December 2004

Browser-Based Applications: Positive or Negative Transference?

Mark Silver
Fordham University

Sidne Ward
University of Missouri, Kansas City

Follow this and additional works at: http://aisel.aisnet.org/amcis2004

Recommended Citation

http://aisel.aisnet.org/amcis2004/398

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
Browser-Based Applications: Positive or Negative Transference?

Mark S. Silver  
Fordham University  
silver@fordham.edu

Sidne G. Ward  
University of Missouri, Kansas City  
wards@umkc.edu

ABSTRACT
Applications that run on top of web browsers dominate the Internet today. Given the many similarities among these applications’ features, positive transference from one to another is often seen as an important source of ease-of-use for such applications. This paper examines the many differences in the way similar features are implemented in different browser-based applications, analyzing the way these inconsistencies can lead to negative transference (interference) that degrades rather than enhances usability.

Keywords
Design, human factors, standardization, interference, transference, browser-based applications, usability, shopping-cart applications.

INTRODUCTION
Browser-Based Applications (BBAs) dominate the world of Internet Applications today. Instead of employing a specialized software client, BBAs run on top of the web browser. Their functionality is provided to users by a combination of the web browser's (limited) functionality and the web server's information-processing capabilities, accessed through server-generated web pages. BBAs are the chief vehicle for on-line shopping and are also widely used for e-trading, e-banking and e-mailing. For example, Amazon.com is a browser-based application for buying books and other physical goods and Hotmail.com is a browser-based e-mail application.

BBAs are popular for several reasons, but one of the most common explanations of BBA popularity is that BBAs are extremely easy to learn and use. The argument has two parts, with the first generally running like this:

All you need to know to operate a BBA is how to use a web browser. And nearly everyone who uses the Internet knows how to use a browser. Indeed, most Internet users are highly proficient browser users. Even Internet novices will not have difficulty because browsers are so easy to use. Operating a browser and, by extension, a BBA only requires being able to enter a URL in the address bar, filling in form fields, and clicking on hotspots.

However intuitively appealing this argument may be, it is nonetheless faulty. Three counter-arguments undermine it:

1. BBAs implement their functionality through server-generated web pages, and users interact with BBAs through those pages. While the browser itself may be simple, the web pages that constitute a BBA could be, and frequently are, quite complex. Similarly, while browsers may be easy to use, a given BBA could have significant usability problems.

2. Many web sites—and consequently, many BBAs—suffer from poor attention to usability in design and inadequate usability testing (Nielsen, 2000; Shneiderman, Lazar, and Ivory, 2003).

3. Web browsers are fundamentally deficient as platforms for building general-purpose applications, so applications built on top of them suffer from a large set of practical problems (Silver, 2003).

Together these counter-arguments suggest that while users may perceive BBAs to be easy-to-use—because they find browsers easy-to-use—BBAs may not be so usable after all. But before reaching this conclusion, the second part of the argument supporting BBA ease-of-use needs to be considered:

People who use a few BBAs typically use many, and since BBAs have many similarities one to another—both because they are all built on the common browser platform and because they implement many similar features—positive transference from currently used BBAs will make it easier for someone to learn and use additional BBAs.
This assertion seems reasonable on the surface, but it too fails under closer examination. Despite some similarities across BBAs there are far more differences. In particular, similar features are often implemented with small or great variations in different BBAs. These inconsistencies lead to an interference (negative transference) that is likely to equal or exceed the positive transference. Consider a simple example, the “printable version” button. This feature differs greatly in how it is implemented across BBAs. For instance, some BBAs open the printable version in a new browser window whereas others open it in the same window. In the former case one simply closes the printable version window after printing. A person accustomed to this approach who encounters an application that does not open a new window is still likely to close the window, in this case closing the browser entirely and aborting the BBA midstream.

