Integration of Multiple Product Models: IFC Model Servers as a Potential Solution

A. Kiviniemi
VTT Building and Transport, Technical Research Centre of Finland, Espoo, Finland

M. Fischer
CIFE, Civil and Environmental Engineering, Stanford University, Palo Alto, USA

V. Bazjanac
Lawrence Berkeley National Laboratory, University of California, Berkeley, USA

ABSTRACT: The development of the Industry Foundation Classes (IFC) started from the vision of an integrated building product model which would cover all necessary information for buildings' whole lifecycle: requirements management, different design activities and construction and maintenance processes. Although the IFC model specification covers a substantial part of the required information its implementations into practical applications have shown several serious problems. One of the main problems is that the internal structures of the different software products do not support the information needs for the whole process. Thus, the idea of lossless, incremental data flow through the different applications used by the project participants has not come true. It is obvious that file based data exchange is not feasible solution, and some other solution for integrated project data model is necessary for the AEC industry. This paper discusses some viewpoints and potential solutions to the above problem.

1 INTRODUCTION

The development of the Industry Foundation Classes (IFC) started in International Alliance for Interoperability (IAI) from the vision of a shared building product model which would cover all necessary information for buildings' whole lifecycle: requirements management, different design activities and construction and maintenance processes (Figure 1).

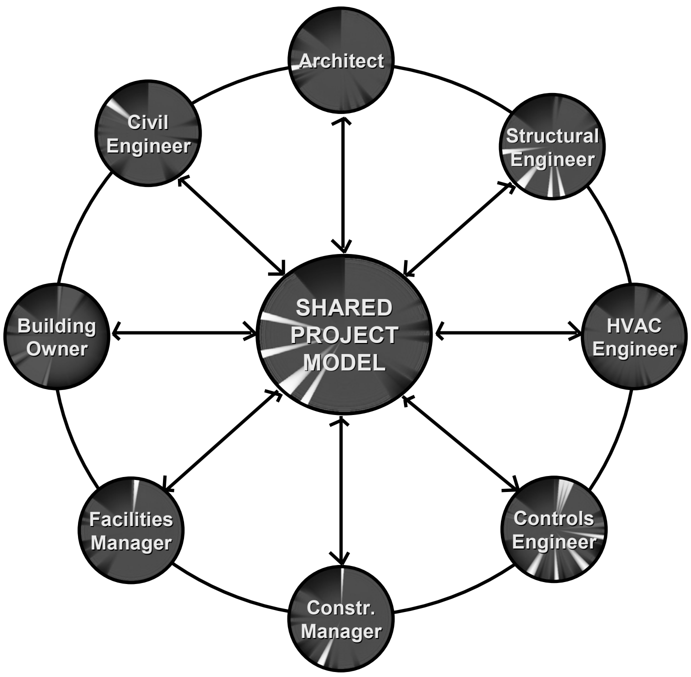

Figure 1: Shared Project Model, IAI 1997

Although the IFC model specification now covers a substantial part of the information required in the design, construction and maintenance processes, its implementations into practical applications have shown several serious problems. One of the main problems is that the internal structure of the different software products does not support the information needs for the whole process. Thus, the idea of lossless, incremental data flow through the different applications, which IAI has presented since 1996, has not yet come true (Figure 2).

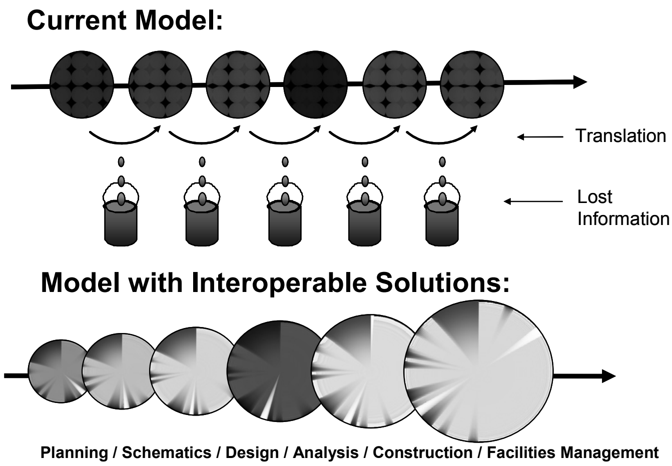

Figure 2: Incremental data flow in the AEC process, IAI 1997

The problems of one building product model and data exchange have been addressed in some earlier research projects. For example, the key problems addressed in the "PM4D Final Report" (Kam &

