REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
10/27/2000	CONFERENCE PROCEEDINGS	

4. TITLE AND SUBTITLE
DESIGNS AND LESSONS LEARNED IN OBJECT-ORIENTED WEB-BASED MAPPING

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Ruth Wilson, Frank McCreedy, Roy Ladner, Miyi Chung and Kevin Shaw

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Research Laboratory
Marine Geoscience Division
Stennis Space Center, MS 39529-5004

8. PERFORMING ORGANIZATION REPORT NUMBER
NRL/PP/7440--00-1012

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
OFFICE OF NAVAL RESEARCH/MARINE CORPS

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This paper presents a summary of designs used in developing an object-oriented web-based mapping application. The Web Mapping Toolkit (WMT) is a Java applet which accesses mapping data from the Geospatial Information Database (GIDB). The GIDB is an object-oriented database developed by the Naval Research laboratory which stores multiple data types from many sources. The motivation for development of the WMT is to allow users to access geographical data over the Internet through a web browser, without the necessity of specialized software or hardware. Main topics of discussion for the design ofthe WMT will include choice of programming languages and data transfer protocols, object-oriented design applied to mapping, use of object-oriented pointers for efficient retrieval, interface definition language (idl) design choices, and factors associated with map display. Lessons learned throughout the design process will also be presented.

15. SUBJECT TERMS
object-oriented, mapping application, geospatial information database, geographical data, interface definition language

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Ruth Wilson
Unclassified	Unlassified	Unclassified		11	19b. TELEPHONE NUMBER *(Include area code)* 228-688-4525

Proceedings of the Southern Conference on Computing
The University of Southern Mississippi, October 26-28, 2000

DESIGNS AND LESSONS LEARNED IN OBJECT-ORIENTED WEB-BASED MAPPING

RUTH WILSON, FRANK MCCREEDY, DR. ROY LADNER, MIYI CHUNG AND KEVIN SHAW*

Abstract. This paper presents a summary of designs used in developing an object-oriented web-based mapping application. The Web Mapping Toolkit (WMT) is a Java applet which accesses mapping data from the Geospatial Information Database (GIDB). The GIDB is an object-oriented database developed by the Naval Research Laboratory which stores multiple data types from many sources. The motivation for development of the WMT is to allow users to access geographical data over the Internet through a web browser, without the necessity of specialized software or hardware. Main topics of discussion for the design of the WMT will include choice of programming languages and data transfer protocols, object-oriented design applied to mapping, use of object-oriented pointers for efficient retrieval, interface definition language (idl) design choices, and factors associated with map display. Lessons learned throughout the design process will also be presented.

1. Introduction. The Geospatial Information Data Base (GIDB) is an object-oriented (OO) digital mapping database system designed by the Digital Mapping, Charting and Geodesy Analysis Program (DMAP) of the Naval Research Laboratory. Development of the GIDB began in 1994 to demonstrate that representation of spatial data is less cumbersome and more optimal in object format versus relational format. Smalltalk was chosen as the programming language for the database, since at that time it was the most robust pure OO language available. Gemstone was chosen as the OO Database Management System (OODBMS). When Java made its debut in 1995, the DMAP Team began monitoring its usefulness in web-based capabilities. The stability of the Corba 2.0 standard, which would allow for communication between Smalltalk and Java across a network, led the DMAP Team to begin work on web-based digital mapping capabilities in 1997. This work has resulted in the GIDB Web Mapping Toolkit (WMT), which gives users access to mapping data via the Internet.

The GIDB is able to store spatial data from a variety of sources. Most of the mapping data in the GIDB is obtained from the National Imagery and Mapping Agency (NIMA). Other data sources include the Naval Oceanographic Office, US Geological Survey, National Ocean Survey, US Census Bureau, and the US Army Corps of Engineers. The most common format of GIS data that can be read by the GIDB is vector data such as NIMA's Vector Product Format (VPF) or ESRI Shapefile format. The GIDB can also read in imagery, audio clips, video clips, and specialized ascii or binary files. The GIDB database stores these multiple types of geographically referenced data in a quadtree spatial indexing scheme.

Each data type is ingested into a common object format for uniform retrieval. This common object format was designed based on the organization of VPF data, the most complex data type in the GIDB. Each dataset is called a database. Each database is sub-divided into libraries, based on the scale of the data. Each library is made up of coverages, which represent different themes such as hydrography, transportation, and utilities. At the coverage level, the individual feature instances are stored in a quadtree, based on whether the feature is a point, line, area, or text feature. This object format structure of the database is summarized in figure 1.

* Digital MC&G Analysis Program, Naval Research Laboratory, Stennis Space Center, MS 39529, ruth.wilson, fmccreedy, rladner, chung, shaw@nrlssc.navy.mil

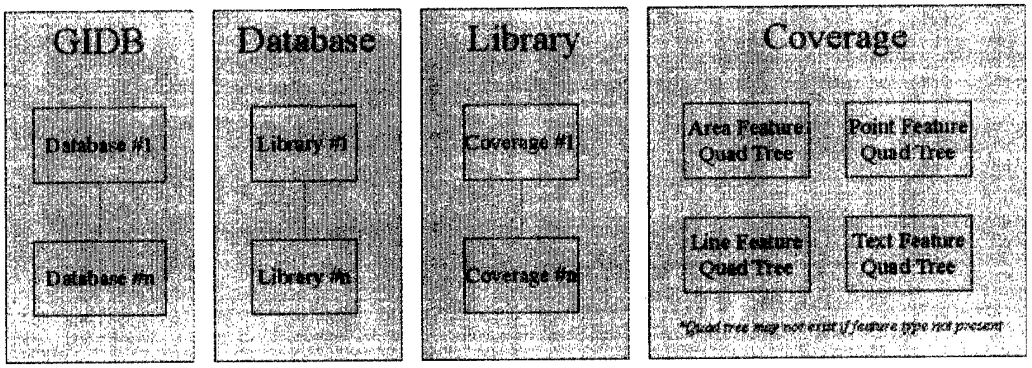

Figure 1. GIDB Common Object Structure.

