REDUCTION, SIMPLIFICATION, TRANSLATION AND INTERPRETATION IN THE EXCHANGE OF MODEL DATA

Vladimir Bazjanac[1], Arto Kiviniemi[2]

[1] *Lawrence Berkeley National Laboratory, University of California, USA*

[2] *VTT Technical Research Centre of Finland*

ABSTRACT: A major purpose of Building Information Models (BIM) is to serve as a comprehensive repository of data that are retrievable by multiple software applications which participate in the same AECO industry project. Data placed in a BIM by one software application are retrieved and used by other applications. Retrieved data are at times not useable by the recipient application in exactly the same form as received; in such cases the received data are manipulated and/or transformed before they can be used.

This paper provides an overview of issues that arise when data transformation is necessary for "downstream" applications that use data authored by model based CAD and/or other interoperable software. These include manual and semi-manual data transformation, as well as rules for data set reduction and simplification, and rules for data translation and interpretation. The rules can be imbedded in data model views and middleware to become part of a seamless process of data exchange and sharing.

KEYWORDS: buildings, data modeling, BIM, data transformation, data reduction, data, simplification, data translation, data interpretation, rules of transformation.

1 INTRODUCTION

The Architecture-Engineering-Construction-Operations (AECO) industry is showing a renewed interest in using information technology (IT) more effectively in daily professional practice. Building Information Models (BIM) and modeling (Bazjanac 2004) are currently a "hot topic" of discussion throughout the industry. New industry consortia and alliances, such as buildingSMART (IAI-NA 2006), FIATECH (FIATECH2007), Construction Users RoundTable (CURT 2007), the Continental Automated Buildings Association (CABA 2007), Open Standards Consortium for Real Estate (OSCRE 2007), to name just a few, have been emerging with increasing frequency in the last few years. Professional societies' new task forces and committees, such as the American Institute of Architects (AIA) Technology in Architectural Practice (TAP 2005) and the Steering Committee on Interoperability and buildingSMART of the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE 2007) among other, have also been emerging with increasing frequency. New initiatives and projects, such as the U.S. General Services Administration Services (GSA) National 3D-4D BIM Program (GSA 2003), CIMsteel Integration Standards – Release 2 (CIS/2 2003), Information Delivery Manual (IDM 2006) and other, are beginning to have a visible industry wide impact. In one way or another, they are all attempting to improve and/or change industry processes and make design, procurement, delivery and operation of buildings more efficient, streamlined and cost effective. They are all trying to accomplish that by capitalizing on opportunities provided by contemporary IT.

Members of all AECO industry disciplines use software in daily work. Each discipline has its own "mission critical" software application or set of applications – professional software that is critical to conducting business in a particular segment of industry (Bazjanac 2002). Most of the applications used in this industry generate information which is then reused in some form by other applications.

Exchange and/or sharing of data among software applications have traditionally been disorganized and inefficient. When possible, data exchange is often based on "pointtopoint" exchange via software interfaces that map parts of internal data structure and sets of one application to the other. Or it is done by integrating applications: A group of software applications, usually from different industry disciplines (often called a suite of tools), share parts of the same data model and thus data are exchanged directly among the participating applications. Mostly, however, direct data exchange is not possible – any acquisition of already existing data is accomplished through (usually substantial and arbitrary) end user intervention, which often involves manual or semi-manual replication of already existing information. This dramatically slows down the involved process, usually introduces errors and omissions in the resulting data base, and can cause misunderstandings and misinterpretations in the process (Bazjanac and Crawley 1997).

Poor handling of computer based information prompted the formation of the Industry Alliance for Interoperability in the U.S. in 1994, which became the International Alliance for Interoperability (IAI) in 1996 (IAI 2007). The IAI has subsequently developed and released several versions of an open and extensible, life cycle data model of buildings – Industry Foundation Classes (IFC), which are still the only life cycle data model of buildings recognized by the International Standards Organization (ISO/PAS 16739). Such a data model provides the necessary fundamental building blocks that enable seamless software data exchange in the AECO industry. Enabling seamless data exchange is the ultimate goal of software interoperability and the IAI (IAI 2007).

