A Global Forum for Naval Historical Scholarship

International Journal of Naval History

Volume 1 Number 1 April 2002

Creation of a Web-Enabled Naval Operations Database

(Or, How I Stopped Worrying and Learned to Love Object–Oriented Design Methods)

Jon Parshall, is.com.

Abstract

The World Wide Web is changing the way the way in which historical information is being presented to the world community. Never before has there been such convenience in being able to turn up information on practically any historical topic almost at the touch of a button. The result has been a faster dissemination of information to a wider, more diverse audience than ever. However, this author suggests that the majority of historical information being presented on Web sites is hardly representative of a "new" media. In fact, most historical sites are simply replicating (badly) the standard hardcopy reference works already available.

The author poses these questions in the context of a larger discussion focused on the design for a new web–enabled naval operations database. This database, while still in the development stage, illustrates techniques for using technology to present standard historical information such as technical characteristics of vessels and Tabular Records of Movement (TROMs) quickly and easily to Web users. More important, the application lends itself well to answering complex naval operational questions, thereby creating "new" information from a synthesis of several "old" data types. The design techniques behind the database are explained in detail, as well as the unanticipated uses of the application, its strengths, and its potential drawbacks.

Project Background

I am the proud owner of The Imperial Japanese Navy Homepage (http://www.combinedfleet.com). Since its establishment in 1995, it has arguably been THE site on the web for finding information pertaining to the Imperial Navy in WWII. It has won numerous awards, including being a Yahoo "Cool Site of the Day" in 1996, has been singled out for notice by several other search engines, and has garnered the attention of amateurs and recognized historians the world round. I owe much of my current activity in the field of naval history to having had the foresight (and luck) to establish it during the infancy of the Internet.

I hate my web site.

Let's just get that out on the table. Did I mention that it's been in operation since 1995? Well, believe me, it looks it. My site is a poster child for everything that was wrong, square, and clichéd in early web design. Its user interface is clunky, it has too little fresh content, the formatting of the text on the pages is not optimized for easy reading, and it is inadequately dense in terms of hyper–linkages within the site itself. In other words, it no longer does what I want it to do. Unfortunately, I am too busy with "real" projects these days to update it, because the body of 400+ HTML pages was originally completely hand–crafted, one page at a time.

This problem is endemic to first generation web sites everywhere, because it derives from a fundamental fact inherent to all hand–crafted HTML sites—while HTML is easy enough to create initially, it is an utter pain to maintain or change. The truth of that statement is all around us in the countless web sites that are stale and never updated because it is simply too hard for the average Webmaster to wade through a sea of HTML to change the appearance and/or substance of the content. In my case, changing the look and feel of the entire site (for example, changing the bottom navigation bar) would require touching each page individually. Life's too short.

I first become aware of this phenomenon in 1997, when I took a job as the marketing manager for a web software development company. Even then, the answer to the hand–crafted HTML trap was evident—Web sites should be driven by a database. Databases can be used to store all manner of Web content—text, data, and even images. By mating individual chunks of content from the database with a small number of standardized HTML style sheets, one can essentially automate the delivery of web content to viewers. This eases maintenance dramatically, since changes to content can be handled directly via the database, rather than opening up an individual page and searching through the formatting and text. Likewise, changes in the way in which content is displayed (i.e. layout and design) can be managed through a discrete number of content templates. This separation of data from design is crucial to the keeping a site fresh.

I began to ponder whether or not a database might have relevance to my site as well. At first glance, the application seemed obvious—build a small system to house data on the various classes of ships in the Imperial Navy and use it to drive the individual ship class pages, since this data comprises the bulk of the content on the site. I set about trying to sketch out how the database model might look, using a spreadsheet. It was then that a fundamental problem emerged, and with it an opportunity to do something unique on the Web.

Go to any naval reference Web page on the Internet—mine, www.warships1.com, www.hazegray.org; any of them—and one will see ship data presented in roughly the same format from site to site:

Ship Name	Ship Class	Ship Type	Laid Down	Launched	Completed	Builder	Blah Blah
Yamato	Yamato	Battleship	11/4/1937	8/8/1940	12/16/1941	Kure DY	blah blah blah…

In other words, ships are represented much as one might envision spreadsheet data: one ship = one row of data. Why is this so? Because every one of us Web masters back in the mid– to late–90’s did the same thing when our sites were in their infancy—we opened up our favorite reference books and began transcribing data into HTML. This is the way the data in our reference books was laid out, of course, since it’s the only way one **can** reasonably lay out such a book. Copying this general format was simple: simple to lay out in HTML, simple to input, and simple for the reader to understand.