Use of a common object format makes it possible to retrieve data collected from many different sources from one central location. One advantage of this is that applications that make use of this data (such as map creation applications) do not need to implement readers for all of these types of formats; they only need to know how to access the database. Also, since all of the data relevant to a feature is encapsulated in an object, accessing the data is much faster and easier than reading it from a relational database implemented in a flat file structure with tables and indexing.

The GIDB WMT is a web-based Java applet developed to access data from the GIDB. This paper will discuss the design process for the WMT, with focus given to why certain choices were made over other choices. Some of the obstacles discovered in the development of the WMT will be detailed, with emphasis on how these obstacles were overcome. Finally, a summary of the design work will be discussed, and direction for future work will be presented.

2. Communication Scheme. The GIDB is accessed by external programs in addition to the WMT using Common Object Request Broker Architecture (CORBA). There are several benefits to using CORBA. One of the benefits is to transfer objects to the client, so that the client can be more than a "dumb" terminal that views server side generated maps. With objects, functionalities like querying features for their information, removing features, and adding features to the map are easily added. CORBA is an abstraction of the communication process so focus can be less on low-level communication mechanisms and more on what needs to be transferred. Initially, communication was performed using the Internet Interoperability Protocol (IIOP), as shown in figure 2.

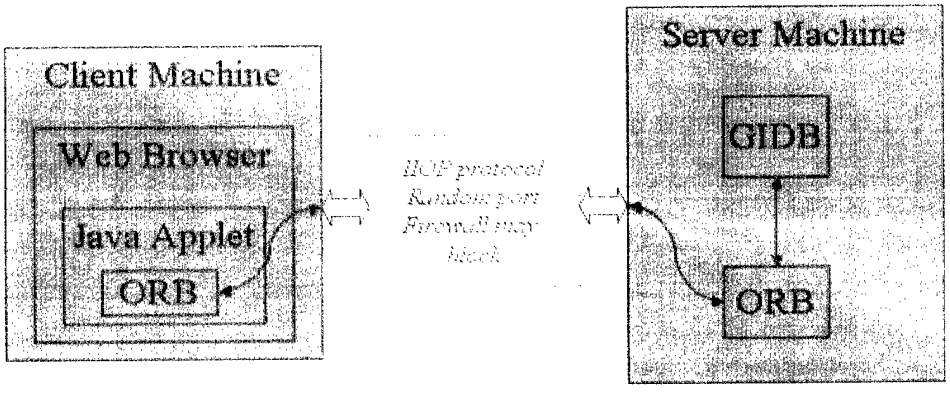

Figure 2. Initial GIDB Communication Scheme.

The initial IIOP communication scheme appeared to work well, but as more users attempted to access the GIDB server machine, some were not able to make a CORBA connection. It was eventually determined that if a user's client machine was behind a firewall, the CORBA connection would fail. Firewalls can filter out or block packets of data based on the port the packets are trying to reach and also the protocol that the packet is using. Most network administrators will only allow commonly used and non-vulnerable services, such as HTTP traffic, through their firewalls. They will also usually restrict network traffic to port 80, typically a web server port. The issue of HTTP protocol versus IIOP protocol could be resolved by a process called HTTP-Tunneling, which takes a CORBA message and wraps it so that it appears to be an HTTP message. HTTP-Tunneling is accomplished using special ORB software, such as VisiBroker Gatekeeper. The GIDB server ORB, Gemstone's GemORB, does not support HTTP-Tunneling, however, so the HTTP protocol is not an option for the GIDB. The other firewall issue of port selection also could not be resolved in the GIDB system. Since Gemstone's GemORB, the GIDB server ORB, randomly selects a communication port, most firewalls would prevent the communication packets from being transferred. At the time this firewall issue was discovered, Java servlets were becoming a mainstream technology which could possibly be used to resolve this issue.

A servlet is server-side Java code called from a web server that does HTTP request processing instead of the web server for certain URLs. Servlets allow data to be transferred to whatever web server is being used, and allow data to be routed to a particular servlet to process a request and return results. This capability allows the developer to create dynamic web content, rather that just feeding the client a file off of the server machine webspace. Since Java has a rich set of IO classes, Java objects can be passed back and forth between the client and servlet. The current GIDB communication scheme is shown in figure 3.