In this context, seamless data exchange and/or sharing mean direct data exchange and/or sharing among (i.e. not only between two) software applications in the AECO industry. "Direct" means that no user intervention (i.e. manual modification of data by the user) is involved in the process. Data exchange and/or sharing refer here to data exchange and/or sharing that take place across industry disciplines and throughout the building life cycle. This implies data exchange and/or sharing among very diverse software application, some of which differ very much in their internal architecture, employ very different internal data models, and may require the definition of the same data in different format(s) or at different granularity.

A key concept in attaining direct data exchange and/or sharing is BIM as the authoritative repository of project information. BIM in this context is a data model of buildings instantiated with data that uniquely define a specific building; it serves as a repository for all project information that is subject to exchange and/or sharing. Populating such a data repository requires the use of software applications that are capable of populating the repository with data or retrieving data from it.

It is often assumed in this industry that it is only necessary to create interfaces to the common data model to make existing software applications interoperable. Such a view represents a somewhat simplistic understanding of issues that arise when trying to make software interoperable and is only partially true, as the subsequent discussion will show.

Different software applications typically reflect different "views" of the same building, as industry disciplines they serve "see" the same building differently – they must each deal with issues that are unique to their discipline. For example, architectural applications (such as the various CAD tools) support the definition of buildings in the way architects "see" the building and model all the pertinent information about the building architects generate. Some of that information, however, is irrelevant to ducting and plumbing and is not part of the ducting and plumbing "view" of the same building. Some of the discipline "views" cane be substantially different; Figures 1 and 2 graphically show the difference between the "architects' view" and the "thermal view" that is required for the simulation and analysis of energy performance of the same building.

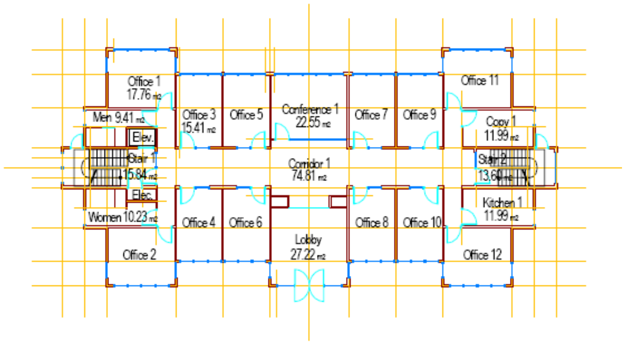

Figure 1. Geometry of the ground floor of an office building, as depicted in a typical architectural CAD tool.

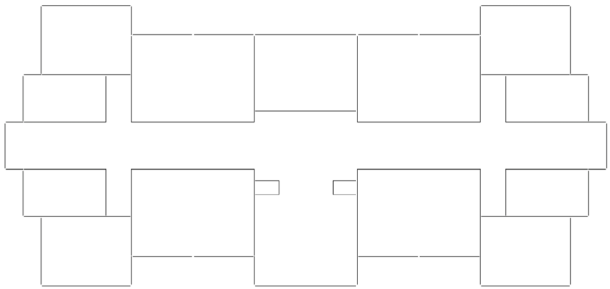

Figure 2. Geometry of the same office building ground floor, as depicted by an energy performance simulation and analysis tool, shows that office and other spaces that exhibit the same thermal behavior are merged into respective thermal zones. This depiction of the building's geometry is considerably simpler than that shown in the previous figure – note the single line representation of walls, omission of doors, and the lack of any detail in the representation of windows.

Views of the data model determine data sets and formats of data for the given industry discipline, industry process, and even a given industry organization that have to be available for exchange and/or sharing among software applications that share that view. The IAI recognized the need for the definition of such views and adopted a formal Model View Definition (MVD) methodology applicable to the IFC data model (Hietanen 2007). U.S. GSA has already published its "spatial program validation view" (GSA 2006) that will be required for all new projects; other views of IFC are under development.

Because of the diversity of software applications that may implement a given model view, some of the exchanged and/or shared data may inevitably have to be transformed before they can be used by a given application. The rest of this paper discusses types of such transformation and issues that arise with it. In this context software integration is a subset of software interoperability (NBIMS 2007). General issues related to interoperability in the AECO industry discussed here apply equally to software integration.

