

Maintaining Information Awareness with Irwin

D. Scott McCrickard
College of Computing and GVI Center
Georgia Institute of Technology
Atlanta, GA 30332-0280
mccricks@cc.gatech.edu

Introduction

The ongoing explosion of Internet usage has many positive effects for the educational community. Email can connect colleagues separated by thousands of miles. Usenet news can allow professors and students to communicate outside of the classroom. The World Wide Web provides both essential information and much needed distractions. But with these positive effects come new problems.

Although this information is readily available, it is updated at irregular intervals, making it difficult to keep track of changes. For example, a professor may want to monitor a class newsgroup on the days leading up to an assignment due date, or a researcher may want to be updated on the score and status of a ball game on the evening before a paper due date. Another problem is that information can take a significant time to download and display. Consider the time required to download a weather report from the Web, complete with colorful icons and advertisements, at a time when a student is late for a class across campus.

People in academics often want or need to be aware of changes in the types of information in these examples, but they cannot (or should not) spend all of their time reading mail, scanning newsgroups, and surfing the Web. One solution is to use visualization and communication tools to help maintain a desired level of awareness. These tools should include an omnipresent display to allow the user to check on the status of resources with little or no effort, yet the tools must use limited screen space as the user will often want to use other applications that require significant screen space. We try to achieve a balance between these competing requirements with our Irwin information awareness system.

Irwin (Information Resource Watching In a Nutshell) is a set of tools that uses graphical, auditory, and textual communication mechanisms to help the user maintain a desired level of information awareness. This paper outlines ways in which Irwin can assist students and faculty in maintaining awareness, and it describes a field study that examines how four academicians used Irwin to maintain information awareness over a five-month period.

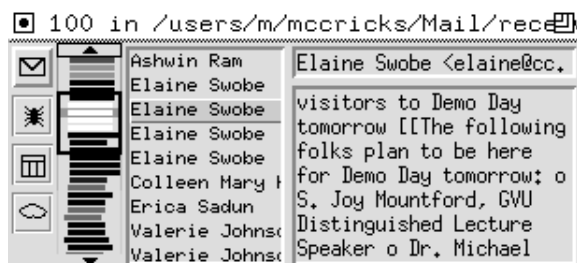
Irwin

Irwin [10] monitors Internet information resources and alerts the user of updates and modifications. Irwin consists of a set of *hypertools*, small reusable programs that can run simultaneously and share information. The central tool in Irwin handles the visualization and user interactions, while the remaining tools process the information from each resource and update the visualization tool when important changes occur.

The information resources monitored by Irwin can include email folders, Usenet newsgroups, Web pages, and weather data. The email and newsgroup hypertools monitor the messages in a folder or

group, alerting the user when new messages arrive. The Web tool summarizes headers, lists, and hypertext links on a Web page, allowing the user to monitor news wires and hotlists. The weather tool monitors the weather conditions and forecast for a given city. When users initially start Irwin, it tries to determine their informational interests by examining their email files and Web bookmarks. Since users are not required to answer a list of questions at startup, they are often more likely to use Irwin. The user can then configure both the information monitored and the way in which it is displayed after getting used to the power and flexibility of Irwin.

To show both an overview and details simultaneously, Irwin uses a *confluent zoom* display. A confluent zoom provides multiple views of a resource simultaneously with each view having a different level of granularity. With a confluent zoom, the user can see overview and details with minimal effort, and the context of the detailed views is maintained at all times by the coarser views. Since intermediate views have proven useful in graphical tools such as map browsers and medical imagery [11, 12], we expect that the intermediate views will smooth the transition from the high-level overview to the detailed zoom view.



In its confluent zoom, Irwin employs an auditory cue, an icons view, a navigation bar, and several textual views. In the icons view (the leftmost view), the state of each resource is given by an icon. When the resource changes, the appearance of the icon changes and an auditory cue is played. If user selects an icon, the other views are updated to show information about the corresponding resource. A navigation bar (second from the left) shows a syntactic

encoding of the resource contents and can be used to select the messages displayed in the textual views (the two rightmost views). Users can configure the orientation and placement of the icons with respect to the other views and can even choose to hide the other views until some event happens, e.g. an icon is clicked.

