DEVELOPING A SECURE GEOSPATIAL VISUALIZATION AND COLLABORATION SYSTEM USING SOFTWARE ENGINEERING TECHNOLOGY

H. Yu,∗ L. Wang,** J. Zhang,*** J. Barksdale,* and X. Yuan*

Abstract

Developing secure web applications is a difficult task because the original purpose of the World Wide Web (WWW) was to freely share information. This paper presents a prototype for a platform-independent secure geospatial visualization and collaboration web application. This system is developed using object-oriented system design methodology. It integrates software engineering, web security, visualization and Java technologies with the WWW to enable production, dissemination, and use of imagery and geospatial information on a hierarchical level through the WWW. Several visualization modules on the client side are developed using the Model-View-Controller design pattern technology. These modules are plug-in and reusable components for other similar applications. A Visualization Web Bridge and Data Providers on the server side are designed and implemented to bridge the gap between the Internet users and the remote geospatial databases. Several Internet security mechanisms are implemented to secure system configuration, to authenticate and authorize data access, and to secure information sharing. The system is an open and extensible source. The application results demonstrate that this system provides a certain level of security, and that users have flexibility to query, visualize, manipulate, and analyze real-time remote geospatial data. It exhibits that multiple users in geographically dispersed organizations can conveniently collaborate and share information as well as 3D images through this system. It also demonstrates that using object-oriented system design and design pattern provides a great chance for software reuse and increases the system flexibility.

Key Words

Software engineering, visualization, web security, WWW, Java technology

* Department of Computer Science, North Carolina A&T State University, Greensboro, NC, USA; e-mail: cshmyu@ncat.edu; jtb32@yahoo.com; xhyuan@ncat.edu
** Comtech EF Data Corporation, Tempe, AZ, USA; e-mail: wanglinsong@yahoo.com
*** Andrew Pearson Industries, Winston-Salem, NC, USA; e-mail: goldiono888@hotmail.com

1. Introduction

Today’s technologies require ever more high-quality, highly accurate geospatial-related data, such as topographic data, aerial photography, cartographic data, satellite and satellite derivative data, earth science data, hydrologic data, forest data, and land-use/landscape data, to be collected and created. Scientists and engineers have been working on visualization technologies for over 20 years, and many sophisticated algorithms and applications have been developed. Thus, imagery/geospatial data visualization technologies have been implemented [1–5]. NIMA’s Terrain Visualization tool PowerScene™ can create, in real time, highly accurate, geo-specific scenes for viewing 3D data on the workstation [5]. Autometric Inc.’s EDGE™ product family can provide multiple functionality to import and display terrain models for terrain delimitation and line-of-sight analysis; to enable interactive 3D simulation, modelling, visualization, and remote sensing of airborne, ground, and space-based objects; and to analyze and display 3D data fields [4]. Visualization technologies encompass the effective capture, interpretation, analysis, representation, and communication of imagery, imagery intelligence, and geospatial information. Most of the current implementations of 3D imagery and geospatial visualization systems do not allow a user to interact with the system in a secure web environment. The user is not able to retrieve remote data, collaborate, and share views with other users using the same system.

With the Internet becoming the popular medium for information publication and sharing, more and more research is being undertaken on web-enabled visualization [6–9]. The World Wide Web is a network infrastructure that allows users to access and disseminate various documents in different forms that can be text, graphics, video, and audio information. Therefore the Internet/intranet can be a key mechanism for better use and distribution of information internally and among customers. Very little research has been done to support web-based 3D imagery and data visualization. Goddard and Sunderam [6] show...
their work on web-based 3D collaboration by ToolSpace. Wood, Brodlie, and Wright discuss several visualizations over WWW models, which are: a publisher creates the visualization as an image, a publisher creates the visualization as a 3D model, and a publisher creates visualization framework and a viewer. They applied one model to environmental data [9].