2 TRANSFORMATION OF IMPORTED DATA

When a software application is the first one to create specific data about a building, it is "authoring" those data. Such data constitute the "original" data that can then be used by other software applications, which at that point can be called “downstream” applications. Downstream applications may, and often do, support different disciplines and have implemented different model views than the authoring application.

CAD applications give shape to a building and, in the process of documenting it, are the first to create original data. Additional data are subsequently authored by downstream applications. Because the need to exchange data among CAD applications is relatively infrequent, the real payoff from software interoperability is seamless data exchange with and among downstream applications. But that data exchange is not always automatic or straight forward.

A software application imports or generates itself all data it manipulates. To obtain valid results, data imported by an application must not only be in a form and format that is readable by the application, but also must represent values the application expects to import. For example, if a downstream application expects to import a value for floor-tofloor space height, the imported value must represent floor-to-floor and *not* floor-toceiling space height. Because of the diversity of applications (and their internal data structures) that may participate in a given data exchange, “original” data must often be transformed before they can be used by a downstream application – data sets must be reduced and/or simplified, or data must be translated and/or interpreted.

2.1 *Data set reduction and simplification*

Complex data sets, as originally defined, are often too “rich” to be imported in their original form by a downstream application – such data sets include data that are important to the authoring application and are perhaps required by applications of the same or similar type, but are irrelevant to other applications. For example, a typical model based CAD application generates all sorts of information it needs for precise visualization of building geometry. To be able to reproduce the same precise visualization, another model based CAD application needs to import this detailed information.

In the case of a wall shown in Figure 3 (a simple and obvious example), the original information will include precise data that describe any openings, protrusions, depressions, textures, etc. on each wall surface (Figure 3, left); to reproduce that wall in detail, another application needs to import the original data set. Other applications reproduce that wall in much less detail, and need only a subset of the original information – perhaps only wall length, height and its “center-line” position (Figure 3, right). For such an application the original data set is *reduced* in size and *simplified* in form before it is imported and deployed by the application.

2.2 *Data translation*

Some authoring applications at times generate data in form or formats that cannot be recognized or accepted by a downstream application. For example, structural concrete in a building may be expressed in ft^3 when originally defined; a downstream application may need it expressed in kg. In that case the original information must be *translated* into form (units in this case) acceptable to the downstream application. Such data translation can be straight forward if the rules of translation and the involved data are unambiguous.

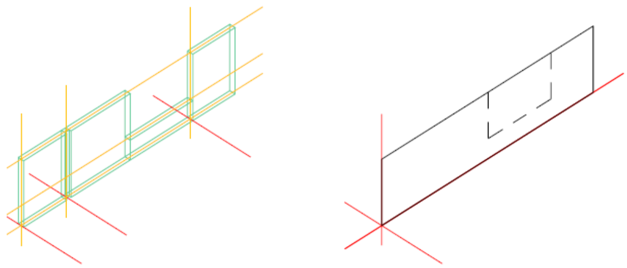

Figure 3. The wall on the left shows typical detail as defined in CAD: geometry in 3-D with openings, surface depressions, 3-D reference grid lines and more; the wall on the right shows the representation of the same wall as defined in a downstream energy performance simulation tool – the wall is positioned along its center line, has only length and height (no thickness), and an origin point (instead of grid lines). The opening is subtracted from wall surface. The data set required to define the wall on the right is smaller and simpler than for the wall on the left.

Sometimes data needed by a downstream application exist, but represent something else in their original form. Data translation in that case involves recognizing that an original datum also represents something else, and renaming and perhaps reformatting it per expectation and requirements of the downstream application.

A good example of data translation is the identification of exterior walls. Numerous downstream applications distinguish between interior and exterior walls, and define and treat them differently. Most building geometry authoring tools do not consider that distinction and do not explicitly include it in data they originate. Data translation in this case involves the detection of walls and/or wall segments that are facing the exterior, and naming them as exterior.

2.3 *Data interpretation*

At other times data needed by an application do not exist in the form needed, but can be derived from available data. For example, the original data sets describing a wall may include specific information about the type of all materials in the compositions of the wall and their thickness, but not specifically include the K-value for the wall. In that case the K-value can be derived from existing information and delivered to the downstream application that needs it – in this case the original information is *interpreted*.