It’s also a dumb way to do it.

The reason this approach is inherently dumb is because it does not address a fundamental aspect of how ships behave. Any naval historian knows that most ships change over time, sometimes dramatically. If someone asks for the configuration of the *Yamato*, the answer **cannot** be given without also knowing the timeframe the user is interested in. *Yamato*, like practically all warships during the Second World War, changed almost every time she entered a naval dockyard. She received radar, her anti–aircraft armament was increased, her crew expanded, and so on. These periodic modifications continued until she was finally sunk in April 1945. The same general process—refit, repair, upgrade—was replicated aboard practically every ship in every navy during the Second World War. Unless one can represent this fundamental facet of “shipness”—the fact that ships change over time—then the only thing one has accomplished by building a database–driven ship database is spoon–feeding the same old data to people who are too lazy to make a trip to the library. Don’t get me wrong; I think that convenience of information is a moderately worthwhile goal. All things being equal, I’d prefer the masses know *something* about the Japanese Navy rather than *nothing*. But the more I thought about it, the more dissatisfied I was with the prospect of building a database to serve up the same old static data.

Here I will digress for a moment, but in my opinion this points to a fundamental flaw of the current state of history on the Web. There are still too many people out there blindly typing away so as to give us all the same data in the same format that we’ve always had, but in a “new” media. Unfortunately, copying books into HTML has two fundamental flaws (beyond trifling issues of copyright). First, this “new” media stinks for serious reading. Computer screens still aren’t a comfortable place to sit and read long tracts of text and data, which is what reference books are all about. Rote copying of books to the Web does nothing except promote the ophthalmologic industry. And indeed, most usability studies have revealed that if someone is really serious about reading the material on a web site, they simply print it out and read it offline. Yet most Webmasters don’t design with this in mind.

Second, from a content density standpoint, book copying efforts are fundamentally doomed to failure. Until practical Intelligent Character Recognition (ICR) technology (preferably for foreign languages as well) becomes A) widespread, B) cheap, and is C) combined with high–speed scanning capabilities (to allow for scanning of large amounts of textual data), no web site will ever be able to replicate the sheer volume of high quality content that is available now in hardcopy. For instance, at present can one anticipate using the Web to deliver a complete corpus on the state of World War II battleship technology, *à la* Dulin and Garzke’s three–volume treatise on the subject?[1] Anyone who attempts to do so with the tools currently at hand is probably certifiable. Yet we Webmasters continue typing madly away, convinced that more is better.

Collectively, we have failed to step back and ask ourselves four fundamental questions:

- What is the overall purpose of putting (Data X) on the Web?
- To whom is the data to be presented?
- How is the data to be consumed?
- Are there any technology tools available that would allow us to present the data in new, and perhaps more useful ways?

Much as it pains me to diverge into business buzzword speak, it is clear that historians need to do a little more "thinking outside the box" before they start designing their sites. If we can't think of more compelling ways to present information on the Web, then it really isn't a "new" media after all, is it?

Back to *Yamato*. The more I pondered, the more I became convinced that a truly satisfactory ship database would have to address the temporal element. Doing so would not only create a more useful representation of the vessel itself, but would also open up possibilities for operational research that had never been feasible before. While it is true that 95% of the audience of my site would be satisfied merely to know what *Yamato*'s armament was on date X, I also want to be able to cater to a more serious type of researcher (myself included). People like me want to know things about how and where the Imperial Navy's ships actually operated, how the Navy behaved as an organization, and what the effects of the conflict were on the structure and operations of the Navy. For instance, a hypothetical naval historian interested in, say, the Imperial Navy's anti–submarine warfare operations might reasonably want answers to any of the following questions:

- "How many destroyers did the Japanese Navy lose to submarines from January through June, 1943?"
- "What was the aggregate tonnage of escort warships launched by the Imperial Navy during 1944?"
- "What were the names and graduating class of every destroyer skipper in the Imperial Navy in May, 1942?"
- "What were the relative operational tempos of the Imperial Navy's destroyers in 1942 compared with 1944?"

With the standard reference materials at hand today, any of these questions could probably be answered. However, doing so would require pawing through reams of books or old *Maru Special* back–issues, hand tabulating the data, and then crunching out the answers into a spreadsheet. This makes the prospect of answering such questions almost more trouble than it's worth. However, with a database that stores the needed information **and** addresses time, such reports could be generated with relative ease.