Fischer 2002) were that 1) the different information content in different software products makes impossible to maintain all the data when transferring a building product model between different software applications, 2) the instantiated models are large, which makes the file exchange of the model time-consuming although usually only a small part of the model has changed and transferring the whole model would not be needed, if partial exchange was available, and 3) versioning and controlling user rights in file exchange are practically impossible.

Also, John Haymaker recognized the need for several models and the linkage between these models in his Ph.D. research (Haymaker et al 2003).

It is obvious that file based data exchange is not feasible solution, and some other solution for integrated project data model is necessary for the AEC industry.

2 INTEGRATED MODEL SPECIFICATION

The complexity of an integrated model specification has lead to criticism against the development of an integrated building product model specification (Behrman, 2002).

Behrman strongly criticizes top-down data exchange standardization efforts, such as IFC. Many of his arguments are valid, such as the difficulty and slow speed of the development and complexity of the implementation of the standard. As Behrman writes, the lack of high-level commitment of a critical mass of key players is a fundamental problem in data standardization efforts in the AEC industry.

However, the bottom-up development - independent minimalist standardization based on each use-case, which Behrman recommends - has not been more successful in the AEC industry or replaced IFC development since the publication of Behrman's report. On the contrary, aecXML, which tried to use the bottom-up approach, has not progressed since 2002, while IAI has published two new versions of the IFC specifications. Both bottom-up examples discussed in Behrman's report - landXML and gbXML - are still the only aecXML schemas and in draft stage almost three years later. Although the development and implementation of the IFC specification has been slow, it has progressed and strengthened its position as a de-facto standard since 2002.

In addition, Behrman's report does not include the latest technologies in IFC implementation: IFC model servers and standardization of their APIs (SABLE 2003). The development of model servers started in 2001 and as of June 2005 at least three products exist (IMSvr 2002, WebSTEP 2002, and EPM 2003). This development would not have been possible without a comprehensive model specification, such as the IFCs. The model servers and their standardized APIs can hide the complexity of the underlying model specification and enable the use of standard protocols in data exchange, such as XML and SOAP in the software implementation, which is one of Behrman's main critiques of the IFC Specifications.

In our opinion, there is no reason why the development of the model specification should not be based on one integrated schema. On the contrary, regardless of the problems caused by the inevitable complexity of such a schema, one integrated model specification helps to build common understanding of the appropriate structure and relations between the objects in a building product model.

The problems are on the level of instantiated models. The data content in any existing software tool used by the AEC industry covers only a small part of the necessary information. Thus, data exchange between different types of applications contains almost always objects which cannot be stored in both applications. This leads to the same data loss scenario which IAI has been trying to fix by the IFC development (Figure 2).

3 SEPARATION OF INSTANTIATED MODELS

These problems can be addressed by dividing the instantiated model of a project, i.e., project's data set, into four separate main models: 1) Requirements Model, 2) Design Model(s), 3) Production Model(s), and 4) Maintenance Model. These linked sub-models form the integrated project information model (Figure 3).

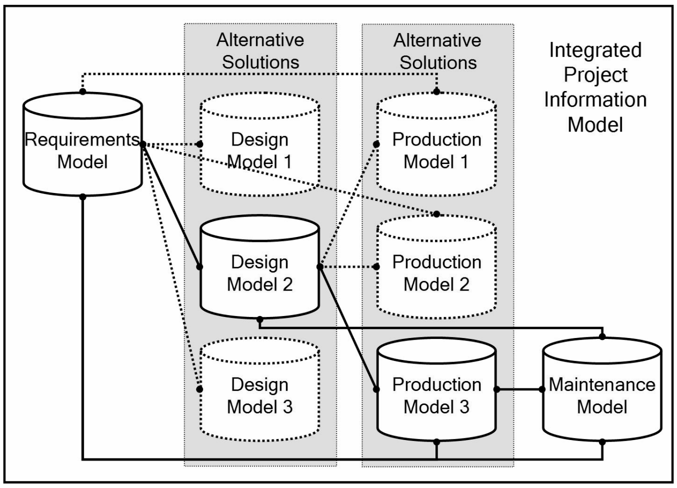

Figure 3: Integrated Project Information Model consisting of the four main models

A crucial issue in the use of separated instantiated model is the ability to link objects in different models to each other. This issue is documented more in detail in the first author's doctoral dissertation (Kiviniemi 2005) and also in a CIB W78 conference paper (Kiviniemi 2005b).

