

Lawrence Berkeley National Laboratory

Lawrence Berkeley National Laboratory

Title

The Implementation of Industry Foundation Classes in Simulation Tools for the Building Industry

Permalink

<https://escholarship.org/uc/item/76c6z6g4>

Authors

Bazjanac, Vladimir
Crawley, Drury B.

Publication Date

1997-06-01

THE IMPLEMENTATION OF INDUSTRY FOUNDATION CLASSES IN SIMULATION TOOLS FOR THE BUILDING INDUSTRY

Vladimir Bazjanac
Lawrence Berkeley National Laboratory
Berkeley, CA 94720

Drury B. Crawley
U.S. Department of Energy
Washington, DC 20585

ABSTRACT

Industry Foundation Classes (IFC) provide an environment of interoperability among IFC-compliant software applications in the architecture, engineering, construction, and facilities management (AEC/FM) industry. They allow building simulation software to automatically acquire building geometry and other building data from project models created with IFC-compliant CAD software. They also facilitate direct exchange of input and output data with other simulation software.

This paper discusses how simulation software can be made compliant with version 1.5 of the IFC. It also describes the immediate plans for expansion of IFC and the process of definition and addition of new classes to the model.

INTRODUCTION

In a fundamental sense, the building industry still operates the same way it has for many decades. It utilizes contemporary computer and information technologies - the backbone of many other industries today - only in the most rudimentary way. Though enormous amounts of information are generated for each project, exchanging that information among participants is inconsistent, usually reduced to a very small subset and only a few of the many participants in the project at a time. Most information is eventually lost. Some is generated in contradiction to other or is unnecessarily duplicated. Computer-based buildings tools are used in a stand-alone manner and cannot exchange data directly, even when they are used by the same party. This often results in omission, repetition, confusion, misunderstanding, error, delay, and, eventually, in litigation. Because of that, buildings take longer to design and build and cost more to construct and operate than necessary.

The potential for the use of contemporary information technology in the industry is enormous, and so are the potential cost savings. In his July 1994 report on the UK construction industry, Sir Michael Latham challenged the industry to use contemporary technology and save up to 30% of the cost of building projects by the year 2000. (Latham 1994) The use of *information* technology is clearly implied in the report.

THE INTERNATIONAL ALLIANCE FOR INTEROPERABILITY

In the spring of 1993, some of the major companies in the building industry of the United States started discussing ways of bringing modern information technology to the industry. This group formed the Industry Alliance for Interoperability in the early summer of 1994 and demonstrated interoperability among a group of CAD and simulation tools at the AEC Systems Show in Atlanta, Georgia in June 1995. The Alliance became a public organization, open to any member of the industry, in September 1995 and formally became a global organization in May 1996. At that point the name was changed to the International Alliance for Interoperability (IAI).

The IAI is an action oriented, not-for-profit organization. Its mission is to define, publish and promote specifications for Industry Foundation Classes (IFC) as a basis for world-wide AEC/FM project information sharing throughout the project life cycle, and across all disciplines and technical applications. IFC define a single, object oriented data model of buildings shared by all IFC-compliant applications. IFC project models define individual buildings for which compliant applications can exchange information accurately and error-free.

IFC are public and "open" for implementation and use by any member, are defined by the industry, are extensible and will evolve over time. Software implementation of IFC is proprietary to protect the data and technologies of member companies that compete in the market. IAI member companies hope that IFC may eventually become a de facto industry standard.

By June 1997, the IAI had seven chapters in North America, Europe and Asia (with three more organizing in Australasia and Europe) and a total of almost 500 member organizations and companies. The organization is governed by the IAI International Council. Each chapter has its own Board of Directors, Coordination Committee and various "domain" committees. Two technical committees - Research/Advisory, and Software Implementation - are international and report to the International Technical Management Committee. Most committee interaction takes place through teleconferences and over the Internet. Individual chapters hold joint face-to-face meetings as often as once a month. International technical meetings take place quarterly.