Unfortunately, explicitly including time in the application design also complicated the representation of a ship immensely. As mentioned, a static representation of "ship" can be conceived of in terms of a single row of data in a spreadsheet. Not so a temporal or "dynamic" representation, which must necessarily be able to be assembled into a "snapshot" of the ship that corresponds with her physical status on any given date. No more "single table" databases here—there would need to be a fairly complex data schema underlying this thing. Initially, I was trepidatious about taking this road, but the more I thought it over, the more inadequate and pointless the alternative appeared. Fortunately, taking the first step down this slippery slope in terms of understanding how to "model" a ship through time actually turned out to be the key to reaching a more satisfactory solution to the larger problem of modeling naval operations as a whole.

Enter The Geeks

Here's where things briefly get a little more technical. I am fortunate enough to work at a Web development shop filled with all manner of helpful technical types and application development tools. One of these tools (Rational Software's *Rational Rose*) was designed to do object–oriented modeling work. "Object Oriented" (often referred to simply as "OO") is a way of organizing application code and the data that is associated with it. There are several coding languages available which utilize some or all of the tenets of OO, Java being the most popular and widely available. In an OO application, real world items (such as ships) are manifested in individual code "objects", i.e., a "*Yamato*" object, and a "*Musashi*" object, each of which is an "instance" of the "Yamato–class" parent object. Each individual object carries along with it the knowledge of all of its own individual attributes (guns, armor, speed, etc.) and the ability to change that data through various "methods". Programmers use object–oriented techniques because they are designed to mimic real–world phenomena, making the problem space describable in plain English, and thereby making the application coding easier as well. Indeed, at its most basic level, object–oriented modeling is primarily concerned with discovering the "nouns" and "verbs" of a particular problem space.

"Huh?" you say. "Nouns and verbs?"

Indeed. For instance, what are the primary nouns that would compose a study of naval operations? Well, "Ship" is a good place to start, obviously, so you probably need an basic object called "Ship". Not only that, but it turns out that most of the physical things that compose a ship are other nouns, too. For instance, what do ships have aboard them? "Guns", "Armor", "Engines", etc. All of these sorts of items are just components of the larger object, the Ship, and each of them can become sub–objects in their own right. Physical dimensions (such as the bore of the guns, the shaft horsepower of the engines, or the length of the ship) can either become attributes of the component objects or of the Ship itself. Thus, a Ship becomes a collection of nouns and their attributes.

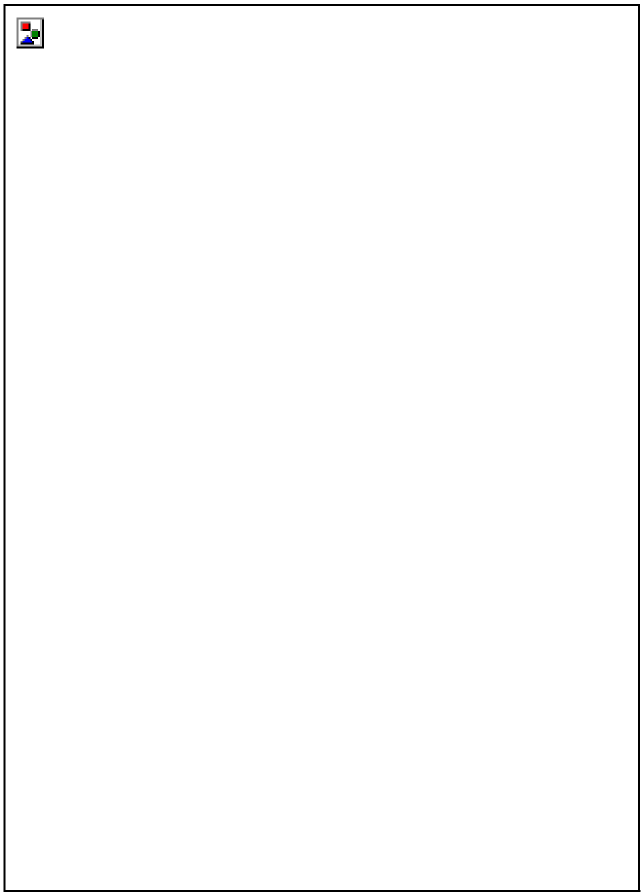
Similarly, one can postulate other related nouns that interact with Ships. Examples of these include an “Officer” (who can command a Ship), a “Flotilla” (a low–level organizational unit to which a Ship can belong), and a “Place” (where Ships can be at a point in time). Most of these objects are fairly compact in terms of the attributes associated with them—we want to capture basic information, but there’s no need to go overboard. So the Officer object might have a fairly limited set of attributes: Name, Rank, Graduating Class, Date of Birth, and Date of Death. By comparison, a Ship might well have dozens of component objects associated with it and potentially dozens of attributes as well.