Auditory cues

Auditory cues provide a means to attract users to a change that has occurred. They may be the only way the user knows about the change if the display is obscured or not in the user's current line of sight.

We selected sounds that are distinctive but are not distracting and, most importantly, are short. Each is less than two seconds in duration, which we expect will be long enough to be noticed but not too long to become an annoyance. In our initial configuration, we tried to select sounds that were indicative of the resource; for example, a dog bark for mail (dogs always bark at the mail carrier) and a splat for news (similar to a newspaper hitting a front porch). Of course, the user can select from a list of sounds (we provide twenty) or create and use their own sounds.

Icons

Icons are used in interfaces because they provide a universal representation in a small amount of space. In a WIMP environment they can both provide information and invite the user to click to obtain more information.



Irwin uses a 16-by-16 pixel bitmap in its icons to show the status of information resources and to provide a gateway to more detailed views. The color of the icons change to reflect the recency of updates to the resource: the resource's icon is originally black, but when a resource is updated its icon changes to red. Over time, if the resource is not updated its icon color fades. It never fades entirely to black until the user clicks on it.

In addition, the bitmap itself changes based on the state of the resource. The weather icon changes its appearance to a sun, a cloud, or rain to reflect the current weather conditions, and the spider icon changes to a dead spider when its Web site is not accessible. The image at the right shows (top to bottom) the mail, sun, rain, live Web, and dead Web bitmaps.

Navigation bar

Irwin needs a method for visualizing and navigating a list of items, such as an email address list or a list of news headers from a Web page. While a scrollbar is a widely-used navigation tool, it shows little information about the content of the list. Irwin's navigation bar leverages the familiar scrollbar metaphor by providing the same basic look and functionality as a scrollbar (arrows, a trough, a thumb), but the trough space of a navigation bar is used not only for navigating the list but also for providing an overview of the contents of the list. Each item in the list is represented by a line in the trough. The thumb (the black box surrounding several graphical lines) indicates the elements that are shown in the textual listbox. To encode information about the list, we used two classes of encodings: syntactic and semantic.

A *syntactic encoding* is based only on the structure of the word being encoded. Advantages of syntactic encodings are that they require no user input and they are consistent between sessions and between users. In Irwin's syntactic encoding, a word or phrase is represented with a series of blocks of pixels, where each pixel block represents a character in the word. To differentiate between words, Irwin colors the blocks that correspond to vowels such that 'a' is red, 'e' is orange, 'i' is yellow, 'o' is green, and 'u' is blue. For example, the image at the left shows a portion of a navigation bar representing several news postings by Annie Anton. Her login (anton) is represented by a red block, two black blocks, a green block, and a black block.

We expect the syntactic encoding to be most useful for mailing lists and newsgroups where a small number of people are responsible for a large amount of the activity. In the previous example, Annie Anton regularly posts computer jobs and internships to a local news group - by using Irwin, her postings can be identified with a glance. Also, the syntactic encoding works well for course newsgroups where the students post questions and concerns and the professor posts responses and clarifications. Since the encoding for the professor would appear quite frequently, it should be easy to recognize.

A *semantic encoding* uses information about the meaning of each list item to create the encoding. In Irwin, clusters of related items are created, and the pixel blocks for items in a cluster are identically colored. The clusters can be created automatically using the Lance-Williams dissimilarity update formula [6], or the user can specify the clusters using keyword matching. The Irwin screenshot earlier in this paper showed a navigation bar using semantic encoding, where email messages are colored based on the sender group (faculty, staff, students, other).

We expect the semantic encoding will be most useful for resources with large amounts of information on a number of different topics. For example, news articles taken from a Web page can be clustered based

on the topics of the articles, with the most frequently occurring topics being colored. Initially, the topics can be created automatically, then later the user may want to specify topics of interest.

We also encode the time at which the message was stamped by indenting the message according to the hour; for example, messages received at 3 PM will be indented three blocks. This is intended to group messages by arrival time, thus facilitating searches and browsing. If users know the time at which a message arrived, they can identify the range of messages in which it must fall using the navigation bar.