The questions of how usable BBAs are and of how positive and negative transference affect overall usability are questions in need of empirical study. But prior to embarking on an empirical study of the latter question, there is value in studying the differences in how similar features are implemented across BBAs. Such is the mission of this paper, which has four objectives:

1. to make a prima facie argument that differences among BBAs can generate significant negative transference that degrades usability.
2. to identify features whose inconsistent implementation across BBAs make them likely sources of interference and, therefore, worthwhile targets for empirical study.
3. to propose an agenda for researching BBA interference.
4. to alert BBA designers to features that are likely to be sources of interference, as a first step toward reducing this interference through standardization.

The sources of interference can be analyzed at two levels:

- application-independent features
- application-specific features

Application-independent features apply to many different types of BBAs. For example, many BBAs require logins and offer printable versions. Application-specific features apply to a given class of applications. Many e-tailing BBAs use a shopping-cart approach defined by a specific set of features, such as adding items to carts. Similarly, on-line banking BBAs implement a different set of features, such as transfers from one account to another. This paper first considers a set of common application-independent features and then focuses on features typically found in retail shopping-cart applications. These analyses are preceded by a brief overview of the concept of transference.

**POSITIVE AND NEGATIVE TRANSFERENCE**

Transference refers to the process of users' bringing their previous experience with them when encountering new applications that resemble ones with which they are already familiar. Although transference might at first seem to be a beneficial process, the effects of transference can be positive or negative. Positive transference occurs when users’ performance is amplified as their previous experience with similar systems assists them in learning and using the new system. Users encountering a new system may find, however, that some system features appear familiar but in fact operate differently than previous experience would predict. In this case, negative transference—also known as interference—may occur as the users’ performance is attenuated.

Many researchers see consistent user interfaces as a means toward positive transference. Nielsen (1989), for example, notes that consistent user interfaces can lead to the transfer of skills from one application to another, increasing both productivity and user expectations of being able to master new applications. By the same reasoning, one could conclude that inconsistency across interfaces can lead to negative transference, decreasing productivity as users attempt to employ features according to their previous experiences and find themselves needing to correct actions and learn different methods of performing tasks. Such users may also become frustrated quickly when their expectations of mastering the new system easily are thwarted.

Empirical research provides support for the effects of positive and negative transference. For example, Polson, Bovair, and Kieras (1987) demonstrated positive transfer effects when subjects were taught to use one text editor and then subsequently were taught to use a similar one. Foltz, Davies, Polson, and Kieras (1988) examined differences among users of different versions of a word processor, finding that subjects encountering commands with identical effects, yet employing different terms, were not able to transfer their previous experiences positively to the new commands. And Ozok and Salvendy (2000)
Silver et al.  

Browser-Based Applications: Positive or Negative Transference?

found that subjects committed more errors when web sites exhibited inconsistent interfaces, suggesting negative transference is at play.

Research on transference is directly applicable to browser-based applications. BBAs in general and especially BBAs within a given class of applications have so many features in common that one expects to find some degree of positive transference. But even within a particular class of BBAs, or even focusing on a particular BBA feature, BBAs are far from homogeneous. Comparing one BBA with another turns up great inconsistencies in both interface and functionality. Moving from one BBA to another provides numerous opportunities for errors to occur as users’ expectations are not met and as functionality fails to perform as anticipated. Ironically, this interference is in part a consequence of the many similarities across BBAs. The similarities promote users’ expectations about how to interact with a new BBA, inviting them to draw on their experience and engage in their accustomed behavior when interacting with the new BBA. And that behavior allows the inconsistencies across BBAs to snare the user, leading to inconvenience, errors, and frustration.

The sections that follow analyze specific BBA features that might have been sources of positive transference but that generate interference instead.

APPLICATION-INDEPENDENT FEATURES

Many BBAs have features in common, even if the BBAs belong to different classes of applications. Some of these application-independent features are actually browser features—such as toolbar buttons—that affect the BBA’s operations. Others are features such as internal search and login/logout that are accessed through links within the BBA’s pages. In both cases, inconsistencies across BBAs can interfere with people’s efficient and effective use of these features.