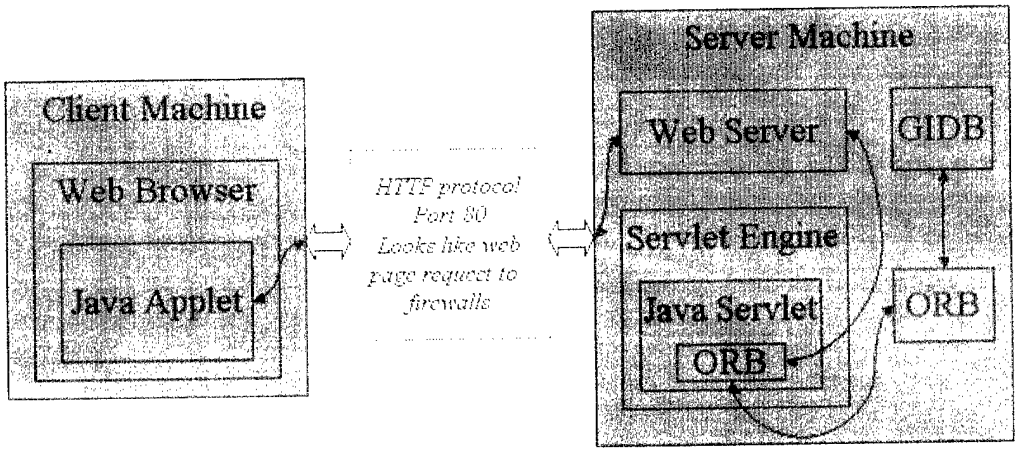

Figure 3. Current GIDB Communication Scheme.

This communication scheme will allow data transfer through all but the most stringent firewalls. A bonus in using servlets has been that the power of Java is now available on the server side. Many utility methods can be easily implemented, such as writing out files to the server machine from data sent by the client. On unix systems, Apache is used as the GIDB web server, with JServ as the servlet engine. On NT systems, Internet Information Server (IIS) is used as the GIDB web server, with JRun as the servlet engine. Note that a servlet engine is a container and

dispatcher of all of the servlets that exist on the server. The servlet ORB is Visigenic VisiBroker. The GIDB database ORB is Gemstone GemORB.

In summary, the current GIDB communication scheme can now operate through most firewalls. Also, the client machine does not need to download ORB Java classes, which have a total of over a megabyte of data. In addition, Java processing on the server machine adds extra functionality to the WMT. There are still some issues which may need to be resolved with this communication scheme, however. The server cannot call directly to the client. If this is needed then some mechanism must be created where the client will poll the server. Also, HTTP protocol is not as efficient as IIOP protocol, so other solutions may need to be explored. Additionally, more layers of communication are involved, with the client passing data using HTTP to the servlet, which then passes data using IIOP to the GIDB server.

3. Object Mapping. At the heart of CORBA is the definition of the data to be transferred and the methods that can be called to access the data. For this object mapping, an Interface Definition Language (IDL) file is created. Although CORBA is used to transfer data between the Java servlet and the GIDB Smalltalk database, the entire class hierarchy of the database is not mapped on the client side. There are a number of reasons for this:

- A large amount of time would be spent writing the IDL by hand since the database has no IDL generator.
- The client would be saturated with class definitions requiring a long download time.
- The client doesn't need most of the database class information in order to display and perform simple queries on the data.
- The client needs to be thin since it is a web application.

Given these limitations, a minimum set of data that would be needed by the applet was defined. The minimum requirements include the geographic points that define the feature, the type of feature (point, line, or area), and a reference into the database for the feature in case more information about it is needed. Initially, the reference was a string comprised of a database name, library name, coverage name, and feature ID. This string had to be parsed by the server for it to find the object in the database. This parsing requirement was eliminated by using the feature's Object Oriented Pointer (OOP) as its reference rather than the string identifier. The OOP is a unique number that the Gemstone OODBMS uses to reference its objects. At first, the OOP was defined in IDL as a "long", which is a 32 bit integer. As the database grew in size, some of the OOP values became larger than 32 bits, causing an exception when these values were attempted to be passed to the client side applet. Although IDL defines a "long long" which is a 64 bit integer, this solution was not viable since Smalltalk does not have a mapping for "long long". To resolve this problem, the OOP number is now sent to the client as a string. This solution is not as efficient as a "long long" would be, but it basically gives an unlimited range. Another solution to be investigated in the future is to switch to a byte array representing an unlimited ranged integer.

The IDL file defined for the WMT contains all of the methods for accessing data in the database. Some examples of these methods are:

- Retrieve all of the databases for a geographic area.
- Retrieve all of the libraries in a geographic area.
- Retrieve all of the coverages and their feature lists for a geographic area.
- Retrieve information about a feature type such as the number of this type of feature and a reference to the collection of these features that are in the client's current area of interest.
- Return a block of size specified by client of all of the features of a certain type in an area.

- Retrieve the attributes for a group of features specified by the client.
- Retieve any multimedia data for this area. This can include pictures, videos, audio, and HTML links.
- Retrieve an image of this area that can be used as a background for the map. This typically includes scanned maps and satellite images.

The WMT applet is designed so that information from the database is only retrieved at a request from the user. In the initial WMT design, a set of features was retrieved in its entirety, requiring a user to wait for the entire dataset to be retrieved from the server before it could be displayed. Often this process would take several minutes, leading the user to believe that an error occurred. To rectify this long wait period, the WMT design was modified so that data is retrieved in blocks of 50. Each block of data is displayed on the client as the client requests the next block of data. A status bar is also provided to the user, showing the number of features being retrieved and approximately how long it will take. The multi-tasking is accomplished using Java threads.

As previously mentioned, the IDL mapping of data is a subset of the original object definition in the GIDB database. Even with a subset of the original data, however, the amount of information being passed between the client and server would sometimes cause an out of memory error on the client. To prevent this error and to speed up retrieval time, it was decided that not all aspects of a feature as defined in the idl would be passed on the initial retrieval. In particular, attributes for features would not be passed initially. Only if the user wanted to examine or query the attributes would they be retrieved. This optimization resulted in an obvious improvement in initial feature retrieval.

