Stocks

A final report on the general design and analysis of peripheral and user notification alert interfaces for a personal financial portfolio tracker.

Ellen Lau
Supervisor: James Tam
Course Instructor: Dr. Brian Wyvill
Final Report for CPSC 503
University of Calgary
December 8, 2003

1. Abstract

The stock system consists of 2 key components: the interface component and the system component. The interface component serves as the front end of the system and this is the portion that the user will see and interact with. The system component is the backend of the system that will read in stock price information from either a comma delimited text file (csv format) or an xml file. The system component is responsible for retrieving and formatting the data so that it can easily be used by the interface. The core of development is in dealing with the front end of the system. User centered design as noted in Soderston and Rauch [7] is used to provide information about the users and their normal habits, while task centered design as noted in Lewis and Rieman [4], is used to provide information on the main (primary) tasks done by the users on a regular basis. Design of the interface is based on the information from the user and task profiles. In addition, low and medium fidelity prototyping techniques are used to create simple and low cost prototypes for quick and easy feedback from the users. Designing, testing and modifying the interface must be done fairly quickly since the interface will go through several iterations of change before a design is chosen for the system. In testing the interfaces, the users are asked to evaluate, and in some cases, interact with the interfaces and give feedback on different aspects of the design and flow of the interaction. That feedback is then used in the design of the next cycle.

2. Introduction

The purpose of this project is to design a stock display system that employs both peripheral awareness techniques as well as immediate or alert notification techniques. The necessity in using both the peripheral interface and the alert notification becomes obvious when, for example, a user is concentrating on a primary task but wants to keep an eye on changing stock prices. In this case, the user does not want to be bothered with every little change in his stocks, and is content to continue to surf the web glancing every now and then briefly at the stock display. The opposite of this is true when a stock hits an all time low. In this case, the user will want to be immediately notified of the plummeting stock price and will want the system to grab his attention right away. Based on those types of scenarios, the stock system contains 2 vital parts: the user interface that is displayed to the user which serves as the front end of the system, and the system that feeds and formats the stock information before presenting to the interface which will serve as the backend of the system.

The interface will depend on three main factors that will affect how they are designed:

I) The display of peripheral stock information (peripheral awareness display)
II) Immediate notification alerts (alert displays).
III) Effective representations of the data and efficient flow of interaction.

The example above explained the importance of using the peripheral awareness display and the alert notification, but equally important is how the information will be represented. It would be pointless for the user to have a very efficient peripheral system running in the background but have no idea what the information means when he glances to look at it. So an integral part in designing both the peripheral interface and the notification alert is having the representation of the data be meaningful yet easy to decipher for the user. Also, when the user needs to interact with the system, the flow of actions should be efficient and logical. That is, there should be a direct causal relationship between what the user does and the response of the system. For example, when the user clicks to 'Exit', the expected result would be a shutdown of the stock system. Also, if an alert were to appear to the user showing a stock hit a 52 – week low, there should be enough information displayed to the user (the stock price, the time of last trade, P/E ratio) for them to make a decision on their next course of action. Logically, if that change were significant enough, it would mean the user would want to either buy or sell the stock. The system interface should offer a direct means that would allow the user to take that action, which might mean a 'Buy' or 'Sell' button available on the interface. All the information that appears when the alert is issued may not need to be shown on the peripheral display. Details such as P/E ratio are only relevant when the user wants to buy or sell a stock; in addition, P/E ratios are rarely changed so placing that piece of information on a constantly updated display would yield very little change in the data presented.

All these concepts must be integrated into the interface design and each design must be tested out on the intended users of the system. To determine the most effective design, a short cycled iterative design approach is used. That is, all the design ideas are tested, feedback is gathered from the test subjects, and based on the feedback the ideas are reworked and then tested again. A number of different test techniques are employed with the 'think aloud' technique being the most frequently used. The 'think aloud' technique, as is used by Soderson and Rauch [4], is one where the users are encouraged to verbalize their thoughts about the design, the interaction flow, and general feel of the interface. Another technique used more in determining preferences in existing systems is the questionnaire method. This is a fast and easy assessment that basically asks the user to rate how much they like certain aspects of an interface on a scale of 1 to 10. The numbers alone however, do not convey enough information and post interviews are needed to find out exactly why the user gave the response they did.

In terms of presenting test content to the users, the ideas are first constructed using very simple or low fidelity methods such as simple paper sketches. As the ideas flourish and mature, a medium fidelity prototype is developed. Such prototypes are simple interface shells that have the main idea of the interface and little of the functionality but it is still simple and quick enough to produce in a very short period of time. Once the interface matures and progresses through many iterations of design and redesign, a high fidelity prototype is constructed that encapsulates many of the functionalities of the systems. For this system, the users were allowed to take the system into their normal work environments and beta test the system. Using real live stock quotes from Yahoo, the users were able to provide feedback through post interviews about the system and its functionalities.

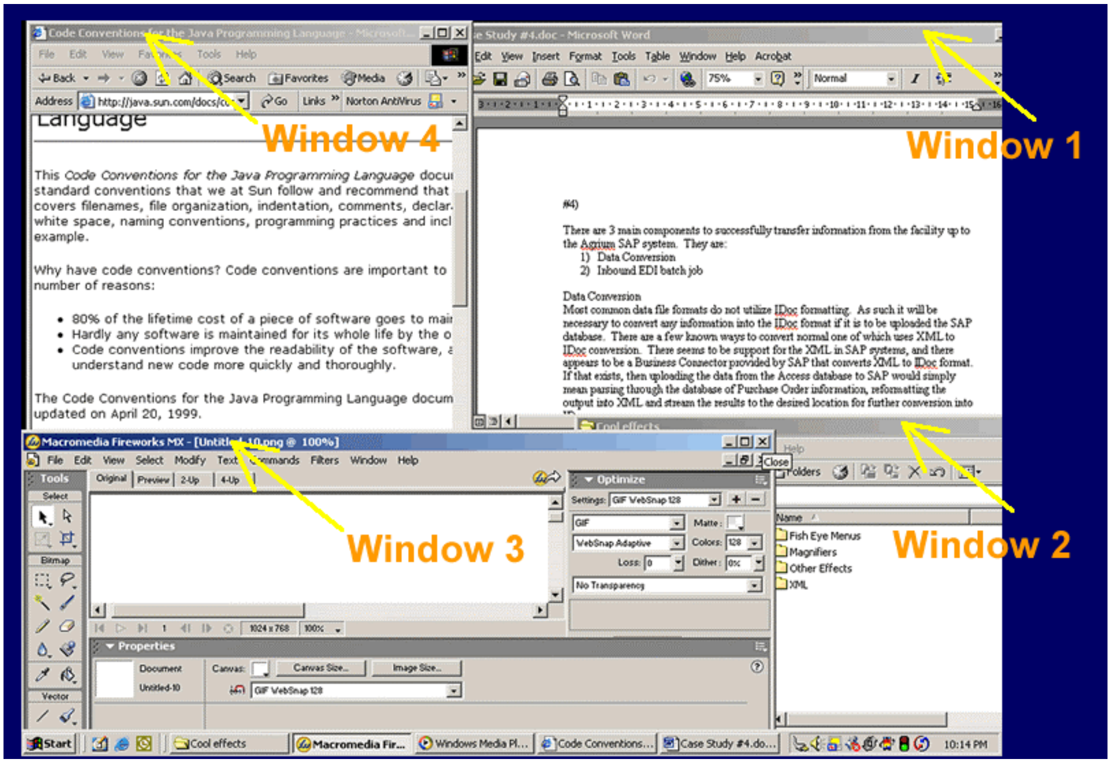
3. Existing Evaluation Methods

There are many different methods to test the usability of an interface. This project employs the observation technique to profile the users of the system. To get feedback from the users, the think-aloud method is used along with questionnaires about look and feel of the interface. In terms of methodologies used for the designing the interfaces, user-centered and task centered design techniques are used. This ensures that the design of the interface is built with the user in mind. Since there is constant feedback needed from the user for each prototype, the use of low and medium fidelity prototypes are used as prototyping techniques to allow for a fast and easy design cycle.

4. Usability Studies

4.1. Observation

The observation technique is simply watching the user interact with an interface. This is an effective technique because it allows the observer to see the user as if they were really using the interface in their own environment. There is no interruption from the observer and all of the users movements are recorded. For this project, the observation technique was used to see what types tasks the users do on their computers. These tasks are considered the user's primary task, which ranges from simple word processing to coding Java applications. Observation was also done on how the user adjusted the layout of the various windows on their screen. This allows some insight into how the user would prefer the layout of their computer desktop. A typical desktop is shown in Figure 1.


Code Conventions for the Java Programming Language - Microsoft...
http://java.sun.com/docs/co
Window 4
Case Study #4.doc - Microsoft Word
Window 1
Macromedia Fireworks MX - [Untitled-10.png @ 100%]
Window 3
Optimize
Properties
Cool effects
Window 2
Fish Eye Menus
Magnifiers
Other Effects
XML
Start
10:14 PM

Language

This *Code Conventions for the Java Programming Language* docu
standard conventions that we at Sun follow and recommend that
covers filenames, file organization, indentation, comments, declar.
white space, naming conventions, programming practices and incl
example.