The “Back” Button

BBAs differ in how they handle the “Back” toolbar button. Some BBAs require it, some forbid it, and some do some of each. Misusing the “Back” toolbar button in a transaction sequence can be disastrous, possibly leading to duplicate transactions, failed transactions, or lost information. For example, the Bureau of Public Debt issues the following warning on its Treasury Direct web site to those buying government notes and bills:

WARNING: If you fail to follow the following directions, you can seemingly conduct a transaction—and even receive a confirmation number—when, in fact, no transaction was processed ...

1. Don't use your browser's "Back" or "Forward" buttons. Rather, use the navigation buttons near the bottom of each page.

Since other BBAs instruct their customers to use the “Back” button, interference from those systems is likely to lead to failed transactions for some of TreasuryDirect’s customers.

Printable Versions

As noted earlier, BBAs differ in how they manage printable versions. Some BBAs open the printable version in a new browser window, while others use the current window. A person accustomed to closing the new printable version window after printing is likely to do the same in BBAs that do not open new windows. But closing the printable version when it is not in a new window exits the browser, aborting the application. Interference does not get much worse than that.

The “Enter” Key

BBAs differ in how they treat the "Enter" key when a person is completing a form. Some BBAs treat “Enter” as a signal to post the form; others do not. These two approaches are likely to interfere one with another. If a person becomes accustomed to pressing the "Enter" key with no effect, he or she is likely to press it mistakenly in BBAs where it does have an effect, prematurely posting a partially completed form. At best, the user is inconvenienced by the delays and discontinuities associated with having the page returned for completion. At worst, the BBA acts on the form as though it were complete, possibly causing all sorts of problems for the user.

Finding Features

Some features are common to many BBAs—for instance, internal search capabilities and privacy policies—but how one finds the links to these features varies greatly from one BBA to another. Differences across BBAs in such attributes as link location on the page, link identification (how the link is labeled), and link salience can cause trouble for people as they go from one BBA to another. Someone accustomed to clicking in a particular spot for a particular feature may have trouble
locating that feature, or may even mistakenly click on a different feature, in BBAs that implement the link differently. Similarly, users accustomed to a particular visual representation of the feature—say, a brightly colored button—may have trouble locating it when it appears in other forms. If the feature is not directly accessible from the homepage, users may expend much effort searching the site for it. And since some BBAs may not implement a given feature at all, the user may spend time and effort searching in vain. Conversely, if users do not find a feature, they cannot be sure whether they missed it or whether it simply is not there.

Logout is an important special case of this problem. Many web sites, especially those of such financial institutions as banks, credit card providers, insurance companies, and investment firms, provide a logout button on their pages. But the button goes by various names and is positioned in varying locations on the page. So a person who uses many financial BBAs, despite being very familiar with such applications, must still search for this important button in any given application. Some BBAs help the user by making the logout button salient—coloring it differently than other buttons or making it larger. Ironically, while improving the given site’s usability, this approach increases the interference when the user accustomed to this salient button works at other sites that do not make the logout button salient in the same way.

Going Home

Many sites have a clickable logo in the upper left corner of every page that takes users directly to the homepage. But some companies—Amazon.com, for instance—put something else clickable in the corner and move the logo a bit to the right. Users trying to go home may select this other feature—perhaps a link to a featured product or an advertisement for another site—by mistake. Such a design choice might simply reflect a desire to put the feature in a prominent place, or it might be an attempt to exploit the interference effect to trick users into selecting this feature. Either way, it degrades usability for the person who is trying to reach the homepage but accidentally ends up elsewhere.

Search Boxes

Many BBAs have a search box on their homepages. Some sites have other fill-in boxes on the homepage, instead of, or in addition to, the search box. Some BBAs, for instance, invite customers to supply their e-mail addresses for the company’s mailing list. Bloomingdale’s has a fill-in box to locate a couple in the bridal registry. But given that search boxes are much more common than these other fill-in boxes, users may mistakenly type a search phrase into these boxes intended for other purposes.

