A comparative study of the IFC and gbXML informational infrastructures for data exchange in computational design support environments

B.Dong[1], K.P.Lam[1], Y.C.Huang[1], Dobbs,G M[2]

[1]Center for Building Performance and Diagnostics, Carnegie Mellon University, Pittsburgh, PA 15213
[2]United Technologies Research Center, East Hartford, CT 06108

ABSTRACT

Significant progress has been made in the area of common data exchange in the building industry with the development of information technology. Currently, the Industry Foundation Class (IFC) and Green Building XML (gbXML) are two prevalent informational infrastructures in the architecture, engineering and construction (AEC) industry. IFC and gbXML are both used for common data exchange between AEC applications such as CAD and building simulation tools.

This paper presents a detailed investigation and comparative study of the differences between IFC and gbXML in terms of their data representations, data structures and applications. It aims to explicitly illustrate the complex data representation through selected examples of the respective schema. Two specific demonstrative cases will include building element specification (i.e.,enclosure geometry) and building sensors (control and operation). Findings will be reported on the following aspects: (1) the strength and challenges of the diametrically opposing approaches between IFC and gbXML; (2) hierarchical structure of the schema in support of extensibility, data extraction, ease of implementation etc.; (3) formal adoption and application.

Based on the results of this study, the gbXML schema is selected for development to demonstrate the features of gbXML. A proposed XML schema for lighting simulation will be presented. It aims to provide a seamless data integration platform between a CAD model (i.e., REVIT) and lighting simulation software (i.e., Radiance) in this study to support concurrent design of high performance buildings.

KEYWORDS

IFC; gbXML; Common data exchange; Lighting simulation.

INTRODUCTION

The challenge to design complex high performance buildings demand a paradigm shift from the traditional sequential to an interactive concurrent design environment. This implies the need to adopt new integrative design processes and the associated computer-based design support tools need to be re-conceptualized and re-engineered within an integrated IT environment. Such a system must not only be capable of supporting concurrent performance evaluation of building design and engineering across domains, but be accessible across geographically distributed environments through both synchronous and asynchronous modes of communication (Lam et al., 2004). With the development of information technology, significant progress has been made in the area of common data exchange in the building industry. The first generation of data exchange models such as STEP (ISO 1992), COMBINE (Augenbroe 1992), ICADS (Pohi and Reps 1988), ARMILLA (Gauchel et al., 1992), and IBOM (Sanvido 1992) focus mainly on building geometry information. It is necessary but evidently insufficient for any performance-based design support. The second generation data exchange models expanded to include information that is required for particular domain-specific performance modeling, such as energy/thermal comfort simulation, HVAC design, lighting design, building life-cycle analysis and building plan checking. The hierarchical models (Bedell and Kohler 1993) and KNODES (Rutherford, J, 1993) for energy and life cycle cost modeling are examples. These "object-oriented" data models begin to address the larger question of interoperability between various performance domains in building design. Perhaps more importantly, it offers new opportunities for integrative concurrent design support (Lam et al., 2002). The current third generation models are able to potentially encapsulate "all" information related to a building in a comprehensive data schema and facilitate the sharing of pertinent information througout the entire building life-cycle, from design (for evaluating total building performance) to construction (for evaluating cost and schedules) to operation (occupancy and environmental sensing and system controls).

One key factor in the success of any data schema is in the wide-spread adoption by all constituencies in the architecture, engineering and construction (AEC) industry. Currently, the Industry Foundation Class (IFC)(IAI 2006) and Green Building XML (gbXML)(gbXML 2007) are two main and prevalent informational infrastructures in the industry. The goal of IFC is to provide a universal basis for process improvement and information sharing in the construction and facilities management industries. EXPRESS-G is used to identify classes, the data attributes of classes and the relationships that exist between classes in IFC (IAI 2006). It is a graphical modeling notation developed within the ISO-STEP community for representing application interpreted models. Some major impementations of this schema include CORENET (Singapore), BuildingSMART (Australia), SARA (Finland), Building Lifecycle Interoperable Cost (BLIS, 2007), Simple Access to the Building Lifecycle Exchange (SABLE, 2007) and BSPro (Bazjanac 2002, 2004)

The gbXML schema was developed by Green Building Studio (formerly GeoPraxis) with the support of the California Energy Commission Public Interest Energy Research (PIER) Program, and the California Utilities (Pacific Gas and Electric Company, Southern California EDISON and Sempra Energy Utility). gbXML currently facilitates the exchange of data among CAD and energy analysis software. Several popular CAD software (e.g., Autodesk's Revit, Architectural Desktop, and Graphisoft's ArchiCAD) and energy analysis applications (e.g., DOE-2. e-QUEST, HAP) can exchange data using this schema, through the Green Building Studio web service (DOE-2, 2007).