With the dramatic growth of the Internet and web applications, web security has become a major concern throughout the world [10–13]. Web servers make an attractive target for attackers for many reasons: publicity, commerce, proprietary information, and network accesses. Therefore, web security becomes one of the biggest issues we face today. Application-level security vulnerabilities are well known, and some articles discuss different ways to solve these problems [10, 11, 13]. Rubin and Geer [12] discuss several problems in their survey. These are server security, securing the host, securing data transport, and mobile code security. Scott and Sharp [13] discuss the application-level vulnerabilities and seek a common way to solve these problems. Probst, Essmayr, and Weipl [11] develop a framework named Generic Authorization Mechanisms for Multi-Tier Applications.

Software engineering is rapidly changing towards the notion of component assembly and integration. It brings into software well-developed ideas that are “plug-and-play.” Object-oriented technique and design patterns are software engineering technologies that have been used to develop real-world applications for years [14–16]. Therefore, using object-oriented system design methodology to build linkages between visualization technology and the WWW has come to fruition.

A secure geospatial visualization and collaboration system has been developed. It takes advantage of the various software engineering, visualization, security, and web technologies, and gives users on the Internet the ability to retrieve geospatial data from remote data source, and manipulate 3D views on their local machines. This system also provides a cooperation capability for a group of people to fulfill specific tasks, as in real-time communication via the Internet.

In the second part of this paper the system architecture will be studied. The secure system configuration will be discussed in Section 3. The modular structure of the client tier will be presented in Section 4. The server tier will be described in Section 5. Experiments and results will be discussed in Section 6 and conclusions given in Section 7.

2. System Architecture

A Secure Geospatial Visualization and Collaboration System (SGVCS) has been developed that is platform independent. Because of the nature of 3D visualization—a user manipulates 3D views that are continuously generated by 3D data processing—the architecture presented in this paper is a three-tier, thick client web application architecture. The architecture is shown in Fig. 1.

These three tiers are the client tier, the server tier, and the database tier. The client tier is a fully functional application that provides rich graphical user interface with the server tier and visualization manipulation. The server tier is responsible for data retrieval and provides data protection by user information authentication and authorization. The database tier includes the user information database and multiple geospatial databases, which may or may not reside on the same physical machine.

A user issues a request that includes the interested area and data type through the Application Module. The Visualization Coordinator receives the request, calculates the values of longitude and latitude of the original point and width and height of the area, packs this information with group information into a DataRequest object, and sends it to the Visualization Web Bridge (VisWebBri). The VisWebBri checks the user’s privilege through the User Authority Control and then decides to give permission to the user or not. If the request is a group activity the VisWebBri sets up a group flag, and then passes the user request to the Data Provider. The Data Provider is responsible to retrieve geospatial data from database(s), and then sends the retrieved results to the VisWebBri. The VisWebBri passes the retrieved results to the Visualization Coordinator, and then the Visualization Module receives the retrieved information and executes the display task.

The system supports multiple-level users: administrator level, group supervisor level, and general user level. Each level’s user has a corresponding privilege to access geospatial databases. Users can hierarchically communicate and collaborate. Through the Visualization Module a user can manipulate the image. The system also provides multiple 3D display windows to give users flexibility to view several interested areas in different aspects at the same time. Through the Collaboration Module a client can share the same image and information with the members in the same group, a group supervisor, and a system administrator simultaneously even if they are geographical dispersed.

3. Secure System Configuration and Data Transmission

In a distributed system the first place that management affects security is the system configuration. An insecurity system configuration can cause many holes and attacks. Also, as the system becomes more complicated the prob-
lem becomes heightened. In general, the web-server configuration files live in the server root. Configuration files are composed entirely of directives and explanatory comments. Directives control the file that includes user names and passwords and the file that contains group names and passwords, as well as access to the file in the document tree, including default permissions and how to override them locally. Running a web server as root is a common mistake that can create vulnerability.