4. Drawing Principles. Most GIS data retrieved from the GIDB server is in the form of vectors, although some raster data that can be used as map backgrounds can be retrieved as well. Vector data contains the geographic location of points that make up the features. Consequently, many different types of features may be displayed, overlapped, and turned on and off. The question that comes up is "how do features keep from covering up other features?" There are three basic types of vector features that are dealt with: points, lines and areas. Area features are polygons and can be quite large. If features are drawn into a single image buffer, and a large area feature is drawn after the other features, some features will most likely be covered up by the large area feature. One way to fix this problem is to sort all of the features before you draw them. If features are drawn using the order areas, lines, points, then this problem can be prevented to some degree.

Initially in the WMT design, no sorting was done at all and features often covered other features. Later, separate image buffers were used for points, lines and areas, as shown in figure 4.

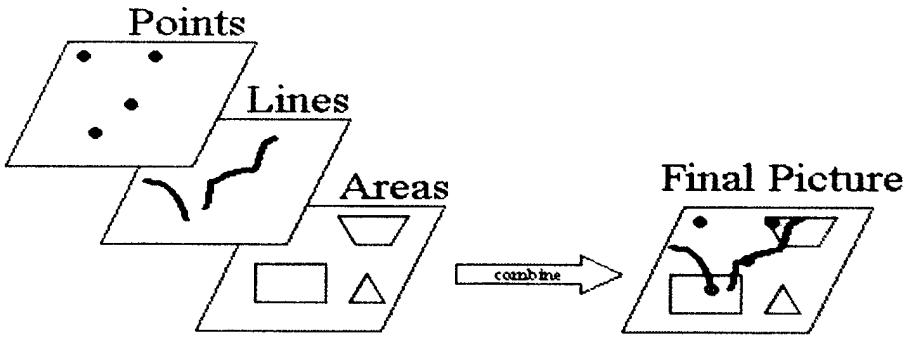

Figure 4. Use of Separate Buffers for Creating a Composite Map.

Note that an image buffer in this context is an offscreen image to which features can be drawn. The buffer technique also can help the case of turning a feature on and off. Instead of having to redraw all of the features, a particular feature can be turned on or off in its respective buffer and the buffers can be recombined which can speed up the drawing if one of the other buffers has complex data. All of the features in the buffer that contains the modification will have to be redrawn, though, so the realized value of this approach is questionable.

One thing to note in using separate buffers is that each of the buffers needs to be filtered for transparency. Assume that the color white will not be used by any features. Pre-fill each buffer with white, draw the features in the buffers, and then filter the buffers so all pixels that are white get an alpha value of 0, making them transparent. If this filtering is not done, each buffer would completely mask the buffers beneath it. In Java 1, alpha seems to be implemented in a non-optimal way. Given a region of pixels that have an alpha value of 127 (half the max) then a grid is made. Only half of these pixels will be drawn which creates a screen door effect. There is no color blending going on. Java 2 does do color blending and may even be able to take advantage of hardware blending so it may be considerably faster.

The current approach for drawing features in the WMT is to draw all of the features into one display buffer. One buffer is used rather than three buffers for three main reasons. First, memory is saved by using only one buffer. Second, filtering the buffers for transparency was taking a very long time, so it is faster to use only one buffer. Finally, drawing an image with transparent values is slow in Java 1, so drawing only one image rather than three is faster.

With the single buffer approach, the features are not sorted based on type, but the user can change the drawing order using a drawing options function as shown in figure 5.

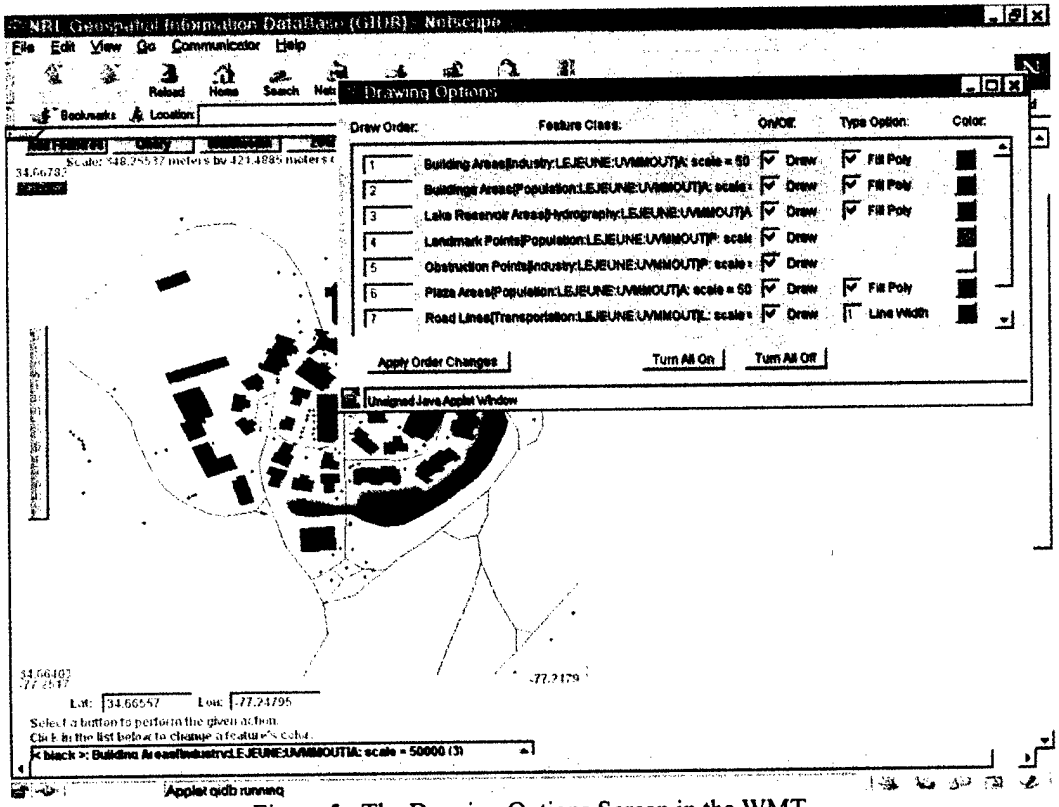

Figure 5. The Drawing Options Screen in the WMT.