Why have code conventions? Code conventions are important to
number of reasons:

- 80% of the lifetime cost of a piece of software goes to mai
- Hardly any software is maintained for its whole life by the o
- Code conventions improve the readability of the software, a
  understand new code more quickly and thoroughly.

The Code Conventions for the Java Programming Language docum
updated on April 20, 1999.

#4)

There are 3 main components to successfully transfer information from the facility up to the Agrium SAP system. They are:

1) Data Conversion
2) Inbound EDI batch job

Data Conversion
Most common data file formats do not utilize IDoc formatting. As such it will be necessary to convert any information into the IDoc format if it is to be uploaded the SAP database. There are a few known ways to convert normal one of which uses XML to IDoc conversion. There seems to be support for the XML in SAP systems, and there appears to be a Business Connector provided by SAP that converts XML to IDoc format. If that exists, then uploading the data from the Access database to SAP would simply mean parsing through the database of Purchase Order information, reformatting the output into XML and stream the results to the desired location for further conversion into

Figure 1. Screenshot of a typical user desktop

4.2. Think aloud

The think aloud method is beneficial because it will highlight deficiencies in the interface designs, as the user will vocalize their difficulties when using the interface.

4.3. Questionnaires

While the observation and the think aloud techniques are important, they rely solely on the user providing that information. Sometimes, there needs to be a further investigation as to why the user said what they did. To facilitate that, questionnaires are usually used in conjunction the techniques mentioned above.

4.3.1. Surveys

Surveys are constantly being used to evaluate the prototypes. Similar to the type of questions used by Cadiz, Venolia, and Jancke [1] in their SideShow application, users are presented prototypes and are asked a series of pertaining to the effectiveness of the peripheral display. Users of SideShow were asked to fill out a survey whether they strongly agreed or disagreed with a list of statements about their application. In the surveys done for this project, the same technique was applied to for all of the prototype designs.

A testing method used by McCrickard, Catrambone and Stasko [5] was used to evaluate the effectiveness of different types of information representation. McCrickard, Catrambone and Stasko [5] conducted a series of empirical tests to evaluate animation in their peripheral system. They asked participants to complete a series of browsing tasks on the Internet while still keeping abreast of the information that was being presented to them in the periphery. Questions would then be asked during the task about information presented to them via the peripheral display. For this project, a similar technique was employed whereby different shapes and colors were displayed in the peripheral system and the user was asked questions about them later to evaluate their retention of the information displayed.

4.4. Interview

Lastly, there is the interview method. The interview is usually done after the testing, and usually when the questionnaires are complete. It is used to fill up any gaps that might have been missed. More usually, it is a chance to get the user to talk freely about the interface.

4.5. User Centered Design

User-centered system design is heavily used throughout this project. User-centered designs allow the users to be an integral part of the design process through the use of prototypes as noted by Redish [6]. There is constant feedback from the user and users may even take part in building the prototypes this project however, will not so far as to include users in the build process.

4.6. Task Centered Design

In addition, task centered prototyping, as is used by Lewis and Rieman [4], was used to complete user centered design for this project. An important part in understanding the user is what type of tasks the user normally performs. As noted above, simple observation was done to see first hand the types of tasks the user performs during the day. Once information about the tasks are collected, design could proceed without heavy input from the user until the prototype is finished. This methodology helps the project proceed at times when users may not be available for immediate feedback.

4.7. Prototyping Methods

Prototyping methods are used to realize the ideas into working interfaces. As noted by Soderston and Rauch [7], prototyping methods move from low to medium fidelity. Starting off with paper sketches and storyboards, the prototypes eventually get more complicated as they allow for more functionality. The key to prototyping in these early stages is low cost and quick implementation. This project heavily employs medium fidelity prototyping techniques. Some example so medium fidelity prototypes used for this project are interface shells that have only a small list of stock price information. The small list (usually 2 or 3 stocks) allows for the user to see where the stock price information would be placed, but still allows the interface designer the flexibility to make changes quickly. A very small number of functionality exists in the medium fidelity prototype.

5. Testing Overview

5.1. User Background

The financial markets are overflowing with an ample amount of information. From annual reports to investment outlooks, any investor will to sort through a myriad of information to make a sound and solid investment. This project will focus on the activities of a speculator in the financial markets. A speculator is someone who may switch investment philosophies when the markets change. Basically, a speculator is someone who is not someone that looks towards a long-term growth investment; rather, he is more focused on the short-term gains of the stock. Investors on the other hand tend to stay with a consistent, long-term strategy in what they invest in. Very little change happens in the investor's portfolio and may hold the stocks for very long periods of time. Clearly, a speculator will need to be more aware of the changes in their portfolio throughout the course of the trading day.

The targeted user for this project then is a speculator of the financial markets that has 5-20 stocks either in his portfolio or on a wish list. A wish list is simply a list of all the possible stocks that he would like to track. While ideally, the system would allow the user to see additional information about the stock such as performance charts and analyst reports, the focus of this project will solely be on the display of the stock price on the user's computer display. The speculators studied are by no means considered professional financial analysts, rather, they are part time investors who have a solid background in financial investments but their primary profession is outside of the financial sector. Because their primary task during the day has nothing to do with the financial markets, there then becomes a need for them to keep an eye on how their portfolio is doing.

As such 4 people were selected as users for this system. They participated in throughout the development of the system and offered feedback of the low level prototypes as well as the high level prototype developed. The backgrounds of the users are varied: Java programmer, security auditor, project manager, and engineer.
In addition, there was a person used as an evaluator of the system. He was not considered to be a user because he had extensive background in commodity trading. His input was primarily used in the evaluation of the high level prototype.

5.2. User Profiling

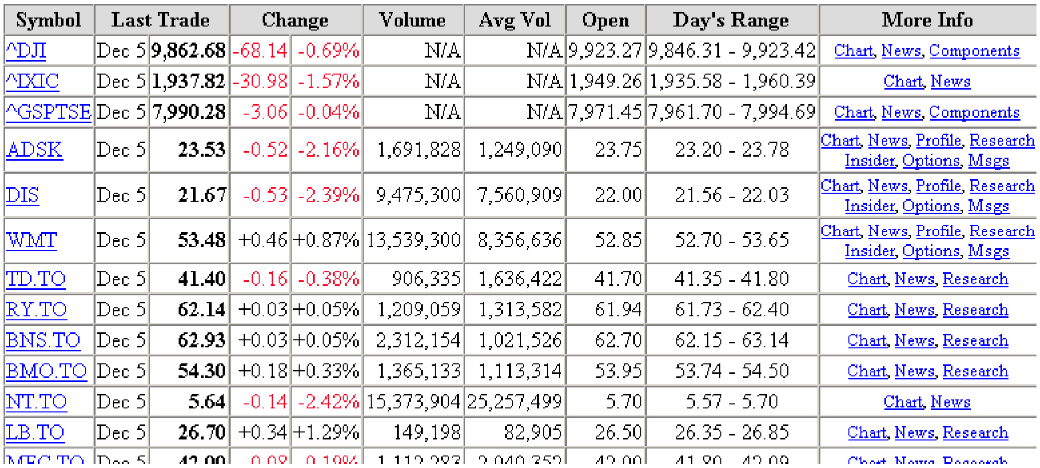
5.2.1. Observing Primary Tasks

Information collected about the users was gathered through the observation technique. On average, users asked as series of questions relating to the type of work they do on their computers. A majority of users simply do word processing such as type up report documents, and a small portion wrote code for software applications. For the most part, users had a considerable number of windows opened on their desktop and many had cascading windows to show more than one window display at a time.

5.2.2. Retrieving Financial Information

Typically, to see a their stock portfolio, users would open up Internet Explorer and browse to the financial pages in Yahoo.com. Since the financial pages on Yahoo do not stream the information, some users refreshed the screen to see the most recent changes on their stock price.

The users collectively wanted to be aware of the changes happening in their portfolio as the trading day progressed. In order to keep aware, the users constantly opened up a bookmarked financial webpage that listed their stock prices. A typical page is similar to the one shown in Figure 2. To update the stock prices, the user would have to constantly refresh the page. At times, users would leave the page open on a portion of their desktops but minimizing the window so as not to interfere with their other tasks. The problem with this solution is that to see the entire list, the user would often have to scroll down using the scroll bar to see the entire listing in the portfolio.