Textual views

Graphics alone might not show all the information necessary to understand a message - at some point a textual view will likely help. Irwin incorporates two textual views: a header list and a message view. Irwin constructs lists of headers for each resource, the senders for email and Usenet news, abbreviated headers for Web pages, and days of the week for the weather forecast. If the user selects an item from the list, the message view shows a more informative summary of the original message.

While the Irwin textual views can give the user an up-to-date peek at the contents of the monitored resource, it is not meant to replace the programs that are designed for reading, browsing, and searching the resources. To assist the user in bridging the gap between awareness and other tasks, Irwin allows the user to jump to various readers and browsers to view the resources of interest. Irwin is automatically configured to jump to a Web page using Netscape Navigator if a URL is referenced, or Irwin can be configured by the user to start or activate various email readers, news readers, Web browsers, or other programs and commands.

User reaction

While studies like our navigation bar experiment are helpful in analyzing performance for certain specific searching and identification tasks, the true test of tools, particularly for a difficult to quantify task like awareness, is the reaction of the public. By offering Irwin to users and observing how they use it, we hope to answer some of these questions.

Nine faculty members and graduate students attended a presentation and demonstration explaining how to set up and use Irwin. Seven people tried Irwin at least once, and four of those continued to use it regularly. The remainder of this section summarizes their comments. (A full report of their comments is available as well [8].)

We asked those who did not use Irwin why they chose not to. Most simply did not feel like they needed such a tool. One person noted that she didn't receive much email and only rarely browsed the Web - she simply had no use for tools like Irwin. One user wanted a less passive visualization tool. He reported that he felt he did not have enough control over when the resources were checked. He seemed more interested in a typical browsing or reading tool.

All of the frequent users configured Irwin to better meet their needs, though most used it for several days first. Common resources to monitor were email, various news and sports Web sites, high traffic Usenet news groups, and the local weather.

Reaction was mixed on the use of audio. All but one user turned the audio alerts off or only used a simple beep, and the one user who did use it only used short sounds (under a second) even though Irwin

provides sounds up to two seconds long. Reactions varied from "I didn't need it" to "It would drive me up the wall." While we feel that audio is not for everyone, if it is kept short and distinctive it can prove to serve its purpose for some people.

Most users did not use the navigation bar very extensively. They typically looked at Irwin only to see the most recent additions, i.e. those visible in the text boxes. One user noted that he could recognize a few patterns and occasionally would notice blocks of identical encodings together (indicating several emails or news postings by the same person) but would rarely click on them and browse them. The problem may be that the navigation bar is better suited for browsing and searching tasks than for awareness, or it may be that the users did not choose resources for which the navigation bar is best suited.

After using Irwin for several months, the users had many suggestions for resources that they wanted to monitor but could not with Irwin. The resources include Unix commands like `lpq` and `finger` and Web images like traffic and weather maps. The users wanted more flexibility over the way in which information is presented - not all resources fit in the list-based concept of the navigation bar. As we extend and improve Irwin, we are exploring new identification schemes and new communication mechanisms to allow the user to specify items of interest and to have the information delivered in the most appropriate manner.

Related Work

A significant amount of research has focused on the filtering and dissemination of information [5, 6]. However, far less work has been done in understanding how people need the information to be communicated. Many systems generate email or Web pages that still require the user to remember to check them. Other solutions have been developed for specific resources, but there is not always a direct application to a broader range of resources. We hope that the Irwin monitoring system will address these concerns by providing a configurable set of tools for monitoring interesting resources and staying aware of changes that occur to them.

The individual components of Irwin were influenced by prior research. Auditory cues have proven useful in both enhancing and replacing visual cues in user interfaces [3, 2]. Studies have shown that changing the appearance of icons can convey additional information about their use; for example, animated icons [1] convey more information about the functionality of tools in a tool palette than a static image. The navigation bar concept of encoding lines of data using graphical lines came from previous systems. Read wear and edit wear [7] represents lines of text with lines of pixels, where characteristics like length and color correspond to properties of the text such as the number of modifications. The SeeSoft [4] and RunView [9] software visualization programs represent code with lines of pixels where the length of the line can correspond to the length of the line or function, and the color can correspond to the author, module name, or last modification date. To our knowledge, our navigation bar was the first to use syntactic encodings, the first to fully leverage the scrollbar metaphor, and the first to empirically demonstrate that this type of encoding improves user performance in search and recognition tasks.

