ISBN 978-83-60810-51-4

A Web Service-Based Framework for an Online 3D Model Viewer

Hany F. ElYamany, Mohamed E. Elawady, Emad E. Abdelhay, Ahmed I. Khalil
Computer Science Department, Faculty of Computers and Informatics
Suez Canal University, Ismailia, Egypt
{hany_elyamany, mohamed_abdelaziz, emad4171, ahmed3943}@ci.suez.edu.eg

Abstract—**Building Information Model (BIM) is an IT methodology to construct a facility virtually in details. The digital format representation is essential to facilitate information sharing and exchange among multiple contractors. Industry Foundation Classes (IFC) is utilized to embed the shared data in an XML standard structure to improve the interoperability among all interrelated participants within the facility construction process. Web Service (WS) is an XML component which encapsulates and transfers a business process and its contiguous data safely, concurrently, and less-cost among several sides through accessing the web. In this paper, a web service-based framework is introduced to enhance the performance of IFC elements exchange and accessibility through the web. The framework employs multiple web service replicas in order to obtain a fast real-time 3D view. Each individual service replica retrieves a specific section of the IFCs information from the DB backend and draws its corresponding object. Experimental evaluation shows that the framework effectively enhances the time required to sketch the overall 3D view for the participant.**

Index Terms—Web Services and Systems, Building Information Systems, Performance and Scalability

I. Introduction

BIM (Building Information Model) is a digital representation of a facility that alleviates exchange and interoperability of information in a digital format [1]. This is quite unlike the past 3D software that had only geometric models. Using BIM tools, diverse participators and contractors (owners, architects, civil engineers, structural engineers, and schedulers) can share facilities' information throughout the lifecycle of a construction project as shown in Figure 1 [2].

Each of the utilized BIM Tools within the industrial market has its own file format (e.g. XML, CAD…etc) and hence, the embedded information inside those files would not be easily exchanged among the different participators. Approximately 4.25% of the entire work costs are wasted in the U.S Capital Facilities Industry due to the problem of incompatibility [3]. As such, the exchange and sharing of construction information are becoming salient issues in BIM Processes. To solve these challenges, IAI (International Alliance for Interoperability) has presented IFC (International Foundation Classes), which expresses facility information with the EXPRESS data modeling language as a standard construction information model [2, 3].

The term IFC is recognized for the underlying schema and for data content structured according to the schema [1]:

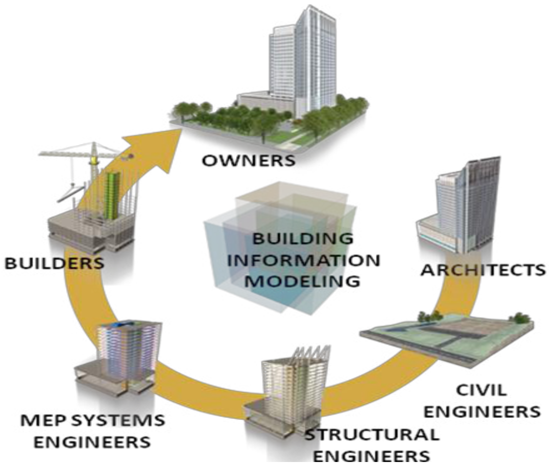


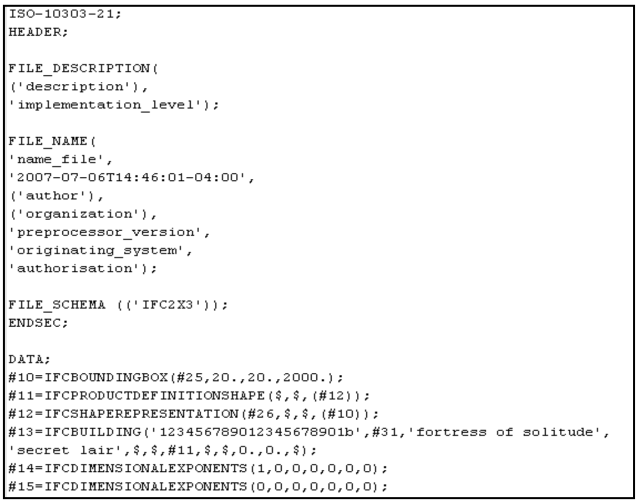
Figure 1. Building Information Model Relations