Fine; we’ve described the nouns. However, ships also have things happen to them, and these actions are usually captured by verbs. For instance, ships can be, “launched”, “damaged”, “repaired” and “sunk”. Each one of these verbs becomes an object–oriented “method” of a Ship. These methods describe what happens to a Ship object when the event occurs. Different types of objects carry with them their own types of methods (i.e., only Ships have a “Sunk” method, whereas Places do not), and know how to execute them. For example, a Ship that is told to execute the Launched method would know that it has physically changed its state from “Building” to “Fitting Out”. It is now available to be counted in totals of warship tonnage launched, as well as in totals of ships that are fitting out, but would probably not be counted in totals of active warships—it would first need to execute the “Commissioned” method for that to happen. Furthermore, if one were modeling shipyard capacity, one would know that the slipway that was building this particular ship is now available for new construction, since when one launches a ship it is no longer on the slipway. In this way, by discussing nouns and verbs, and their interactions, a model of the application can be arrived at. [Readers interested in the actual object model underlying the application are referred to the application design documentation.]

As laid out above, the Ship object can get pretty complex. It is apparent that many of the items that compose the Ship are components—guns, radar, torpedoes, etc. These components are all modeled as separate entities, making revisions to the physical state of the ship more realistic. For instance, when *Yamato* received an upgraded anti–aircraft suite in 1943, she did so at the cost of landing her two triple 155mm wing turrets. It is far more realistic to conceive of these changes as having occurred due to the reduction of two triple 155mm turret components, followed by the addition of six dual 127mm AA turret components, rather than simply decrementing the total number of 155mm barrels by six and incrementing the 127mm barrels by twelve. What this approach adds in complexity it more than makes up for in terms of realism. The outcome of this approach, though, is that very few attributes of the ship are kept on the main Ship object itself. The hull dimensions and speed are about the only thing stored on this highest–level object.

As it turned out, however, the most important object to hone was not the “Ship” or the “Officer”, but rather the “Historical Event”. The Historical Event object forms the temporal glue that holds the entire application together and acts as a trigger for all sorts of methods. Why was *Yamato*’s anti–aircraft armament increased in 1943? Well, there was probably a “refit” Event that led to her landing her 155mm secondary battery. Why was *Akagi* sunk? Well, there was this little Event called “The Battle of Midway,” and so on.

Two important design decisions drove the final state of the Historical Event. The first was the realization that multiple ships should be able to be associated with a single Historical Event. In other words, there should not be a standalone Battle of Midway Event object for *Akagi*, and another for *Kaga*. Instead, there should be one Historical Event called “The Battle of Midway”, to which could be associated every ship that participated in the battle. In this way, assembling accurate orders of battle (OBs) becomes trivial—The Battle of Midway object (like all Historical Event objects) has a method called “Create OB” which gathers information on the collection of Ship objects that are associated with the Event. Thus, the OB is created without having to search each ship in the database to see if they happen to have their own Battle of Midway Event.



The second decision was that each ship should be allowed to record its own, subsidiary Ship Events for each Historical Event. In other words, *Akagi*’s individual history for the Battle of Midway event is different from *Kaga*’s, although both carriers were present at the battle. This approach to loosely coupling Ship Events to a higher level Historical Event object allows individual ship activities to be associated with others, but also kept segregated for reporting purposes. Thus, the history of the Battle of Midway is to be found mainly at the level of the individual vessels present, rather than at the Historical Event level. In the same way that a Ship object is a collection point for an assortment of **physical** attributes that are provided by lower–level component objects, the Historical Event acts as a collection point for **temporal** data stored in the individual ship's histories.

Taking this approach to the design of the Historical Event completely transformed the way in which the ship data was organized inside the application. In contrast to a standard reference book, a Ship was no longer viewed simply as a collection of physical attributes. Instead, a Ship was more properly understood in terms of its being a collection of temporal events, some of which may have triggered changes in the ship's physical state, others of which added to its history. Ships could now be related to other ships on the basis of the Events in which they mutually participated, rather than merely being (say) members of the same class of warships.

