Design of a Service-Based Framework for Generic 3D Information Visualization

Xiangchuan Wang, Gerald Jaeschke, Matthias Hemmje
Fraunhofer IPSI
Darmstadt, Germany
{xwang, jaeschke, hemmje}@ipsi.fraunhofer.de

Abstract

A generic framework based on services for 3D information visualization is presented. The framework can support a wide range of applications and offers the flexibility to be tailored to specific application needs by configuration. The design and implementation details are discussed as well as a concrete application based on this framework.

1. Introduction

Information has no shape or color. Thus, one of the advantages of using diagrams is that virtual shape can be given to information to aid our comprehension and understanding [1]. Information visualization systems convert information to a visual format that is easier for us to perceive and understand. As applications vary, the information to be visualized is also different. Information visualization systems must deal with different source data. At the same time, the presentation is also changing with different preferences and criteria. It is not surprising that most information visualization systems are application-specific and developed from scratch, although the basic mechanisms remain always the same. This is especially true for 3D information visualization as 3D user interface standardization has not yet evolved as far as its 2D counterpart. The lack of adaptability of current 3D information visualization systems results in specific implementations. There is a strong need for a generic 3D information visualization framework that provides high-level functionality empowering application developers to rapidly create 3D information visualization applications. With the rapid growth of the World Wide Web, increasing massive information confronting a business or an individual opens up new opportunities and challenges for generic 3D information visualization frameworks.

In this paper, we present the work of our research group towards a generic framework to support 3D information visualization developments. After briefly discussing information visualization in section 2, the design of the framework is presented in detail. Finally, a specific application is given as well as deficiencies are considered for future improvements.

2. Information Visualization

Information visualization is the use of computer-supported, interactive, visual representations of abstract data to amplify cognition [2]. The strategy is to convert abstract data to a visual form that exploits human skills in perception and interactive manipulation. By perceiving and interacting with visual representations, it enhances our abilities to understand complex data and their relations. The key point here is that by using pictorial representations, information visualization can significantly enhance our ability to understand complex relations and structures that originally are hard for us to perceive and even more empower us to directly interact. Information visualization deals with how the data and functionality provided by an application are presented to users. Since information visualization is applied widely but its implementations are application-specific, there is a great demand to design a generic information visualization framework that can be applied to various applications.

On the other hand, 3D is another common topic on information visualization for its attractive large display space as well as natural and cognitive aspects [3] [4]. However, except for some special areas such as simulation, training, Computer Aided Design (CAD), 3D has not been applied as largely as expected. Besides the complicated and time-consuming 3D modeling, weak support of existing environments is another obstacle to the wide use of 3D [5] [6]. It is generally agreed that 3D would be the next main interface to the users. However, as it is still to be explored how to fully take advantage of 3D on the one hand, providing tools and frameworks that support the development of 3D information visualization applications is another aspect. This paper does not go into detail about 3D displaying and related technologies but elaborates on the framework that supports 3D visualization systems on the Web.

3. Design and Implementation

To develop a generic framework to serve applications of diverse domains, general scenarios that hold true for most information visualization systems have to be identified. First, most systems today are Web-based since HTTP has become a pervasive technology. Second, data to be visualized generally comes from databases and somehow is not semantically and syntactically independent. To improve reusability, such a framework should be independent on any application specific knowledge such as structure or content of the business data and concrete representations. Third, the representations should not be fixed and constant, but configurable instead.

3.1. Existing Technologies

Let's look at the common approaches to information visualization first. Because of the popularity of the World Wide Web, the browser/server paradigm has become the most commonly used architecture in today's applications. One can hardly imagine a business that does not handle HTTP data. In this way, most of the processes are carried out on the server. Various Server-Side-Includes (SSI) mechanisms appeared to support server-side processing. Usually, it includes data retrieval and creating relating visual representations. Data retrieval, processing, and translating are all computed directly on the server. It is unavoidable that in the outcome delivered to the client for display purposes, data and visualization are tightly coupled. Furthermore, different Web servers support different SSI mechanisms. For example, Microsoft advocates its Active Server Pages (ASP) while Apache Web Server plus PHP (which stands for "Hypertext Preprocessor") is popularized in the realm of Linux. The Java Servlet mechanism attracts more and more interest for its platform independent feature. In addition, to promote the interaction aspects on the client side, Java Applets form an appropriate approach. Combining Applets with Servlets seems to be a coherent and independent solution for most applications both on the server side and on the client side. Besides Java Applets, there are other techniques to improve the dynamic and interactive aspects on the client side. However, concerning large applications, Applets appear to be more powerful and well-supported on a variety of platforms.

As to the 3D representation, though the Extensible 3D (X3D) is under its way to become the standard and sooner or later will replace the Virtual Reality Modeling Language (VRML) [7], VRML is perhaps the most supported and used 3D format nowadays. Java 3D is another effective means to explore three-dimensional interactive world except for its demand for additional run-time environments.

XML (eXtensible Markup Language) [8] is especially useful for publishing on and data exchange over both Internet and Intranet. Since its appearance on the scene, it has been widely deployed for a variety of applications. A variety of tools have been marketed to support database contents publishing in XML [9].

