kirsh, D. and P. Maglio. On distinguishing epistemic from pragmatic action. *Cognitive Science* **18**. pp. 513-49, 1994.
15 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.
16 Lyman, P. and H. R. Varian, *How Much Information?* 2003. http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/
17 Maldonado, H., B. Lee, and S. R. Klemmer. Technology for Design Education: A Case Study. *CHI 2006: ACM Conference on Human Factors in Computing Systems (Extended Abstract)*.
18 Matthews, T. Designing and Evaluating Glanceable Peripheral Displays. *DIS 2006: ACM Conference on Designing Interactive Systems (Doctoral Consortium)*.
19 Naaman, M., S. Harada, Q. Y. 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 Nichols, J., B. A. Myers, M. Higgins, J. Hughes, T. K. Harris, R. Rosenfeld, and M. Pignol. Generating remote control interfaces for complex appliances. *UIST 2002: ACM Symposium on User Interface Software and Technology*. pp. 161–70.
21 Norcio, A. F. and J. Stanley. Adaptive human-computer interfaces: a literature survey and perspective. *IEEE Transactions on Systems, Man and Cybernetics* **19**(2). pp. 399–408, 1989.
22 Paterno, F. Model-based design of interactive applications. *intelligence* **11**(4). pp. 26–38, 2000.
23 Pirolli, P. and S. K. Card. Information foraging. *Psychological Review* **106**(4). pp. 643–75, 1999.
24 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.
25 Rhodes, B. J. The wearable remembrance agent: A system for augmented memory. *Personal Technologies* **1**(4). pp. 218–24, 1997.
26 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*.
27 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.
28 Szekely, P. Retrospective and Challenges for Model-Based Interface Development. *DSV 1996: Design, Specification, and Verification of Interactive Systems*. pp. 1–27.
29 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.
30 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.
31 Weiser, M. The Computer for the 21st Century. *Scientific American*. pp. 94-104, 1991.
32 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.
33 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.
34 Yeh, R. B., C. Liao, S. R. Klemmer, F. Guimbretière, B. Lee, B. Kakaradov, J. Stamberger, and A. Paepcke. ButterflyNet: A Mobile Capture and Access System for Field Biology Research. *CHI 2006: ACM Conference on Human Factors in Computing Systems*. pp. 571–80.