Lawrence Berkeley National Laboratory

Software interoperability for energy simulation

Author
Hitchcock, Robert J.

Publication Date
2002-07-31

California Energy Commission
Public Interest Energy Research Program

LBNL No. 51244
HPCBS # E5P2.3T1b

HPCBS
High Performance Commercial Building Systems

Software Interoperability for Energy Simulation

Element 2 - Life-Cycle Tools
Project 2.3 - Interoperability

Robert J. Hitchcock, Ph.D.
Ernest Orlando Lawrence Berkeley National Laboratory
July 31, 2002

LBNL-51244
IO-486

ASHRAE Transactions, Vol. 109(1), January 2003.

Software Interoperability for Energy Simulation

Robert J. Hitchcock, Ph.D.
Commercial Building Systems Group
Building Technologies Department
Environmental Energy Technologies Division
Ernest Orlando Lawrence Berkeley National Laboratory
University of California
1 Cyclotron Road
Berkeley, California 94720-8134

July 31, 2002

This work was completed under contract to Lawrence Berkeley National Laboratory as part of the High Performance Commercial Building Systems program. This program is supported by the California Energy Commission's Public Interest Energy Research (PIER) Buildings Program and the Assistant Secretary for Energy Efficiency and Renewable Energy, Building Technologies Program, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

Software Interoperability for Energy Simulation

Robert J. Hitchcock, Ph.D.

Abstract

This paper provides an overview of software interoperability as it relates to the energy simulation of buildings. The paper begins with a discussion of the difficulties in using sophisticated analysis tools like energy simulation at various stages in the building life cycle, and the potential for interoperability to help overcome these difficulties. An overview of the Industry Foundation Classes (IFC), a common data model for supporting interoperability under continuing development by the International Alliance for Interoperability (IAI) is then given. The process of creating interoperable software is described next, followed by specific details for energy simulation tools. The paper closes with the current status of, and future plans for, the ongoing efforts to achieve software interoperability.

Introduction

Whole building integrated design can result in buildings that are energy efficient and of overall higher quality. To be successful, this design approach requires the use of multiple analysis tools capable of evaluating alternative design solutions from a variety of performance perspectives, of which energy efficiency is only one. However, the time and resources necessary for setting up completely separate analyses have historically limited the use of sophisticated tools such as energy simulation during the building design phase. All of the details of a design solution must be extracted from existing design documentation such as CAD drawings and supporting text documents, and reentered into the energy simulation tool. This procedure is not only costly in time and resources; it is also prone to error. The turn-around time for evaluating each proposed design solution can mean a critical delay of days or even weeks. The alternative of waiting to evaluate and correct a nearly completed design solution, leads to sub-optimal design corrections.

Furthermore, even in those projects that employ energy simulation during design, the use of these tools is difficult to dynamically carry forward to later stages of the building life cycle. The input to, and output from, energy simulation is static. Re-simulation of the building as it changes over time (e.g., from alterations in occupancy, use, or operation; or more substantial renovation) requires significant data reentry. Even if the input files and output results from the design stage have been archived, it is a difficult task to accurately collect all of the information on building changes, reconcile these changes with the earlier input, and compare new analysis output with the previous results. These difficulties mean that although energy simulation could be a useful tool in assuring optimum building performance over time, it is employed even less frequently during building operations and maintenance stages than during design.

But, what if a complete description of the building, adequate not only for energy simulation, but also for analyses such as construction cost estimation, occupant comfort and health evaluation, and chemical-biological vulnerability assessment, was easily accessible by multiple software programs? What if the building information created within CAD tools could be easily shared with energy simulation and cost estimation tools, and the additional information input by users of these tools could be easily propagated to others? What if all of this information was archived in a manner that it could be dynamically updated to reflect changes over time, and made accessible to authorized participants in the building process throughout the life of the building? This is the promise of software interoperability that is now beginning to be realized.

Software interoperability is the ability to share data amongst a variety of related yet different software programs. Data that are input into, and generated by, each of the individual software programs, can be readily transferred to the other programs. This data sharing can be achieved by mapping the relevant data within each program to a data model that comprises a superset of the information required by the full set of programs.

The Industry Foundation Classes (IFC) Common Data Model

The International Alliance for Interoperability (IAI) was formed in 1994 with the mission of defining, publishing, and promoting specifications for a common data model for the AEC/FM (architectural, engineering, construction, and facilities management) industry that would support interoperability between the wide variety of existing and future software tools used by industry participants (IAI 2002). The IAI is a non-profit organization, of which ASHRAE is a member.