When data required by downstream applications have not been *explicitly* defined before, it may be possible to derive them from original data *which themselves may not be needed by these applications*. The interpretation of original information that yields derived data then becomes a process that can involve recognition, extraction, sorting, and calculation.

Some data interpretation is relatively simple and quick. A good example is determination of efficiency factors for a building, such as ratios of building exterior wall area to the building gross floor area (needed by a prescriptive building energy code compliance application), or the cost of plumbing per fixture (needed by a value engineering application). In the former case, the determination of the ratio requires the identification and summary of all exterior walls and their surface areas from the original building geometry data sets, as well as of all gross floor area; the ratio is obtained by simple division. In the latter case, determining the ratio involves isolating the total cost of plumbing in the cost estimate and counting all plumbing fixtures, then dividing the cost by the number of fixtures.

Other cases of data interpretation require much more effort to complete. For example, different industry disciplines calculate the building "net area" using different rules. Determining the area in the building which is useable for a particular purpose (that a facilities management application may need) requires detecting all spaces intended to serve that purpose in a building, and adjusting individual detected space's net area to fit the area definition rules incorporated in that particular application.

There is another kind of data interpretation: *by end user*. Such interpretation is usually subjective and unpredictable, resulting in different interpretation results of the same data by different users. It is based on individual end user's world view, depth of knowledge, educational background, understanding of the problem, set of skills, preferences, available resources, etc., and is often not reproducible. Seamless data exchange eliminates opportunities for this kind of data interpretation.

3 CURRENT PRACTICE: MANUAL DATA TRANSFORMATION

Seamless data exchange is employed in current industry professional practice relatively rarely, even though the use of interoperable software is increasing (CIFE 2007). When deployed, it is seldom deployed throughout the entire project or industry process, let alone throughout a building's life cycle. Without seamless exchange, data transfer among applications is often affected by human intervention – end users transform data and data sets *themselves* before importing them into downstream applications.

Manual data transformation can be highly subjective and unpredictable. If different users each transform the *same* data or data set, chances are that each will come up with a different result. This is not surprising – it is inevitable, as stated above.

Results from manual data set reduction and simplification usually include at least a few errors and/or omissions. The same is true of manual data translation and interpretation; these can also result in data misrepresentation and misinterpretation, where original data are given additional meaning not originally intended. Manual data transformation can, unwittingly, even result in contradiction – data with values that contradict each other – that inevitably renders results from software that used such data unreliable or meaningless. Finding and correcting erroneous, misrepresented and contradictory data, and finding and adding missing data can be very laborious and frustrating tasks.

Results from other than simple and straight forward manual data transformation are usually not reproducible and can be questioned. Justifying such transformation results requires the documentation of qualifications and explanation of assumptions made during the transformation, a laborious process that is seldom performed.

Substantial manual data transformation inevitably delays the industry process or task within which it is performed, and increases its cost. It also delays the productive use of affected downstream applications, which is why results from such applications are often ignored – they are generated too late to have an effect. These are also the major reasons why some of the downstream applications are not employed as often as they could and should be (Bazjanac 2007).

4 RULES FOR DATA TRANSFORMATION IN DATA TRANSFER

The need to transform original data is real and evident: it is encountered and performed when downstream software applications are used throughout the life cycle of a building. If the transformation of original data is performed in an arbitrary manner, the results too can only be arbitrary.

Seamless data exchange can work properly only if needed data transformation is an *integral part* of the seamless data exchange process. This means that data transformation must also be automated, and that it must be an *unambiguous* transformation that in each instance yields the *same data form and value* for *all* applications that need to import and deploy those transformed data and/or data sets.

Each data transformation should be governed by rules that unambiguously specify how the transformation is performed. These rules should account for every instance of data transformation. They should be *agreed to* per industry discipline by its professionals as well as developers of each discipline's software; they should be *accepted and deployed* in each discipline's software applications.

Rules of transformation should be coded in software. The coding would prevent any end user intervention (such as unauthorized change of rules and any manual transformation of data and data sets), and would assure that the same results are obtained from the same data transformation process in each and every instance of its use with the same original data.