IFC 1.0

The IAI announced the release of Version 1.0 of the IFC (IFC 1.0) in June 1996 and published the *End User Guide* (IAI 1996a) and a "pre-release" *IFC 1.0 Specifications*. (IAI 1996b) The complete documentation, available to IAI members through their respective chapters or the Internet, contains four additional volumes which describe:

- Domain processes enabled with the model
- The complete IFC 1.0 model specification
- Static file exchange
- Runtime interfaces

IFC 1.0 consist of an object oriented core model, four independent resources models, and four initial domain extensions (architecture, building services, construction management, and facilities management). The core model defines objects, attributes and relationships common to domain extensions. Resource models document the definition of geometry, units and common utilities. Extension models define objects, attributes and relationships specific to domains.

Volume 1, *AEC/FM Processes Supported by IFC*, documents AEC/FM domain processes supported by Release 1.0. It effectively defines the scope of this release from the users' point of view.

Volume 2, *IFC Project Model Specifications*, defines the IFC project model - all information required by AEC/FM processes described in Volume 1, structured as object classes, data types and standard interfaces. This volume also discusses several key IFC design concepts, such as design intent, sharing of semantic relationships, model extension by application developers, static file exchange and runtime interfaces.

Volume 3, *IFC Model Exchange Specifications*, documents the data model view of the IFC project model. It contains the complete EXPRESS (ISO 1994a) and EXPRESS-G definitions of the core model, the four independent resources models (attribute-driven and explicit geometry, measures and utilities), the four domain models, and the description of file-based exchange (early vs. late bound toolbox implementation). Exchanged information can contain the entire project model or only a part of it. This volume also includes sections devoted to conformance testing. It provides sufficient information for application developers to use existing CASE tools (that automate the process of software implementation) directly with the EXPRESS definition.

Volume 4, *IFC Model Software Interfaces* contains a discussion and reviews of object models and languages and the documentation of Microsoft Interface Definition Language (MIDL) files for the core, resources and domain models in IFC. (Microsoft 1996)

The geometry resource models allow multiple representations of objects:

- Reference geometry
- Bounding box
- Attribute-driven geometry representation
- Explicit geometry representation

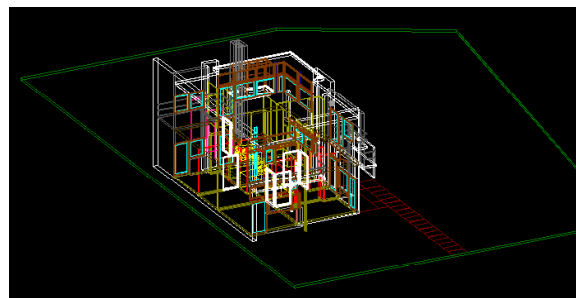
Reference geometry (oriented vertex) defines the object's origin point and orientation in three-dimensional space. The bounding box defines the rectangular envelope in which the physical object fits completely. (The shape representation of all HVAC equipment is limited to bounding box in IFC 1.0.) Attribute-driven representations define location, orientation and dimensions of building elements that have shape (such as walls, windows, etc.). Explicit geometry representations define building elements that have shape as solids; they are based on a subset of STEP Application Protocol 225, known as AP225. (Haas 1996)

PILOT IMPLEMENTATION

With the publication of IFC 1.0, 26 companies in the U.S., Canada and Europe announced their intent to make their software IFC-compliant. These companies include the major international CAD vendors Autodesk, Bentley, Nemetschek, and IEZ.

A smaller group, the IAI Pilot Implementers, showed proof of concept for IFC 1.0 at the ACS Show in Frankfurt in November 1996. Four CAD companies (Autodesk and Bentley from the U.S., and Nemetschek and Softech, German subsidiary of Softdesk, from Europe) exchanged files that contained geometry data. The exchange took place among special IFC-compliant versions of their commercial CAD software. These were "live" exchanges of fairly complex building representations, displayed as two- or three-dimensional drawings.

