HAIL: Hierarchical Adaptive Interface Layout

John Magee and Margrit Betke

Computer Science Department, Boston University
111 Cummington St, Boston, MA 02215 USA
{mageejo,betke}@cs.bu.edu

Abstract. We present a framework to adapt software to the needs of individuals with severe motion disabilities who use mouse substitution interfaces. Typically, users are required to adapt to the interfaces that they wish to use. We propose interfaces that change and adapt to the user and their individual abilities. The Hierarchical Adaptive Interface Layout (HAIL) model is a set of specifications for the design of user interface applications that adapt to the user. In HAIL applications, all of the interactive components take place on configurable toolbars along the edge of the screen. We show two HAIL-based applications: a general purpose web browser and a Twitter client.

Key words: Adaptive User Interfaces, Video-Based Interfaces, Camera Mouse, Web Browsers, Social Networking

1 Introduction

Severe paralysis can have several causes. Degenerative neurological diseases such as Amyotrophic Lateral Sclerosis (ALS) and Multiple Sclerosis (MS) can result in progressive impairments. Brain and spinal cord injuries due to cerebral palsy, brain-stem stroke, or trauma can cause paralysis and impairments that remain static or may improve with therapy over time. It is important that the assistive technology that individuals with these movement impairments use can adapt to changes of the individuals' conditions. We present a framework for adapting software applications that individuals use with mouse substitution interfaces. We tested our work with the Camera Mouse[1], a video-based interface that detects and converts head movements into mouse pointer movements.

The Camera Mouse has emerged as a popular assistive technology used by individuals with severe motion impairments. Connor et al. [1] conducted experiments with several individuals using the Camera Mouse. We here address some of

---

[1] Available as a free download at `http://www.cameramouse.org`

the observations from these experiments: individuals with disabilities have widely ranging abilities, and their abilities may change over time. We propose interfaces that change and adapt with the users and their individual abilities. Our target audience generally has difficulty with precise control of a mouse pointer. This makes tasks such as clicking on small buttons or web links difficult. With the Camera Mouse, a click command is issued when the mouse pointer has dwelled over the button or link for a certain amount of time.

A great amount of effort has been devoted to making the web accessible to blind or vision-impaired computer users. Individuals with motion-impairments generally have few options. Previous efforts to create accessible web browsers involved zoom-able interfaces, designing larger buttons on standard web browsers, adding confirmation boxes before following a link (e.g., [2]), dynamically changing the layout [3], or growing the size of links. Other work on adaptive interfaces present interfaces that are too complex for some of our users (e.g., [4]).

We propose the Hierarchical Adaptive Interface Layout (HAIL) model, which is a set of specifications for the design of user interface applications that adapt to the user, rather than requiring the user to adapt to the interface. We present two applications designed to the HAIL model specification: a general purpose web browser and a Twitter client. These were chosen to compare the number of user inputs needed to complete certain tasks. For the purposes of this paper, the applications are not fully featured, rather they were made to serve as a proof of concept for HAIL interface applications. After conducting experiments and receiving user feedback, we plan to expand the applications by adding features such as text input via on-screen keyboards or bookmarking.

Although we designed these interfaces for users of the Camera Mouse, users of other mouse substitution devices, such as mouth joysticks, may also benefit from applications that are based on the HAIL model.

2   Hierarchical Adaptive Interface Layout Model

The hierarchical concept of the HAIL model means that each increasing level of the interface is slightly more complex than the previous level. There are two important specifications for HAIL applications. First, all of the interactive components take place on toolbars along the edge of the screen. The center of the screen is by default reserved for display only and can be used as a rest area for the mouse pointer. This helps avoid the "Midas touch" problem of unintentionally selecting every item the mouse happens to stop near, and gives users a large area in the center where they do not have to worry about the mouse position. Second, each level of the HAIL model specifies minimum button dimensions as a percentage of screen size (Table 1). This assures that a user comfortable with a certain level in one application would be able to use other applications with the same setting.

We built upon experiences in designing user interface programs that offer configurable complexity and difficulty. In particular, Magee et al. [5] designed the game BlockEscape to evaluate user interactions by allowing settings such

Table 1. Minimum button size as a percentage of screen width. Intermediate levels can be used to add or subtract buttons.