Symbol	Last Trade		Change		Volume	Avg Vol	Open	Day's Range	More Info
^DJI	Dec 5	**9,862.68**	-68.14	-0.69%	N/A	N/A	9,923.27	9,846.31 - 9,923.42	Chart, News, Components
^IXIC	Dec 5	**1,937.82**	-30.98	-1.57%	N/A	N/A	1,949.26	1,935.58 - 1,960.39	Chart, News
^GSPTSE	Dec 5	**7,990.28**	-3.06	-0.04%	N/A	N/A	7,971.45	7,961.70 - 7,994.69	Chart, News, Components
ADSK	Dec 5	**23.53**	-0.52	-2.16%	1,691,828	1,249,090	23.75	23.20 - 23.78	Chart, News, Profile, Research Insider, Options, Msgs
DIS	Dec 5	**21.67**	-0.53	-2.39%	9,475,300	7,560,909	22.00	21.56 - 22.03	Chart, News, Profile, Research Insider, Options, Msgs
WMT	Dec 5	**53.48**	+0.46	+0.87%	13,539,300	8,356,636	52.85	52.70 - 53.65	Chart, News, Profile, Research Insider, Options, Msgs
TD.TO	Dec 5	**41.40**	-0.16	-0.38%	906,335	1,636,422	41.70	41.35 - 41.80	Chart, News, Research
RY.TO	Dec 5	**62.14**	+0.03	+0.05%	1,209,059	1,313,582	61.94	61.73 - 62.40	Chart, News, Research
BNS.TO	Dec 5	**62.93**	+0.03	+0.05%	2,312,154	1,021,526	62.70	62.15 - 63.14	Chart, News, Research
BMO.TO	Dec 5	**54.30**	+0.18	+0.33%	1,365,133	1,113,314	53.95	53.74 - 54.50	Chart, News, Research
NT.TO	Dec 5	**5.64**	-0.14	-2.42%	15,373,904	25,257,499	5.70	5.57 - 5.70	Chart, News
LB.TO	Dec 5	**26.70**	+0.34	+1.29%	149,198	82,905	26.50	26.35 - 26.85	Chart, News, Research
MFC.TO	Dec 5	**42.00**	0.08	0.19%	1,112,283	2,040,352	42.00	41.80 42.09	Chart, News, Research

Figure 2. Typical financial page from Yahoo.com showing a listing for a stock portfolio

Ultimately, the users expressed that their current method of retrieving stock information was rather cumbersome and required significant interruption to their primary task. Therefore, the stock system interface needed to display the entire portfolio to the user. When asked what information about the stock they would like to have displayed, the users gave varied answers. One user expressed he wanted upwards to 8 different items about the stock displayed (stock price, index change, open price, volume, market cap, percent change, 52 week range, P/E ratio), while most other users were content with 2-3 items (stock price, percent change, 52 week range). When asked if they were willing to sacrifice screen real estate to show more stock information, the users were reluctant. While they still would like to have the information available, the users were satisfied with directly interacting with the interface (E.g. Right clicking on the interface) to get more information.

The users were also asked what they first look for when they see a listing of their portfolio, and most replied that they look for the amount of red that is listed. One user replied, that when he first opens the Yahoo financials page, he glances at it to see how many of his stocks dropped that day. Also, he would then look for the ones that dropped a significant amount by checking the percent change of all the 'red' coloured stocks (see Figure 2 of a sample of a portfolio listing). He would conversely do the same thing for all the black coloured stocks (which on the Yahoo page means either a gain in stock price or no change).

5.3. Peripheral Prototype Testing

There are 2 key things in testing of the peripheral system: how the information should be displayed peripherally and where on the screen the information should be displayed.

5.3.1. Basic Primary tasks

Two basic tasks, or test runs have been designed and set up to test the peripheral aspect of the prototypes. Since the prototype being tested is a peripheral system, there needs to be a standard primary task the user performs when evaluating the prototype. This is to ensure that the feedback received from the user is mostly because of the prototype design and not on any other factor. The two main primary tasks used for conducting prototype feedback tests are:

1. Browsing the web to look for information on a specific topic.
2. Copy a selected passage into MS Word.

These two basic tasks reflect the real life tasks users are asked to perform while on the computer.

5.4. Screen Layout

5.4.1. Bar Displays

For a peripheral system to be effective, it must be persistent and locatable. The early portion of usability tests were designed to determine the best screen orientation and layout best suited for these users. As noted, these users have a considerable amount of data appearing on their screens at any point in time therefore it is pertinent that the layout for the system be positioned so that it is minimally intrusive.

As a low fidelity prototype design, a simple sheet of paper was placed over a portion of the user's computer screen. The paper was essentially placed in 3 different positions: along the top, left and right of the screen. This is illustrated in the following Figure 3, 4 and 5 where the blue bar represents where the paper was placed of the computer screen

They were then asked to perform they're normal, regular tasks as if nothing had changed. A post interview was done to ask the users how the positioning of the paper affected their task. A bottom bar was not placed since the Windows task bar already occupied the lower portion of the screen. This would not a good test to run using low fidelity prototyping as there is too much functionality tied in with the Windows bar and blocking it would be an extreme disruption to the primary tasks.

The general consensus from the users was that there was a general preference in the vertical bars as opposed to the top horizontal bar. The feeling was that there already exists the windows task bar at the bottom of the screen. A further addition of a horizontal bar on the top would reduce the amount of screen real estate available. The vertical bars placed on the sides were more acceptable as most documents have automatic word wrapping features. In choosing either the right or the left vertical bar, more users chose the right bar. Users felt it was less intrusive when placed on the right as opposed to the left since the general tendency is to look first from the left then to the right. The users expressively felt more comfortable doing their primary task when the bar was on the right hand side of the screen.

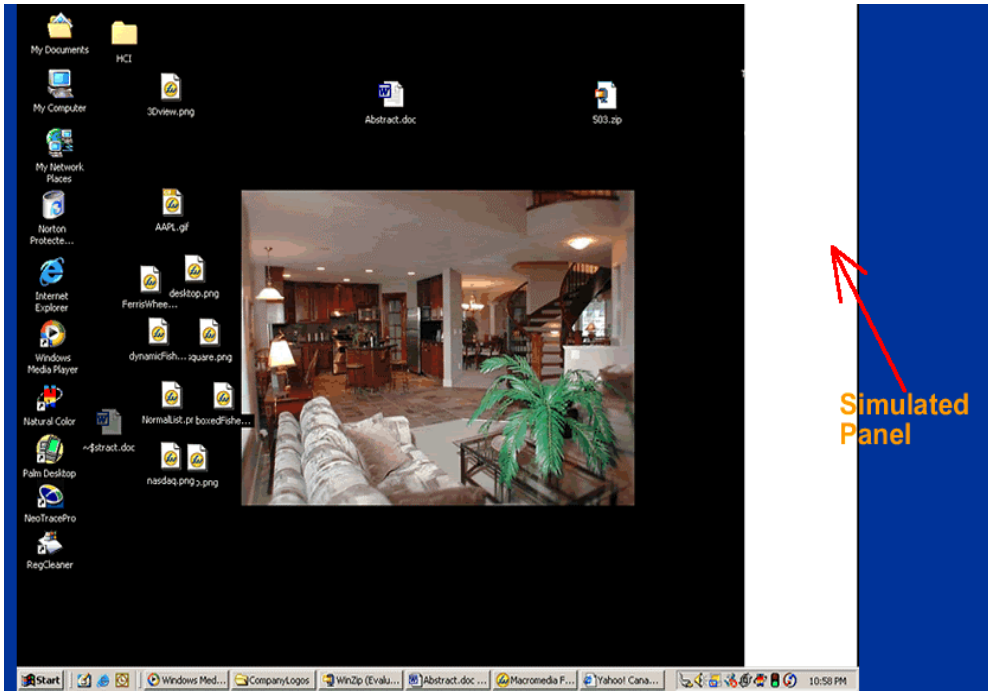


Figure 3. Simulated Panel on the right side of the screen.