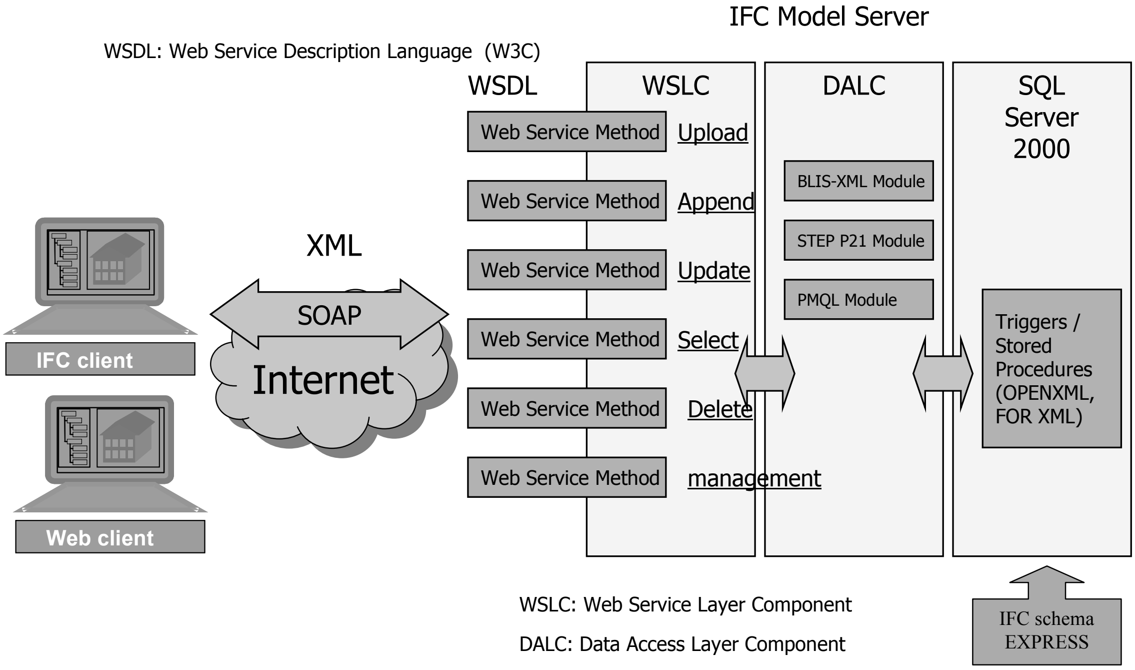

Figure 4: IFC model server architecture © Yoshinobu Adachi 2002

4 MODEL SERVERS

As mentioned earlier, the file based exchange is not a feasible solution for shared building product models, and partial model exchange is also a necessity. In addition, the complexity of the IFC specifications has been one of the bottlenecks for implementation, and easier access to the model data using simple queries would improve the usability of the IFC specifications. To solve these problems, several projects have been developing IFC model servers since 2001: IMSvr (Figure 4), WebSTEP and EPM 2003. All these model servers provide partial model exchange and simple query access to the model using standard technologies such as XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), and STEP (STandard for the Exchange of Product model data) (Adachi, 2002). As mentioned in Section 2, at least three IFC compliant model servers exist as of June 2005.

However, from the implementation viewpoint, the different application interfaces to different model servers are a problem, because they either limit the use to one model server or require implementation of several application interfaces for each domain (Figure 5)

5 STANDARDIZED MODEL SERVER API

A standardized application interface for each domain can solve the problems mentioned above. The SABLE project is developing such interfaces based on SOAP (SABLE 2003, Figure 5). Each domain-specific API handles the information exchange needed by the client applications for each domain, which logically corresponds with the BLIS views (BLIS 2001). The SABLE project is ending in summer 2005 after which its results can be taken into wider use.