- The IFC schema is: an open-source, well-defined and industry-wide standard for identifying and exchanging construction and facility management precise information among different software applications, its main structure composed of the following basic components:
  - ✦ Class diagram that defines the existence of components and ideas in the construction and FM (Facilities Management) domain and the relationships that exist between – i.e. the graphical representations of the IFC specification.
  - ✦ Computer interpretable representation of the class diagram with additional constraints and requirements that can be utilized by solution providers to produce a specific executable code – i.e. the textual representation of the IFC specification.
  - ✦ Documentation that demonstrates the content and use of the IFC schema in details and that makes it easy to navigate within the IFC schema – i.e. the HTML based IFC schema documentation.
- The IFC model is: a data file or database whose structure is established by the IFC schema, usually the IFC file produced by software applications that are IFC compatible. It involves a well-defined illustration of the building (or engineering structure) that is an openly viewable BIM of that project and that may contain 3D, 2D or no graphical representations of the elements as shown in Figure 2.

Most of the BIM Tools from all the leading CAD vendors (e.g. AutoDesk, Graphisoft, and Visio) can export or import

978-83-60810-51-4/$25.00 © 2012 IEEE

IFC file formats, allowing users the ability to transform each BIM model into an IFC file format to exchange and share with each other. when IFC models were used in actual projects, three problems emerged: (1) increased time to exchange files due to increased size of IFC files as a result of increases in the size of models; (2) loss of information in IFC files due to the use of different IFC translators in diverse kinds of software; and (3) version control issues and the authority to use files. Due to these problems, information models became less consistent and were unable to provide necessary information throughout the entire processes of the projects [2, 4].



```
ISO-10303-21;
HEADER;

FILE_DESCRIPTION(
('description'),
'implementation_level');

FILE_NAME(
'name_file',
'2007-07-06T14:46:01-04:00',
('author'),
('organization'),
'preprocessor_version',
'originating_system',
'authorisation');

FILE_SCHEMA (('IFC2X3'));
ENDSEC;

DATA;
#10=IFCBOUNDINGBOX(#25,20.,20.,2000.);
#11=IFCPRODUCTDEFINITIONSHAPE($,$,(#12));
#12=IFCSHAPEREPRESENTATION(#26,$,$,(#10));
#13=IFCBUILDING('12345678901234567890 1b',#31,'fortress of solitude',
'secret lair',$,$,#11,$,$,0.,0.,$);
#14=IFCDIMENSIONALEXPONENTS(1,0,0,0,0,0,0);
#15=IFCDIMENSIONALEXPONENTS(0,0,0,0,0,0,0);
```

Figure 2. IFC Sample Data [5]

World Wide Web Consortium (W3C) [6] describes Web Service (WS) as “a software system designed to support an interoperable machine-to-machine interaction over a network. It has an interface described in a machine format; specifically WSDL. Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards”.

This paper describes Web Service-Based IFC Viewer framework which shares IFC data visually among different end users through web services in order to solve the previous problems: (1) Allow real-time partial data exchanging needed to be processed by participating users of the same team, (2) Provide the same web-based tool for all users to standardize IFC translations, and (3) Support centralized version control and several security roles in such a way that would let the administrator grant privileges for each user to access some (or all) IFC data.

This paper provides specific details about our Web Service-Based IFC Viewer framework and its service-based architecture. Related work will be presented in section 2. Section 3 describes the structure of Web-Based BIM framework. Section 4 illustrates how our framework is implemented and states the hardware & software specifications used in its application. Section 5 contains a summary and presents future work.

II. Related work

Several research projects have implemented client-server architecture for collaborating on BIM models. Perhaps the most notable of these was the IFC Server from the VTT Technical Research Centre (Finland) and IMSvr project of SECOM (Japan) [7]. The project was an Internet-based model server that built upon two significant standards, the IFC building data format and the Simple Object Access Protocol (SOAP) web services protocol. Leveraging these existing industry standards made the task of creating software simpler because developers could reuse existing knowledge, documentation and tooling. Unfortunately the project never got beyond the research lab, but its concepts still alive for further projects.