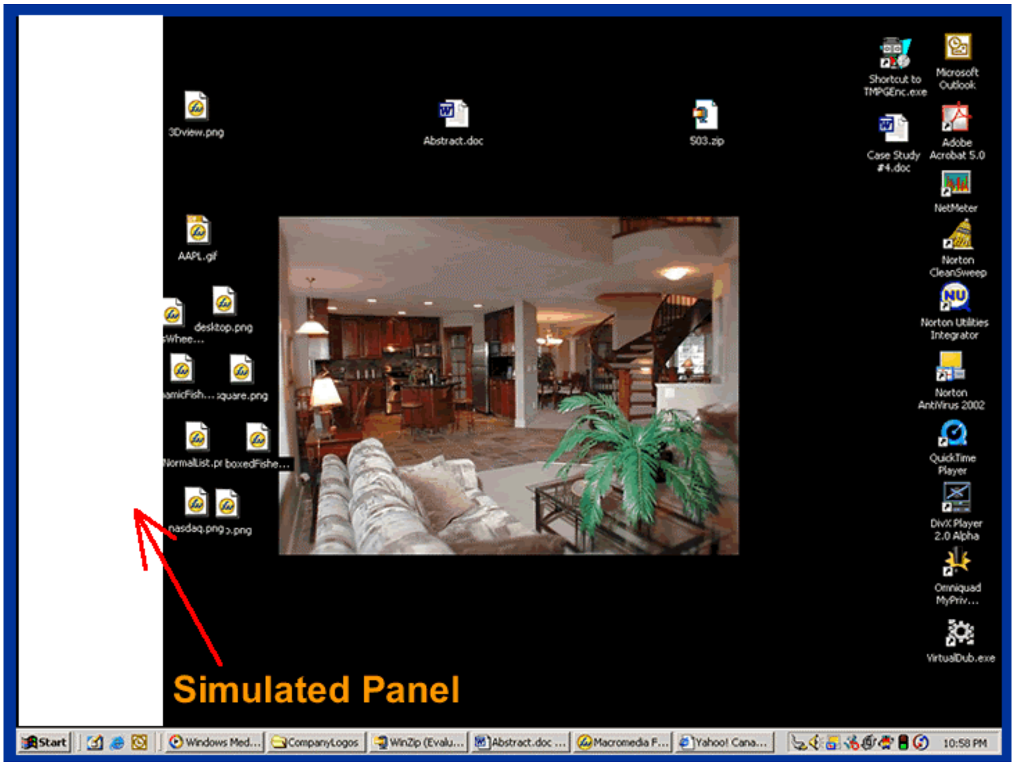


Figure 4. Simulated Panel on the left side of the screen

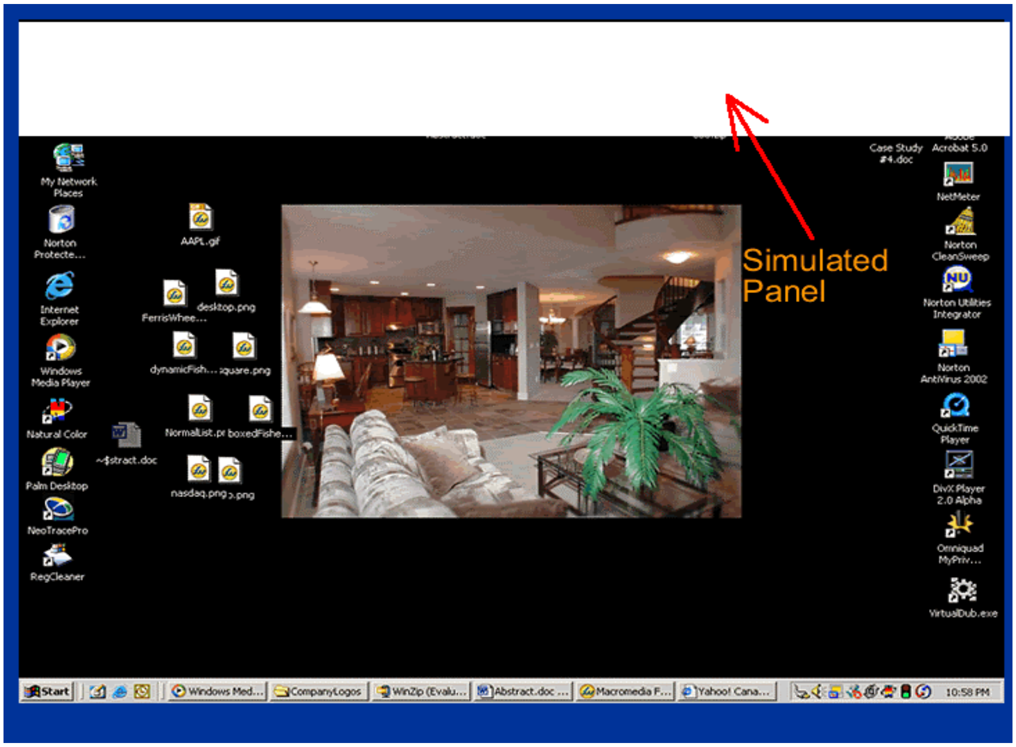


Figure 5. Simulated Panel on the top of the screen

5.4.2. 3D Box displays

There were some experiments using a 3-D box perspective (Figure 6). The idea was that the side panels could be used to display information while the main screen still display the information for the primary task. The information on the side panels would display the stock information in a perspective giving the illusion of a 3D effect. This idea was not pursued further because in order to achieve the 3D effect, 4 side panels would have to be used and this was just too space consuming for the user. There was a refined design to still try to achieve the 3D effect using only two panels, however using 2 sides of the screen is unnecessary since information can be effectively displayed using just one panel.

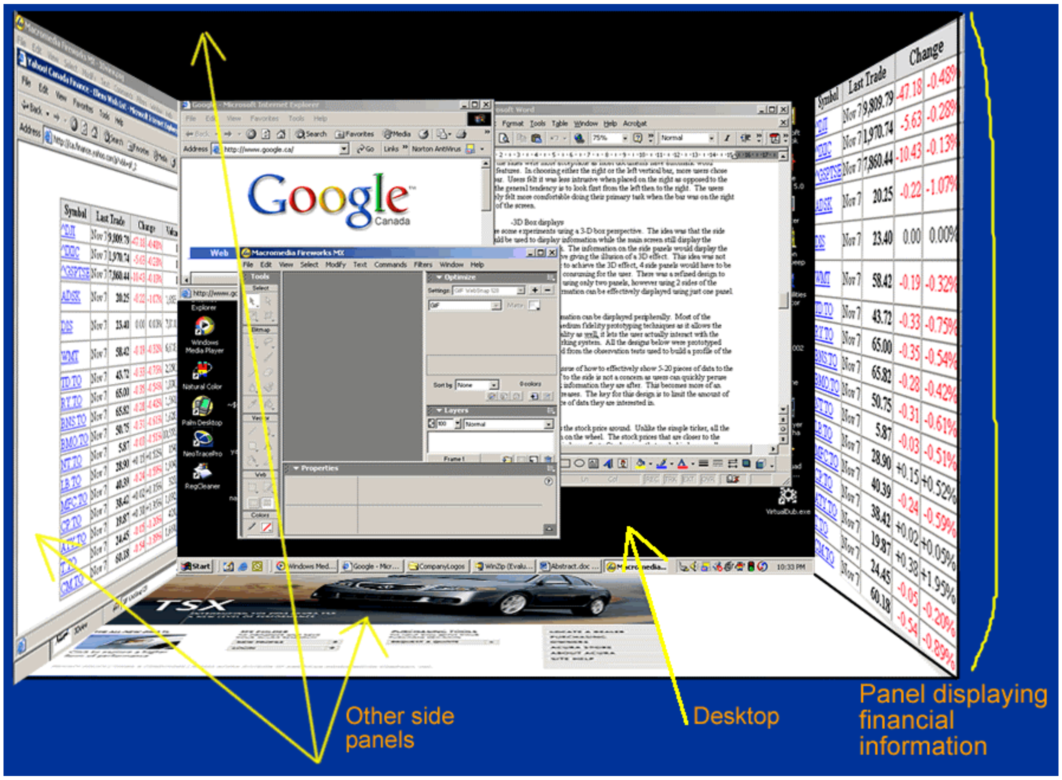


Figure 6. A mock up of a possible 3D layout

5.5. Peripheral Displays

There are many ways that information can be displayed peripherally. Most of the prototypes are designed using medium fidelity prototyping techniques as it allows the user to see more of the functionality. In addition, it lets the user actually interact with the interface as if it was really a working system. All the designs below were prototyped based on the information gleaned from the observation tests used to build a profile of the user.

Aside from layout, there is the issue of how to effectively show 5-20 pieces of data to the user. Listing five items of data off to the side is not a concern as users can quickly peruse through the list to find the stock information they are after. This becomes more of an issue as the number of items increases. The key for this design is to limit the amount of item a user has to locate the piece of data they are interested in.

5.5.1. Ferris Wheel Design

The Ferris wheel display rotates the stock price around. Unlike the simple ticker, all the prices are visible at any position on the wheel. The stock prices that are closer to the front are displayed brighter and in larger font. Stock prices that are behind are smaller and more faded. This perspective of the wheel saves some room on the screen since the Ferris wheel is not displayed 'flat out' like a 2D circle. Also, the stock prices are color coded green, red and black. The stock information text is green if the last change in stock price was a positive gain, red if the last change was negative, and black if there was no change whatsoever. This color-coding immediate alerts the user of the type of change happening to their portfolio. An entirely red wheel will tell the user that their portfolio is decreasing in value, and vice versa. This design is better than the normal ticker as all the stock information is visible at all times. While it may not be the case that the stock is in the front, the user can see how far back the stock is and check back in a few seconds when the information will be more prominent.

To that end, a Ferris wheel display was conceived. A simple paper model was constructed to illustrate the idea, but the bulk of development will have to focus on a medium fidelity prototype as is seen in Figure 7. The idea behind this prototype is to have all the stocks visible as they move around the wheel, the front information would be more opaque, while the information in the back would be darker and more faded. Since all the stocks are visible, the idea is that the user can see when the stock will move to the forefront and check back later when the stock becomes more visible. Also, if the stock is not directly behind another stock, the price for that stock is easily visible.