Figure 1. Three-dimensional model of Rietveld's Schroder House© (courtesy of the Foundation Gerrit Th. Rietveld c/o Beeldrecht, the Netherlands)

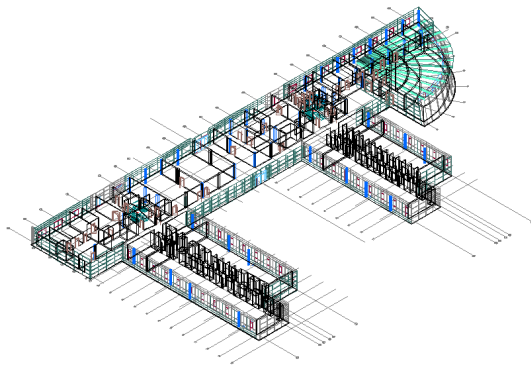


The three-dimensional model of an existing historical building that is a celebrated example of De Stijl architecture (Figure 1) was displayed by one CAD program and slightly modified. The modified, IFC-compliant CAD file was then sent to another CAD program that subsequently redisplayed the building showing the modification exactly as it was displayed by the previous CAD program. Separately, building elements (such as walls) were drawn "from scratch" and "passed" to the next CAD program, modified (e.g., by the addition of a window), "passed on,"

modified again and redisplayed with no loss of accuracy.

In addition, Autodesk showed "live" file exchange among four applications by their third-party developers (architecture, structural, HVAC and FM), running on top of an IFC-compliant version of AutoCAD 13 (Release 13c4a with special ARX extensions). The Autodesk demonstration followed a script in which a portion of a fairly large office building situated in Innsbruck (Figure 2) is remodeled. It showed how designers, engineers and facilities managers can work in an interoperable environment and effectively exchange information on the ensuing problems and solutions. Once again, the building and information exchange were non-trivial.

Figure 2. Three-dimensional model of the office building situated in Innsbruck (courtesy of AcadGraph)



The November 1996 demonstration of interoperability in Frankfurt showed without any doubt that IFC project model data can be exchanged effectively and without loss of information. It showed that building geometry data can be automatically exchanged among applications and platforms *now* without any loss of accuracy.

The demonstration was repeated at the AEC Systems Show in Philadelphia, Pennsylvania in June 1997. The same implementers that participated in the demonstrations by the IAI and Autodesk in Frankfurt in November 1996 joined forces and demonstrated "live" exchange among a larger group of applications. The exchange followed a revised and more complex script of interoperability.

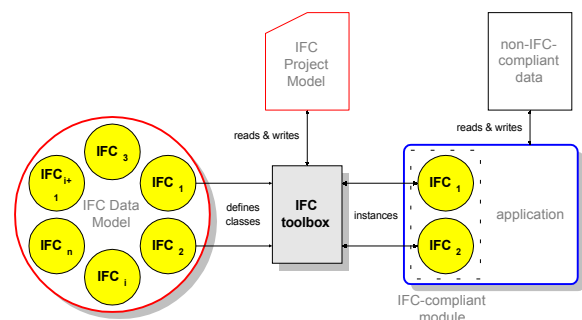
IFC TOOLBOX

To aid implementation, the IAI sponsored the development of a toolbox that facilitates writing IFC objects/ attributes (defined in EXPRESS in the IFC model) in C++ applications. The toolbox function and properties are perhaps best understood if one thinks of the toolbox as a layer above the file format that makes handling IFC-based data and programming IFC-compliant applications much easier.

Figure 3 shows a diagram of the IFC toolbox environment. The IFC Data Model is defined in EXPRESS; the IFC Project Model (which represents

a specific building) is an exchangeable ASCII file. The toolbox contains all classes and methods included in the IFC Project Model. The application programmer deals only with classes in the toolbox and can ignore the details of the (original) complex representation of the building and other data - the programmer creates a module (within the application) that only translates toolbox objects to application objects. Data specific to the application or not contained in the IFC Project Model (non-IFC-compliant data) remain unaffected and are ignored by the programmer.

Figure 3. Toolbox environment (for early binding)

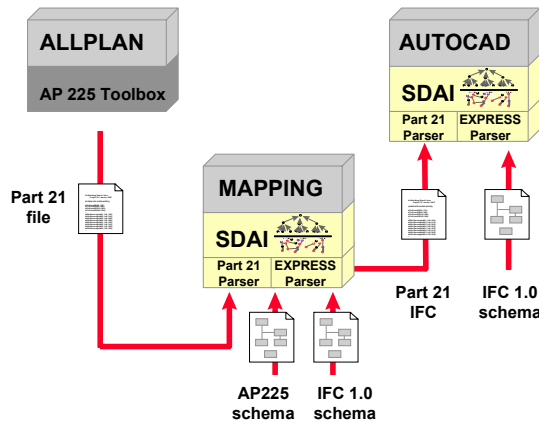


The application's IFC-compliant module receives existing object instances from and sends new instances to the toolbox. The toolbox itself is a library of functions that are present at runtime - the toolbox is integrated into the application. When the information from the Project Model (IFC-compliant data) is present when an application is compiled, the binding is called "early." When it is present at runtime, the binding is called "late."