The platform-independent secure visualization and collaboration system is implemented based on the client-server architecture that is currently running in a Windows environment. Because of the platform-independent feature of Java, this system can run on any other operating system. Apache® software is used as the web server and Tomcat® as the servlet container. In this type of application system the server is the central system and the repository of information resources. Therefore the server is the locus of threats. In order to secure system configuration the Apache® server runs as a genuine user 3DVisServer with both a unique user ID and membership in a special group SecureVisua. The home directory of 3DVisServer is the server root and includes the document root. Therefore, the web server runs with a group ID SecureVisua and has the right to access all files it needs. It may include the server log and server configuration files.

Packets of information travelling through networks are vulnerable. To ensure that no one has tampered with or monitored the information that you requested, the Secure Sockets Layer (SSL) is used in this system. SSL relies on certificates that include the name of the certificate authority who issued the certificate, the name of the entity to which the certificate was issued, the entity’s public key, and time stamps that indicate the certificate’s expiration date. The public key is used to encrypt information and the private key is used to decrypt it. With Java Secure Socket Extension (JSSE) [17] packages, Rivest-Shamir-Adelman for authentication and key exchange, Data Encryption Standard for bulk encryption are easily plugged into the system to enable secure network communications. Before any data transfer, a secure sockets layer handshake authenticates each other and establishes the encryption method and a unique session key to guarantees data privacy and data integrity.

4. The Modular Structure of the Client Tier

The client tier is a Java application that provides multiple GUIs for an Internet user to interact and communicate with the server based on request and response patterns using the HyperText Transfer Protocol Secure (HTTPS) protocol. The returned geospatial data is visualized and manipulated on the client machine. According to the functionality, the client tier application is composed of four modules: the Application Module, the Collaboration Module, the Visualization Module, and the Visualization Coordinator. The modular design of the client tier is shown in Fig. 2.

The Model-View-Controller (MVC) design pattern is used in the design of the Application Module, the Visual-

Figure 2. The modular structure.

ization Module, and the Collaboration Module [16]. These modules can be reused in similar applications. In addition, the communication among these modules is maintained by an instance of observable/observer object model, the Visualization Coordinator is the observable object, and the other three modules are the observer objects. The Visualization Coordinator provides function calls sending requests to the server; and the other three modules are notified by the Visualization Coordinator with the object returned from the server. All the information exchanged within a module or between two modules is wrapped into a Java object.

4.1 The Application Module

The Application Module provides basic user interaction to start and set up the application environment. It allows a user to log in, log off, and exit from the system; select an area of interest on the 2D map and request geospatial data; and select specific data types to be returned from the server. It also allows a supervisor to add, delete, and update user information. The structure of this module is shown in Fig. 3. All these functionalities are invoked from different interfaces of the Application GUI.

Figure 3. The application module.

The Model of the Application Module contains the Current Logged User, the User List, and the Data Type Selected. Each interaction involves data exchange between components within the Application Module and between the Application Module and the Visualization Coordinator. All server interaction is taken care of by the Visualization Coordinator. The Application GUI is responsible for making sure the clients are authorized to use the system. A user can log in as an administrator, supervisor, or general user. Once authorization is established the user information (user name, password, and access privilege) can be retrieved from the User Information Database. This
information is passed through the system as long as that user is logged in. The Application GUI provides function bars and a 2D map in which the user can select an area of interest. The function bar supports operations such as Log In/Off, Data Type Selection, 2D Map Request Data, and Collaboration Mode. Once a user clicks the Data Type Selection button a pop-out window lists all available data sources for this user. A user can check the interested data type, and then use the mouse to select a region from the 2D map for retrieving the corresponding geospatial information. Fig. 4 is a snapshot of the Application GUI.