gbXML is developed based on the XML (Extensible Markup Language) format. XML provides a robust, non-proprietary, persistent, and verifiable file format for the storage and transmission of text and data both on and off the Web (W3C, 2006). XML allows groups of people or organizations to create their own customized markup language for exchanging information within and between their respective domains of interest. It has a global "language" format, with consistent syntax and can potentially represent any computational building model through translation using appropriate mapping engines. Notwithstanding the consistent language format, the implementation of the actual data model or schema, with their associated semantics, can still vary significantly. This presents a challenge to advancing interoperability within the industry. The following sections illustrate this point by comparing and contrasting the data structures and representations of building geometry and sensor.

COMPARISON BETWEEN IFC AND GBXML

Geometry information

IFC Representation

As one example of geometry information representation, Figure 1 shows how the IFC schema represents the coordinates and dimensions of an *IfcWall* object.

IfcWall is a subtype of *IfcBuildingElement.*
IfcBuildingElement is a subtype of *IfcElement*, which generalize all components that make up an AEC product such as walls, windows or doors.

IfcElement is a subtype of *IfcProduct. IfcProduct* has two attributes named ObjectPlacement and Representation. ObjectPlacement defines the starting point of *IfcWall*. It can be given by (1) absolute value relative to the world coordinate system by *IfcGridPlacement*; (2) relative, value relative to the object placement of another product by *IfcLocalPlacement*; (3) by grid reference i.e., by the virtual intersection and reference direction given by two axes of a design grid through *IfcGridPlacement*.

The followings give an example of how *IfcWall* is represented as *IfcProductRepresentation* to elaborate the relational and organized data representation of IFC.

The *IfcProductRepresentation* defines a representation of a product, including its geometric or topological representation. It has two attributes: *IfcProductDefinitionShape*, and *IfcMaterialDefinitionRepresentation.* The *IfcProductDefinitionShape* defines all shape relevant information about an *IfcProduct*. It allows for multiple geometric shape representations of the same product. The *IfcRepresentation* defines the general concept of representing product properties. (IAI, 2006). *IfcPlacement* locates a geometric item with respect to the coordinate system of its geometric context. It has two attributes called Location and Dim, where Location refers to the geometric position of a reference point such as the center of a circle, of the item to be located, Dim refers to the space dimensionality of this class and is a *IfcDimensionCount* object, derived from the dimensionality of the location. *IfcPlacement* has three subtypes: *IfcAxis1Placement*, which defines the direction and location in three dimensional space of a single axis. *IfcAxis2Placement2D* is used to locate and originate an object in two dimensional space and to define a placement coordinate system. *IfcAxis2Placement3D* is used to locate and originate an object in three dimensional space and to define a placement coordinate system.

Hence, a wall gains its geometric position and orientation by virtue of a reference to axis2_placement (*IfcAxis2Placement*) that in turn references a cartesian_point (*IfcCartesianPoint*), several directions (*IfcDirection*) and its starting point (*IfcVirtualGridIntersection*). *IfcCartesianPoint* has an attribute called Coordinates, which is a list of 1 to

3 *IfcLengthMeasure* objects. This is where the coordinates are represented.

gbXML representation

In order to compare with IFC approach, the same examples are given in gbXML approach. As mentioned before, gbXML is developed based on XML, which captures data information representation but not the relationships among them. Figure 2 shows gbXML schema of geometry information representation. All the geometry information imported from CAD tools are represented by the “Campus” element. The global child element “Surface” represent all the surfaces in the geometry. There are several attributes defined in “Surface” such as “id” and “surfaceType”. Every “Surface” element has two representations of geometry, “PlanarGeometry” and “RectangularGeometry”. They both carry the same geometry information. The purpose of this is to double-check whether the translation of geometry from the CAD software is correct or not. Every “RectangularGeometry” has four “CartesianPoint” elements which represent a surface. Every “CartesianPoint” has a three dimensional representation, three coordinates (x,y,z).

