Building Simulation

Seventh International IBPSA Conference
Rio de Janeiro, Brazil
August 13-15, 2001

DISTRIBUTED SIMULATIONS: AN OBJECT ORIENTED APPROACH

Eleni Primikiri and Ali Malkawi, Ph.D.
University of Michigan
Taubman College of Architecture and Urban Planning
Ann Arbor, MI, 48109, USA

ABSTRACT

During the past few years, our research team has developed techniques to solve problems associated with designing a collaborative evaluation performance tool environment. This paper will describe the development of a system (Web-IBEDO) that makes use of emerging communication technology. The system is being used as a model for a larger implementation of distributed multi-simulation environment.

Web-IBEDO is a computer program that supports the simultaneous integrated use of simulation tools and multiple databases, over the World Wide Web (www). This energy performance simulation system has been developed using the Microsoft Visual Basic ActiveX DLL technology that supports Object Linking and Embedding (OLE). The system's engine resides on a web server and is controlled by a web-browser. The goal is to deliver this tool to all the design actors via the network and thus enhance the capabilities of sharing specialized knowledge among the project participants.

INTRODUCTION

Recently, the common interest in the field of architecture is heading towards design-oriented and interactive computer models for the study of building energy use. The need for multiple accesses to simulation engines for building performance calculations over the network has become increasingly important, as the Internet has gained acceptance in science engineering and management. The applicability of these accesses is dependent on their implementation aligned with the level of understanding the designer has, as well as the desire to use these tools. As a result, several efforts were established in order to facilitate the communication among the design process participants, to conduct a better building representation that yields to an improved understanding of the design stages. Each of these approaches is focused on various aspects of the concept of shared design scenarios in architecture, and promotes in some sense collaborative design.

One approach is the development of collaborative environments with a focus on data exchange models. An example of such approach was the European COMBINE projects (I and II, 1990-95). The main purpose of this approach was to produce the first prototype of a future generation of Intelligent Integrated Building Design System (IIBDS) (Augenbroe, 1994). The COMBINE project succeeded in developing and demonstrating the functionality of integrated building design systems. It sought to play the role of the intermediate between emerging data technology and the supply of tools that use this technology in an industrial setting (Augenbroe, 1997). In general, the COMBINE project comprised the basis for many of the other approaches developed later and was the first step towards data exchange concept.

Another approach is information distribution with a focus on the documentation and communication process. An example of this is Web-PHIDIAS (McCall et al., 1994), a hypermedia-based, intelligent CAD system that delivers both CAD functionality and Design information on the web. It provides private (individual) as well as collaborative (group) drawing and text authoring spaces, with a variety of types authoring and viewing privileges for groups (McCall, 1999). Furthermore, it consists of an interactive Web-based client that serves as an interface to the PHIDIAS hypermedia server.

Several approaches have also been developed that allow the integration of multiple building models and databases used by analysis and visualization tools, through an object-based representation of building components and systems. Building Design Advisor (BDA) as an example of this approach operates as a data manager and process controller, allowing building designers to benefit from the capabilities of the multiple analysis and visualization tools throughout the building design process (Papamichael et al., 1998). Other attempts with similar structure that are currently functioning on the web as on-line calculation tools for energy analysis are the Home Energy Saver (HES) (Milles, 1997), and the Residential Ventilation Simulation Program (RVS) (Forowich, 1997). Both work under the same communication

framework, the Common Gateway Interface (CGI), which transforms the web from a collection of static documents into an interactive medium.

Other approaches seem to incorporate some of the aforementioned models. An example is the P3 project (Kalay, 1998). It supports an integrated collaborative design environment, which consists of three computational systems: a shared-product engine, a performance-evaluation system and a process-based component. It is intended to foster three computational constructs: Representation tools, Communication and Evaluation tools, and Negotiation tools (Kalay, 1999).

Finally, the development of a collaborative performance-based building design between geographically distributed users is the approach that is used by S2 project (Mahdavi, 1999). It constitutes an active, multi-domain application that incorporates an object-oriented design and a space-based environment for integrated building performance modeling (Mahdavi, 1999). The communication framework -the Common Object Broker Architecture (CORBA)- enables an efficient mode of communication among the different components of the application.