FEATURES SPECIFIC TO SHOPPING-CART APPLICATIONS

Thousands of retail e-commerce web sites are designed as shopping-cart BBAs. While these applications all have much in common—the add to cart, view cart, checkout, and other features—one finds numerous differences in how these features are implemented. Some of these inconsistencies may seem small, but since most on-line consumers use many such sites, the potential for interference is great. The consequences of such interference can be confusion, errors, and lost time for the consumer as well as lost sales for the site. Even applications built using the same development system—for example, the many on-line stores implemented through Yahoo! Merchant Solutions—have significant differences from one to another. Here are some of the more significant inconsistencies.

Counting Items

Many shopping-cart sites show the count of items currently in the cart somewhere on each catalog page. But different sites count items differently. Some BBAs include the number of distinct products while others count the total number of items, including multiples of some products. Consumers who uses both types of sites are likely at some point to interpret the item count incorrectly, leading them to fear that the cart does not contain what they put in it.

Keep Shopping

Shopping-cart sites differ in where they take users who click on the “Keep Shopping” button when viewing the shopping cart’s contents. Some BBAs take users to the homepage, some take them to a master catalog page, some to the most recently viewed catalog page, and some to whatever catalog page the server deems most appropriate given the path traversed. If a catalog is sufficiently complex, taking a customer to a different place than he or she expects may lead that customer to getting lost in the catalog.
Update Quantity

Interference aside, a major usability problem for shopping-cart BBAs is the mechanism customers use to change the quantities ordered when viewing their shopping carts’ contents (Silver, 2003). If a customer simply changes the value contained in a fill-in box seen in the browser, the browser may continue to display out-of-date totals and the server and browser may not agree on what is in the cart.

Most shopping-cart sites work around these problems by requiring customers to press an “Update Quantity” button after making any changes. But this solution still degrades usability. Each time users press the button they suffer through a delay and discontinuity while the revised web page is received from the server and rendered. If users forget to press the button there can be a discrepancy between what the user and the server think is in the cart. Until the page is updated, any subtotals and totals on the page will be incorrect. The BBA can be coded to update automatically if the customer terminates his or her entry with the “Enter” key, but this leads to even more delays and discontinuities if the person has many items in the cart to be updated.

Because shopping-cart sites differ in how they work-around the update quantity problem, this feature can also be a source of interference:

- If a customer proceeds to checkout without pressing the “Update Quantity” button, some sites process the change; others ignore it. Negative transference from the BBAs that update automatically will likely lead some people to fail to update when required.
- Similarly, some sites automatically update when customers terminate entries with the “Enter” key; others do not.
- The “Update Quantity” button looks different, and is located in different places, at different sites, so a customer might not notice the button at some sites and consequently not realize it needs to be clicked.
- Some BBAs place an “Update” button labeled in a very small font directly beneath the quantity fill-in box. While this is a logical place for the feature, other BBAs place a “Remove” or “Delete” button in this location. A user accustomed to clicking beneath the quantity after changing it could easily delete items by mistake.
- Some sites work around the update problem by not allowing users to enter numeric values directly into the form. For instance, Americannutrition.com has “increase quantity” and “decrease quantity” buttons, instead of a fill-in box. Macy’s has a drop-down box that allows customers to select a quantity from a list and automatically triggers a posting to the server. These sites solve some—not all—of the usability problems associated with changing quantities but introduce even more inconsistencies across shopping-cart sites, opening the door to even more interference. In particular, neither of these sites even has an “Update Quantity” button, so frequent users of these sites are likely to become accustomed to not looking for or clicking such a button.