HAIL Level	Button Size as a Percentage of Screen Width
1	14%
2–4	12%
5–7	8%
8–10	4%

as changing the speed of the game, or varying the number of objects that must be navigated around. Adjustment of the settings resulted in a game that was easier or harder for the user to interact with. Following this approach, the HAIL model offers a range of interface layouts from easy-to-use and uncomplicated at one end, to having more features and being more complex at the other end. The intent is that individuals using mouse substitution interfaces should be able to set their user interface at the level of complexity that is right for them. Users who are adept at clicking on small buttons will be able to use the more complex interface with more buttons and thus be able to accomplish tasks faster. Users who have difficulties selecting small buttons will be able to choose interfaces with increasingly large buttons (and as a result, fewer buttons on the screen). Although tasks may take longer with the simpler interface, they can now be accomplished without requiring the mouse-pointer accuracy of the more complex interface. HAIL model levels are shown graphically on a sliding scale in Fig. 1.
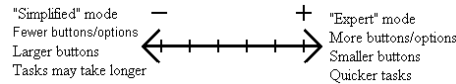

Fig. 1. Graphical depiction of HAIL levels. Low levels on the left, high levels on the right, intermediate steps in the middle.

Traditional pointer-based interfaces allow movement of the pointer in any direction, allowing the user to select any interface element on the screen directly. HAIL interface applications instead highlight the currently selected interface element. Buttons on the toolbars navigate between those interface elements, and provide a mechanism for activating the currently selected element or executing an event on the selected element. Navigation between interface elements is more akin to keyboard arrow keys – Up, Down, Left, Right – than a mouse pointer. Due to this, HAIL applications work best with information and interfaces that can be organized into linear lists or grids.

3   The HAILbrowser

The HAILbrowser is an implementation of a web browser following the HAIL model specifications. Since the HAIL model is a generalization, this implementation is an application to test the model presented. There are several good reasons for choosing a web browser as a testing platform for the HAIL model. The web browser could be called the most important piece of software for today's computer users. It has become the ubiquitous application not only for accessing information, but also as a gateway for communication applications like email and social networking.

Specialized web browsers like the Zac Browser (Zone for Autistic Children) [6] have made the web an accessible tool for people with autism. The goal of the HAILbrowser is to open up the web to users of accessibility technology like the Camera Mouse. We have previously used standard web browsers with the Camera Mouse and found that only the most adept Camera Mouse users can successfully navigate the world wide web with these browsers. The standard web browser assumes that users have pinpoint accuracy with their mouse pointer to select links. This is often an unrealistic assumption for users with movement impairments. Even Camera Mouse users without disabilities may have trouble keeping the mouse pointer hovering over small hyperlinks in order to select them. The HAILbrowser offers a different mechanism to select hyperlinks. It displays toolbars to allow individuals to navigate through links on a page and select one of the links to follow (Fig. 2). At the high end of the layout hierarchy, the HAIL framework yields a browser with smaller toolbars and more buttons; at the low end of the hierarchy, the HAIL framework yields a browser with larger toolbars and fewer buttons.
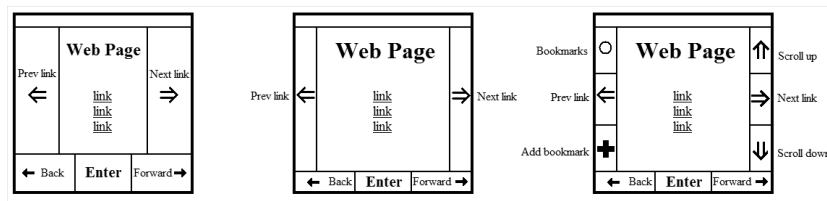
Fig. 2. Layout of three layers of the HAILbrowser.

The web pages can have links and interactive elements arranged arbitrarily, which can make the task of navigating to a specific link difficult and time consuming for Camera Mouse users. The lower complexity levels of the HAILbrowser therefore display the mobile web version of web pages by default (Fig. 3). These pages are designed for display on small electronic devices like mobile phones that have small screens. The layout of the web pages is thus simplified, less cluttered, and often easier to navigate. These pages often present navigation links in a

linear list format, which is easier to navigate by iteratively selecting subsequent links. The obvious trade-off of this format is that the page may display less information than the non-mobile counterpart. The mobile web default is configurable by the users.
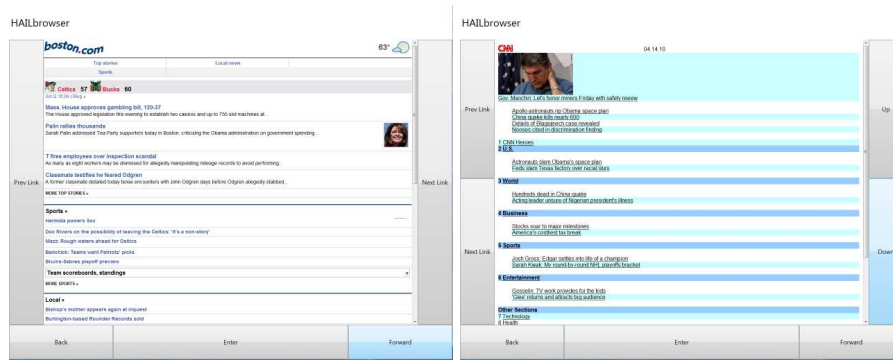

Fig. 3. HAILbrowser screen shots.