The common data model now under continuing development by the IAI is called the Industry Foundation Classes (IFC), and is an object-oriented specification of the attributes of, and relationships between, building related entities. The IFC data model represents tangible building entities such as walls, windows, ducts, and chillers, and intangible entities such as project, task, and budget, as objects with sets of attributes. Several versions of the evolving IFC model have been released by the IAI beginning with Release 1.0 in 1996. Commercial software capable of importing and exporting IFC data is now available. The first of these IFC-compliant software tools on the market were primarily CAD, and are compliant with IFC Release 1.5.1, which was the first IFC version supported by official compliance testing defined by the IAI. A number of additional tools for various applications have now been certified as compliant with either IFC Release 1.5.1 or Release 2.0, and are commercially available. For up-to-date information on software that is compliant with the IFC model see the IAI Implementation Support Group website (IAI-ISG 2002a) and the Building Life-cycle Interoperable Software project website (BLIS 2002).

The most recently released version of the IFC data model is IFC2x, which was officially released by the IAI in London on October 27, 2001. The documentation and specifications for IFC2x are available for download from the Internet (IAI-UK 2002). The majority of IFC-compliant software developers have publicly committed to implementing support for the IFC2x release in 2002.

Creating Interoperable Software

Barriers

There are several chicken-and-egg problems associated with creating interoperable software and getting it adopted by the marketplace. First, it is difficult to develop IFC-compliant software prior to having a stable, robust IFC data model. Yet, it is difficult to define a stable, robust model until there are software tools that can be used to test and improve the model. Second, software users who do not fully understand the benefits of interoperability do not create a market demand for interoperable tools. Yet, software vendors who do not perceive a strong market demand are unwilling to commit resources to developing new capabilities. Interoperable software is like a fax machine, useless without other fax machines with which to send and receive data. Third, it is very time consuming to develop new software capabilities based on new technology before software development toolboxes have been created to facilitate the process. Fourth, it is impossible to verify the cost and time savings from interoperable data exchange prior to gaining experience with full-function interoperable tools on real-world projects. Yet, it is difficult to convince users to employ new tools that do not have well-documented benefits. Fifth, it is problematic to get the AEC/FM industry, which is itself characterized by fragmentation, to move beyond the mindset of using standalone software tools that cannot communicate outside each specialized discipline.

These issues have resulted in a slow progression from specifying the IFC data model to deploying interoperable software based on this model. We have by no means reached the end goal of this effort, however substantial progress has been made. The extant releases of the IFC model have reached a sufficient level of maturity and stability. Over twenty IFC-compliant software tools are now available including CAD, cost estimation, energy simulation, HVAC design, constraint checking, and software development tools. These software tools are not yet all fully functional, but they are being used in pilot projects to assess and improve their use.

Software Implementation

Defining a common data model is the first step toward creating and deploying interoperable software for the AEC/FM community. With the release of IFC 1.5.1, 2.0 and 2x, that step has been taken. The second step is implementing the ability to exchange data between software tools based on this model.

Each software tool inevitably defines its own unique data representation of information that is relevant to using that tool. This is nearly impossible to avoid since each tool is tailored to its specialized application, and the tool developers are focused on the specific capabilities required by that application. The principal behind software interoperability based on the IFC data model is that each software tool must map its internal data model to the common IFC model. For example, the internal representation of a wall within an energy simulation tool must be mapped to the representation of a wall in the IFC model. This is a non-trivial task since the IFC representation of a wall must accommodate the full variety of ways in which various tools view a wall. The implementation of data mapping requires considerable developmental effort to assure that each tool correctly interprets data elements in a consistent way. This difficulty becomes more clear when you recognize that a wall can be alternately viewed as a multistory building façade or a sub-section of this façade that bounds a single interior space, and must be described using thermal, structural, and aesthetic attributes, among others. Also, the mapping must be two-way so that data can be both imported into a software tool from an IFC building representation, and exported from the tool to an IFC representation.