Rules of data set reduction and simplification should typically be incorporated in data model view definitions. Downstream applications that support such view(s) will then automatically receive properly reduced and simplified data sets. If no applicable model view exists yet, rules should be built into middleware that serves respective applications. The middleware should also include applicable data translation and interpretation rules, and should be automatically executed as part of the seamless data exchange process.

4.1 *An example: geometry simplification tool*

Geometry Simplification Tool (GST) is middleware that transforms building geometry and construction materials data authored by IFC compatible model based CAD tools, such as ArchiCAD, Revit, AutoCAD Architecture, MicroStation, or Allplan Architecture, into geometry and construction materials definitions directly readable by applications that support gbXML (Kennedy 2002). The primary target of GST is IDF Generator, a preprocessor for building energy performance simulation engine EnergyPlus (LBNL 2001).

GST is developed jointly by Graphisoft and the Lawrence Berkeley National Laboratory (LBNL). It reads any valid IFC file that, as part of building geometry data, includes definitions of space boundaries; it extracts pertinent information from the IFC file and creates input data that describe building geometry and construction materials in form acceptable to "whole building" energy performance simulation and analysis applications.

Several rules for data set reduction and simplification, as well as rules for data translation and interpretation have been imbedded in GST. These include rules for data set reduction and simplification for walls and slabs, as described in Figure 3. GST rules for data translation include the case described earlier in Section 2.2: DATA TRANSLATION. Some of the other rules imbedded in GST are rules of data interpretation. These include:

- Ordering all vertices that define a surface or an opening in clock-wise sequence. Such vertices usually appear in inconsistent order when originally defined and have to be re-ordered for use in applications like EnergyPlus.
- Calculation of surface outward normal. Original data, as transmitted in an IFC file, currently do not include definitions of surface outward normals. These are unambiguously derived from the sequence of surface vertices.
- Rules of defining concave polygons. Concave polygons have to be split into two or more rectangular or convex polygons before they can be used by EnergyPlus and similar downstream applications. These rules determine how and in which order such division is done.
- Identification of spaces on the other side of space boundaries (i.e. on the other side of walls and slabs). Original data transmitted in an IFC file associate a surface with the space that surface belongs to, but not with the space on the other side of that surface. Detection of spaces "on the other side" involves multiple tracing of Global Unique ID (GUID) components in the IFC file employing rules of tracing developed for this purpose.
- Definition of construction materials layering sequence. Applications that author definitions of composite constructions usually obtain such definitions from software libraries. Such definitions do not always consistently identify the outside or the inside, nor do they always define layers of construction in the order needed by downstream applications. Rules incorporated in GST define the sequence of construction layers in composite constructions from the outside in.

GST is currently undergoing rigorous testing and debugging. Preliminary tests have shown that, with minor appropriate additions to its rule set, GST can generate usable building geometry input for other types of downstream applications as well.

5 CONCLUSIONS

Data exchange and/or sharing in the AECO industry often require reduction and/or simplification of original data sets, as well as data translation and/or interpretation, when downstream applications are involved. Such data transformation is currently mostly done manually or semi-manually by end users, which often causes errors, omissions, contradictions and/or misrepresentations that are difficult and costly to detect and correct. Consequently, manual or semi-manual data transformation delays the process to the point where any subsequent productive use of downstream applications becomes irrelevant because it is too late.

Definition of data transformation rules can rectify this if the rules are incorporated in data model views or in middleware that is designed to prepare data and data sets for use by downstream applications. Such rules unambiguously define the necessary data transformation; when imbedded in the process of seamless data exchange, they eliminate any opportunity for manual or semi-manual data transformation by end users. Thus, such rules can eventually lead to much more frequent and productive use of downstream "mission critical" software applications.

The "promise of software interoperability" will be fulfilled by seamless data exchange and sharing among *downstream* applications in the future. Seamless data transformation, based on accepted rules, will inevitably play an integral role in achieving software interoperability.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the contribution of Janos Maros, Gyula Kiss and Istvan David from Graphisoft, and Dr. Robert J. Hitchcock from the Lawrence Berkeley National Laboratory in the definition of data transformation rules incorporated in GST.

