Jumping Through Java

December 10, 2009

Originally published February 7, 1997 in Web Review magazine.

Since our cave dwelling days, humans have felt an overpowering need to employ new technologies and will jump through hoops of fire to do so. Imagine the scene when man discovered the rock. People immediately created all sorts of applications including rock pillows, rock beach balls, and rock candy. After a few years of painful experimentation, people realized that rocks were good for two things: building walls and throwing at other people. Thus the beginning of modern warfare.

A survey of today's Internet environment demonstrates that people haven't changed much in the past few hundred thousand years. We're still at the mercy of new technologies.

Last year, frames were the big thing. After a frantic period of horrendous Frames-related experimentation, we're finally beginning to see a few elegant uses of the technology and lots of sensible choices not to use Frames. Even **Netscape** removed their Frames-based navigation system.

Java is the reigning technology du jour. The software has become so stable that applets sometimes don't cause the browser to crash. Many users now have Java-capable browsers. Multitudes of newly minted Java application developers stand ready to create dynamic, interactive applets. The Java wave has begun. How can a warm blooded Web site designer resist? After all, everyone else is doing it.

Well, there are two major problems with current implementations. First, people are building Java applications upon poorly designed information architectures. Second, they are making some major interface design blunders. The results are often quite horrendous. Let's review one of the major culprits.

Sun's Hyperactive Home Page

The **main page** (Java version) of the Sun Microsystems Web site is overwhelming. Links of all shapes and colors fill the screen. Some are black, some are blue. Some are static, some scroll, and some pop-up. To be exact, there are 47 static links, 46 pop-up links, and 12 scrolling links. The complexity of the page induces a hypnotizing deer-in-the-headlights state upon unsuspecting users.

By example, Sun has clearly demonstrated how Java can be employed to radically increase the link density of Web pages. This is not necessarily something to be proud of. When we architect Web sites, we strive to **reduce** the numbers of links on the main page. We try to organize information into a manageable number of content areas. The last thing we want to do is turn our users into candidates for roadkill.

In the traditional design world, it has often been said that form should follow function. Well, in the world of Web site development, interface design should follow information architecture. Even the best designed pop-up menus won't make up for a weak architecture.

In Sun's case, the content has simply not been divided or "chunked" in a way that is meaningful and manageable for users. There are eight main categories that merit their own pop-up menus. The distinctions between some of these such as *Products and Solutions* and *Sales and Service* and *Technology and Research* are somewhat blurry. There are six secondary categories such as *Sun on The Net*, *Past Stories*, and *Buy* which seem to have nothing in common other than not being quite as important as the major categories. There are two different kinds of news: *Hot Topics* and *News in Brief*, which should be combined to reduce user confusion. And there is a search interface with no explanation of what you are searching or how you can search. With such a confusing architecture, it's no wonder that the interface overwhelms.

To make things worse, Sun has implemented three versions of the home page: a Java version, a non-Java graphics version, and a text-only version. Not only does this add to the clutter on the main page, but also introduces two of the major problems with using Java. First, many users don't have Java-capable browsers. The audience is divided into the Java haves and have-nots. Second, there is a substantial impact on development cost. The fact that Sun's own Web site only employs Java on the main page and not throughout the site speaks volumes about the cost and complexity of integrating Java.

A Time and a Place

I'm not such a technophobe that I can't see the obvious strength of Java. As Bill Gates would say, it's cool! There are a few sites that use it well. I really like CNET's **news** applet and the SIGGRAPH 96 conference **billboard** applet. Both present dynamic and useful information in an elegant, non-intrusive manner.

However, my favorite Java-enhanced site is the **American Museum of Natural History**. First of all, the site is built with a simple yet solid information architecture. Five main options at the bottom of the page allow the user to find important information quickly and easily. As far as coolness factor goes, the colorful graphics and animation convey a sense of fun and the interactive pop-up descriptions communicate a metaphor of exploration. These work together to create just the right image for a museum.

So, I really don't hate Java. I just think that 95% of Java-enabled Web sites are interface blunders built on top of architecture disasters.