The software implementation of this data mapping can be accomplished in different ways, as illustrated in **Figure 1**. A software application can implement the import/export of IFC data as either an internal/integrated set of methods, as illustrated by Application A in Figure 1, or as an external utility, as illustrated by Application B. These approaches both directly access data within an IFC Project File, which is an ASCII text file that uses either the ISO standard (ISO 1998) EXPRESS language, or XML the Extensible Markup Language (W3C 2002), to represent data based on the IFC specification. Although the data contained within an IFC Project File could also be archived in a specialized database, this is not the current approach in use by most software vendors. An alternative to direct access of the IFC Project File is to make use of a client/server model. In this approach, a data server that has been developed to the IFC standard provides indirect access to the IFC data. Clients to this server can then either be implemented externally to an application as illustrated by Application C, or embedded within the application as in Application D.

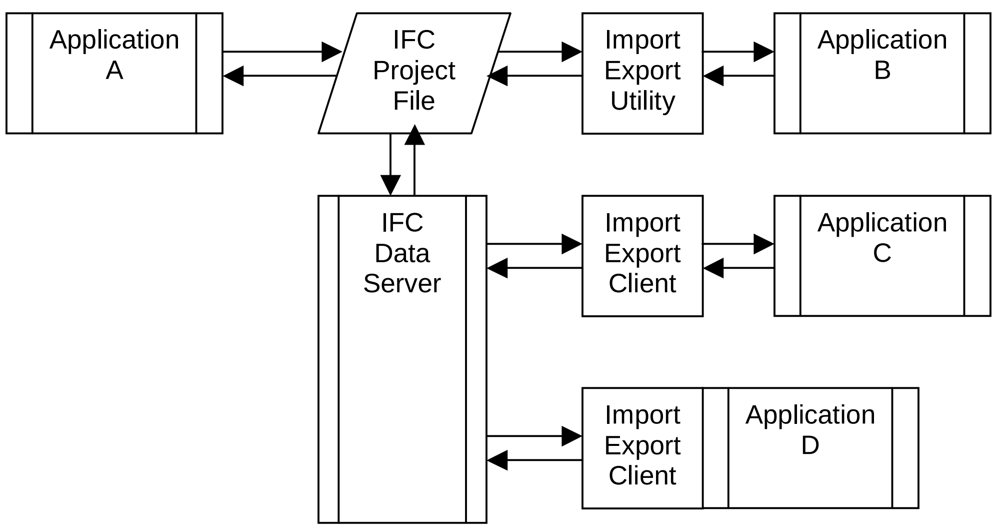

Figure 1. Implementing IFC-compliant software interoperability.

The client/server model offers the advantages of a middleware toolbox (the server) that eases the development process. For example, the server can add capabilities or modify the methods by which it accesses IFC data, without forcing the developers of Applications C or D to modify their client code. There are now several such IFC data servers available to software developers (IAI-ISG 2002b). These toolboxes make the previously very difficult task of directly importing/exporting IFC data much simpler to implement (Karola, et al. 2001).

Interoperability For Energy Simulation

The data requirements for performing sophisticated energy simulation are substantial, including building geometry, construction material characteristics, HVAC component and system characteristics, and internal loads and schedules. Historically these data have been collected from a variety of sources for a given building project and manually input to a standalone energy simulation tool. The user interfaces to these standalone tools have improved over the years, but this has only served to make the data entry less painful. Furthermore, having entered the data into the energy simulation tool has not help make these data accessible to other analysis tools, even closely related tools such as lighting/daylighting analysis.

This problem is being addressed by software that is compliant with the IFC data model (Karola, et al. 2001). Building geometry has been the primary focus of the initial software implementation for energy simulation, as has been the case for virtually all other IFC-compliant implementation efforts. A building can now be drawn within an IFC-compliant CAD tool and saved to an IFC project file instead of to a file in the native format of the CAD tool. The geometry of building entities such as walls, windows, roofs, floors, and spaces can then be imported into an energy simulation tool from the IFC project file. The overall flow of data for these initial implementations is illustrated in **Figure 2**. This level of interoperability can achieve a huge savings in the amount of time that has traditionally been consumed by this process. Note however that the flow of data is currently only one way, from the IFC project file into the energy simulation application. This is not the case for all international software implementation efforts, but it is the case for U.S. implementation efforts (Hitchcock 2001).

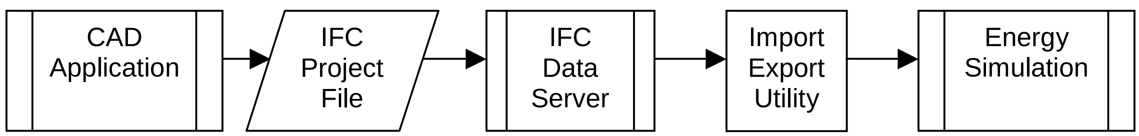

Figure 2. Flow of data from CAD through IFC to energy simulation.