The IAI Pilot Implementers (with the exception of Bentley, which used STEP tools) used an early-binding toolbox in the development of IFC-compliant software for the demonstration at the 1996 AEC Show in Frankfurt. They all reported that the toolbox was invaluable - it saved substantial time and programming resources. This toolbox was revised to include the stabilized core model and other revisions in Release 1.5 (see below), and became available in July 1997. *concad* (the German company that developed the toolbox) will convert the toolbox to work with FORTRAN, C or other application programming languages for a fee. A late-binding toolbox (from another vendor) will be available before the specifications for IFC 2.0 are released.

CSTB of France demonstrated an example of a late-binding toolbox at STEP meetings in San Diego, California in June 1997. The tool is an SDAI C++ late-binding platform with persistent storage and transactional services that allows the exchange of data via STEP physical (Part 21) files. (ISO 1994b) It is a building model server for applications (such as mapping) that contains a module which maps from one EXPRESS schema to another. The mapping in the demonstration was from AP225 to IFC 1.0 schema (Figure 4). Software modules that map between other schemata can be added to the tool.

Figure 4. Late binding: exchange of geometry between AP225- and IFC-compliant applications (adapted, courtesy of CSTB)



In the CSTB demonstration in San Diego AP225-compliant building geometry was generated using Nemetschek's Allplan FT. The tool was then used to map from AP225 to IFC 1.0 and import that geometry into AutoCAD. The building displayed by AutoCAD was identical to that displayed by Allplan FT. A group of French software developers plans to demonstrate the implementation of this tool in Clermont-ferrand in September 1997.

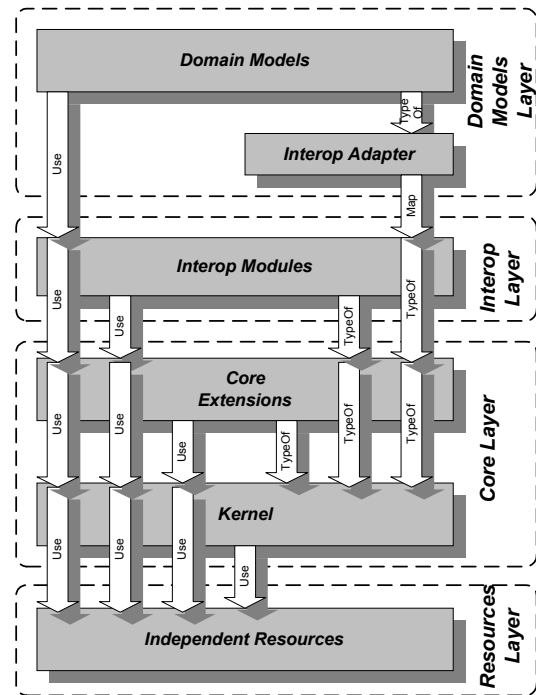
IFC DATA MODEL ARCHITECTURE

Work on IFC 1.5 is being completed as of this writing. This is an interim release for developers of commercial IFC 1.0-compliant software applications and the development of IFC 2.0. It allows developers to start implementing IFC in their software with greater ease and provide them with a stable environment for the future. The implementation will be based on the 1.5 version of the core and the 1.0 version of extension models. IFC 1.5 include:

- Stabilized core model
- Refined independent resources (revised geometry)
- Revised IFC model architecture

The stabilized core model is based on extensive review and on experience with pilot implementation of IFC 1.0. It provides a stable environment for development and implementation. The revised model architecture is based on decomposition into four conceptual "layers" (Figure 5). This allows for a modular structure, provides a framework for sharing of information among different domains, facilitates reuse of model components (as well as of software components), makes development and maintenance of the model easier, and permits upward compatibility between model versions. (IAI 1997a)

Figure 5. IFC 1.5 architecture



The basic layer (resources layer) contains independent resources that are grouped into "families:" general resource (classes identification, measure, time, and actor), geometry (shape representation, explicit and attribute-driven geometry) and business concepts family (classes property, classification, cost, and material). The geometry family will include a geometry library in IFC 2.0, and the business family will be expanded with history, state, version, status and approval resources.