Uses of the Application

As designed, the application ought to serve multiple audiences on my site very well, since it:

- Acts as a basic ship reference manual (thereby serving the needs of the "average" browser);
- Provides links to detailed technical information on the individual components on a given ship (which caters to the technical "rivet–turner" audience);
- Provides operational information (which is aimed at the hardcore researcher crowd).

The latter audience is the most interesting in the long term, and it is here that I hope the application will return the most meaningful benefits, since it can provide a great deal of operational data in several different formats. At a basic level, the database can easily assemble a ship's tabular record of movement (TROM). A TROM, to the application, is nothing more than a chronological listing of all the events associated with an individual ship. Generating a TROM is no big deal in and of itself. One can already find TROMs for most WWII warships in hardcopy. However, since ships can be associated with the same historical events, the database takes the TROM concept one step further by interrelating all of the TROMs. Now, compiling a list of the operations in which, say, two destroyers worked together requires simply reporting on the subset of events to which they are mutually associated, rather than having to search separately through each ship's operations.

On another level, the database can also be queried on larger operational issues. For instance, returning to the list of questions that our Pacific War ASW researcher might have asked, one can begin to intuitively see what reporting approaches would be required to answer his/her questions:

Question	High–level Technical Answer

"How many destroyers did the Japanese Navy lose to submarines from January through June, 1943?"	For the given date range, collect the ships of Type "Destroyer" whose state changed to "Sunk" as a result of damage type "Submarine torpedo."
"What was the aggregate tonnage of escort warships launched by the Imperial Navy during 1944?"	For the given date range, collect the ships of Type "Destroyer" and "Destroyer Escort" whose state changed from "Laid Down" to "Launched."
"What were the names and graduating class of every destroyer skipper in the Imperial Navy in May, 1942?"	For the given date range, for Ships of Type "Destroyer", return the names and graduating class attributes of every associated commanding Officer object.
"What were the relative operational tempos of the Imperial Navy's destroyers in 1942 compared with 1944?"	This one is trickier, but still feasible (assuming the ability to compute distances between Lat./Long. coordinates). For the date range in question, for each ship of type "Destroyer", compute the total miles from Place to Place for each sortie, and provide a yearly average.

The result is that this design will allow the generation of new information on the topic of Pacific war operations that has either been too difficult to tabulate, or simply not obvious due to the lack of an explicit association between ships, men, and their connecting events. By integrating several different types of reference data, new, higher order data is created, leading to greater insights into the workings of a navy at war.

Extensions to the Basic Framework:

Mapping

One of the most intuitive ways to view information (particularly military movements) is through a map, and this application has been designed with future mapping applications in mind. Modern Geographic Information Systems (GIS) allows one to associate a visual representation (a map) with an underlying database. GIS would be perfectly suited to this naval operations database, and would provide a number of added benefits to a Web audience:

- The ability to locate where an Historical Event took place rapidly.
- The ability to see important locations and bases
- The ability to plot a ship's location visually over a given time range.

- The ability to display the answers to questions like, "Show me the sinking locations of all Japanese destroyers lost to submarine attack in 1943". Programming such "visual reports" would be relatively straightforward.

Incorporation of other Navies

This application was designed with a single navy (Japan's) in mind. However, the basic object architecture is so generalized that incorporating additional navies into the framework would be relatively simple to do, and would provide even greater benefits. For instance, if the U.S. Navy's combatants were to be incorporated into the design, generating OBs for **both** sides on a given Historical Event would be straightforward. Likewise, the application would be useful for performing comparative analyses, such as determining which navy had better availability of combat assets, what the leading causes of damage were to ships of a given class in each navy, etc. The only real impediment to implementing the framework for multiple navies would involve the additional loading of data, which for a navy as vast as the USN in World War II, could be formidable. However, even loading of data for a discreet set of important events (Midway, Coral Sea, i.e., "The Biggies") would produce useful results in and of themselves.

Limitations of the Framework

No technology project is without its limitations, and this one is no different. Several of these stumbling blocks are discussed below.

Data Loading and Input Time

It is clear that loading the data into the system will be a significant undertaking, even if only major combatants are modeled. Loading basic physical data is relatively easy, since many ships are of a given class which will facilitate the mass loading of "as–built" data. However, each ship will then need to have its individual TROM loaded, event by event. Historical–level Events will need to be created, participating ships associated with them, and individual Ship–level Events created to document their actions. Confusion will undoubtedly arise for some of the more obscure events—"Did I already create an event for Convoy Number Such–and–Such?" In short, loading the application will require quite a bit of effort and time to accomplish.