The IFC model represents not only the geometry of each of the individual building entities, but also additional data such as the entities' thermal characteristics. The initial focus on exchanging only the geometry of these entities has not been dictated by shortcomings in the IFC data model, but rather by the decision to begin by implementing a subset of the IFC model that is of use to the widest range of active software implementors. We are currently working to extend the capabilities of these early implementations to address the additional data requirements of energy simulation software, including representation of HVAC components and systems, and two-way flow of data both into and out from energy simulation.

Current Status and the Future

The effort to achieve software interoperability for the AEC/FM industry has made considerable progress over the last several years. A relatively stable and robust common data model has been developed by the International Alliance for Interoperability. There are now three extant versions of this model that have been released. The most recent version, IFC2x, is intended to be a stable platform for software development over the next couple of years. Incremental extensions to this platform will be made available as they are developed, before being integrated into a new platform release in the future.

Software vendors and other developers have implemented interoperability for a wide variety of application areas including CAD, visualization, energy simulation, energy code compliance, HVAC design, cost estimation, construction formwork planning, building code checking, and general constraint checking. A number of software development toolboxes are now also available to assist in the implementation of new interoperable software. This is truly an international effort with a large portion of the currently available software targeted to countries other than the U.S.

Pilot building projects are now being undertaken employing these available tools. These projects tend to be carefully constrained since there is not yet a lot of experience in using these tools in active projects. This is the next necessary step in working toward the deployment and industry adoption of this technology that has such great potential, but has yet to prove itself in the marketplace. As the benefits of these early applications of interoperability are recognized by end users, and as current limitations are exposed and overcome, the barriers to creating and deploying this technology will continue to fall.

Work also continues on extending the IFC data model to fulfil the data requirements of various disciplines within the AEC/FM industry. In particular, work is well along on extending the HVAC portion of the model to more fully support sophisticated energy simulation (LBNL 2002). This work will include the enhanced implementation of interoperability for energy simulation mentioned above.

There is still much work that needs to be done before software interoperability fully supports the myriad information needs of AEC/FM professionals across the complete life cycle of a building, from pre-design planning, through design, construction, commissioning, operations and maintenance, to final demolition. This work requires input from the full spectrum of users of this information, including the members of ASHRAE.

Acknowledgement

This work was completed under contract to Lawrence Berkeley National Laboratory as part of the High Performance Commercial Building Systems program. This program is supported by the California Energy Commission's Public Interest Energy Research (PIER) Buildings Program and the Assistant Secretary for Energy Efficiency and Renewable Energy, Building Technologies Program, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

References

BLIS. 2002. Building Life-cycle Interoperable Software project, www.blis-project.org.

Hitchcock, R.J. 2001. "EnergyPlus™ Interoperability: Acquisition of Building Geometry from IFC-Compatible CAD Tools," in Building Energy Simulation User News, Vol. 22, Number 4, July-August 2001.

IAI. 2002. The International Alliance for Interoperability, www.iai-na.com.

IAI-ISG. 2002a. The International Alliance for Interoperability – Implementation Support Group, www.bauwesen.fh-muenchen.de/iai/iai_isg/.

IAI-ISG. 2002b. An International Overview of IFC Development Activities. The International Alliance for Interoperability – Implementation Support Group, www.bauwesen.fh-muenchen.de/iai/ImplementationOverview.htm.

IAI-UK. 2002. The International Alliance for Interoperability – United Kingdom, http://cig.bre.co.uk/iai_uk/.

ISO. 1998. Product data representation and exchange: The EXPRESS language reference manual. ISO 10303-11:1994/Cor.2:1998. ANSI, New York, New York, USA, (also see www.nist.gov/sc4/wg_qc/wg11/n070/wg11_n070.htm).

Karola, A., et al. 2001. "BSPro COM-Server Interoperability Among Software Tools using Industry Foundation Classes," in Proceedings of Building Simulation 2001, August 13-15. International Building Performance Simulation Association.

LBNL. 2002. The Building Services 8 Project, Lawrence Berkeley National Laboratory, Berkeley, California, USA, http://eetd.lbl.gov/btp/iai/bs8/index.html.

W3C. 2002. The World Wide Web Consortium, www.w3.org.