Hard to Find Features and Information

Users of shopping-cart sites expect to find certain standard information within the web site, including such items as shipping rates, return policies, privacy policies, order status, and how to locate a physical store. But not every shopping-cart site implements all of these features and one finds dramatic differences in terms of where these features are located in sites and how they are identified. So the “Finding Features” problem described previously for all BBAs is especially severe for shopping-cart applications. For instance, order status might be at the top of the page, the left side, or the bottom. It might blend in as part of a menu or might be made to stand out in some way. It might be located on a subordinate page such as “My Account.” Or it might not be there at all. Consider also shipping rates and policies, which are typically of great interest to many on-line shoppers. Some sites have a link from the homepage to shipping rates and policies, some include such links on a subordinate page (which page it is varies by site), and others do not make this information available until the consumer is well into the checkout process. The latter case is especially frustrating for users since until they check out they are searching for information they think is available based on their experience with other sites but that is not. The wasted time and effort by the user may translate into lost sales for the vendor.

Unit Prices and Extended Prices

One would think that distinguishing the unit price from the extended price (unit price times quantity) would be a simple matter. But shopping-cart displays are surprisingly diverse in how they treat price data. Some carts have columns for both the unit price and the extended price, while others have a column for one or the other, but not both. Moreover, some sites use the column heading “Price” for the unit price and others use it for the extended price. Inconsistent labeling across sites can
lead to interference. Someone accustomed to looking at the “Price” column for the unit price might easily confuse the unit price with the extended price on a site such as Barnes and Noble that has only an extended price column and labels it “Price.”

**SOME PRELIMINARY DATA**

While the preceding analysis makes the case that inconsistencies across BBAs pose usability risks, empirical studies are required to determine the extent of such inconsistencies and their consequences for usability. As a first step in this direction, we examined the “logout” feature in 25 BBAs—a convenience sample of widely used sites in various domains including financial services, travel, and telecommunications—finding that the location and appearance of this feature varies substantially. Twenty-two of these BBAs place logout in the top portion of the page; the remaining three put it in a panel on the left side. More notably, the twenty-two show substantial variability in its location (see Table 1). The logout feature is most frequently found in these BBAs on the right side, near—but not at—the very top, but only about half put it there (12 of 22). The others place it in various positions within the top of the display. Horizontally, the positions range from the left to the center to the right of the display. Vertically, the positions range from the left to the center to the right of the display.

All told, the data suggest that people who employ many BBAs must often search carefully across a broad section of the display to find the logout button.

<table>
<thead>
<tr>
<th>Location</th>
<th>Left</th>
<th>Center</th>
<th>Right</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extreme Top</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Near Top</td>
<td>2</td>
<td>4</td>
<td>12</td>
<td>18</td>
</tr>
<tr>
<td>Total</td>
<td>3</td>
<td>6</td>
<td>13</td>
<td>22</td>
</tr>
</tbody>
</table>

**Table 1. Location of Logout within the Top Portion of the Page**

The 25 BBAs also vary in terms of how sharply the logout feature stands out. Nearly half the BBAs (12 of 25) make the logout feature salient, but the techniques they employ vary substantially. Bank of America—among others—distinguishes the logout feature with a different text color, while institutions such as Wachovia use a different background color. And the distinguishing colors vary from BBA to BBA. Still other sites—such as American Express—use empty space on the page to set the logout button apart spatially.

Sixteen of the twenty-five BBAs (64%) include the logout feature in a menu among other features. While it is unclear whether being embedded in a menu makes this feature easier or more difficult to find in a given BBA, the inconsistency across BBAs (roughly a 2/3 to 1/3 split) is likely to be problematic for users trying to logout.

The labeling of the logout feature also varies widely across BBAs (Table 2). The differences in labeling between “Log out” and “Logoff” or even “LOGOUT” and “Sign off” may be slight enough not to produce much interference by themselves. But given significant differences in the buttons’ locations (on what are often cluttered web pages), given that some are made salient while others are not, and given that some are in menus while others are not, the added burden of not even knowing what text to look for adds appreciably to the difficulty of searching for this feature.