**4.2 The Collaboration Module**

Various visualization applications that have special requirements for collaboration may be in different levels. One example is military and/or hierarchical order of command and control structure. In this kind of structure subordinate agents will literally be in the field, while supervisors can and will be many miles away. Transmitting data back and forth will allow the supervisor to “see” what the clients in the field are viewing on their individual computers, and/or instruct or command that the supervisor’s view be seen or projected onto all the clients in the field. In this scenario, the supervisor will have authority/control over a fixed number of clients or a listing of potential clients. Another example is real-time collaboration. In a battle field or emergency scenario clients need to view and get updates and feedback through 3D imagery on the progress of both friends and foes in the field.

The secure collaborative visualization and communication framework named Collaboration Module establishes uniform collaboration among a group and supports real-time collaboration. This module allows an authorized user to communicate and share 3D geospatial map information with other users in the same group. There are two architectures by which this collaboration may be achieved: the client/server architecture and the peer-to-peer architecture. The Collaboration Module integrates the above two architectures. It enables users to exchange information over the Internet without any server involvement using the collaboration GUI, but each user has to log on to the server to authenticate himself/herself to join the collaboration mode. The peer-to-peer communication is implemented with secure socket connections, and the client/server communication goes through the HTTPS protocol.

Fig. 5 shows the structure of the Collaboration Module. Users exchange messages through the Collaboration Controller, and log on to the server through the Visualization Coordinator. The data model contains Joined Users representing other users who have already joined the collaboration mode, and the Current User representing the currently logged user in collaboration mode. The information exchanged among joined users in the same group is wrapped into an object of one of the three classes: Message, DataRequest, or ChatUser. Users in the same group can use an object of class ChatUser to notify each other of a join or leave of the collaboration mode, use an object of class Message to chat with each other, and use an object of class DataRequest to synchronize the 3D views.

**4.3 The Visualization Module**

The Visualization Module is loosely coupled into this application. It only receives the geospatial data from the Visualization Coordinator and does not exchange data with other modules. The Visualization GUI gives a user the capability to do view manipulations provided by the Visualization Engine, which include zoom in, zoom out, pan,
and rotate the 3D image with the mouse and keyboard; exaggerate terrain elevation and enable/disable longitude and latitude box, grid, and multiple windows; and enable the contour line mode to do terrain analysis. Fig. 6 shows the internal structure of this module.

Figure 6. The visualization module.

The data sent to this module are wrapped into an object of class *DataRendering*, which may include several different data types covering the same geographic area. The Visualization GUI is composed of a mobile 3D control panel and 3D visualization windows. To allow users the flexibility to display 3D images in a large area on the screen, the mobile display control bar is used. The 3D control panel allows a user to select display type such as smooth surface or contour line. A user can also change display mode and display effect, such as exaggerate terrain elevation, enable/disable box, grid, and multiple windows. Using a mouse or a keyboard a user can manipulate the 3D image, such as zoom in, zoom out, pan, and rotate the image.

The Visualization Engine, which is platform independent, is the key component for image display. It consists of a Data Adapter, Terrain Generator, Road Generator, Grid Generator, Colour Controller, and Visualization Manager. Fig. 7 shows the structure of the Visualization Engine. It receives the *DataRendering* Object from the Visualization Coordinator and displays the information as a 3D model according to the user’s requirements. It supports three layers: the terrain layer, road layer, and grid line layer. The terrain data is sent to the Terrain Generator and highway information is sent to the Road Generator. The Terrain Generator creates a 3D terrain layer based on the terrain data. The Road Generator creates the highway layer based on the received vector data of highways. The displayed terrain and roads are automatically updated according to the user’s selection. The Grid Generator creates grid lines to help a user locate the area. The Visualization Manager handles all manipulation operations. It consists of a transform class, a feature class, and a coordinative class. The transform class implements functions such as zoom in/out and rotate the image. The feature class implements feature changes such as box on/off, grid on/off, and road on/off. The coordinative class communicates with the Display Generator to implement functions such as display surface or contour line. The actual on-screen rendering is done by a single dedicated render thread. This thread runs asynchronously from all other threads to provide maximum performance and interactivity.