The user can also turn polygon fill off, which allows features beneath them to show through. Anytime a feature is turned on or off then all of the features must be redrawn into the single buffer. In most cases this is not a problem and they are all redrawn very quickly. The worst case seems to be with a textured polygon that has thousands of points. Textured means that some image is tiled across and clipped to the polygon's region. Texturing also involves transparency filtering, meaning that every pixel not in the polygon becomes transparent. The texturing takes a long time since the method that checks to see if a pixel is inside the polygon must look at each point of the polygon for each pixel of the map. If the map is 400x400 pixels, and the polygon has 10,000 points, then there are 400 * 400 * 10,000 (1.6 billion) calculations that must be performed. Each calculation consists of several if statements so a large number of machine instructions are being carried out. If there are several such polygons, performance is drastically impaired. If the polygon does not need to be textured, then the polygon can be drawn directly to the buffer, instead of creating an image of it. This approach is much faster. Again, Java 2 may provide a better solution for texturing polygons. Another solution would be to use a separate buffer only for area features or even only for area features that need to be textured.

Most of the difficulties that have been encountered in drawing features in the WMT could be eliminated by using Java 2. There are many more graphics methods available in Java 2 than in Java 1. Instead of having to implement many drawing functions in the WMT code, a switch to Java 2 would provide most everything needed by the WMT in the pre-defined Java classes. For instance, DMAP developers had to write an image rotator for the WMT, whereas Java 2 lets a user define transforms for any drawing routine, so developers could make a transform that rotates an image and then does any number of other types of transforms to it before actually drawing it. There is also anti-aliasing support in Java 2 which improves the map appearance in some cases by eliminating jagged lines. Many other enhancements and improvements in Java 2 could be useful in the WMT for developers.

5. Status of Java. The WMT applet has been written using Java 1.1.x. Both Netscape and Internet Explorer (IE) have a Java Virtual Machine (VM) as part of the web browser. These are both Java 1.1.x VMs. While each one is a well made VM, they do have their differences. Often a change in code would work in one VM and not work in the other one. VM differences are a major problem for a developer. Trying to make a GUI appear just right and then having it look correct in only one VM is extremely frustrating. The solution to this problem is to use the Java plugin.

The Java plugin is, as it sounds, a plugin to a web browser. Several examples of common plugins include Shockwave, Quicktime, and Cosmo Player. The same Java plugin is used for Internet Explorer, Netscape, and any other web browser that implements the plugin architecture. What this means is that applets run within the plugin should function identically regardless of the browser from which they are invoked. Also, the Java plugin uses the latest VM from Sun Microsystems without having to wait for Netscape and IE to update their VMs. The major reason that the plugin has become a focus of attention is that it provides Java 2 support. Java 2 has a much larger set of APIs built in, and packages used by the WMT from other sources will most likely need a Java 2 VM to run in.

One of the best aspects of the WMT is that it will give users access to map data using their existing web browser and no other software. Given this fact, the developers of the WMT have been hesitant to use the Java plugin since it would require a user to download and install the plugin software, which is about a 5 megabyte download. However, Microsoft has begun to not include their VM in a default IE download. Also, Netscape will soon no longer have their own VM implementation. Beta releases of Netscape 6 ship with the Java 2 plugin. The Java plugin is the future of Java on the client side, at least as far as applets are concerned. One other nice thing about the plugin is that it also installs a Java 2 Java Runtime Environment (JRE) on the system.

This allows a user to run any other Java applications on their system. Given the benefits of the Java plugin and the direction that browser vendors are taking with regards to the VM, the WMT will soon be implemented for use with the Java plugin.

Migration of WMT code from Java 1 to Java 2 has begun. While this migration does not require a major coding change, there are other issues which need to be addressed. One such issue is that the IDL compiler for Java 2 would not compile the IDL file provided for Gemstone. The reason for the failure was a conflict in the names defined, so several objects in the IDL file were renamed so that the file would compile. Another issue was that the ORB in Java 2 was not as full featured as the Visibroker ORB that we had been using. For instance, the Java 2 ORB had no support for transactions. The only way that Java 2 could communicate with GemORB was to recreate the Java 2 runtime Java Archive (JAR) file without any of its CORBA classes. This had to be done rather than just adding the VisiBroker classes to the classpath since Java 2 will look in its runtime JAR for classes before any other source. By removing CORBA classes from the JAR, Java 2 will not find them there and will look for them in the VisiBroker JAR file. This approach works, but it may still be a problem in some respects because now any package that is dependent on an ORB will have to be compatible with VisiBroker.

Another migration issue is that Java 2 provides the Swing package, which makes GUIs have a better appearance. Swing seems to be considerably more complex than Abstract Window Toolkit (AWT). Consequently, it will require much time and effort to convert WMT code from AWT to Swing. The improved user interface for the WMT, however, will be worth this added effort.