A consortium of European organizations (TNO, VTT, and CRISO) committed to developing the concept further by producing BIMServer based on JAVA technologies [8]. Alongside these research institutions, several business companies have expressed an interest in making compatible client software to explore the potential of the concept in practice.

Several companies (GraphiSoft, Autodesk…etc) has released a commercial version of the previous mentioned product BIMServer to enhance some presentation and security issues. Moreover, it supports project migration from the oldest versions of their products implemented by earlier projects such as the Revit project in Autodesk to be compatible with it [9].

WISPER project [10] has introduced a web and IFC-based distributed computer integrated environment that supports CAD design, visualization, estimating, planning specifications and supplier information. WISPER has enabled users to exchange project information through STEP standard and share the information through the IFC database.

University of Salford’s School of Construction and Property Management has developed an IFC application as a part of the British Engineering and Physical Sciences Research Council (EPSRC) in order to implement BIM methods and technologies to preview the entire building designs on different ways of construction [11]; including integrating the lifecycle facility costing into nD modeling .

However, the main drawbacks of all listed previous projects are: (1) Lacking compatibility with modern technologies and most recently software products. (2) Lacking in interactivity and producing a non-immersive virtual world. (3) Requiring a necessary installation such as the case in [10] as a particular software component must be installed on participants’ machines before beginning any project use. This paper solved these drawbacks by: (1) The proposed framework is compatible with the recent software technologies. (2) Direct, easy and interactive access to the 3D facility graphical representation within considering the security, performance and scalability requirements. .

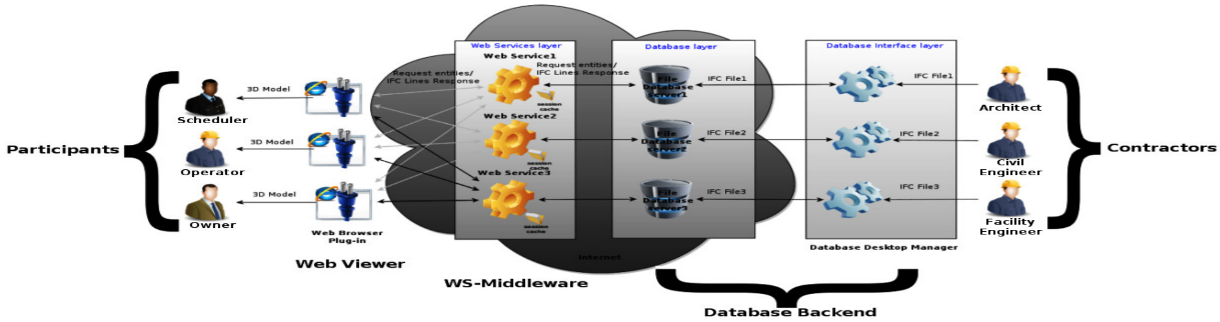


Figure 3. The Web Service-Based IFC Viewer Framework

III. Framework Description

The architecture of Web Service-Based IFC Viewer framework is constructed based on a three-tier client–server infrastructure to define the relationship between the Database Backend, Web Service-middleware, and Web Viewer that allows partial retrieving and querying of IFC project. Figure 3 shows a general overview of the Web Service-Based IFC Viewer framework. Each tier of the suggested framework is designed to be isolated for providing several benefits such as maximum control, scalability, and flexibility [12, 13]. A detailed description of each tier is discussed as follows:

The Database Backend: It accepts and holds contents of the IFC files, and then facilitates data retrieval. It includes two main components: Database Storage and Database Desktop Manager. The Database Storage has the following specifications:

- Saves multiple IFC files; each file is identified with a unique identity.
- Searches the IFC files' contents by different parameters (IFC entity, IFC Line number, or entity parameters).
- Performs analytic storage for each IFC file that establishes all possible relations among its contents.

The Database Desktop Manager basically connects providers' IFC input files to corresponding Database Storage and generates an HTML code, sent as a message, necessary for participants to monitor the graphical presentations of a requested IFC file through a web viewer.