This paper describes an approach that utilizes objects to represent simulation engines. These objects reside in Client/Sever environment that facilitates potential distributed simulations.

PROJECT DESCRIPTION

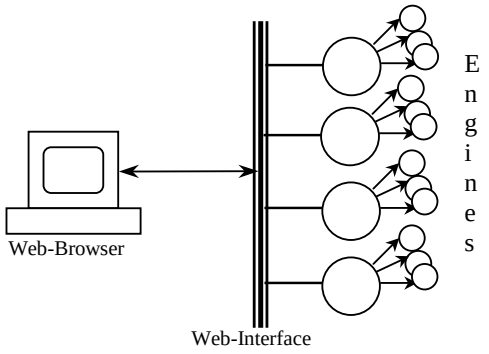
The project described in this paper is focused on the development of an automated technique that allows building simulations to be shared over the Internet. Furthermore, its purpose is to examine the operational basis of Object-Oriented-Programming (OOP) as a technique of an interface design that mediates the information exchange between the user, the application and the Internet.

Web-IBEDO is a computer program that supports the simultaneous integrated use of simulation tools and multiple databases, over the World Wide Web (www). Through the use of an Object Oriented approach, adopted for the realization of the simulation tool IBEDO (Malkawi, 1997), this web-application establishes all the communications of its designed architecture.
The objective is to generate a prototype technique for creating Web-Applications wherein the different components can reside on a Web-Server and can be connected with each other in an object-to-object interface design. For the purposes of testing and examining different techniques, IBEDO software was used. The reason for this choice was that the application was structured along the object-oriented programming approach.

TECHNICAL DESCRIPTION

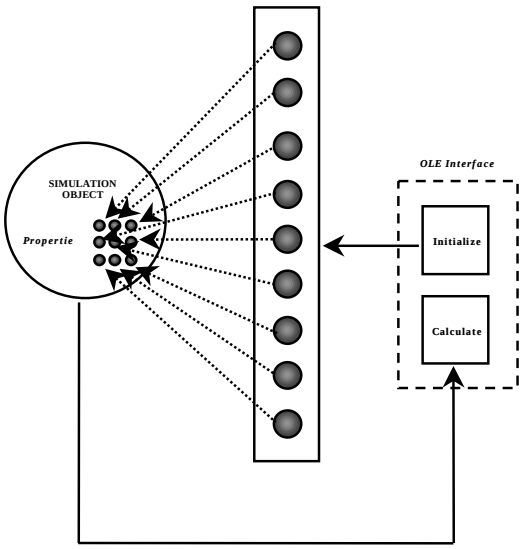
Web-IBEDO is being implemented to support its use as both a stand-alone structure as well as a module that can be connected to a larger environment. In its stand-alone form, the system provides layer to a single engine that is implemented as an object. The layer communicates with the object interface through its own interface functions. This would allow the object to reside on a web-server and then be invoked by a web-browser.

The module has an open structure that provides possibilities for integration with other modules and environments (Figure 1). Using this technology of having multiple engines as objects and interfacing with these objects through a layer of communication allows multiple simulations to run on the web.



(Figure 1: the communication of a larger environment that involves multiple engines.)

Web-IBEDO is a reengineered version of IBEDO that can be easily adopted to support Intranets. IBEDO is an energy performance simulation tool whose computational engine exists as an object. Modules and databases exist as distinct “objects” in memory providing their data and function to each other upon request via an object-to-object interface. The simulation object has two public “methods” or functions that can be called to give or receive information and properties from the simulation engine. The first method, initialize, takes in an object that is a collection of simulation input values that represent the properties of a class. These properties can be set separately. The second method, calculate, performs the thermal simulation output for the building specified by the first data object’s description. (Figure 2).



(Figure 2: Structure of IBEDO, (Malkawi, 1997))

The system used in Web-IBEDO's communication framework was Object Linking and Embedding (OLE). This allowed an esoteric communication of multiple components (data files, Dynamic Link Libraries (DLL), applications) that have been developed individually. The same strategy was adopted in the effort to enable IBEDO to work as a Web-application by the use of a new technology of OLE in Visual Basic, called ActiveX DLLs.