The core layer contains the kernel and core extensions. The kernel provides all basic (non-AEC/FM specific) concepts required by the current IFC (classes object, relationship, attribution, and type definition), while core extensions provide AEC/FM specific extensions to concepts rooted in the kernel (classes product, process and modeling aids). Core extensions include classes space, element, site, building and story, as well as grid (a modeling aid).

The interoperability layer contains modules that facilitate interfaces with domain models: building elements and building service elements. The former includes classes wall, roof slab, floor, beam, column, built-in, door, window, and covering, ceiling. The latter includes equipment, fixture, and electrical appliance.

The domain models layer includes three domain models (architecture, building services and facilities management) and application models. These will be extended with limited versions of construction, structural, codes and standards, and cost estimating domain models in IFC 2.0. This layer also includes "interoperability adapters" that facilitate exchange with application models that are topologically different from IFC (i.e., that have a software architecture different from IFC).

A class may use or reference another class only within the same layer or a lower layer. Same layer references are limited to independent resources and core layers. Inter-domain references (within the domain models layer) are resolved through interoperability adapters and core extensions.

IAI ROAD MAP

The plan for development of IFC is defined in the IAI Road Map. It provides the schedule of future releases, defines the new processes to be enabled with each new release, and identifies technologies needed to enable the newly defined processes. It also identifies new opportunities to market specific new functionality. The content of the Road Map is constantly evolving.

Future releases of IFC will include new domain extension models and expansion of the existing models. Release 2.0 will include three different types of additions:

- New object/attribute/relationship sets
- New IFC technologies
- Subsets of existing, non-IFC based domain models

New object/attribute/relationship sets will be added to the existing domain extensions as well as the new domain extension models. The new technologies (new to IFC and not yet widely used in the AEC/FM industry models) enabled in the next release will include general networks, access to external libraries and data bases, and semantic associations (a general purpose element aggregator). A subset of an external structural steel model (CIMsteel) is also slated for inclusion, if a collaborative agreement can be reached with its authors.

New additions to IFC are planned and executed as "IAI projects." A project defines the general scenario in which the task that requires the specific addition is placed relative to AEC/FM practice, and defines what exactly needs to be added to IFC and the resources that are required and available to perform the work. (Rules of project formulation are defined in Volume 2 of IFC 1.0 documentation.)

Several projects planned for Release 2.0 and beyond are of particular interest to building simulation:

- Near completion of the architecture extension model
- General networks
- Further development of the building services extension model: HVAC air-side and water-side delivery systems, pathway design, and power and lighting systems
- Access to external libraries and data bases

Release 2.0 will include one project from the simulation domain, which will enable high-resolution visualization. The project will result in the addition of two new object/attribute sets to the IFC: "light

source" (with attributes spectral power distribution, luminaire geometry and photometric output distribution) and "surface" (with explicit shape representation, dimensions, material and parameterization). These additions will make it possible for ray-tracing models to become IFC-compliant and benefit from automatic acquisition of geometry and pertinent building data. This, in turn, will reduce the time and cost of input preparation for such models, and make their use in daily practice more likely.

The current schedule for IFC events is:

- Release 1.5 June 1997
- Revised *concad* toolbox (based on IFC 1.5) July 1997
- Prototype commercial implementations based on Release 1.5 model November 1997 (at the ACS Show in Frankfurt)
- Final Release 2.0 end of 1997
- Commercial implementations of Release 1.5 on the market early 1998
- Prototype commercial implementations based on Release 2.0 model June 1998 (at the AEC Systems Show)

At present, all IFC data exchange is limited to the exchange of physical files as defined in STEP Part 21. The IAI plans to move to server-based (client-server) exchange, mostly as defined in STEP Part 22 by 1999. Direct object-to-object exchange is expected by year 2000.