6. Mapping in 3D. Recently progress has been made on allowing the user to render their WMT map in 3D. This is accomplished by taking the 2D map in the WMT display, and overlaying it on a Virtual Reality Modeling Language (VRML) elevation grid. The grid is made by accessing NIMA Digital Terrain Elevation Data (DTED) that is stored in the GIDB. After the VRML file is created, the user can "fly through" their map. An example of a VRML display is shown in figure 7. The 2D map from the WMT used as the overlay is shown in figure 6.

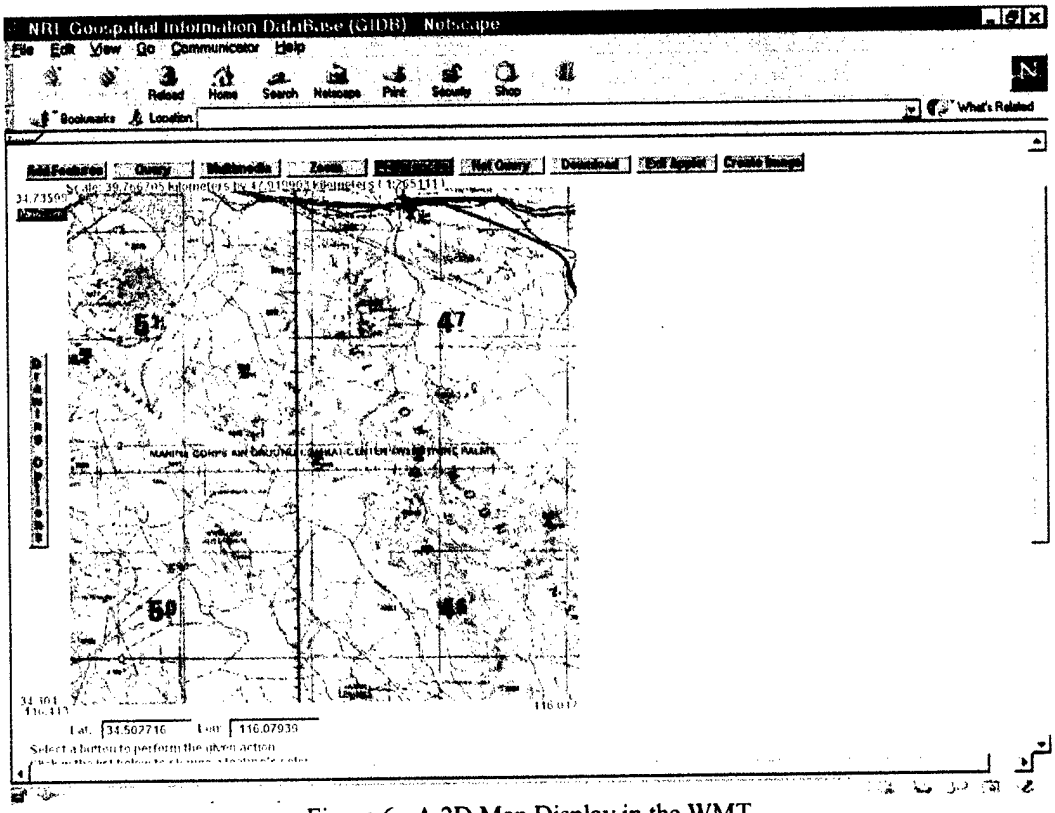

Figure 6. A 2D Map Display in the WMT.

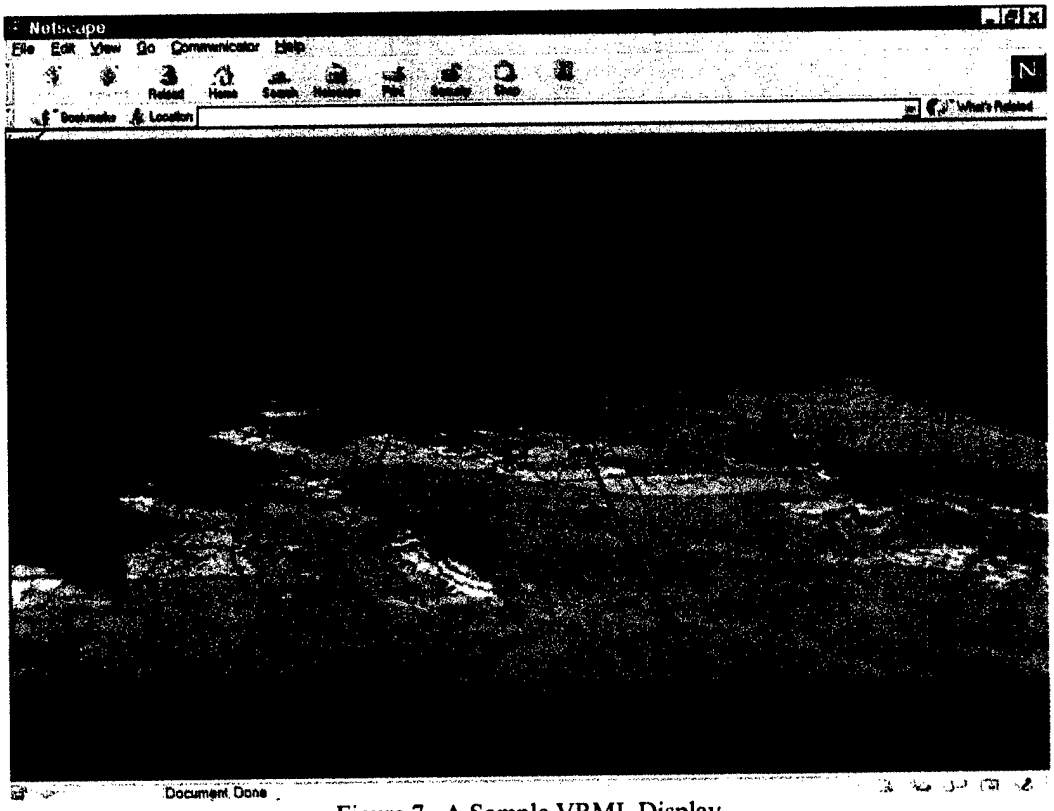

Figure 7. A Sample VRML Display.