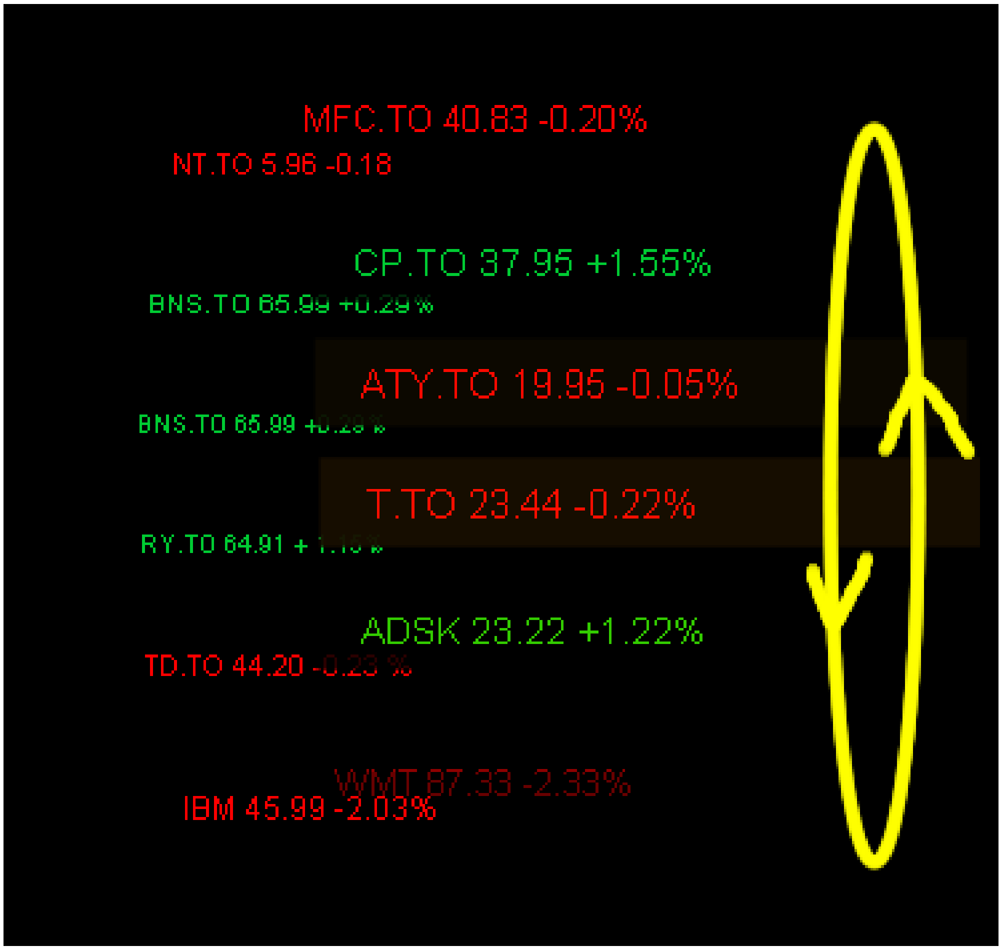


Figure 7. Ferris wheel display. The yellow line represents the direction of rotation

5.5.2. Square Layout

Another idea to display the information is to have the stocks arrayed in a box shaped arrangement as seen in Figure 8. This way, the user is aware spatially of where the stocks are in the arrangement and can easily direct their attention to that area. This is layout is more efficient than a list since it is able to show 9 different items in a small area which makes it easy to pick up information in a quick glance of the interface.

In terms of alert notification, the cube layout can use the fish eye method to display a particular square to alert the user to changes. As seen in the Figure 8, the important square is enlarged and the other squares are squished up to allow the important square to grow in size.

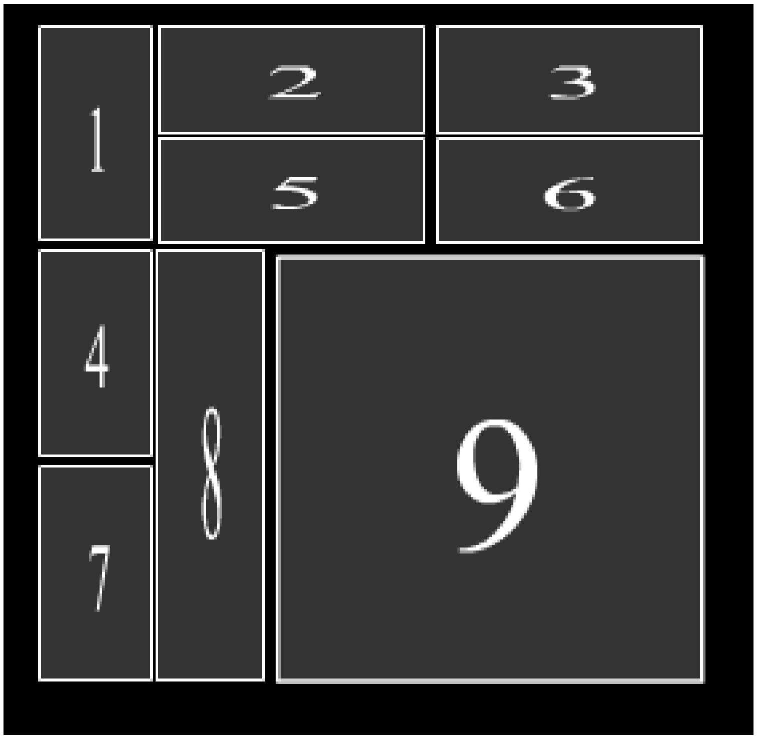
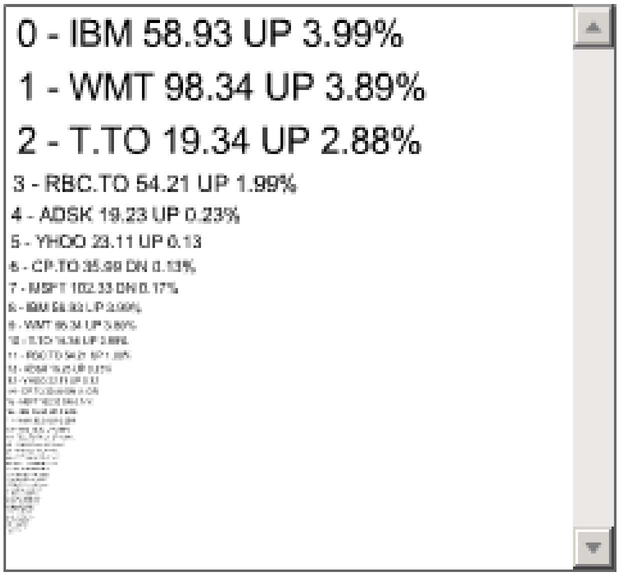


Figure 8. Square display

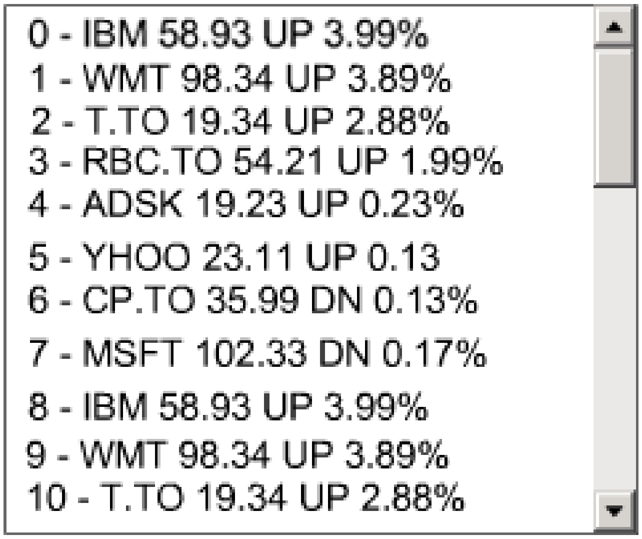
5.5.3. Dynamic List using Fish Eye

This prototype is designed to show the stocks using a fish eye lens, the concept that was first developed by Furnas [3]. In the Figure 9 below, the stock with the most change is displayed at the top so in Figure 8, that stock would be IBM, other stocks that have less change such as CP.TO are smaller and further down the list. This prototype is designed so that at a glance, the user can see which few stocks are experiencing the most changes during the trading day. There is no need to go through all the data to determine that information. This is more efficient than the simple ticker because the information is immediately sorted with the more significant information displayed prominently in the list. This design reduces the time needed by the user to locate the information they want. This is very evident when comparing a normal list to the fisheye display as seen in Figure 9 and 10. Also, the sorted fish eye display has the added advantage in that it automatically relegates unimportant data to the bottom of the list where it is less visible whereas the normal list makes no such differentiation.



0 - IBM 58.93 UP 3.99%
1 - WMT 98.34 UP 3.89%
2 - T.TO 19.34 UP 2.88%
3 - RBC.TO 54.21 UP 1.99%
4 - ADSK 19.23 UP 0.23%
5 - YHOO 23.11 UP 0.13
6 - CP.TO 35.99 DN 0.13%
7 - MSFT 102.33 DN 0.17%
8 - IBM 58.93 UP 3.99%
9 - WMT 98.34 UP 3.89%
10 - T.TO 19.34 UP 2.88%
11 - RBC.TO 54.21 UP 1.99%

Figure 9. Screenshot of a Dynamic List Fish Eye display



0 - IBM 58.93 UP 3.99%
1 - WMT 98.34 UP 3.89%
2 - T.TO 19.34 UP 2.88%
3 - RBC.TO 54.21 UP 1.99%
4 - ADSK 19.23 UP 0.23%
5 - YHOO 23.11 UP 0.13
6 - CP.TO 35.99 DN 0.13%
7 - MSFT 102.33 DN 0.17%
8 - IBM 58.93 UP 3.99%
9 - WMT 98.34 UP 3.89%
10 - T.TO 19.34 UP 2.88%

Figure 10. Screenshot of a normal list

5.5.4. Boxed Fisheye display

The idea behind this interface is to have a list of stocks display horizontally or vertically. Each stock would be surrounded by a square that is one of three colors: green, red, or black. The colors represent the change in stock price, green means the stock price went up, red means it went down and black means no change (Figure 11). The brightness of the color increases or decreases with the amount of change the stock is experiencing. So for example, a 10.0% increase in the stock price would show the stock price in a bright green square, while a 1.00% increase would display a duller green. When a sudden event occurs, the box that has the stock price on it would enlarge and come to the forefront of the display. This is effective in that when there is a dramatic change, the intensity of the color of the box will increase as well as the size. This interface uses the two different types of visual variables to convey information: color and size. As noted by Carpendale [2], these variables can be used as a basis for information visualization. While the size of the box is not an effort to convey any quantitative information about the stock price, it is effective in representing ranking or order within the list. For example, if there are 2 boxes that display stock price, one can easily see which of the two stocks experienced a greater gain by comparing the brightness and the size of the two boxes. This display is different than the dynamic list fish eye display in that the positioning of the stock price doesn't change. This allows the user to always refer to the same area on the screen for the stock information. So for example, if there are 3 stocks in a portfolio, and they are ordered: Autodesk (ADSK), Microsoft (MSFT), and Canadian Pacific Railway (CPR) on the boxed fisheye display, the user will always be have to find MSFT between ADSK and CPR. The position of the stocks will not change, only the box that surrounds the stock price.