The Visualization Engine supports multiple independent views that can be mapped in different windows. Each view may display the scene from a different point of view such as the 3D view, 2D overview map, and so on, or may display different aspects of the scene such as the surface of the terrain, polygon terrain, or as contour terrain. For each view there is a view model that contains data and attributes and a thread that executes rendering tasks. Within each view, the scene that is to be rendered is well buffered. When the user switches the display mode the view can reuse the content in the buffer to improve the performance.

The Visualization Engine is designed and implemented using object-oriented techniques, Java 3D graphic package, and Geo 3D package [18]. It can be treated as a black box and reused in other applications. In order to improve the system performance, a threshold value is set to keep the value of point’s number limitation. The interpolation algorithm is used to guarantee the display quality.

4.4 The Visualization Coordinator

In the observable/observer architecture, the Visualization Coordinator (as in Fig. 8) is the observable object. Whenever it sends out a notification object, its observer objects, the Application Module, the Collaboration Module, and the Visualization Module, will all receive this notification object. They take action according to the type of the notification object and the information in this object.

Figure 7. The visualization engine.

Figure 8. The visualization coordinator.
The Visualization Coordinator serves as a liaison between the client and the server. It provides function calls, which receive an object from the Application Module or the Collaboration Module, and connects to the server to serialize and transfer the object over the HTTPS connection. Then the Visualization Coordinator retrieves the returned object in the HTTPS response from the server and notifies all of its observers with this retrieved object.

During the notification process, the Visualization Coordinator performs some extra work to add more functionalities of the client Java application and maintain the independency between three observer objects. The Visualization Coordinator communicates with the server based on a request/response pattern over the HTTPS protocol.

5. The Server Tier

The server tier provides functions to respond to those requests from the client tier, to access databases, to extract data information according to the user’s requests, and to pack and send retrieved data back to the client tier. The server tier also provides functions to implement user authentication, user authorization, and user information management. The server tier consists of three web components: the Visualization Web Bridge, the User Authority Control, the Data Providers, and the Shared Data Buffer object. The three web components take advantage of Java Servlet technology, which provides multithread capability to improve server performance. The structure of the server tier is shown in Fig. 9.

![Figure 9. The server tier.](image)

5.1 The Visualization Web Bridge

Geospatial Data Sets (GDS) contain rich and diverse information that is stored in various formats, such as tables, images, and text. Typical GDS software maintains, manipulates, views, and displays information in dynamic and graphical ways. GDS and the WWW are two independently developed application areas. A WWW user cannot access GDS information directly.

The Visualization Web Bridge (VisWebBri) is developed to bridge the gap between the Internet users and the geospatial databases. It consists of three components: Request Analyzer, Retrieving Controller, and Data Integrator. The client tier Visualization Coordinator communicates with the server tier by sending HTTPS requests to the VisWebBri. The data transferred in the HTTPS request are a Java object of class UserData, ChatUser, or DataRequest. Based on the received Java object, it provides three major interactions with clients: User Account Management, User Collaboration Management, and Geospatial Data Retrieving. If it is a login activity, the Request Analyzer checks the user’s access privilege through the User Authority Control. If the user does not have permission to access the interested data the request will be denied. If it is a shared data request the Request Analyzer checks the shared data table that lists all DataRendering objects in the Shared Data Buffer. If the requested data are in the buffer, the Request Analyzer reads the DataRendering object from the buffer and sends it to the Visualization Coordinator in the client side. Otherwise, the Request Analyzer passes the DataRequest object to the Retrieving Controller. The Retrieving Controller breaks the request into several objects according to the selected data types and generates a corresponding RetrieveType object. Then the Retrieving Controller passes these objects to the Data Providers to retrieve requested data. Once the Data Integrator receives retrieved data from the Data Providers, it packs these data into a DataRendering object, and then sends it to the Visualization Coordinator to notify the Visualization Engine for display. The Request Analyzer also checks if the current request is a shared data request; it updates the shared data table and saves the DataRendering object in the Shared Data Buffer for other members in the same group to use. This design avoids continued repeating of the same data and therefore improves the system performance.