Visual Basic supports Object-Oriented Programming, with the ActiveX DLLs that allows the communication among the different objects. In addition, Visual Basic's Internet Information Server (IIS) applications can be easily designed and function on the Internet as well as facilitate the communication of the object-engine with the Internet interface.

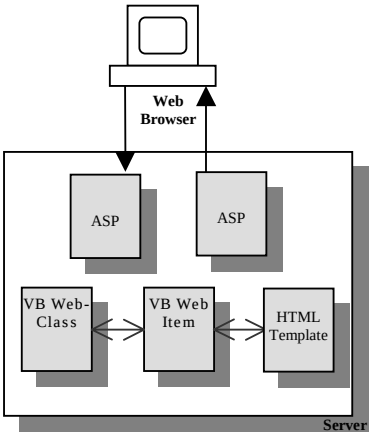
IIS applications are used for the development of server-side web-based applications. While this technology is new, it provides a combination of features that are extremely useful to the Visual Basic programmers seeking to develop server-side applications for use on Intranets, the Internet, or any IP-based networks.

IIS applications projects include a project reference to the Microsoft WebClass Library, which brings up the core of an IIS Application, a WebClass object. A WebClass object is a Component Object Model (COM) component that functions in conjunction with a WebClass Run-time Manager COM object provided by Visual Basic. The WebClass Run-time Manager COM component is provided by the Dynamic Link Library (DLL) files utilized by a compiled Visual Basic IIS application project. The WebClass Run-time Manager object provides the necessary run-time communication channels to IIS to allow WebClass object to become the new recipient of HTTP request. (Willis T. et al, 1999)

WebClasses contain a series of resources, which are used to send responses back to the client making the HTTP request. These resources are referred as WebItems. WebItems are child objects within the WebClass component that are typically associated with external resources. The external resources, such as an HTML template file, can be packaged up and delivered as HTTP packets in response to a client HTTP request. Typically, WebItems are HTML templates that are used as the foundation for the user interface, which is presented to the client. External HTML templates, which could be built by User Interface specialists, can be imported into the IIS application project. They can then be modified if necessary to fit the project requirements. (Willis T. et al, 1999)

The WebClass is encapsulated within an ActiveX DLL, and resides on the web server. By residing on the web server, the WebClass COM component can be easily instantiated in response to a request by multiple clients. The actual instantiation of the WebClass object is performed within a host Active Server Page, generated automatically by Visual Basic.

Active Server Pages (ASP) act as an interface application within IIS. Upon instantiation of the ASP, an instance of the COM component resides in the memory of the web server. By default the WebClass object only remain instantiated for the duration of the HTTP request. (Figure 3)



(Figure 3: Communication Framework of Visual Basic IIS application)

Web-IBEDO follows a fairly simple structure. It consists of 2 components, a browser and a server. All input and output occur on the browser. The IIS application resides on the server, connecting the browser, the data files, the engine of IBEDO and the application itself. The WebClass of Web-IBEDO undertakes the role of translating the input file, sending all the necessary information to the engine, launching the appropriate data files, and finally preparing the output file. The data files include Weather data, Weighting Factors data, Transfer Function Factors data, etc. The engine exists as a DLL file, which consists of all the modules that contain the solar calculations, Heating Load calculations, etc.

The application exists on the server as a dll file, and connects all these different components together, under a common interface design creating the appropriate input and output form.

Web-IBEDO provides a robust communication framework similar to the original IBEDO. The communication architecture between the different components, data files, DLLs and application of Web-IBEDO is illustrated in figure 3. In terms functionality, the user provides the building properties through the Web-Based Interface using an input form, or a data object. Once the information is posted, the web-application is called and initialized in parallel with the engine that resides as a DLL object on the server. The appropriate data files are loaded, and the calculations are initiated. Subsequently, the web-application provides the user with an output form. Upon completing this procedure, the user will be able to examine the thermal behavior of the given design. In addition, the user has the option to either reiterate the process or end the simulation.