As stated earlier, WMT development is migrating to Java 2. Java 3D classes are available for download with the Java 2 release. There may be advantages to using Java 3D over VRML for our WMT 3D implementation. VRML is a file format that requires a browser plugin such as Cosmo Player to display the VRML scene in the browser. Two different VRML plugins may display the same VRML file in different ways. Java 3D, on the other hand, is an API that gives a developer greater control over the scene generation. There may also be imroved performance using Java 3D. As the migration to Java 2 advances, these issues will be investigated.

7. Summary and Future Work. The WMT applet allows users to access mapping data over the Internet using a web browser. The design of the WMT has evolved through many changes based on speed and size of data transfer, taking advantage of object-oriented principles, memory usage on the client machine, and overall performance. New functionalities are added to the WMT on an ongoing basis, and the WMT has many other capabilities that have not been mentioned in this paper. The major area for future work on the WMT relevant to the topics covered in this paper lies in the migration to Java 2, which will greatly enhance the capabilities and performance of the WMT.

8. Acknowledgments. We would like to thank the Marine Corps Warfighting Lab, under Program Element number 0603640M, and the Office of Naval Research, under Program Element number 0603238N, for sponsoring this work.

9. References.

[1] A.B. Chaudhri and M. Loomis, *Object Databases In Practice* Chapter 15, 234-253 (New Jersey: Prentice Hall PTR, 1998).

[2] Defense Mapping Agency, *Military Standard: Vector Product Format. Draft Document No. MIL-STD-2407* (Fairfax, VA: Defense Mapping Agency, 1993).

[3] K. Shaw, M. Cobb, M. Chung, and D. Arctur, Managing the US Navy's First OO Digital Mapping Project, *IEEE Computer*, 29(9), 1996, 69-74.

[4] R. Wilson, T. Lovitt, M. Cobb, M. Chung, B. Ray, and K. Shaw, Design of a Java Interface to a Smalltalk OO Mapping Database Utilizing CORBA, *Parallel and Distributed Computing and Systems: Proceedings of the Tenth IASTED International Conference,* Las Vegas, NV, October 28-31, 1998, pp.60-63.

[5] Object Management Group, *The Common Object Request Broker: Architecture and Specification*, (Object Management Group, Inc., July 1996).

[6] T. Mowbray and W. Ruh, *Inside CORBA: Distributed Object Standards and Applications*, (Massachusetts: Addison Wesley Longman Inc., 1997).

[7] M. Cobb, H. Foley, R. Wilson, M. Chung, and K. Shaw, An OO Database Migrates to the Web, *IEEE Software*, 15(3), 1998, 22-30.

[8] M. Cobb, M. Chung, K. Shaw, and D. Arctur, *Object-Oriented Database Design and Implementation Issues for Object Vector Product Format*, NRL FR/7441-95-9641, Naval Research Laboratory, Stennis Space Center, MS, 1996.

[9] K. Shaw, M. Abdelguerfi, E. Cooper, C. Wynne, H. Miller, B. Ray, R. Broome, T. Lovitt, *Virtual World Reconstruction Using the Modeling and Simulation Extended Vector Product Prototype*, NRL/FR/7441--96-9654, Naval Research Laboratory, Stennis Space Center, MS, 1997.

PUBLICATION OR PRESENTATION RELEASE REQUEST

NRLINST 5600.2

1. REFERENCES AND ENCLOSURES	2. TYPE OF PUBLICATION OR PRESENTATION	3. ADMINISTRATIVE INFORMATION
Ref: (a) NRL Instruction 5600.2 (b) NRL Instruction 5510.40D Encl: (1) Two copies of subject paper (or abstract)	( ) Abstract only, published ( ) Abstract only, not published ( ) Book ( ) Book Chapter ( ) Conference Proceedings (refereed) (X) Conference Proceedings (not refereed) ( ) Invited speaker ( ) Multimedia report ( ) Journal article (refereed) ( ) Journal article (not refereed) ( ) Oral Presentation, published ( ) Oral Presentation, not published ( ) Other, explain	STRN NRL/PP/7440--00-1012 Route Sheet No. _____ Job Order No. _____ Classification X U _____ C Sponsor ONR/Mar Corps approval obtained _____ yes _____ no

4. AUTHOR

Title of Paper or Presentation
DESIGNS AND LESSONS LEARNED IN OBJECT-ORIENTED WEB-BASED MAPPING

Author(s) Name(s) *(First, Mi, Last)*, Code, Affiliation if not NRL

RUTH WILSON (CODE 7440.2), FRANK MCCREEDY(7440.2), ROY LADNER (CODE 7440.2), MIYI CHUNG (CODE 7440.2), KEVIN SHAW (CODE 7440.2)