5.2 The Data Providers

The Data Providers are actually responsible for the geospatial data retrieving from geospatial databases. Once a Data Provider receives objects of class RetrieveType from the VisWebBri, it launches multiple data-retrieving threads, accesses the databases, and retrieves geospatial data. These data are usually well organized into a file system with a hierarchical structure. Each Data Provider deals with one geospatial data type, which may be 3D terrain data or layered 2D data, and reads the geospatial data files and filters, reformats, and loads them into memory. The Data Providers monitor the progress of these data-retrieving threads. After all these threads finish, the Data Providers collect all information, such as terrain data and road information, and send them to the VisWebBri with the data coverage based on the request it receives.

5.3 The User Authority Control

The User Authority Control is a security component to guarantee that only permitted users can log into the system and access geospatial data. The functionality of the User Authority Control is composed of a login and user authentication process, a user authorizing process, and a security data access control process.

The login process allows the system to authenticate a user and to track vital information about the current user.
The Application Module provides a login interface to let a user key in his/her user name and password. Once a user enters his/her user name and password on the client side, Visualization Coordinator processes this information, generates a login control object, and passes it to the User Authority Control on the server side. The User Authority Control accesses the User Information Database and checks the login information. If the information is correct, the User Authority Control sends the login control object back to the Visualization Coordinator with granted access information. Otherwise the user will receive a reject message.

In general, an application requirement is to associate different privileges with different groups of users. The privilege is the permission to access different security-level data. This system supports multiple-level users, as mentioned above: administrator level, group supervisor level, and general user level. After checking the user’s privilege in the User Information Database, the User Authority Control authorizes the user’s privilege by assigning what kind of geospatial data this user can access. This information is included in the login control object.

It is not necessary to visit the User Authority Control every time a user requests geospatial data, because it will decrease the system performance. In this implementation the User Authority Control is accessed in two cases: the first case is when a user logs into the system, the second is when a user requests collaboration and sharing of information.

5.4 The Shared Data Buffer

The Shared Data Buffer provides cache and look-up of DataRendering objects. The use of the Shared Data Buffer can avoid unnecessary geospatial database accesses and improve the overall server performance. It also performs self-cleaning to avoid too much memory occupation.

The Shared Data Buffer consists of name-value pairs, in this case, syncID—object of class DataRendering pairs. Each object of class DataRendering can be referenced by a unique syncID that is created by the Collaboration Module on the client tier. Before a new object of class DataRendering is saved in the Shared Data Buffer, it will check the timestamp of each syncID already in the Shared Data Buffer. If the timestamp shows this syncID has already existed more than the given time, the syncID and associated object of class DataRendering will be deleted from the Shared Data Buffer.

6. Experimental Results

The Secure Geospatial Visualization and Collaboration System has been designed and implemented. Software engineering, object-oriented system design, visualization, web security, and Java technologies as well as the World Wide Web are used in the implementation [10, 15, 17–20]. During system development, the Java 2 Enterprise Edition (J2EE) was chosen as the development platform of this web application. Model-View-Controller design pattern was applied to optimize the system design and improve the system reuse. The system is currently running on a Window 2000 platform. The Apache/Tomcat server was chosen as the Servlet application server. Because of the platform-independent features of Java, this system can run on any other operating system without modification. The modular design also makes it easy to reuse primary components. Every module in this system can be treated as a black box and can be integrated into other application systems. Java Web Start is used as the deployment tool for this system because it allows applications to be launched independently from a web browser. Once the Secure Geospatial Visualization and Collaboration System is updated the Java Network Launching Protocol [20], that Java Web Start implements, enables the downloaded application to be updated automatically without any user interaction.

