So, What Do You Want To See?

Designing the User Interface for "Ships and Aircraft of the U. S. Navy" An Interactive Multi-media Database Display

John Loven
Skunkwork Multimedia Inc.
Trappe, Pennsylvania

Abstract

Producing the "Ships and Aircraft of the U.S. Navy" interactive multi-media database display for the U.S. Navy Memorial Visitor Center raised interesting design problems:

How should over 3,000 historical photographs and accompanying data be classified?
How should these classifications be developed to anticipate the needs of a very general public audience?
How will the classification system be executed in software?

A careful audience analysis and a very flexible software approach ultimately yielded a successful design.

Introduction

At first glance our assignment seemed straight-forward: Put 3,500 historical images of ships and aircraft, with data about them, into a kiosk so visitors to the U.S. Navy Memorial Visitor Center can brows through them.

But when the first, and most obvious question was asked...

"Who are the browsers, and what do they want to see?"

... a significant design challenge emerged.

The Memorial staff had primarily envisioned the proud Navy veteran, going to the kiosk to show his or her progeny "My Ship". These veterans would be conversant with the Navy's venerable type-class-hull number identification system for ships and would feel quite at home with the pictures.

However, as parts of the memorial opened to the public, it became clear that a major component of the visitors were school children with no real experience of anything military, much less sea-faring.

How might we make this wonderful historical collection equally accessible, equally pleasing and informative to these two extremely different populations, not to mention everybody in between?

We tried to step back and derive some useful design principles for database displays in general which would apply to "Ships and Aircraft."

1. Formal / Informal / Spontaneous Inquiries

It seemed to us that a very diverse audience might come to a display with three kinds of inquiries described as follows:

Formal inquiries come from those users conversant with a classification system established along Aristotelian lines: Order, Genus, Species, etc. In our case these individuals would ask to see the "Iowa Class Battleship BB62 New Jersey."

Informal inquiries come from those users with a functional relationship to the data. These users might just want to see the currently active battleships. It's similar to the inquiry above, but related to time and function more than the official classification system.

Spontaneous inquiries come from the most naive users. Having little or no personal experience of the subject matter, they formulate an inquiry based on some personal criteria of meaning or emotion. "Show me the ones with the biggest guns!" is a typical spontaneous inquiry.

It is worth noting that the above inquiries become broader as they become less formal, but they do overlap. The formal inquiry nets pictures of the "New Jersey." The informal inquiry gets pictures of the "New Jersey", the "Iowa" and the "Wisconsin". The spontaneous inquiry gets the three Iowa class battleships, plus several earlier battleships and some heavy cruisers.

A useful database display must allow for formal, informal, and spontaneous inquiries, without making any of the three seem like a less desirable or intelligent approach.

2. Loven's Theorem of Data Dynamics

A. In a database of given size and power, there is a constant amount of data/time.

B. Data/time has two manifestations: Make-it time and Take-it time.

ERGO: The more time you spend putting in the data and structuring the database, the less time it takes to get something useful out, and conversely, the faster the database is thrown together, the longer it will take to retrieve anything interesting.

This is a silly formalization of a principle that everybody knows instinctively, but which is too often disregarded in practice. The general population has made a startling leap in attitudes about the speed of computer response time. Just 10 years ago, if a computer could pull up a document in less time than it took to go to the library and find the book, it was a miracle of technology. Today, the buzz word is "think speed." If the computer doesn't keep up with one's spontaneous intentions, it is obviously obsolete.

A useful database display must combine clever anticipation of the user's intention with great craft and technological skill in building the software, to maximize response time as far as the hardware will allow.

3. Redundancy is Not a Naughty Word.

Computer programmers have a deep, perhaps genetically rooted, desire to write programs that provide just one way to do each thing that the program does. They were surely shamed in their computer science classes if their code became convoluted or - heaven forfend - redundant.

Non-redundant software, however, places a burden on the user: one must learn how to work the software before using it. And that, as we are all aware, is unacceptable in the museum or exhibit environment. The user will only spend a few minutes at a display, and that time must be spent using the display, not learning it.

The more ways there to do a thing, the higher the chances that the user will intuitively take a satisfactory course, and be rewarded with an interesting experience.

It is useful to note that the user will never know that there were several ways to see the pictures of the battleship "New Jersey." People who say "Won't it be confusing if there are several ways to get to the picture?" have lost track of the actual user's experience. They know too much about the program. The user leaves saying "Hey! I did what seemed logical to me and it worked. What a nice program."

With these principles in mind, we designed the user interface for the program.

In the case of the ships program we user 6 major choices on the main menu:

1. Look at ships by ship name
2. Look at ships by hull number
3. Look at ships by type and class
4. Look at ships by time period
5. Look at ships by mission
6. Look at extraordinary ships

Choices 1, 2, and 3 reflect the formal inquiry scheme, although #1, ship names, also addresses some informal of spontaneous inquiries because some ship names, the Enterprise, the Wisconsin, the Pueblo, may be current in pop culture for various reasons.

Choices 4 and 5 address informal inquiries. Many users want to see a general category, like submarines or ships of the Civil War.

Choice 6 tries to anticipate spontaneous inquiries, especially those that children often make: the biggest, the smallest, the fastest, the oldest, the biggest guns, ships that save lives, and ships that are explorers are available as browsing categories.

The aircraft program is similar in structure offering:

1. Aircraft by nickname
2. Aircraft by designator
3. Aircraft by mission
4. Aircraft by time period
5. Extraordinary aircraft

The concept of redundancy is realized in the menuing scheme. For example, the battleship New Jersey can be found in at least seven ways:

1. By name: New Jersey
2. By hull number: 62
3. By Type and class: Battleship - Iowa class
4. By looking for its mission: Battleship
5. By time period: 1945-present
6. By looking at the list of biggest ships
7. By looking at the ships with the biggest guns.

From choice 4 (mission) on, the New Jersey appears as one of a list of ships. At the extremes, there are 3 Iowa class battleships and many hundreds of ships commissioned since 1945.

The reflection of our design in software hinges on the need for the user to perceive quick system response. It also reflects the understanding that users expect to wait a bit at some points, and expect instant response at others. At one extreme of this proposition, we believe that when a major menu change takes place, the user pauses to re-orient visually, so the system can do some work without the user noticing. At the other extreme, once the user is scrolling down a list or flipping through pictures, any delay is bothersome.

We therefore invoked the theorem of data dynamics in our design. There is, of course, a master database, containing all data. The program itself, however, accesses dozens of smaller databases, derived from the master, which reflect the structure of the inquiry mode selected by the user. When the database is updated, the new master database is fed to the

system, which then takes almost 25 minutes of processing time to generate and index all the sub databases.

This large amount of Make-it time is handsomely rewarded in short Take-it time. When a user asks to see the ships with the biggest guns, there is already a stand-alone database of just those ships, which makes flipping through it seem instantly responsive.

Especially given the relatively low horse power of the computer, the response is very satisfactory.

For those interested in the technical details, the program runs on a Sony Viw 5000 system, which includes a 10mz 286 AT type computer. We wrote the program in C, the Master database was entered using DBase, and the utilities which generate the sub-databases is created in FoxPro.

The program currently holds 3600 images, and we look forward to updating and expanding the collection over several years.