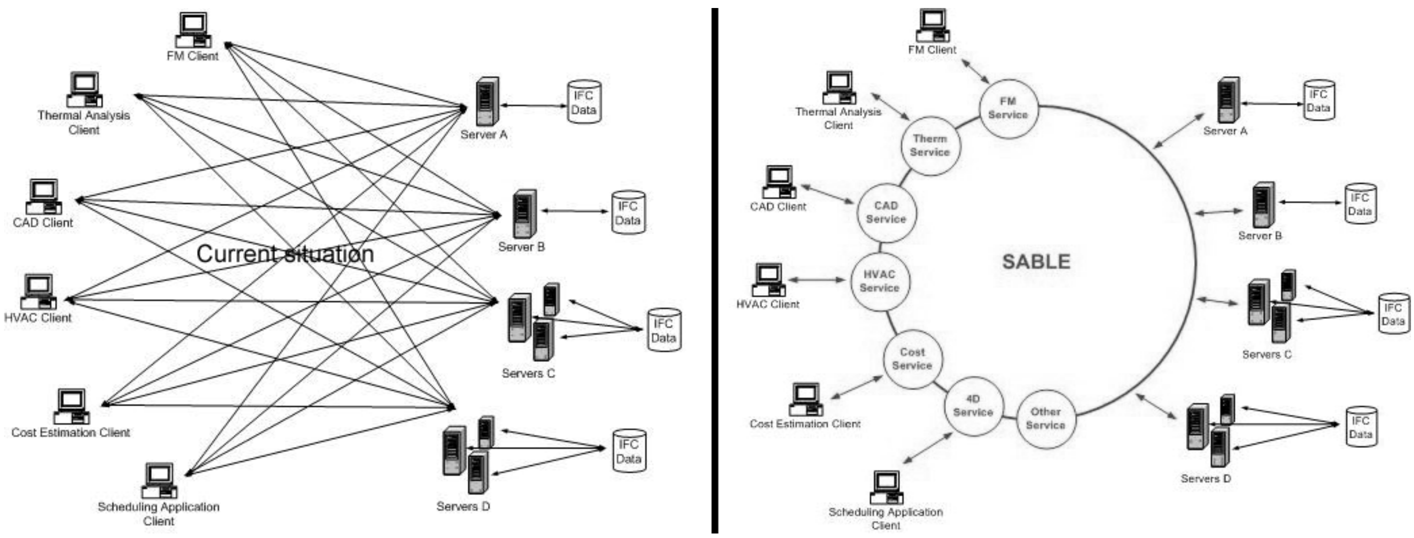

Figure 5: Advantage of the standardized interface approach © BLIS & SABLE 2003

6 FURTHER DEVELOPMENT: A PROOF-OF-CONCEPT

Although some IFC compliant model servers exist and SABLE API is at the finalizing stage, the interfaces for end-user applications are still missing. Thus, the concept of a multi-model environment has not been tested in real projects.

VTT, CIFE and LBNL formed the Virtual Building Environment (VBE) consortium in 2002. The aim of the VBE consortium is to develop and facilitate the use of building product model technologies in the AEC industry.

One of the efforts starting in the VBE II project in summer 2005 will be the development of some interfaces for end-user applications and testing them in a multi-model environment using real project data, SABLE API and existing model servers (Figure 6). One potential partner in this development will be TNO where Peter Bonsma has developed a 3D component as a part of his IFC Engine Series (Bonsma 2005).