The Web Service (WS)-Middleware: This layer is built on utilizing the technology of web services in order to facilitate the intercommunication between the participant's Web Viewer and Database Backend. The Web Service-Based IFC Viewer framework dynamically performs the following functionalities through the usage of web services:

- *Session Management:* Any participant could open more than one browser page simultaneously. In other words; there is more than one browser plug-in for each user at the same time. Each Web Service opens new sessions in the beginning of execution for every browser plug-in to keep track of what objects transferred to each currently executed plug-in.
- Data Transfer: Any browser plug-in receives the size of data packets of requested IFC file, and consequently request IFC elements in such a way that repeated objects inside all elements should be sent only once during the same session.

The Web Viewer: Every participant has a web browser installed with the operating system, activating a particular designed browser plug-in that will run as a utility component to draw and show the 3D graphical model for target facility. Moreover, it will enable the IFC provider admin to collect the required credentials about the involved participants in order to preserve the necessary level of security needed during the IFC data transfer process. The suggested plug-in component will particularly accomplish the following operations:

- *It* takes parameters of the web service endpoint including the IP address and port number of the server which publishes the web service itself (e.g. *IP:* 10.23.43.3, *port:* 2030) in order to interact with the specified WS-Middleware.
- It takes the parameter name, which expressed as a hash key to preserve the security of the IFC file, of the target IFC file maintained inside the target Database Storage in which the involved web services within the WS-Middleware interacts with (e.g. *IFC file name:* g34wds45fc23q).
- It receives IFC file objects of a particular facility through parsing the incoming SOAP messages from the involved web services running inside the WS-Middleware. Specifically, it enables a participant to view every distinct facility after collecting all its related IFC objects. For example, it receives the walls, windows, doors…etc as objects constructing the target facility that a participant aims to recognize.
- It converts the transferred objects of the target facility from IFC format into graphics format, and then renders the drawing context. Typically, it produces a 3D facility viewer as requested by each participant.

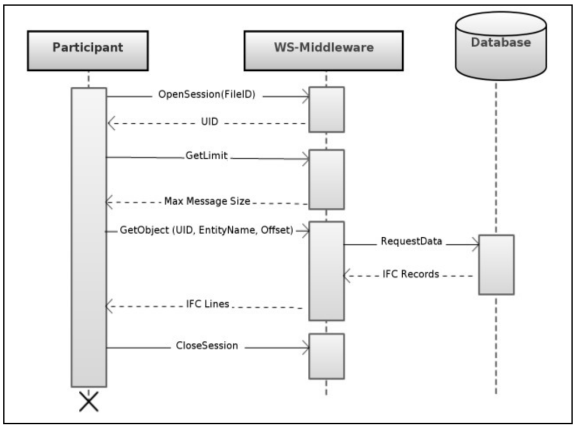


Figure 4. A Sequence Diagram showing the 3-tier framework Interaction

Figure 4 introduces a sequence diagram for demonstrating the main executed activities between the three layers involved inside the Web Service-Based IFC Viewer framework. Every time, each participant implements the following activities in such an order to obtain the facility data he/she aims to check:

1. The participant initializes the communication with the WS-Middleware. It requests opening a new session and determining the target IFC file. The running web services inside the WS-Middleware execute the specified operation named **OpenSession** in order to open new session, and then replies with new unique session identity for that session.

2. The maximum size of transferred SOAP messages between the WS-Middleware and participants have to be initially estimated based on some metrics including the hosted web services servers' performance and the network capabilities. The participant can get this size through consuming the operation **GetLimit** embedded within the running web service in the WS-Middleware.

3. The main activity presents transferring the IFC data from the Database Storage to a participant passing through accessing the operation entitled **GetObject** within the running web services in the WS-Middleware. Participant passes the following parameters to the operation **GetObject** in order to effectively accomplish the data transfer process:

   - **Session Identity**: to recognize each participant's requests.

   - **Entity Name**: to specify the needed objects representing the different entities located in the required IFC file. A detailed example of this process is shown later in implementation section.

   - **Offset**: to bookmark the last IFC line transferred to participant.

   The WS-Middleware processes the incoming parameters, and then sends the needed query to the Database Storage located inside the Database Backend in order to get the required facility data. Therefore, it saves the transferred data in a temporary file using the specified IFC format. Finally it sends SOAP messages encapsulating the required data which don't exceed the pre-determined size limit.

4. Before session termination, the participant enquires the WS-Middleware to close his/her opening session through calling the web service operation **CloseSession** which consequently, delete the session data and empty the session slot for another participant access.