Work resulting in this paper was partly supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technology, Building Technologies Program of the U.S. 11 Department of Energy under Contract No. DEAC03-76SF00098. It was also partially funded by Tekes, the Finnish Funding Agency for Technology and Innovation.

REFERENCES

ASHRAE (2007). The American Society of Heating, Refrigerating and Air-Conditioning Engineers. http://www.ashrae.org/.

Bazjanac, V. (2007). Impact of U.S. National Building Information Model Standard (NBIMS) on Building Energy Performance Simulation. Accepted for publication and presentation at the Building Simulation 2007 conference, Beijing, September 2007.

Bazjanac, V. (2004). Virtual Building Environments - Applying Information Modeling to Buildings. In A. Dikbaş and R. Scherer (eds), *eWork and eBusiness in Architecture, Engineering and Construction, Proc. fifth Euro. conf. product* *process modelling, Istanbul,TR*:41-48. Balkema. ISBN 04 1535 938 4.

Bazjanac, V. (2002). Early Lessons From Deployment of IFC Compatible Software. In Ž. Turk and R. Scherer (eds), *eWork and eBusiness in Architecture, Engineering and Construction, Proc. fourth Euro. conf. product process modelling, Portorož, SLO*: 9-16. Balkema. ISBN 90-5809-507-X.

Bazjanac, V., and Crawley, D.B. (1997). The Implementation of Industry Foundation Classes in Simulation Tools for the Building Industry. In J.D. Spitler and J.L.M. Hensen (eds), *Proceedings of Building Simulation '97, Prague*, 203-210. IBPSA. ISBN 80-01-01646-3.

CABA (2007). The Continental Automated Buildings Association. http://www.caba.org/index.html.

CIFE (2007). Center for Integrated Facility Engineering, Stanford University. Results of the VDC/BIM Use Survey. April 2007. http://cife.stanford.edu/VDCSurvey.pdf.

CIS/2 (2003). CIMsteel Integration Standards. http://www.cis2.org/.

CURT (2007). Construction Users RoundTable: Mission.http://www.curt.org/2_0_about_curt.html.

GSA (2006). General Services Administration Public Buildings Service (PBS), Office of Chief Architect OCA): Spatial Program Validation. http://www.gsa.gov/Portal/gsa/ep/contentView.do?programId=12122&channelId=-18161&ooid=20917&contentId=21768&pageTypeId=8195&contentType=GSA_BASIC&programPage=%2Fep%2Fprogram%2FgsaBasic.jsp&P=PMBIM.

GSA (2003). General Services Administration Public Buildings Service (PBS), Office of Chief Architect (OCA):National 3D-4D BIM Program. http://www.gsa.gov/Portal/gsa/ep/channelView.do?pageTypeId=8195&channelPage=%252Fep%252Fchannel%252FgsaOverview.jsp&channelId=-18161.

FIATECH (2007). FIATECH: Fully Integrated and Automated Technology: Mission. http://www.fiatech.org/about/mission.htm.

Hietanen, J. (2007). IFC Model View Definition. http://www.iaiinternational.org/software/mvd.shtml.

IAI (2007). International Alliance for Interoperability. http://www.iai-international.org.

IAI-NA (2006). International Alliance for Interoperability: buildingSMART. http://www.iai-na.org/bsmart/.

IDM (2006). The Information Delivery Manual. http://www.iai.no/idm/index2.html.

Kennedy, J. (2002). Green Building XML. http://www.gbxml.org/about.htm.

LBNL (2001). EnergyPlus, A New-Generation Building Energy Simulation Program. http://gundog.lbl.gov/EP/ep_main.html.

NBIMS (2007). The National Building Information Standard, Version 1 – Part 1,: Overview, Principles and Methodologies. National Institute for Building Sciences, Washington, DC. 2007. http://www.facilityinformationcouncil.org/bim/pdfs/NBIMSv1_ConsolidatedBody_11Mar07_4.pdf.

OSCRE (2007). Open Standards Consortium for Real Estate.: Mission. http://www.oscre.org.

TAP (2005). American Institute of Architects, Technology in Architectural Practice. http://www.aia.org/tap_default.