As the standardization of XML and Document Object Model (DOM) has evolved, it has become rather easy to get or write an XML/DOM parser that analyzes XML documents. However, though the parser can read any well-formed XML document, the application semantics is still in need of dedicated program logics to process. In this sense, it is not surprising that each application has its own XML structure or definition as well as corresponding handlers. These kinds of work are more or less domain or application specific as the structure of the business data and the target representation are usually different.

3.2 System Architecture

Based on the above discussion, we persist in the browser/server architecture to support most of the Web applications (Figure 1). The advantages of XML are taken and exploited for data exchange and transmission. Java Applets are used for the interaction on the client side and Java Servlets are adopted on the server side for their platform independency, allowing to with the same programming language for both client and server. Considering the wide acceptance of VRML, we choose it to represent 3D information on the clients but go further. XML with embedded VRML is transferred to the clients so that we can get better control of the 3D scene.

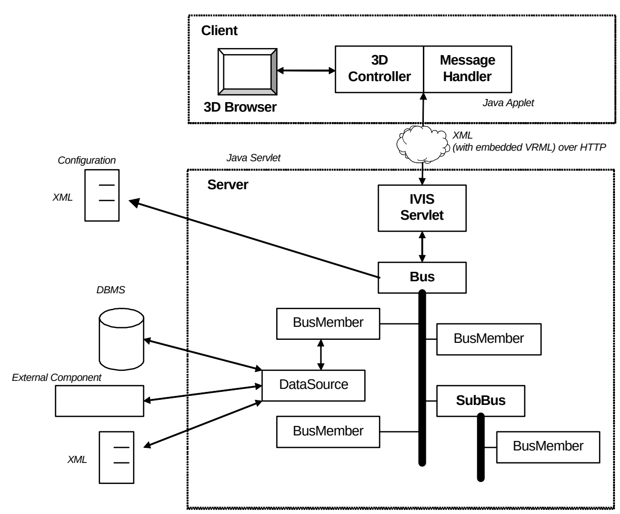

Figure 1. Architecture outline

Traditional communication between the server and the client is through various server-side programs (CGI, ASP, PHP, etc.). It takes usage of computing power of the server and is convenient for server management without affecting the clients. However, since the client still needs to know exact locations of the resources on the server and the resources are related with each other through links inside, the maintenance of documents on the server remains complicated. All links must be carefully checked and kept valid all the time. To decrease the dependence and make the requests transparent, server functionalities are provided in terms of services: Services are sets of server operations accomplishing specific functionalities. Since the communication between the client and server is based on services and not on explicit file locations, interrelationships between files on the server are reduced. By providing a general method of representing request/response interactions over HTTP protocol, the approach allows the server to be utilized as a common platform supporting other Web applications and leaves a large space for future extensions.

Requests for services and results returned are packed into messages that conform to a XML schema defined for the communication between the server and the client. Each communication side has its own XML parser to translate XML messages received and sent over the Web. Considering the congesting network and strained patience of the users, a different XML parser is used on the client side to minimize the memory footprint of the Applet.

On the server side, the Bus is used to manage and mediate different components that provide services. These components are called service-providers or BusMembers that are connected to the Bus or SubBusses. Generally, each BusMember provides a kind of service while each service can be served by more than one BusMember. Inputs and outputs of these service providers are DOM trees. The whole process forms a chain of DOM tree manipulations. The bus takes charge of maintaining the registration information of the services each BusMember provides. Moreover, it controls the data flow among BusMembers either sequentially or in parallel. When the IvisServlet starts up, the Bus is initialized and registers all services provided by the BusMembers according to the external configuration file. After the IvisServlet receives a service request, it hands it over to the Bus. The Bus extracts the service requested and generates a BusEvent according to the service name and BusMembers that register to provide such a service. Then the Bus calls the related BusMembers to get the required services accomplished. The final result is transmitted to the client.

On the client side, HTTP response messages are first interpreted by the XML parser. Then Message objects are generated and delivered to the right processes. Regarding to the VRML, there are different actions defined, including Avatar, Clear, Detail, Function, Insert, Viewpoint, etc. This kind of information is extracted from the response messages and used for communication with the server as well as controlling the 3D scene.

To deal with different 3D browsers on the client side, The BrowserControl class offers an interface that abstracts from the 3D renderer/browser that is actually deployed. Most of the general functions, such as bind / addViewpoint, get / addRoute, get / addNode, are included in this interface. The Controller class is inherited from Applet and is the main interface that can be operated from outside.

3.3 Features

The current implementation accomplishes the whole framework for communications on server side, client side, and between server and client. Application developers do not need to care about details of this communication but can concentrate on just defining services on the server side that will be requested by the client.

The whole framework is independent of any specific business domain. Currently, we mainly use it for supporting the fair business. The combination of the open XML encoding style and the pervasive HTTP protocol makes the framework extendable and suitable for most of the 3D information visualization applications on the Web.

Another feature of the design is the Bus. As its name implies, the Bus serves as the connection and transmission of data for various BusMembers. It leaves a large space for extension. Busses can be nested by making BusMembers a Bus themselves, forming a SubBus. This design allows structuring complex processes and arranging them either sequentially or in parallel.