IV. Implementation and Disscussion

As shown in Figure 3, the suggested framework consists of 3 layers: the Database backend, WS-middleware and participant. First, the Database backend has been designed with SQL Server 2005; connected to a database desktop manager which has been implemented by Java technologies in order to be compatible with any platform. Second, the WS-middleware has been constructed and implemented through using Visual Studio 2008, SQLite Database and ORM: Entity Framework [14]. The Visual Studio is utilized to building the Web services, SQLite Database for storing the transferred data during runtime and ORM: Entity Framework for optimizing the query coding. Finally, the participant is viewing the 3D graph of the target facility through a web browser which embeds an ActiveX control that consequently utilizing the Directx9c Graphics Library for drawing the 3D facility model. Each vertex within the model is sketched using TNO's IFC Engine [15].

Although IFC is constructed using an XML standard language to facilitate the facility information exchange, however, the XML language lacks from redundancy in storage, transmission, and processing [16] that leads to use a relational database engine such as SQL Server 2005 for IFC representation in this work. The database schema for the IFC representation within the database backend layer is depicted in Figure 5.

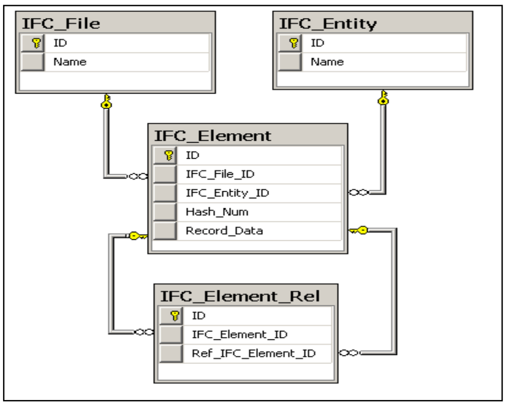


Figure 5. The IFC Database Schema

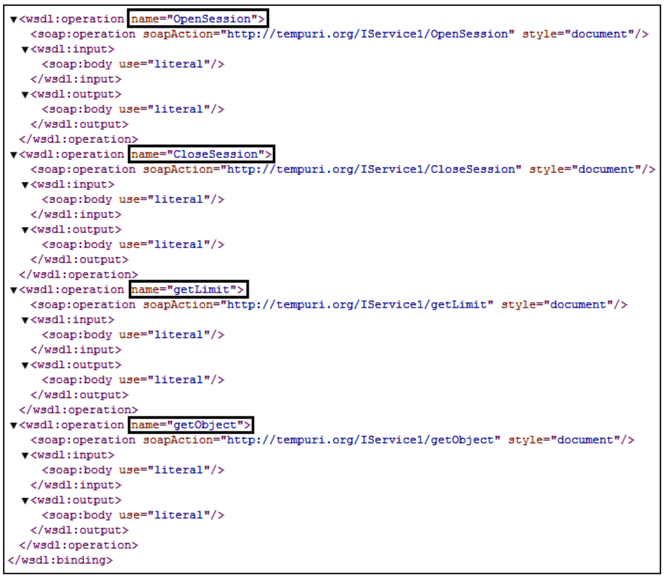
The IFC Database schema contains four entities:

- **IFC_File**: Holds names of IFC files.

- **IFC_Entity**: Stores all names of entities that can be used by any IFC file. For example, a door in a facility is considered an an entity within the IFC file.

- **IFC_Element**: Saves the data of an IFC element contained in an imported IFC file and referred by IFC entity. Each element inside an IFC file has a hash number and record data. For example, as shown in Figure 2, the record (IFCDIMENSIONALEXPONENTS(1,0,0,0,0,0,0)), which expresses the quantities of length, mass, time, electric current, thermodynamic temparture, amount of substance and luminous intensity of the entity dimensions of the product of powers is enumarated by a hash number (#14).

- **IFC_Element _Rel**: Represents the relationships between elements of each imported IFC file. For example, as shown in Figure 2, the IFC element with hash number (#12) is referred to another IFC element with hash number (#10).