At the least-complex level of the HAIL hierarchy, the browser incorporates a set of basic features for browsing the Internet, as described in Table 2. At a more advanced level of the HAIL hierarchy, the browser incorporates a set of additional features that are convenient for browsing such as bookmarking and jumping links (Table 2).

Table 2. HAILbrowser required and optional features.

Basic Features	Advanced Features
Select Next/Previous Link	Bookmarks
Follow Selected Link	Text Entry / Keyboard
Back/Forward	Jump 10 Links
Page Up/Down	Smart navigation (not implemented yet)

4 HAILtwitter

HAILtwitter is a second application designed to be a proof of concept. HAILtwitter uses the application "Twitter." Twitter is a social networking and microblogging service that can be accessed through web browsers, portable electronic devices, or custom client programs that communicate with Twitter servers over

the Internet. As of April 2010, it has over 100 million registered users. Users make accounts and designate a list of other users they wish to receive updates from. Messages posted to the site are limited to 140 characters in length. Users can choose to view messages only from others that they follow, or can search for messages containing a word or phrase, choose from a list of trending topics, or can see all messages from a specific user. The small message size and the information organization in list form makes Twitter an ideal application for users with disabilities. Navigating up and down a linear list of information is an interaction style that is well-suited for a HAIL application.
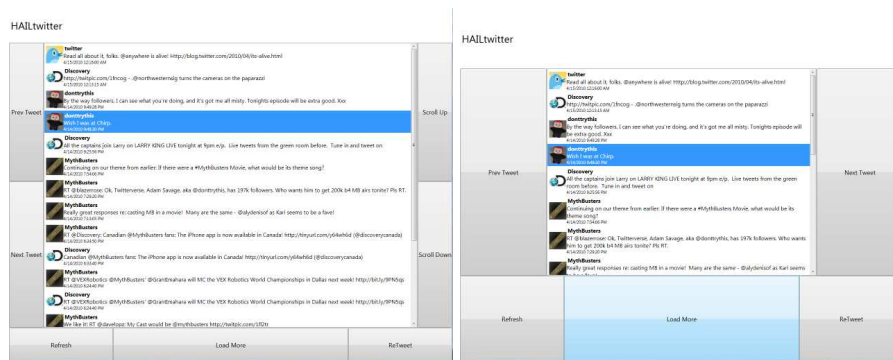

Fig. 4. HAILtwitter screen shots.

Two screen shots of the HAILtwitter application are shown in Fig. 4. Each message occupies a row in the list of messages, along with the sender's profile picture, the sender's name, and the time that the message was posted. The toolbar buttons are used to navigate up and down the list of messages to select one to interact with. Once a message is selected, the user can choose to interact with it in a number of ways. The user can "retweet" the message – essentially rebroadcast it to their friends. Other options are to see all messages from the sender of the current message, or to add or remove the sender from the user's following list. Users can also refresh the page to see newer messages, or load another group of older messages from the current set of messages they are viewing.

We developed the HAILtwitter application with a limited feature set that nonetheless gives the user quick access to social networking and communication. With the addition of an on-screen keyboard, this application can also allow two-way social interaction for some users. Many users use Twitter to post web addresses, links to other Twitter users, or links to pictures. Adding the ability to follow these various kinds of links can provide a richer interactive environment for users (see Table 3).

Table 3. HAILtwitter required and optional features.

Basic Features	Advanced Features
Get more / Refresh	View other timelines
Page Up/Down	Text Entry / Keyboard
Select Next/Previous Msg	Trending topics
Retweet (forward msg)	Follow/Unfollow friends

5   Discussion

Preliminary testing of the applications with able-bodied Camera Mouse users
was positive. Users observed that it was much easier to select the buttons in our
interface compared to standard-sized links in traditional web browsers.