The Digital Terrain Elevation Data (DTED) level 0 and the Vector Smart Map (VMAP) level 0 data of the state of North Carolina have been downloaded from the National Imagery and Mapping Agency’s website to test this system [5]. The DTED level 0 describes the values of average, minimum, and maximum for the entire subcell, as well as the terrain elevation data. The DTED data file describes terrain as a one-degree by one-degree cell defined by whole degrees of longitude and latitude lines on the World Geodetic System. VMAP Level 0 is produced in Vector Product Format (VPF), which provides a standard format for storing digital vector cartographic data.

The SGVCS has been used to retrieve and visualize DTED and VMAP information. If a client who does not have privilege tries to log into the system, the user will be rejected and receive an error message. The system allows a user to try three times. The system also allows a user to use the mouse and the keyboard to select an interested region and to change the view of the 3D display. After “log in,” the users can choose the data types, such as DTED or VMAP, and select the area of interest from a 2D map. Fig. 10 shows the displayed image of a selected region. The red lines are the highways in that region. The system allows a user to manipulate a 3D image in various ways such as rotate, zoom, and transform. Fig. 11 exhibits the result after a user zoomed in, rotated, and exaggerated the original display. A user can see the details of the selected region. Fig. 12 demonstrates the display of the terrain with the values of longitude, latitude, and elevation. One

Figure 10. A selected region.
7. Conclusion

A Secure Geospatial Visualization and Collaboration System is developed that makes the wealth of information stored in geospatial databases available through the World Wide Web. The system integrates software engineering, visualization, web security, and Java technologies with the World Wide Web to enable production, dissemination, and use of 3D imagery and geospatial information on a hierarchical level through the WWW. MVC design pattern is applied to the Application Module, the Visualization Module, and the Collaboration Module. These modules can be “plugged in” and reused for other similar applications. The promise of Java as a “write once, run anywhere” technology is leveraged in this system.

The application results demonstrate that users have flexibility to query, visualize, manipulate, and analyze 3D geospatial data. It also shows that multiple users in geographically dispersed organizations can securely communicate and share information and images through this system. The system provides a certain level of security to prevent password attacks, secure information transmission, and guarantee authorized and authenticated data access.

Acknowledgements

This work was supported by the National Imagery and Mapping Agency under award number NMA201-01-1-2013 and by the National Science Foundation under award number DUE-0415571.

References

Biographies

Huiming Yu received her Ph.D. degree in computer science from the Stevens Institute of Technology, U.S., in 1992. Currently she is a professor in the Department of Computer Science at North Carolina A&T State University (NC A&T SU). She is the principle investigator responsible for several visualization and web security research projects. Her research interests include software engineering, information assurance, web security, visualization, and web applications.

Linsong Wang received his B.Sc. and M.Sc. degrees in mechanical engineering from Nanjing Aeronautical Institute, China, and Hefei University of Technology, China, in 1992 and 1999 respectively. He also received his M.Sc. degree in computer science from NC A&T SU in 2003. He is currently with Comtech EF Data Corporation, Tempe, AZ. His current research interests include object-oriented strategy in embedded system testing and software quality assurance.

Jing Zhang received his B.Sc. degree in computer engineering from Beijing Institute of Technology, China, in 1995. He received his first M.Sc. degree in industrial technology from Morehead State University in 2001 and his second M.Sc. degree in computer science from NC A&T SU in 2004. He is currently with Andrew Pearson Industries, Mount Airy, NC. His research interests include 3D display algorithm in GIS and human–computer interaction.

Jeremy Barksdale received his B.A. degree in broadcast production from NC A&T SU, Greensboro, NC, in 2000 and an M.Sc. degree in computer science from NC A&T SU in 2002. He is currently with Booz Allen Hamilton, McLean, VA.

Xiaohong Yuan received her Ph.D. degree in computer science from Florida Atlantic University, USA, in 2000. Currently Dr. Yuan is an Associate professor in the Department of Computer Science at NC A&T SU. Her research interests include software security, software engineering education, and visualization tools.