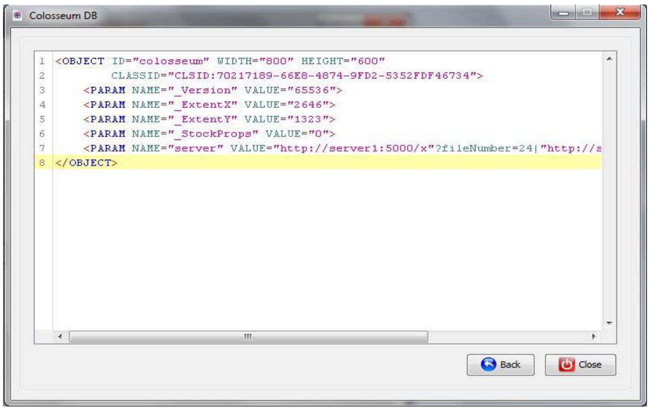


```
▼<wsdl:operation name="OpenSession">
   <soap:operation soapAction="http://tempuri.org/IService1/OpenSession" style="document"/>
  ▼<wsdl:input>
     <soap:body use="literal"/>
   </wsdl:input>
  ▼<wsdl:output>
     <soap:body use="literal"/>
   </wsdl:output>
 </wsdl:operation>
▼<wsdl:operation name="CloseSession">
   <soap:operation soapAction="http://tempuri.org/IService1/CloseSession" style="document"/>
  ▼<wsdl:input>
     <soap:body use="literal"/>
   </wsdl:input>
  ▼<wsdl:output>
     <soap:body use="literal"/>
   </wsdl:output>
 </wsdl:operation>
▼<wsdl:operation name="getLimit">
   <soap:operation soapAction="http://tempuri.org/IService1/getLimit" style="document"/>
  ▼<wsdl:input>
     <soap:body use="literal"/>
   </wsdl:input>
  ▼<wsdl:output>
     <soap:body use="literal"/>
   </wsdl:output>
 </wsdl:operation>
▼<wsdl:operation name="getObject">
   <soap:operation soapAction="http://tempuri.org/IService1/getObject" style="document"/>
  ▼<wsdl:input>
     <soap:body use="literal"/>
   </wsdl:input>
  ▼<wsdl:output>
     <soap:body use="literal"/>
   </wsdl:output>
 </wsdl:operation>
</wsdl:binding>
```

Figure 6. A portion of the WSDL of Web service in the WS-middleware

Figure 6 shows the Web Service Description Language (WSDL) of the deployed Web service running in the WS-middleware. The implemented Web service provides four different business operations to satisfy the participants' requests:

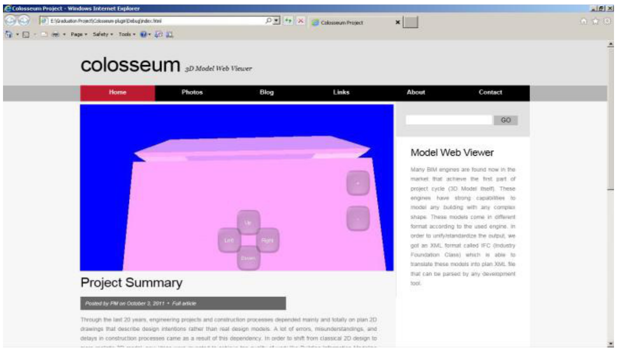
- **OpenSession**: is consumed to create a session for a participant request, determine the IFC file needed to be viewed and perform the needed pre-processing actions for the session establishment including unique identity generation and creation of temporary file database using the SQLite database.
- **CloseSession**: is accessed to close a participant session using its unique identity after sending all the required data of requested IFC file and destroy the temporary file database in order to preserve the security of the transferred data as well as enhance the performance of the utilized servers.
- **GetLimit**: is requested by a participant to obtain the maximum bytes of data that can be sent by the deployed Web service.
- **GetObject**: is used to send the IFC element data to the web browser seen by the participant through using the following parameters: participant identity, IFC element name, and last position of the sent data inside the target element since the data might be divided and sent over multiple times.



```
<OBJECT ID="colosseum" WIDTH="800" HEIGHT="600"
        CLASSID="CLSID:70217189-66E8-4874-9FD2-5352FDF46734">
    <PARAM NAME="_Version" VALUE="65536">
    <PARAM NAME="_ExtentX" VALUE="2646">
    <PARAM NAME="_ExtentY" VALUE="1323">
    <PARAM NAME="_StockProps" VALUE="0">
    <PARAM NAME="server" VALUE="http://server1:5000/x"?fileNumber=24|"http://s
</OBJECT>
```