3.4 Application

The original idea for this framework comes from the 3D Engine of FAIRWIS, an EU funded project to support online publishing and management of trade fairs. For the sake of supporting users in their struggle in getting a complete overview of the fair and an intrinsic impression of the place and organization, three-dimensional scenes of the fair venue and its elements are proposed. It supports visualizing all stands in a specific hall as well as high-lighting sets of stands that result from user queries. Considering the inconvenience to users to install various plug-ins or runtime environments, further requirement is zero-installation for most novice users. For veterans, it is also possible to use more expert tools for better performance and more sophisticated 3D rendering.

The framework presented above was successfully applied to the FAIRWIS 3D Engine. After having set-up the main services on the server such as “overview”, “hall1”, “highlight”, the client can explicitly request 3D scenes in terms of services. It has to be emphasized that these services are defined in the external configuration files exclusively, putting adaptation by configuration into practice. Thus it leaves the sources intact and grants application independency.

On the server side, once the IvisServlet receiving the service request, it calls the XML parser to parse the requesting XML messages. As described previously, the Bus keeps a list of services. For each service, there is also a list of service providers. The Bus then searches for all the service providers and asks them to process one by one. The final result of such chain-process is returned to the client via IvisServlet. In the FAIRWIS case, we have the XMLDataBusMember class and the DatabaseBusMember class to get the original DOM tree from either XML files or directly from the database. Then this DOM tree is processed by the XSLTransformBusMember class to translate it to a VRML-embedded DOM tree.

On the client side, the service is requested via function calls to the Applet. In line with the requirement of having a generic framework, users can ask for whatever services and pass the related parameters. As the current application scenario mainly aims at supporting Internet users, the adopted approach is to deploy an Applet that supports VRML for 3D displaying. This approach trades a longer download time for the need of pre-installed VRML browsers or silent plug-in installation.

To sum up, though the framework is mainly designed for the FAIRWIS, contrary to limiting the framework to this specific domain or specific application, the system is designed and implemented as a generic, flexible framework for common 3D information visualization.

4. Future Improvements

Currently, the whole framework supporting 3D information visualization developments has been set up based on Java Applet and Servlet technology as well as XML, XSL and VRML. What the application developers need to do is to provide the business data either as XML files or directly from the database and extensible style sheet files for transforming. To design the system as generic as possible, it is unavoidable to leave the business logic in the hands of the application developers. As a result, the framework itself does not include any domain specific knowledge or processes. But there is a lot of work for the application developers to fully accomplish a 3D information visualization system.

The next thing to do is to provide mechanisms that abstract from the information and visualization models and let the application developers define proper visual representations by means of configuration. This would be accomplished by providing a collection of predefined general 3D metaphors and mapping logics so that developers only need to determine what data to be visualized and what target 3D objects or attributes will represent the source data. For individual purposes and corporate-design friendly applications, developers have to be able to define and use their own visual models for the information visualization.

Especially for time critical cases like stocks, statistics, etc., information visualization systems have a demand for immediately reflecting changes of underlining business data presented to the end users. How to efficiently reflect changes on the generated user interface has not come to a conclusion yet. There are several general approaches concerning this issue. We consider database trigger mechanisms [10] as one of the most reasonable solutions to this.

5. References

[1] Hideki Koike, “The Role of another Spatial Dimension in Software Visualization”, *ACM Transactions on Information Systems*, vol. 11, no. 3, 1993, pp. 266-286.

[2] Card, S.K., J.D. Mackinlay, and B. Shneiderman, *Readings in Information Visualization: Using Vision to Think*, Morgan Kaufmann Publishers, 1999.

[3] Colin Ware, Glenn Franck, “Evaluating stereo and motion cues for visualizing information nets in three dimensions”, *ACM Transaction on Graphics*, vol. 15, no. 2, April, 1996, pp. 121-140.

[4] Andy Cockburn, Bruce McKenzie, “3D or not 3D? Evaluating the effect of the Third Dimension in a Document Management System”, *Proc. of CHI 2001*, pp. 434-441.

[5] D. B. Conner, S. S. Snibbe, etc. “Three-Dimensional Widgets”, *Proc. of symposium on Interactive 3D graphics*, 1992, pp. 183-188.

[6] Ralf Dörner, Paul Grimm, “Three-dimensional Beans – Creating Web Content Using 3D Components In A 3D Authoring Environment”, *Proc. of VRML 2000*, pp. 69-74.

[7] http://www.web3d.org/.

[8] W3C, Extensible Markup Language (XML) 1.0 (Second Edition), http://www.w3.org/TR/REC-xml.

[9] Elisa Bertino, Elena Ferrari, “XML and Data Integration”, *IEEE Internet Computing*, vol. 5, no. 6, November/December, 2001.

[10] Martin Leissler, Matthias Hemmje and Erich Neuhold, “Automatic Updates of Interactive Information Visualization User Interfaces through Database Triggers”, *5th IFIP 2.6 Working Conference on Visual Database Systems*, 2000.