We found that buttons on the edge of the screen were easier to select than
interface elements in the middle of the screen. The reason was that once the
mouse pointer reached the edge of the screen, for example, the bottom, selection
of buttons placed along the bottom mostly involved control of horizontal move-
ments. The control of the vertical pointer movement did not require as precise
pointing abilities from the user, because the user could simply push the mouse
pointer downwards, and this would result in the pointer remaining at the edge of
the screen. A similar situation exists with corners: the user only needs to move
towards the corner and the edges of the screen "catch" the pointer and direct it
onto the corner button.

Our experimentation with buttons at the corner of the screen brought our
attention to a problem that Camera Mouse users sometimes encounter – un-
intended simulated mouse clicks that wreak havoc with the users' interaction
experience. The top-right corner of an application window typically contains a
button with which a user can close the application. Similarly, another corner
may be dedicated to menu buttons. The bottom edge of the computer screen
may contain a task bar. We have observed that Camera Mouse users of tradi-
tional applications sometimes unintentionally hover over these buttons and thus
issue unintended selection commands that close their applications or switch to
other applications. To avoid these scenarios, we therefore designed our HAIL
applications to utilize the full screen.

6   Current and Future Work

We are currently testing the prototype implementations of our HAILbrowser
and HAILtwitter applications. Feedback from the first round of testing will be
incorporated into our software. We will make the software available as a free
download so that Camera Mouse users will be able to try it out and provide
additional feedback.

To become a successful end-user application, the software must include fea-
tures beyond what we implemented initially as a proof of concept. Integration

with an on-screen keyboard or other text-entry methods is one focus. Text-to-speech abilities would also be useful for some users.

The HAILtwitter application would benefit from incorporating all of the features available on the Twitter website, such as trending topics and lists. For the web browser application, the browser could exploit navigation advancements of semantic understanding of web content used by screen reader navigation. Our browser could leverage the work presented by Takagi et al. [7], using collaborative web accessibility information to aid in navigating the content of a web page. Other approaches navigate by groups of links [3] or via a tree structure [8]. In these approaches, the user would navigate between sections of a web page before navigating through individual links.

We are moving the development of our web browser and Twitter applications to a common HAIL framework with a plug-in architecture. Similar to how a smartphone device provides a basic interface for a variety of smaller "apps," the HAIL framework would allow HAIL applications to share the same user interface program. A similar approach is proposed by Andrews et al. [9], where a common user interface framework interprets interface elements defined within metadata. Users will be able to switch between applications and have the framework remember their preferred settings. This will ease the development of future applications that would benefit from the HAIL model. It would also allow links in one application to open in another application. An email client, news-feed aggregator, Facebook interface, and media player application are currently planned.

References

1. C. Connor, E. Yu, J. Magee, E. Cansizoglu, S. Epstein, and M. Betke, "Movement and recovery analysis of a mouse-replacement interface for users with severe disabilities," in *HCI International 2009*, (San Diego, CA), July 2009.
2. H. Larson and J. Gips, "A web browser for people with quadriplegia," in *Universal Access in HCI: Inclusive Design in the Information Society* (C. Stephanidis, ed.), pp. 226–230, Mahwah, NJ: Lawrence Erlbaum Associates, 2003.
3. B. Waber, J. Magee, and M. Betke, "Web mediators for accessible browsing.," in *9th International ERCIM Workshop "User Interfaces For All" UI4ALL*, (Königswinter, Germany), Sept. 2006.
4. K. Z. Gajos, J. O. Wobbrock, and D. S. Weld, "Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces," in *SIGCHI Conf. on Human Factors in Computing Systems (CHI '08)*, pp. 1257–1266, 2008.
5. J. J. Magee, M. Betke, J. Gips, M. R. Scott, and B. N. Waber, "A human-computer interface using symmetry between eyes to detect gaze direction," *IEEE Transactions on Systems Man and Cybernetics: Part A*, vol. 38, pp. 1261–1271, Nov. 2008.
6. "Zac browser Zone for Autistic Children." http://www.zacbrowser.com/.
7. H. Takagi, S. Kawanaka, M. Kobayashi, D. Sato, and C. Asakawa, "Collaborative web accessibility improvement: challenges and possibilities," in *Proceedings of the*

11th International ACM SIGACCESS Conference on Computers and Accessibility (Assets '09), pp. 195–202, 2009.

8. E. Walshe and B. McMullin, "Browsing web based documents through an alternative tree interface: The webtree browser," in *Computers Helping People with Special Needs, ICCHP 2006, LNCS 4061*, pp. 106–113, Springer Verlag, 2006.

9. J. H. Andrews and F. Hussain, "Johar: a framework for developing accessible applications," in *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility (Assets '09)*, pp. 243–244, 2009.