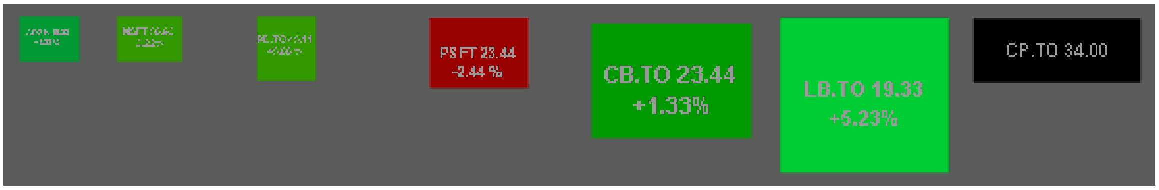


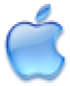
Figure 11. Boxed Fisheye Display

6. Information Visualization

The representation of information has to be as effective as the peripheral design if the user is to understand the information presented to them. Tests were run comparing different representation of data.

6.1. Company Logos

A simple test was run to determine the amount of time it took for a user to identify the name of a company. Users were shown the company logo and the ticker symbol. While most company logos bear the name of the company, users identified quicker with the logos than the ticker names. There was a general preference to using the company logo rather than textual representation (E.g. full company name, ticker symbol, abbreviations).

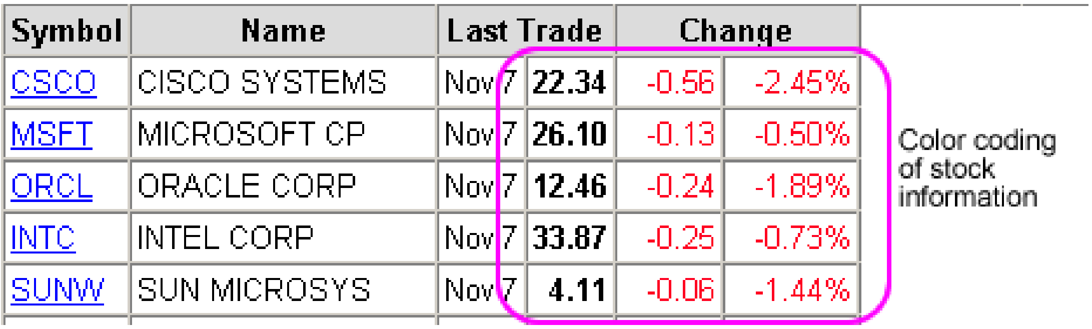


Vs. AAPL

Figure 12. Sample of a Company Logo

6.2. Color representations

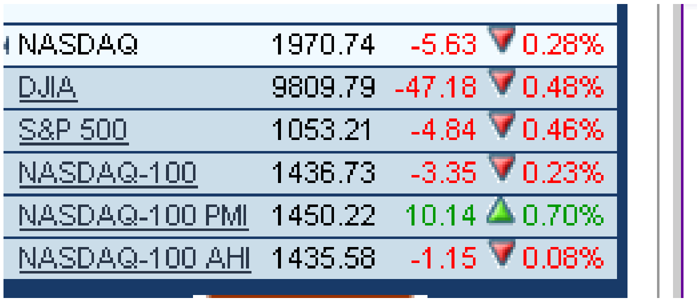
Different color schemes were applied to stock information and the general consensus is that the industry standard of green, red, and black should be used. It is well regarded in the financial sector that green represents positive change, red represents negative change and black means no change. Below, Figure 13 and 14 display some stock listings as is shown on popular web pages.



Symbol	Name	Last Trade		Change	
CSCO	CISCO SYSTEMS	Nov 7	22.34	-0.56	-2.45%
MSFT	MICROSOFT CP	Nov 7	26.10	-0.13	-0.50%
ORCL	ORACLE CORP	Nov 7	12.46	-0.24	-1.89%
INTC	INTEL CORP	Nov 7	33.87	-0.25	-0.73%
SUNW	SUN MICROSYS	Nov 7	4.11	-0.06	-1.44%

Color coding of stock information

Figure 13. Screenshot of a financial display on Yahoo Finance



NASDAQ	1970.74	-5.63	▼	0.28%
DJIA	9809.79	-47.18	▼	0.48%
S&P 500	1053.21	-4.84	▼	0.46%
NASDAQ-100	1436.73	-3.35	▼	0.23%
NASDAQ-100 PMI	1450.22	10.14	▲	0.70%
NASDAQ-100 AHI	1435.58	-1.15	▼	0.08%

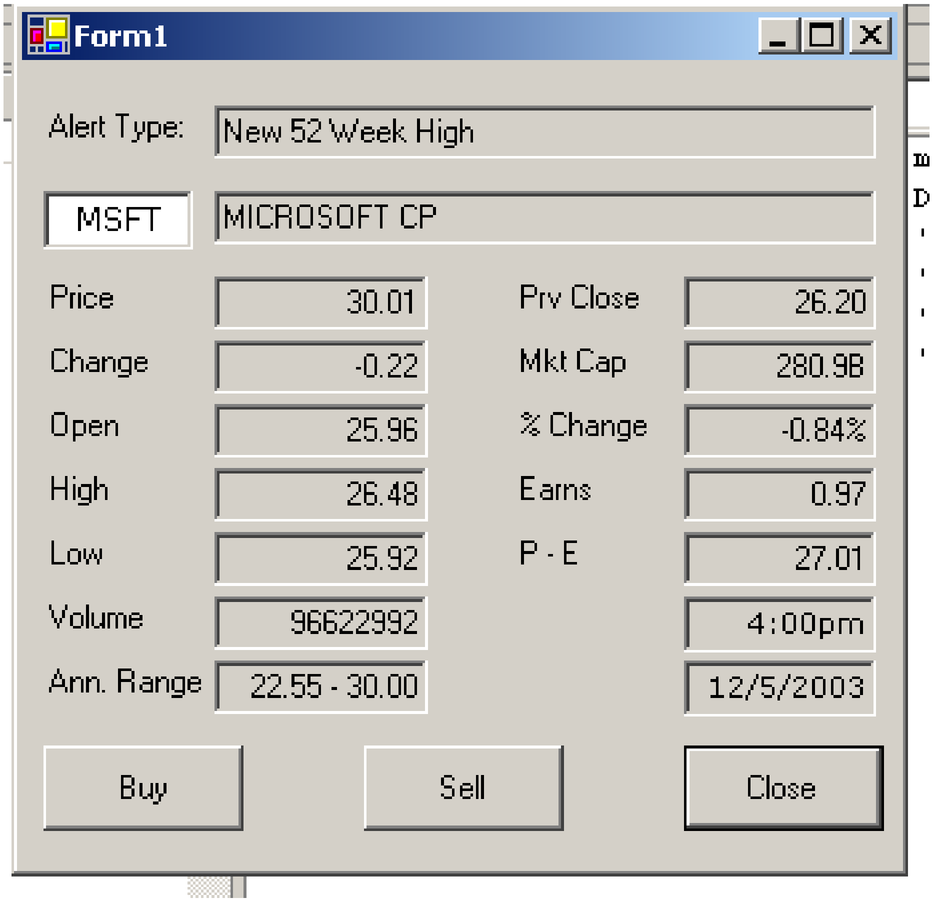
Figure 14. Screenshot of a financial display on NASDAQ.com

7. Immediate Notification Alerts

The design of the immediate notification alerts was consciously left after the peripheral displays were well under way. This was to ensure that the notification alerts would work efficiently with the peripheral system. For example, a pop up window is an efficient way of grabbing the user's attention. The MSN messenger popup window that appears when a user logs in is a good example of this type of alert display. An efficient use of the alert notification interface would require the window to pop up once a stock hit a new threshold. However, the display in the pop up window should relate somehow to the listing in the dynamic fish eye display. For example, if the RBC stock hit a new 52 week high with a price of 54.21 (as is seen in Figure 9), a pop up window would appear alerting the user of the change. The listing in the fish eye display would flash as well indicating to the user which stock in the list has changed. The flashing helps tie the information contained in the pop up window with the peripheral display. This integration of displays help the system appear to function as one unit rather than two disparate functionalities that are just thrown together. Simply put, the effectiveness of the notification alerts depend on the context in which they operate.