It is intended to offer this paper to the SOUTHERN CONFERENCE ON COMPUTING
(Name of Conference)

THE UNIVERSITY OF SOUTHERN MISSISSIPPI, OCTOBER 26-28, 2000, HATTIESBURG, MISSISSIPPI
(Date, Place and Classification of Conference)

and/or for publication in _____ _____
(Name and Classification of Publication) *(Name of Publisher)*

After presentation or publication, pertinent publication/presentation data will be entered in the publications data base, in accordance with reference (a).
It is the opinion of the author that the subject paper (is____) (is not X) classified, in accordance with reference (b).
This paper does not violate any disclosure of trade secrets or suggestions of outside individuals or concerns which have been communicated to the Laboratory in confidence. This paper (does____) (does not X) contain any militarily critical technology.
This subject paper (has____) (has never X) been incorporated in an official NRL Report.

RUTH WILSON, NRL CODE 7440.2 — Ruth a. Wilson
Name and Code *(Principal Author)* (Signature)

5. ROUTING/APPROVAL

CODE	SIGNATURE	DATE	COMMENTS
Author(s) Wilson	Ruth a. Wilson	10-27-00	
Section Head Shaw		10/27/00	
Branch Head Harris		10/27/00	
Division Head ACTING VALENT		10/27/00	1. Release of this paper is approved. 2. To the best knowledge of this Division, the subject matter of this paper ~~has~~ ____) (has never X) been classified.
Security, Code ~~1221.1~~ 7031		10/27/00	1. Paper or abstract was released. 2. A copy is filed in this office. SSC-141-00
Office of Counsel, Code ~~3008.2~~ 1008.2		10/27/2k	
ADOR/Director NCST			
Public Affairs *(Unclassified/ Unlimited Only)*, Code ~~1230~~ 7030.4		10/27/00	add I E # is acknowledged
Division, Code			
Author, Code 7440.2			

HQ-NRL 5511/6 (Rev. 12-98) (e) THIS FORM CANCELS AND SUPERSEDES ALL PREVIOUS VERSIONS

6. DISTRIBUTION STATEMENTS *(Author to check appropriate statement and fill in reason as required)*

[X] A - Approved for public release, distribution is unlimited.

[ ] B - Distribution authorized to U.S. Government agencies only *(check reason below):*

[ ] Foreign Government Information [ ] Contractor Performance Evaluation [ ] Critical Technology
[ ] Proprietary Information [ ] Administrative/Operational Use [ ] Premature Dissemination
[ ] Test and Evaluation [ ] Software Documentation [ ] Cite *Specific Authority _____
(Identification of valid documented authority)

Date statement applied _____

Other requests for this document shall be referred to _____
*(Insert Controlling DOD Office *)*

[ ] C - Distribution authorized to U.S. Government agencies and their contractors *(check reason below):*

[ ] Foreign Government Information [ ] Software Documentation
[ ] Administrative/Operational Use [ ] Critical Technology [ ] Cite *Specific Authority _____
(Identification of valid documented authority)

Date statement applied _____

Other requests for this document shall be referred to _____
*(Insert Controlling DOD Office *)*

[ ] D - Distribution authorized to DOD and DOD contractors only *(check reason below):*

[ ] Foreign Government Information [ ] Critical Technology
[ ] Software Documentation [ ] Cite *Specific Authority _____
(Identification of valid documented authority)
[ ] Administrative/Operational Use

Date statement applied _____

Other requests for this document shall be referred to _____
*(Insert Controlling DOD Office *)*

[ ] E - Distribution authorized to DOD components only *(check reason below):*

[ ] Critical Technology
[ ] Proprietary Information [ ] Premature Dissemination [ ] Direct Military Support
[ ] Foreign Government Information [ ] Software Documentation [ ] Test and Evaluation
[ ] Administrative/Operational Use [ ] Contractor Performance Evaluation [ ] Cite *Specific Authority _____
(Identification of valid documented authority)

Date statement applied _____

Other requests for this document shall be referred to _____
*(Insert Controlling DOD Office *)*

[ ] F - Further dissemination only as directed by _____
*(Insert Controlling DOD Office *)*

Date statement applied _____ or higher DOD authority _____

[ ] X - Distribution authorized to U.S. Government agencies and private individuals or enterprises eligible to obtain export-controlled technical data in accordance with regulations implementing 10 U.S.C. 140c.

Date statement applied _____

Other requests for this document shall be referred to _____
*(Insert Controlling DOD Office *)*

*For NRL publications, this is usually the Commanding Officer, Naval Research Laboratory, Washington, DC 20375-5320

7. OTHER LIMITATION

[ ] Classification only [ ] NOFORN [ ] DTIC exempt (explain) _____

22 Oct-00
Classification Review (Initial/Date)

Substantive changes made in this document after approval by Classification Review and Public Release invalidate these reviews. Therefore, if any substantive changes are made by the author, Technical Information, or anyone else, the document must be returned for another Classification Review and Public Release.

8. INSTRUCTIONS

Author completes and submits this form with the manuscript via line channels to the division head for review and approval according to the routing in section 4.

1. NRL Reports.....Submit the diskette (if available), manuscript, typed double-spaced, complete with tables, illustrations, references, draft SF 298, and proposed distribution list.

2. NRL Memorandum Reports.....Submit a copy of the original, typed manuscript complete with tables, illustrations, references, draft SF 298, and proposed distribution list.

3. NRL Publications or other books, brochures, pamphlets, proceedings, or any other printed publications.....Handled on a per case basis by Site Technical Information Office.