Figure 7. A part of a HTML Message Resulting From Database Desktop Manager

As demonstrated in Figures 3 and 4, the contractors such as Architect and civil engineers add the IFC files of the target facility to the Database backend store through the Database Desktop Manager. Consequently, the Database Desktop Manager initiates the execution of the embedded ActiveX control within the web browser through generating a particular HTML message as shown in Figure 7 which specifically, in that case, determines the drawn area shown in the web browser, in addition of a "server" parameter defining the connection details to a particular web service endpoint as well as the target IFC file. Therefore, the participants including owner and operator are ready to access the ActiveX control in order to starting the viewing of the 3D model of the target IFC representing a facility as depicted in Figure 8.



Model Web Viewer

Many BIM engines are found now in the market that achieve the first part of project cycle (3D Model itself). These engines have strong capabilities to model any building with any complex shape. These models come in different format according to the used engine. In order to unify/standardize the output, we got an XML format called IFC (Industry Foundation Class) which is able to translate these models into plan XML file that can be parsed by any development tool.

Project Summary

Posted by PM on October 3, 2011 • Full article

Through the last 20 years, engineering projects and construction processes depended mainly and totally on plan 2D drawings that describe design intentions rather than real design models. A lot of errors, misunderstandings, and delays in construction processes came as a result of this dependency. In order to shift from classical 2D design to

Figure 8. A Snapshot of 3D Model Web Viewer

The ActiveX control consumes the Web services running within the WS-middleware in order to retrieve the IFC data. Each Web service receives the incoming request which determines the IFC file name and starts to retrieve the data part by part during runtime. Moreover, the Web service does not retrieve any element that has been sent before, while, it uses the first copy of the element in any requested drawing. Notably, a participant can open several browser web pages and ask for drawing different IFC files during runtime. In that case, multiple replicas of the main Web service would be generated dynamically in order to answer all incoming requests. The management of Web service replication is out of scope in this paper. An adaptive Web Services replication framework suggested in [17] could be utilized to manage the required web services replication process.

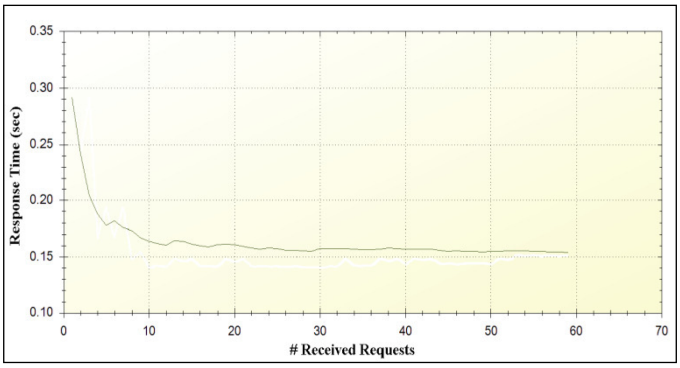


Figure 9. The Average Response Time of GetObject operation in the Utilized WS

Figure 9 describes the relationship between the incoming requests regarding the operation **GetObject** and the average response time. The suggested framework has been run 1000 times; in each time, a number of requests is chosen randomly. After finishing the experiment, the average response time is calculated for each number of requests.

As depicted in Figure 9, at the beginning of the framework initialization, the average response time is high due to the time needed firstly for starting the required sessions and figuring the saved database elements. Later, after the initialization point, the average response time is slightly similar whatever the number of requests is and how much the size of the transferred data is. As a recap, the proposed framework improves the performance and scalability of drawing the 3D facility model.

V. SUMMARY AND FUTURE WORK

In this work, a 3-tier Web Service-Based IFC Viewer framework is described and fully discussed. The suggested framework basically enhances the intercommunication among the different participants and contractors through facilitate the accessibility to the target facility information that both participants and contractors aim to construct.