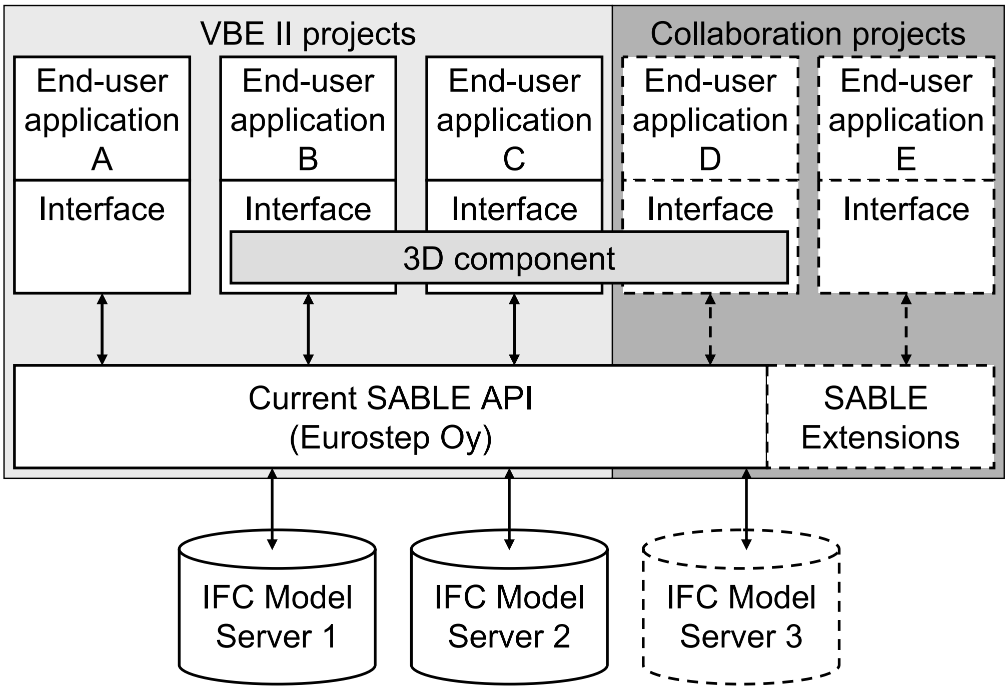

Figure 6: Planned interface development within the VBE II project

An additional potential in the concept presented in Figure 6 would be the possibility to "standardize" an international platform for multi-model environment research. This platform would enable efficient collaboration between different universities and research institutes and efficient use of resources, because a standardized API and underlying model server would simplify the implementation efforts, and the use of software modules developed in other projects would enable simultaneous testing of several features of a shared model without the need to program all necessary functionalities in each project.

REFERENCES

Adachi, Yoshinobu: Introduction of IFC Model Server, 2005. URL: http://cic.vtt.fi/vera/Seminaarit/2002.04.24_IAI_Summit/Adachi.pdf

Behrman, William: Best Practices for the Development and Use of XML Data Interchange Standards. CIFE Technical report TR131, Stanford University 2002, available also at http://cife.stanford.edu/online.publications/TR131.pdf

BLIS: View definitions: http://www.blis-project.org/views/

Bonsma, Peter: IFC Engine Series, 2005. URL: http://www.ifcbrowser.com/

EPM: Express Data Manager by EPM Technology, 2003, URL: http://www.epmtech.jotne.com/products/

Haymaker, John; Suter, Ben; Kunz, John; & Fischer, Martin: PERSPECTORS: Automating the Construction and Coordination of Multidisciplinary 3D Design Representations. CIFE Technical Report TR145, Stanford University 2003: http://cife.stanford.edu/online.publications/TR145.pdf

IMSvr: IFC Model Server project by Secom (Japan) and VTT (Finland). Adachi, Yoshinobu 2001-2002, URL: http://cic.vtt.fi/projects/ifcsvr/

Kam, Calvin & Fischer, Martin: PM4D Final Report. CIFE Technical Report TR143, Stanford University 2002: http://www.stanford.edu/group/4D/download/c1.html

Kiviniemi, Arto: Requirements Management Interface to Building Product Models. Ph.D. Dissertation and CIFE Technical Report TR161, Stanford University 2005: http://cife.stanford.edu/online.publications/TR161.pdf

Kiviniemi 2005b: Kiviniemi, Arto, Fischer, Martin & Bazjanac, Vladimir: Multi-model Environment: Links between Objects in Different Building Models, CIB W78 Conference proceedings 2005

SABLE: Simple Access to Building Lifecycle Exchange project by Eurostep. Houbaux, Patrick, 2003. URL: http://www.blis-project.org/~sable/

WebSTEP: WebSTEP IFC Model Server project by Eurostep. Karstila, Kari and Hemiö, Tero, 2002. URL: http://www.eurostep.com/prodserv/ems/ems.html