<table>
<thead>
<tr>
<th>Label</th>
<th># of BBAs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign Out</td>
<td>4</td>
</tr>
<tr>
<td>Sign Off</td>
<td>2</td>
</tr>
<tr>
<td>Sign Off Account Center</td>
<td>1</td>
</tr>
<tr>
<td>Log Out</td>
<td>10</td>
</tr>
<tr>
<td>Log Off</td>
<td>4</td>
</tr>
<tr>
<td>LOG OUT</td>
<td>2</td>
</tr>
<tr>
<td>LOG OFF</td>
<td>1</td>
</tr>
<tr>
<td>Exit Account</td>
<td>1</td>
</tr>
</tbody>
</table>

**Table 2. Labeling the “Logout” Feature**

Two cases merit special mention. Oxford Health makes its “Exit” button extremely prominent by isolating it in the upper-left corner, separated from the rest of the page by empty space. Ironically, this approach may introduce even greater interference...
since the overwhelming majority of BBAs use that location for a clickable logo that takes users to the home page. Verizon—a company that wants to be its customers’ sole provider of telecommunications services—has separate sites for its Internet, wireless, and local phone service businesses. Across the three BBAs one finds evidence of every inconsistency documented in this section.

The degree of inconsistency within this sample of BBAs strengthens the argument that the potential for negative transference across BBAs merits empirical study. In particular, these data suggest that “logout” is one of the features worth studying.

CONCLUSION

The availability of so many browser-based applications that have so many features in common and yet have so many inconsistencies in how those features are implemented sets up a classic situation for negative transference. Because of the many similarities among BBAs, each BBA seems comfortable and familiar to the user. But as that user transfers his or her behavior from one BBA to another, even small differences can become big problems. Links to features are not where they are supposed to be or the features do not operate as expected. The consequences are the ones we have come to expect from degraded usability: user errors and wasted time. So, while BBA commonalities certainly afford some benefits, the substantial inconsistencies across even similar BBAs impose significant costs.

A prima facie case has been made here for studying empirically negative transference across BBAs. For each BBA feature to be studied, the research has three stages:

1. A template or schema must be created for characterizing the feature’s implementation in a given BBA and for contrasting implementations across BBAs. For instance, the schema for “logout” would include such elements as its location on the page, how it is labeled, and whether it is contained in a menu.
2. The schema must be used to describe and contrast an appropriate set of widely used BBAs to determine the nature and extent of inconsistencies across BBAs.
3. Behavioral experiments are required to determine if, how, and to what extent the identified inconsistencies lead to interference that degrades usability.

In the likely event that empirical studies find significant interference, the next issue is how to reduce such interference. Establishing BBA standards for various common features would likely decrease inconsistencies and increase BBA usability. Of course, establishing standards poses difficulties of its own:

1. Some industry forces would favor establishing standards while others would oppose it. Those companies that benefit from differentiation would likely resist such efforts. Some companies differentiate by offering superior features—for example, Amazon’s patented one-click approach—and others simply differentiate as part of their branding strategy.
2. Some differences across BBAs that reflect differences in the product being sold or differences in the nature of the application (for instance, e-mail versus e-trading) may make sense even if they lead to interference.
3. Differences in firm strategy may lead to differences across sites that would otherwise be more similar. One finds great variance across credit card companies when making an electronic payment. In part these differences reflect some companies’ attempts to minimize the amount of debt their customers pay off.

Personalization is another approach to reducing interference across BBAs. If a given feature can be implemented several different ways, a BBA could support all of them, allowing the user to select his or her preferred method. For example, some shoppers might prefer to return to the main catalog when they continue shopping, while others might prefer to continue where they left off. Of course, this does introduce the overhead of personalizing each BBA one invokes. But personalization need not be an alternative to standardization. We could standardize our approach to personalization, carrying in our electronic wallets information that could personalize BBAs automatically.

Given the pervasive position of BBAs in today’s world of Internet Applications, we cannot afford the negative transference and concomitant usability losses that follow from the substantial inconsistencies across BBAs. To date this issue has been largely ignored. It is time to study this phenomenon more formally and to act upon what we learn.

REFERENCES