IMPACT OF IFC ON BUILDING SIMULATION

Building simulation tools are currently used only occasionally in projects. High cost of entering data and inability to reuse information contained in the tool are some of the reasons. For example, it can take two or more weeks to enter building geometry for a fairly complex building into DOE-2, a sophisticated building energy performance simulation model. (Winkelmann et al. 1993) Most projects simply cannot afford the cost. For another tool to use the information generated by DOE-2, that information has to be interpreted and then transferred, which can take as much time as preparing the DOE-2 input. This information is lost to tools not engaged in the interpretation and transfer.

Without a common data model, software applications in the AEC/FM industry can directly exchange information only through interfaces that "translate" the data from one format into another. A unique (and costly) interface is required for each pair of applications in the exchange (Figure 6). IFC offer an environment for true interoperability in which multiple applications can exchange information directly, and in which no multiple, unique interfaces are needed.

Figure 6. Software interoperability: IFC-compliant applications can exchange information directly, while non-compliant applications must have separate interfaces

Compliance with IFC will allow building simulation software, for the first time, to:

- "Talk" to each other directly and instantaneously
- Share and exchange information of common interest and/or reference

This can result in substantial and quite tangible benefits for everyone involved:

- Virtually cost-free access to building geometry and other related data
- Much shorter simulation cycle time
- Better participation of other relevant disciplines in simulation and analysis
- Better use of results of simulation

Automated, error-free acquisition of building and component geometry originally defined with IFC-compliant CAD software, coupled with automated access to IFC-compliant external libraries and data bases, can reduce input preparation effort to a fraction of what it is now. Manual quantity take-off and transfer of information from textual sources can be eliminated. No information will be lost. This can reduce simulation cost and turn-around time by orders of magnitude and make the use of simulation in daily practice a feasible reality. This makes the U.S. Department of Energy's goal of using energy performance simulation on every building attainable.

In addition, *all* parties involved in the design, construction and/or building operation can have direct and timely access to project information, including the results of simulation. This can result in higher quality of both the simulation and the whole building.

IMPLEMENTATION OF IFC IN BUILDING SIMULATION SOFTWARE

All simulation tools need data to perform their task. The proportion of data available from IFC Project Models will determine the extent to which a simulation application will be IFC-compliant: fully, partially or not directly compliant at all.

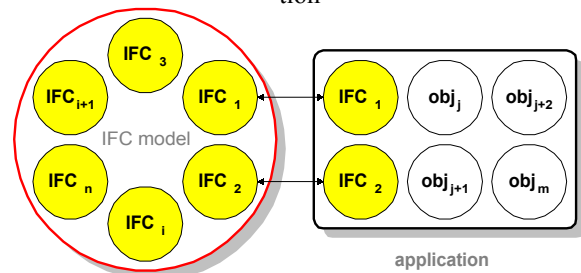
New simulation tools should be conceived from the beginning as object oriented and fully IFC-compliant, to reap all the benefits of potential interoperability. In the case of a fully IFC-compliant simulation tool (Figure 7) there must be a 1:1 relationship between the objects defined in the IFC Project Model and those defined in the tool. With the toolbox and IFC documentation the mapping of IFC onto a new, object oriented software structure should be relatively "pain-free." All data necessary for simulation can be obtained from IFC-compliant sources.

Figure 7. Fully IFC-compliant software application

Often, a simulation tool uses only a limited set of the data available in the IFC Project Model. In such

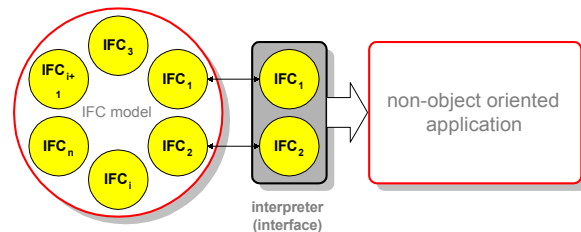
cases it makes more sense to make the tool only partially compliant (Figure 8), even if the tool's structure is object oriented. Only those classes in the IFC Project Model that contain data pertinent to the simulation are mapped to the simulation tool. The remaining data needed for the simulation are obtained from other (either internal or external) non-compliant sources. If such a simulation model needs to exchange these "non-compliant" classes and/or data with another partially compliant model, that exchange is facilitated by IFC interoperability adapters and modules (see IFC Data Model Architecture above). Most existing simulation tools, object oriented or not, that become IFC-compliant fall into this group.