Data Integrity: Garbage In, Garbage Out

One must also be aware of the pitfalls of the data being loaded. While most of the major Japanese ships have reasonably complete TROM data available for loading, some of the data contained in the individual entries may be in doubt. How to reflect the existence of data that is suspect or tentative in nature is a design issue for the application that is still unresolved.

One approach is to simply display a "tentative data flag" for those entries that are questionable. This would allow a researcher to see at a glance that an event needs to be treated carefully. And setting flags is relatively easy to implement.

A second approach is to allow the creation of multiple versions of a single entry, each containing a version of the events in question. For instance, if one wanted to compare, say, S.E. Morison's version of events with those of a contemporary scholar such as Kimata Jiro, two competing sets of events—"Morison" and "Kimata"—could be created. One could then design the system to allow filtering of which events are displayed based on the set they belonged to. This second approach, while perhaps more flexible and powerful, is also much more complex to implement.

Order of Battle Limitations

At present, the highest level organizational entity that the application is designed to handle is a flotilla– or division–level unit (i.e. CarDiv, CruDiv) to which an individual ship is directly attached. Thus, the application cannot be used as a complete Order of Battle database for higher–level units, such as squadrons, task forces, or fleets. This is simply a matter of scope control. It would certainly be possible to extend the object model to encompass these higher level units, as well as the officers associated with them. However, a fundamental aspect of any successful information technology project is scope management, which is guided by a brutal fixation on achieving the primary purposes of the application at the expense of auxiliary features. It is important not to try and build *über*–applications, because such efforts rarely leave the starting blocks. The fundamental point of establishing this database is study naval operations at a granular level, as manifested in the actions of ships. It is not to get bogged down in the details of who was in charge of the 11th Backwater Fleet on Date X. This database has been tough enough to get off the ground as it is.

Changes In Technology

In the course of the year and a half that this application has been under semi–active development, the tools available to Web developers have changed dramatically. Questions as to which technologies are the most appropriate to deploy on, which are the least expensive, and (most important) which are the most fun for my volunteer developers to play with, are a never–ending source of complications to the effort as a whole. This, too, is endemic to technology projects of all kinds, and any historian attempting to venture into the world of application development must be aware of these issues.

Shortage of Technical Resources

By far the biggest hurdle to be overcome in implementing the application is securing the necessary technical expertise. While I am a fine technical analyst, I am not a developer. At present, I am relying on volunteers, and securing their continued attention to a project as complex and involved as this one is difficult. Despite the progress that has been made in creating the initial object design, the actual coding work has proceeded slowly over the last year and a half. It remains to be seen if the application will ever be fully deployed.

Conclusion

The increase in availability (and usability) of technologies such as the Web, databases, and object–oriented programming languages puts powerful new tools in the hands of historical researchers. The old truism, though—that technology is neutral—certainly applies in the case of historical research. Without a vision for its employment, technology is powerless to do anything useful. Not surprisingly, generating this vision typically also requires at least a willingness to familiarize oneself with the tools available.

The power that technology bestows also delivers a concomitant responsibility on the part of the historian to actively seek out innovative approaches for using it. As we have seen, re–cycling and rote presentation of well–known reference data, while convenient to the masses, does no great good for the advancement of naval history as a craft. Instead, we need to be searching for ways to synthesize known types of information into new, higher order information. As historians living in a technological era, we must constantly be exploring the intersection between what our audiences want, and what technology can supply, and then pushing those boundaries outward. It is there that unforeseen (and exciting) advancements lie. Just as this database application originally started as a simple tool for displaying physical ship data, and evolved into a complex tool for studying naval operations, other higher–order applications surely await future researchers. With insight into a given topic, and a solid design, technology offers the possibility of presenting information in ways that have never before been possible. Through it, we can now resolve complex, interesting questions that heretofore were unanswerable via conventional means. By doing so, technology will have enabled us to create truly new knowledge.

[1] William H. Garzke, Jr. and Robert O. Dulin Jr. *U.S. Battleships in World War II.* (Annapolis, Md.: Naval Institute Press, 1976.)
William H. Garzke, Jr. and Robert O. Dulin Jr., *Allied Battleships of World War II.* (Annapolis, Md.: Naval Institute Press, 1980).
William H. Garzke, Jr. and Robert O. Dulin Jr., *Axis and Neutral Battleships in World War II.* (Annapolis, Md.: Naval Institute Press, 1985).