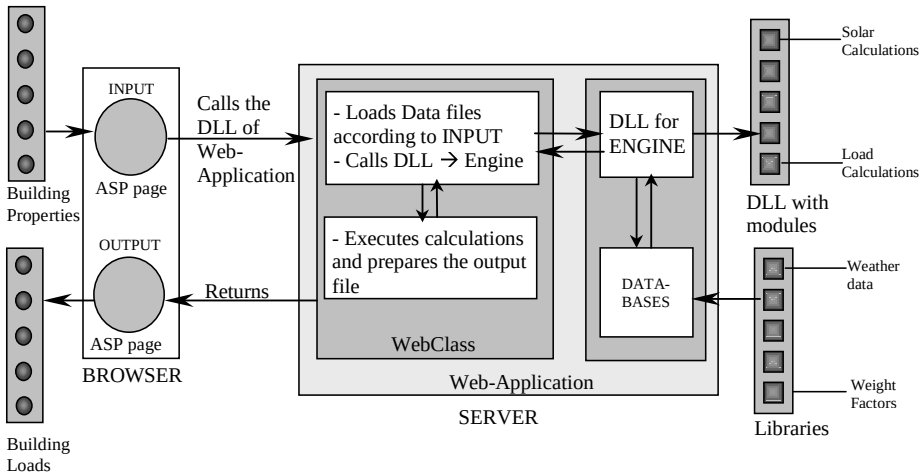
CONCLUSIONS AND FUTURE WORK

The benefits of this approach are established through its applicability, speed, ease of use and flexibility. The system developed demonstrated a methodology for creating web-based simulations.

As designers demand fewer constraints early in the design, environments that provide interface simplicity and accessibility become a necessity. The system presented in this paper attempted to solve such problems. It allows the user to navigate through a simplified interface that is linked to a server-based application that can be easily updated.

The utilization of the off the shelf object oriented development tools allowed the rapid development of the system that makes use of the object communications standards available. This communication allowed external databases to interact and hence, it permitted real-time operations within this application.

Web-IBEDO has been designed and pilot tested and is currently under implementation. Although the system demonstrated a success of utilizing off the shelf software development tools for creating web based simulations, future work needs to address data input and output as well as communication with other simulation tools.



(Figure 3: System Architecture with its major components and their underlying communication mechanism)

REFERENCES

Augenbroe G.L.M., " The COMBINE project", An overview, Proceedings of the first ECPPM Conference, Dresden, 1994.

Augenbroe G.L.M. & Sten de Wit, "Communicating Performance Assessments of Intermediate Building Design States", Proceedings of the 5th International Building Simulation Conference, Prague, Czech Republic, September 1997.

Forowicz T., "New Generation of Software? Modeling of Energy Demands for Residential Ventilation with HTML interface", Proceedings of the 5th International Building Simulation Conference, Prague, Czech Republic, September 1997.

Home Energy Saver Program web site, Lawrence Berkeley National Laboratory
http://hes.lbl.gov

Kalay E.Y., "P3: Computational environment to support design collaboration." Automation in Construction, 8 (1), 1998.

Kalay E.Y., "The Future of CAAD: From Computer-Aided Design to Computer-Aided Collaboration", Proceedings from the xth Computer Aided Architectural Design Conference, Atlanta, GA , August 1999.

Mahdavi A., "The Architecture of S2", Proceedings of the 6th International IBPSA Conference (BS'99), Kyoto, Japan, September 1999.

Malkawi A., "Platform Independent Simulations: Thermal Simulations as an object", Proceedings of the 5th International Building Simulation Conference, Prague, Czech Republic, September 1997.

McCall R., P. Bennett, E. Johnson, "An overview of the PHIDIAS II HyperCard System", Proceedings of ACADIA, 1994.

McCall R, "The Future of CAAD: From Computer-Aided Design to Computer-Aided Collaboration", Proceedings from the xth Computer Aided Architectural Design Conference, Atlanta, GA, August 1999.

Mills E., "Internet as a new Tool for Implementing Energy Efficient Lighting", Proceedings of Right Light 4, Lawrence Berkeley National Laboratory, US Department of Energy, CA, 1997.

Papamichael K., J. LaPorta, H. Chauvert, "Building Design Advisor: Automates Integration of Multiple Simulation Tools", Automation in Construction, 6 (4), 1998.

Residential Ventilation Simulation Program web site, Lawrence Berkeley National Laboratory,
http://www-epb.lbl.gov/ventilation/progrm.html

Willis T. et al, "Professional VB6 Web Programming", Wrox Press Ltd., Birmingham, BH, 1999.