7.1. Existing Systems

Existing financial displays such as Yahoo Finance do have the capability to sent notification alerts to the user. The notifications are set by the user, which then signals the system to send a notification when a stock hits the specified price. The alerts however, may not be an immediate notification as the message is sent via email to a user specified account. It is only if the user is in his email account will that message be received. This may prove to have serious consequences for the user as immediate may be warranted. As such using the existing system would not be sufficient. Below, Figure 15 is a prototype of a pop up window displayed to the user when a new 52 week high is reached for a stock.



Alert Type:	New 52 Week High		
MSFT	MICROSOFT CP		
Price	30.01	Prv Close	26.20
Change	-0.22	Mkt Cap	280.9B
Open	25.96	% Change	-0.84%
High	26.48	Earns	0.97
Low	25.92	P - E	27.01
Volume	96622992		4:00pm
Ann. Range	22.55 - 30.00		12/5/2003

Figure 15. Sample pop up window indicating new 52 week high alert

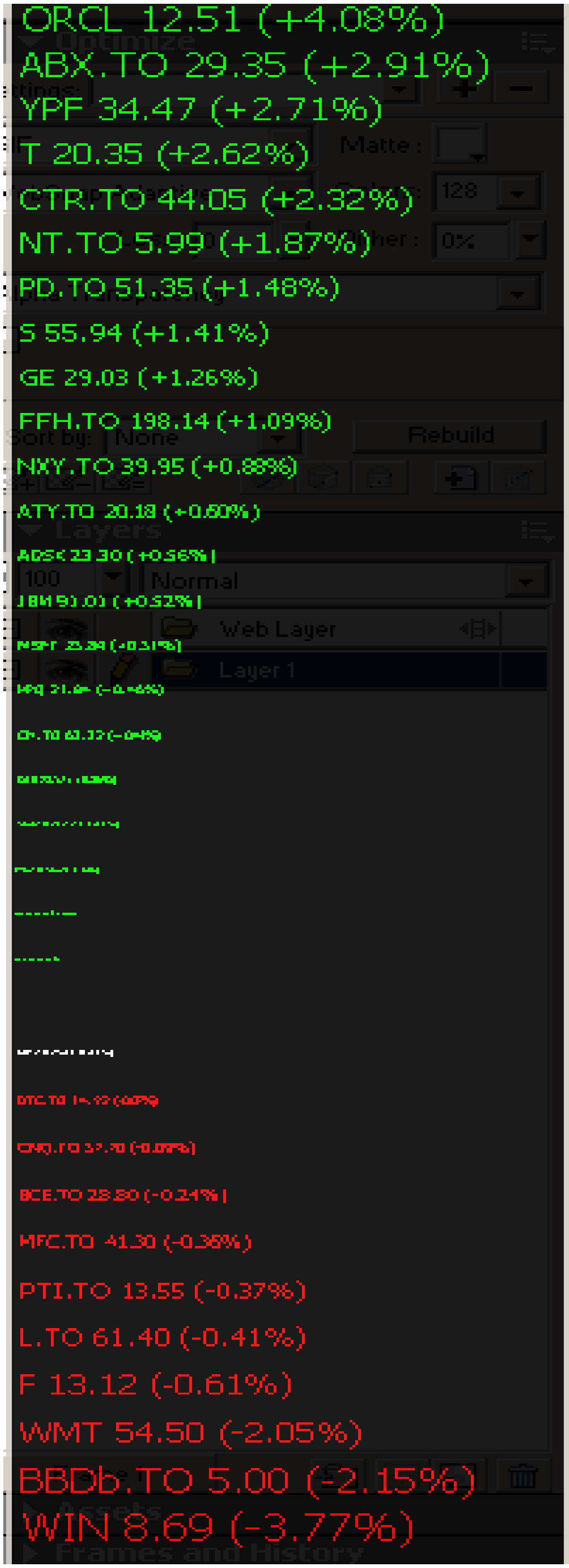
8. Final Peripheral Interface Design Decisions

8.1. Dynamic Fish Eye List Design

The decision on the interface for the system must ultimately be based upon the guiding principles that were dictated by observing the users behaviour and tasks (Section 2 and 5.1.2).

The small area by which the interface can inhabit limits the amount of information that can be shown to the user through the peripheral display. As such, only 3 pieces of information about the stock will be shown: the stock symbol, the percent change, and the stock price. As noted in Section 5.1.2, the users were content to directly access additional information by interacting with the interface via a mouse click.

The users all wanted to have a general idea of how well or poorly their entire portfolio is performing. As such, the interface displays the portfolio to the user in a list type format. However, the interface is quite different than that of a simple list (see Figure 16).



```
ORCL 12.51 (+4.08%)
ABX.TO 29.35 (+2.91%)
YPF 34.47 (+2.71%)
T 20.35 (+2.62%)
CTR.TO 44.05 (+2.32%)
NT.TO 5.99 (+1.87%)
PD.TO 51.35 (+1.48%)
S 55.94 (+1.41%)
GE 29.03 (+1.26%)
FFH.TO 198.14 (+1.09%)
NXY.TO 39.95 (+0.88%)
ATY.TO 20.18 (+0.60%)
ADSK 23.30 (+0.56%)
IBM 91.01 (+0.52%)
MSFT 25.29 (+0.31%)
HPQ 21.0[illegible] ([illegible]%)
CP.TO 63.37 ([illegible])
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
DTC.TO 14.92 ([illegible])
CNQ.TO 37.70 (-0.09%)
BCE.TO 28.80 (-0.24%)
MFC.TO 41.30 (-0.36%)
PTI.TO 13.55 (-0.37%)
L.TO 61.40 (-0.41%)
F 13.12 (-0.61%)
WMT 54.50 (-2.05%)
BBDb.TO 5.00 (-2.15%)
WIN 8.69 (-3.77%)
```

Figure 16. Dynamic Fish Eye listing of a portfolio

Firstly, the stocks in the portfolio are sorted. The stock with the biggest percent gain is listed at the top, while the stock with the biggest percent lost is at the bottom of the list. The intent in organizing in this manner is to minimize the amount of time the user takes to get the information he needs. Since the list is sorted, the user can just glance at the top of the list and at the bottom to find the 2 most changed stock in his portfolio. Subsequently, he can glance top to bottom to find out which stocks are performing well and which are not.

In addition to the sorted list featured, the interface also utilizes a well-known and widely accepted color scheme of green/red/white. As noted in section 6.2 the decision to represent changes with these colors and not any other colors was a conscious and deliberate choice. Since the background was already black, stocks, which indicated no change, were represented with white to make them visible. The choice to have a black background was also a conscious decision as well. There were some prototypes done with a white background (see Figure 9), but feedback received seemed to arrive at the consensus that the green and red against a white background was sometimes very hard to read. When a prototype was developed which had a black background, the users preferred that to the white background interface.
The color representation also has an added advantage in the display. Since the list is sorted, the color scheme for the display will always be green on top, white in the middle, and red on the bottom). As such, the user can glance peripherally at the display and get a general idea of how well or poorly their portfolio is performing by the amount of green or red displayed.

Finally, as described in Section 5.4.3, the interface has another advantage compared to a normal listing. As the percent change increases, so does the size of the text in the display. This is true whether the percent change is a positive gain or a negative gain, so again referring to Figure 16, it can be seen the stocks with the biggest text size also have the biggest change attributed to them. Noted as one of Bertin's visual variables and cited as an important factor in information visualization by Carpendale [2], the sizes of the text help indicate the amount of change for that stock.

8.2. Design Issues

Important to note, that while Section 6.1 discussed the effectiveness of the inclusion of logos, the choice to include them for the system interface was consciously omitted. A key requirement in including logos in the interface was that all the stocks would have a logo available and that the logo be consistent in size and representation (the logo is placed on the same coloured background and in the case of Figure 17, the logos are all on a white background).

Figure 17. Sample of a set of consistent display company logos

This posed 2 different unforeseen problems. First, the logos needed to be added in dynamically as the users change the stocks in their portfolio. So for example, a new logo would need to be input into the system when the user adds in DIS (Disney) to their portfolio. Second, the appearance of all the logos needs to be consistent.