Figure 8. Partially IFC-compliant software application



Few existing simulation tools are object-oriented or amenable to incorporation of internal, IFC-compliant interface modules. To make them object oriented or add new internal interface modules would most likely involve a *complete* re-write, which is often not plausible. For such tools there is but one option: make them indirectly and only partially compliant (Figure 9).

Figure 9. Non-compliant software application



This requires writing an *external* object oriented interpreter that contains the same objects as those pertinent to the simulation in the IFC Project Model, mapped 1:1. The essential function of the interpreter is to "translate" data from the IFC Project Model into the simulation tool data format, and vice versa. The simulation still runs as before and continues to acquire data not contained in IFC from the same sources as before, but can now *also* obtain data from IFC Project Models. Through the interface (which is a separate executable), the simulation tool can now exchange IFC-compliant data with other IFC-compliant applications or interfaces. In that fashion the interpreter converts the non-compliant into a partially IFC-compliant simulation tool.

This approach has been used at LBNL to make several simulation tools for different aspects of building performance (such as energy performance, daylighting, lighting and air-flow) IFC-compliant. Most of these tools are large, quite old and not object oriented. A new tool, the Building Design Advisor (BDA), will map IFC that contain building geometry and other information pertinent to LBNL simulation tools to its own object oriented model. BDA will also contain the mechanisms to convert data imported from IFC Project Models to the formats of the tools that use the particular data, and vice versa. (Papa-michael et al. 1996)

The approach will allow users of tools linked to BDA to automatically acquire building descriptions from project models generated by IFC-compliant CAD software, as well as limited non-geometrical information already contained in IFC. As the IFC Data Model becomes more complete, new classes containing non-geometrical information required by other tools can be added to BDA.

HOW TO ADD NEW OBJECT/ ATTRIBUTE/RELATIONSHIP SETS

Most building simulation tools use substantial volumes of data in addition to building geometry, data that cannot yet be contained in the IFC Data Model. New object/attribute/relationship sets have to be added to the IFC Data Model to allow the inclusion of such data. (IAI, 1997b) The responsibility to define what needs to be added to IFC falls on simulation users and developers. (Bazjanac and Selkowitz 1997)

If not already a member of the IAI, one should join the appropriate chapter. Information on joining is found on the web at <http://www.interoperability.com>. Once a member, one can propose new IAI projects through the domain committees that will:

- Identify object/attribute/relationship sets needed for simulation that have not yet been defined for inclusion in IFC
- Define an explicit IAI domain project that includes those object/attribute/relationship sets

After that, one should work within one's domain committee to get the proposed project on the IAI Road Map.

RELATIONSHIP OF IFC TO STEP AND COMBINE

The IAI and the International Standards Organization's (ISO) STandard Exchange of Product model data (STEP) effort complement each other. The IAI is actively using some of the technology developed by STEP and is providing STEP with testing ground in the AEC/FM market place. Most IAI technical experts are also members of STEP. STEP granted IAI official liaison status in June 1997.

It is important, though, to recognize the difference in mission between STEP and the IAI. While STEP is

setting standards, the IAI is responding to immediate needs of the AEC/FM industry. By definition, results of STEP work must be proven and robust. IFC are continuously evolving, and are occasionally neither proven nor robust. STEP *must* take as much time as necessary; the IAI *must* act quickly. That is why the two organizations complement each other so well, even if they do not always use common architecture or methodology. (Bazjanac and Selkowitz 1997)

When appropriate STEP technology is available, the IAI uses it. When it is not, the IAI develops its own solutions. For example, the IAI is using STEP Part 21, and has borrowed from STEP General Resources (Parts 41, 42 and 43) and AP225 (explicit shape representation). It will probably also borrow from AP230 (structural steelwork) when it is adopted. STEP Part 106 (the Building Construction Core Model) and the IFC core model are being developed in parallel. The goal is for both organizations to use identical models. The development of STEP AP228 (HVAC) involves several members of IAI Building Services domain committees.