The proposed framework contains a particular Web Service-Middleware in order to dynamically retrieve the facility information efficiently and therefore, draw its 3D model to be virtually available online for the related participants. The implemented experiment and outcome results demonstrate that the Web Service-Based IFC Viewer framework improves the performance and scalability of sketching 3D facility model through the web.

The future work can be summarized in three different aspects. First, convert the Database Desktop Manager from a desktop application needed to be installed on a particular contractor side into a web application that would be available to all involved contractors. The target web application will be managed with an authorization process to organize the interaction among the contractors and participants in order to increase the security of the transferred data. Second, replicate the data stored within the Database Server into other servers, when the main Database server is loaded or failed in order to guarantee a faster data retrieving process during drawing the 3D facility model. Finally, establish an application server to publish dynamically the web pages which contain a generated HTML code to draw 3D model for the target facility, and replies its corresponding links to the participants directly.

ACKNOWLEDGMENT

We would like to thank Eng. Mohamed Yamani, the CAD Tech. Manager, of Kemet Corp. in Egypt for his outstanding technical assistance and support in this work. Also, we would like to thank Ahmed Ghanem, Ahmed Hamed, Ahmed Shawky and Rami Abd El-Latif for their contribution in implementing the suggested framework in this research.

REFERENCES

[1] J. Wix et al, Guidance Report: IFC Support for Sustainability “STAND-INN Integration of Performance based building standards into business processes using IFC standards to enhance innovation and sustainable development”, EU project 2006-2008 under “Structuring the European Research Area”, 2007.

[2] H. Kang and G. Lee, “Development of an Object-Relational IFC Server”, ICCEM/ICCPM 2009, Jeju, Korea.

[3] M. Gallaher, A. O'Connor, J. Dettbarn, and L. Gilday, “Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry”, NIST Advanced Technology Program, Information Technology and Electronics Office,Gaithersburg, Maryland 20899, 2004.

[4] M. Fischer and C. Kam, “PM4D Final Report, CIFE Technical Report(TR143)”, CIFE, Stanford University, 2002.

[5] IFC Sample File of STEP Tools Co., http://www.steptools.com/support/-stdev_docs/express/ifc2x3/demos/geometry_out_stp.txt. [last seen: June 2011].

[6] Web Services Architecture, W3C Working Group Note 11 February 2004, http://www.w3.org/TR/ws-arch/. [last seen: Novemeber 2011].

[7] Y. Adachi, "IFC Model Server development Project-official webpage" VTT building and SECOM, Finland, 2002, http://cic.vtt.fi/projects/ifcsvr/index_exc.html. [last seen: October 2011].

[8] BIM Server, http://bimserver.org/.[last seen: October 2011].

[9] Revit plugin to connect to BIM Server, http://bimserver.org/blog/-2009/05/25/revit-plugin-to-connect-to-bimserver/. [last seen: October 2011].

[10] I. Faraj, M. Alshawi, G. Aouad, T. Child, J. Underwood, “An industry foundation classes Web-based collaborative construction computer environment: WISPER”, Automation in Construction, Volume 10, Issue 1, November 2000, pp.79–99.

[11] C. Fu, S. Kaya , M. Kagioglou, G. Aouad, “Development of an IFC-based Lifecycle Costing Prototype Tool: Integrating Lifecycle Costing to nD modelling”, Construction Innovation, Volume 7 Issue 1, pp. 85-98, January 2007.

[12] R. Campbell, “It all ends in Tiers”, SIGC Application Development Advisor, November, 1997.

[13] D. Pountain and J. Montgomery, “Web components”, BYTE, August, 1997, pp.56–68.

[14] Microsoft’s ORM Framework, http://msdn.microsoft.com/en-us/data/ef.aspx. [last seen: December 2011].

[15] IFC engine, http://www.ifcbrowser.com/ifcenginedll.html. [last seen: January 2012].

[16] R. Bourret, C. Bornhvd, and A. Buchmann, “A generic load/extract utility for data transfer between XML documents and relational databases”, WECWIS, 2000, pp.134–143.

[17] M. Mohamed, H. ElYamany, M. Hussein, N. Yhiea, H. Nassar, “An Adaptive Replication Framework for Improving the QoS of Web Services”, in the proceeding of the 2nd International Conference on Cloud Computing and Services Science (CLOSER 2012), Porto, Portugal, 18-21 April 2012.