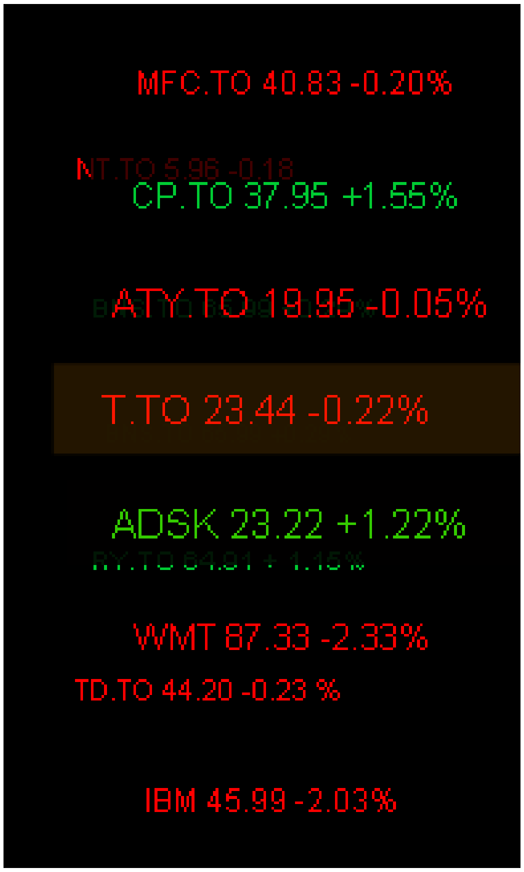
Fortunately, the xml stream from NASDAQ has an element available that provides a URL to a standardized logo (E.g. http://content.nasdaq.com/logos/AAPL.GIF**)** for some of the companies. The logos from the URL are the ones seen in Figure 17. The problem however was that the users for the system have stocks which are listed on the Toronto Stock Exchange (TSX) and other indices. The intent of the system is to build working problem which would take in any stock information, retrieve a stock stream from a real financial service provider, and redisplay that information back to the user. Using the Nasdaq stream, while it may provide a source of standardized graphics, was not a feasible solution since the users could not use it to track their actual portfolio.

Striving for consistency, and after receiving some feedback from users, it was better to either use just company logos to represent the company or just the ticker symbol and not mix the two representations together. As such, the choice to use the ticker symbol was made rather than the company logo.

8.3. Other Prototype Critiques

8.3.1. Ferris Wheel

The Ferris Wheel design allowed all the stocks to be shown on the interface (see Figure 7). Since all the stocks were visible, it was possible to immediately find the price of any stock in the portfolio. Also since the text was color coded according to findings listed in Section 6.2, it was instantly recognizable which stocks produced a gain or a loss. Unfortunately, the design needed quite a bit of space on the screen, and the users complained that their workspace was considerably more cramped. In an effort to minimize the space used the angle of the wheel altered. While this reduced the amount of space, the stocks became very hard to read. It also became very difficult to tell which stocks were in front and which were in the back. Ultimately, it was decided that other prototypes were better in satisfying the needs of the users.



MFC.TO 40.83 -0.20%

NT.TO 5.96 -0.18

CP.TO 37.95 +1.55%

ATY.TO 19.95 -0.05%

T.TO 23.44 -0.22%

ADSK 23.22 +1.22%

RY.TO 64.91 + 1.15%

WMT 87.33 -2.33%

TD.TO 44.20 -0.23 %

IBM 45.99 -2.03%

Figure 18. Modified Ferris Wheel Display

8.3.2. Square Layout

The square layout ran into the same problem as the Ferris Wheel design. With a small list of items, it is very efficient in conserving space, however, as the items grow, so too will the square. The square layout is similar to the system available on the smart money website [8].

8.3.3. Boxed Fisheye Display

Essentially, this is a horizontal version of the dynamic fish eye display. The differences being that the stocks are not sorted. Also, comments from users found the display a bit scattered since there could be 2 green boxes that are relatively the same size located in no particular order (for example, going from left to right, box 1 could be the same size and color as box 5). The user said that it was difficult to know where to look and focus his attention when he glanced at the interface.

9. Dynamic Fish Eye List Interface Assessment

9.1. Testing Techniques

The same users that participated in the development of the interfaces were used in the testing of the dynamic fish eye list interface; their profiles are described in Section 5.1.

9.1.1. Test Types

To verify the effectiveness of the interface, 2 types of tests were done on the system.

The first type of test was essentially a beta release of the system to the users. Since the system is designed to take in real streaming stock data, the users were given beta builds of the system to use in their actual work environments. Interviews were done after a one-week period to gather feedback from the users.
The second type (controlled test) of test is a controlled direct comparison between the interface and a normal stock ticker.

The plan was to allow all the users to take the system and use it in their work environments and later, run the second type of test where the system is tested against a normal stock ticker.

9.1.2. Basic Tasks for Controlled Test

For basic tasks, the same types are used as mentioned in Section 5.2.1 (1. Browsing the web to look for information on a specific topic, 2. Copy a selected passage into MS Word). The users are first to perform one of the two tasks with the normal ticker as the peripheral display.

Important to note, the content of the tasks are changed each time the user performs another test. That is, if they are copying a passage into MS Word, the content of the passage is changed each time the user performs another test. This is to prevent the user from getting too familiar with the task and ensuring results and feedback obtained from the interfaces are a result of the design change and not other external factors.

9.1.3. User Feedback for Controlled Test

Once they have completed the task they will be asked a series of questions related to the prices of the quotes displayed on the ticker (E.g. What was price did you notice for Walmart (WMT)? How well was your portfolio doing in general?). Also they will be asked about the general look and feel and most importantly, the effectiveness of the normal ticker being used as the peripheral display (I.e. How disruptive was the peripheral display to your primary task?). The same thing will be repeated but this time, for the dynamic fish eye interface.

It would be extremely difficult to ask questions about the values of stocks if they were actual real stock information. This is because the value of the data is dependent on the state of the market at that particular time. To ensure that all the stocks stay within a reasonable range, the system was altered slightly from taking in streaming stock information to reading in a set of 20 pre-made text files. The pre made text files also allow for certain immediate notification alerts to take that normally would not occur.

9.2. Alert Notification Testing

Testing the immediate alert notifications was difficult. While Yahoo Finance does provide that functionality, it was very rare that any notification would be sent since it is not a common event for stocks to reach new 52-week highs or lows. As such, the effectiveness of the display was determined through feedback from the users during post interview sessions after a test run.

10. Results

10.1. Benefits in the dynamic fish eye display

In using the dynamic fisheye display, the users were able to able to determine a better assessment of the general performance of their portfolio. That is, at a glance, they could tell whether a large portion of the stocks were experiencing gains or losses. While the users said it was possible to get a general feel of how the portfolio is performing using the ticker by mentally noting how many red or green symbols have passed by, the dynamic fish eye provided the information more efficiently.

Also, when using the dynamic fisheye display, users were better able to get a feel of how certain stocks in their portfolio were performing. One user remarked, that he could occasionally glance down the list to see where one particular stock was in relation to his other stocks. He found that it was easier to track a particular stock using the dynamic fisheye display than the ticker display since he had to frequently look at the ticker to see if that quote had passed.

Also, when asked about the disruptiveness of the displays, the users found that the dynamic fish eye was more 'quiet' than the ticker. Despite some flicker, the users found the dynamic fish eye list worked more effectively in the periphery than the scrolling text of the ticker.

10.2. Deficiencies in the dynamic fish eye display

Depending on the number of stocks in the users portfolio, some of the lesser-changed stocks may not be that visible. Users who beta tested the system remarked that it would still be nice to see that stock information even if there was little or no change in the stock price. Also, some users noted that the slim display of the ticker was more space efficient than the dynamic fish eye display.

11. Further Work

A large portion of this project was spent developing and designing the peripheral system. This is with good reason as the peripheral display is what the users will be seeing a majority of the time. Only when a special event occurs does the immediate event notification appears. It would have been nice to perform more usability tests on the immediate alert notifications. While the popup window was sufficient for the task the users required, it would have been nice to explore different designs that could be implemented for a user notification alert.

In terms of expandability, the system and interface is capable of displaying more than 20 stocks. However, with a significantly larger number of items to track, the effectiveness of the dynamic fish eye display may be diminished since the text can get to the point that it becomes unreadable. More tests and design iterations can be done to allow the interface to better display a long list of stocks.

12. Conclusion

It is difficult to say that any one particular design of an interface will be suited for all users in any situation. From this study, it can be seen that an effective system can be created using user-centered and task centered design techniques. While the interface may still pose some deficiencies to some users, it serves to satisfy the needs of the majority of the users of the system.

13. References

[1] Cadiz, JJ. Venolia, G. D., Jancke, G., Gupta, A., (2001) *Sideshow: Providing Peripheral Awareness of Important Information.* Technical Report, Microsoft Research, Microsoft Corporation. (Redmond, WA), September

[2] Carpendale, M.S.T (2003) *Considering Visual Variables as a Basis for Information Visualization,* Technical Report, Department of Computer Science, University of Calgary. January 30.

[3] Furnas, G.W, (1986) *Generalized fisheye views.* Proceedings of Human Factors in Computing System., CHI '86 (New York, NY) pp18-23

[4] Lewis, Clayton, Rieman, John. (1994) *Task-Centered User Interface Design,* website: http://hcibib.org/tcuid/ (Accessed Dec 2003) Chapter 1

[5] McCrickard, D. S., Catrambone, R, Stasko, J. T., (2001) E*valuating Animation in the Periphery as a Mechanism for Maintaining Awareness,* Proceedings of INTERACT 2001, Tokyo, Japan, July 2001, pp. 148-156

[6] Redish, J.C. (1994) *Involving Users Throughout the Information Development Process* Proceedings of Society of Technical Communication

[7] Soderston, C., Rauch, T. L. (1996) *The Case for User-Centered Design*, Proceedings of Society of Technical Communication

[8] http://www.smartmoney.com/marketmap/ (Accessed Dec, 2003)