While many parts of the Irwin system have been researched individually, we feel that our true contribution comes in the integration and evaluation. We hope that the lessons we have learned have built on the work of previous researchers to provide another step towards understanding information awareness.

Conclusions and Future Work

Based on our experience with Irwin, we would like to discuss a few of the lessons we learned which should help in the future development of information awareness tools.

Irwin users were willing to sacrifice a little bit of space to increase their awareness of information resources. Even those who did not use Irwin cited reasons other than space limitations. While we do not expect that everyone will need awareness tools, it appears that there is a significant population that does.

When using Irwin, most users utilized some combination of feedback from auditory cues and icon changes to know when to check a resource. Though we provided auditory cues that were up to two seconds in duration, all of the sounds configured by our users were under one second, and most used only a simple beep. Longer sounds seem to be too distracting, and a simple visual change may suffice to maintain awareness. The navigation bar seems to be better suited for searching and browsing tasks than for general awareness, though users began to recognize a few patterns over time, and for certain situations (such as monitoring a course newsgroup or a listing of jobs) the navigation bar may yet prove to be essential.

Too often visualizations will provide complex graphical views of a data space but will make it difficult to see the actual data that is represented. Ideally, a visualization will provide a view that combines graphics and text dependent on the available space, the needs of the user, and the amount of information that needs to be shown. The confluent zoom allows the user to stay aware of the status of several resources but also to view details of a selected resource. By providing both focus and context in a compact manner, users can be free to focus more of their attention on other tasks.

References

- 1 Ronald M. Baecker, Ian Small, and Richard Mander. Bringing icons to life. In *Proceedings of ACM SIGCHI '91*, May 1991.
- 2 Steven A. Brewster. *Providing a structured method for integrating non-speech audio into human-computer interfaces*. PhD thesis, University of York, UK, 1994.
- 3 Megan L. Brown, Sandra L. Newsome, and Ephraim P. Glinert. An experiment into the use of auditory cues to reduce visual workload. In *Proceedings of the ACM SIGCHI '89*, May 1989.
- 4 Stephen G. Eick, Joseph L. Steffen, and Eric E. Sumner Jr. SeeSoft--a tool for visualizing line oriented software statistics. *IEEE Transactions on Software Engineering*, 18(11):957-968, November 1992.
- 5 Peter W. Foltz and Susan T. Dumais. Personalized information delivery: An analysis of information filtering methods. *Communications of the ACM*, 35(12):51-60, December 1992.
- 6 William B. Frakes and Ricardo Baeza-Yates. *Information Retrieval: Data Structures and Algorithms*. PTR Prentice Hall, Englewood Cliffs, NJ, 1992.
- 7 William C. Hill and James D. Hollan. History-enriched digital objects: Prototypes and policy issues. *The Information Society*, 10(2), April 1994.

- 8 D. Scott McCrickard. Information awareness on the desktop: A case study. Technical Report 95-24, Georgia Tech GVU Center, Atlanta, GA, 1997.
- 9 D. Scott McCrickard and Gregory D. Abowd. Assessing the impact of changes at the architectural level: A case study on graphical debuggers. In *Proceedings of ICSM*, November 1996.
- 10 D. Scott McCrickard and Thomas H. Rowan. Monitoring and visualizing information resources. Technical Report ORNL/TM-13193, Oak Ridge National Laboratory, July 1996.
- 11 C. Plaisant, D. Carr, and H. Hasegawa. When an intermediate view matters: A 2d-browser experiment. Technical Report CS-TR-2980, U. of Maryland, October 1992.
- 12 C. Plaisant, D. Carr, and B. Shneiderman. Image-browser taxonomy and guidelines for designers. *IEEE Software*, March 1995.