The JOULE project Computers Models for the Building Industry in Europe (COMBINE) is perhaps the best known previous attempt at achieving interoperability in the AEC/FM industry. (Augenbroe 1994) Two important factors separate COMBINE from the IAI:

- COMBINE's interoperable environment is, by design, limited to a selected group of software applications
- COMBINE preceded the IAI by several years

COMBINE and IFC differ conceptually and methodologically. The COMBINE team limited the scope of its work to eight software applications, and the exchange was designed for data pertinent specifically to those eight applications. In contrast, the scope of the IAI explicitly includes the dealing with *any* software application. This necessitated a different approach and method. (Bazjanac and Selkowitz 1997) Because COMBINE preceded the IAI by several years, STEP technology available at the time did not include all the elements relevant to the AEC/FM industry it includes today, such as AP225 or the currently developing Building Construction Core Model (BCCM). (Wix and Liebich 1997)

The IAI benefited greatly from the experience of those who preceded it. The IAI Research/Advisory Committee studied the available documentation on COMBINE and learned from that - some of it was later reflected in the development of the IFC.

CONCLUSIONS

Building simulation software can reap major benefits from IFC *now*. Automatic, error-free acquisition of building geometry and other data available from project models developed with IFC-compliant CAD software can dramatically reduce the simulation input time and cost, and can shorten the simulation cycle. It can facilitate the direct exchange of data among IFC-

compliant applications, and can make it possible to use building simulation in daily practice in the AEC/FM industry.

To benefit from IFC, simulation software developers need to make their software IFC-compliant, directly or indirectly. With complete documentation of IFC 1.0, a refined and stabilized model architecture, the associated toolbox and experience from pilot implementation, this is possible *now*.

With major AEC/FM software developers and industry forces behind it, the IAI will continue the development of IFC. Future releases of IFC will include additional domain extension models. Existing models will be completed. This will eventually provide an environment of true interoperability for building simulation tools.

ACKNOWLEDGEMENTS

The authors wish to thank Jim Forester, Wolfgang Haas, Thomas Liebich, Patrice Poyet, and Jeff Wix of the IAI Research/Advisory Committee, as well as Stephen Selkowitz of the Lawrence Berkeley National Laboratory for their contributions to this paper.

REFERENCES

Augenbroe, G. 1994. "The COMBINE Project, An Overview", Proceedings of *the First ECPPM Conference, Dresden*.

Bazjanac, V., and S. Selkowitz. 1997. "The International Alliance for Interoperability: Its Mission and Its Method", LBNL, August 1997.

Haas, W. 1996. "Product Data Representation and Exchange - Part 225 Application Protocol: Building Elements Using Explicit Shape Representation", ISO/DIS 10303-225.

ISO. 1994a. *ISO-10303-Part 11*, ISO-10303-1:1994

ISO. 1994b. *ISO-10303-Part 21*, ISO-10303-1:1994

International Alliance for Interoperability. 1996a. "End User Guide to Industry Foundation Classes", IAI, August 1996.

Industry Alliance for Interoperability. 1996b. "IFC Project Model Specifications, Version 0.94", IAI, May 1996.

International Alliance for Interoperability. 1997a. "IFC Model Architecture", IAI, March 1997.

International Alliance for Interoperability. 1997b. "A Guide to the Development of Industry Foundation Classes", IAI, May 1997.

Latham, Sir M. 1994. "Constructing the Team", H.M.S.O. c1994, 011752994x, 692.068 LAT, July

1994.

Microsoft. 1996. *Visual C++ 4.0*, Microsoft Press.

Papamichael, K., J. LaPorta, H. Chauvet, D. Collins, T. Trzcinski, J. Thorpe, and S. Selkowitz. 1996. "The Building Design Advisor", LBL-38584, March 1996.

Winkelmann, F. C., B. E. Birdsall, W. F. Buhl, K. L. Ellington, A. E. Erdem, J. J. Hirsch, and S. Gates. 1993. *DOE-2 Supplement, Version 2.1E*, LBL-34947, November 1993, Lawrence Berkeley Laboratory. Springfield, Virginia: National Technical Information Service.

Wix, J. and T. Liebich. 1997. "ISO 10303 - 106 WD: Building Construction Core Model", NIST, Doc. Ref. N599, June 1997.