Below is an example of a partial XML file generated by gbXML with surface type “Exterior Wall”. “RectangularGeometry” defines the starting point of this surface. “PlanarGeometry” defines all the coordinates. Again, there are only fives levels to transverse to get the all the coordinates of an “Exterior Wall” location. It is also easy to add up other surfaces according to the schema defined in Figure 2. In addition, every polyloop, which contains a list of coordinates that makes up a polygon in three-dimensional space, follows a right-hand rule defining the outward normal of a surface.

```
<gbXML>
        <Campus>
  <Surface id="su1" surfaceType="ExteriorWall">
<AdjacentSpaceId spaceIdRef="sp1" />
<RectangularGeometry>
        <Azimuth>0</Azimuth>
<CartesianPoint>
        <Coordinate>2224</Coordinate>
        <Coordinate>1529.75</Coordinate>
        <Coordinate>4</Coordinate>
        </CartesianPoint>           <Tilt>90</Tilt>

<Height>96</Height>
<Width>42.81</Width>
  </RectangularGeometry>
<PlanarGeometry>
 <PolyLoop>
    <CartesianPoint>
        <Coordinate>2224</Coordinate>
        <Coordinate>1529.75</Coordinate>
    <Coordinate>4</Coordinate>
        </CartesianPoint>
   </PolyLoop>
    </PlanarGeometry>
    </Surface>
</Campus>
</gbXML>
```

Sensor information

IFC representation

Figure 3 shows the UML diagram of the IFC schema representations of sensors and sensed data using temperature and CO2 parameters as examples. An *IfcSensorType* defines a particular type of sensor which is used for detection in a control system such as a building automation control. Parameters of the sensors are specified through property sets that are enumerated in the *IfcSensorTypeEnum* data type. *IfcSensorType* has an attribute called PreDefinedType, which is an object of *IfcSensorTypeEnum*. *IfcSensorTypeEnum* defines the range of different types of sensor that can be specified. A temperature sensor and CO2 sensor type are instances of *IfcSensorTypeEnum*. The usage of *IfcSensorType* defines the parameters for one or more occurrences of *IfcDistributionControlElementType*, which inherited from the *IfcDistributionElementType*. This is how sensors are connected with the equipment.

Here is an example of how the sensor values are represented. *IfcPropertySingleValue* is one of the subtypes of *IfcSimpleProperty*. It has two attributes: NorminalValue and Unit. The NorminalValue is an object of *IfcValue*, while Unit is an *IfcUnit* object, where “parts per million (ppm)” is stored. *IfcValue* is a SELECT type for selecting between more specialized types *IfcSimpleValue*, *IfcMeasureValue* and *IfcDerivedMeasureValue*. Attributes of *IfcMeasureValue* include *IfcThermodynamicTemperatureMeasure* and *IfcRatioMeasure* which are REAL types. *IfcThermodynamicTemperatureMeasure* is where the sensed temperature value is stored, while *IfcRatioMeasure* is where the sensed CO2 value is stored (IAI, 2006).

gbXML representation

As shown in Figure 2, gbXML has an element named Meter which is a description of a resource measurement. Any measured data can be defined in this element. It has children elements: Name, Description, UtilityRate and others. “Name” defines a name of sensor, while “other” could contain the value and unit of this sensor and it has any occurrences. Using the same temperature and CO2 parameters, the generated XML file could be as follows:

```
<gbXML>
        <Meter>
                <Name>TempSensor1</Name>
                <Description>Temperature Sensor
_1</Description>
                <Value>23</Value>
```

```
			<Unit>Celsius</Unit>
		</Meter>
		<Meter>
			<Name>CO2Sensor1</Name>
			<Description>Co2 Sensor
No.1</Description>
			<Value>600</Value>
			<Unit>PPM</Unit>
		</Meter>
</gbXML>
```

There are only three levels from root element “gbXML” to data element “value”. In addition, there is no need to contain “UtilityRate” element in “Meter” element if there was no UtilityRate information. Furthermore, a simple XML stylesheet can extract the information from the above XML file. Hence, to store and obtain the sensor data using gbXML is relatively simple and straightforward.

Comparison and Discussion

Firstly, IFC adopts a comprehensive and generic approach to represent an entire building project. From IAI online information website, it shows a holistic approach of IFC representation, covering domains from building construction to building operation. IFC representation was also extended in the building commission domain and implemented in several cases studies (Akin, 2004). The application of gbXML deployed by Green Building Studio Inc. is currently only on the energy simulation domain. Users can upload a well-formatted gbXML file to get a quick summary of simulation results from DOE-2 based on contextually related assumptions. It is also possible to get input files for EnergyPlus and EQuest from GBS afer the simulation runs. However, as it is shown in Figure 2, gbXML has the ability to carry building environmental sensing information. An effort to extend the gbXML schema to conduct lighting simulation is discussed later. In terms of geometry, the generic approach of IFC has the ability to represent any shape of building geometry, while gbXML only acceptes rectagular shape, which is enough for energy simulation.

Secondly, IFC uses a “top-down” and relational approach, which yields in a relative complex data representation schema and a large data file size. gbXML adopts a “bottom-up” approach, which is flexible, open source, and a relatively straight-forward data schema. The “top-down” approach can trace back all the semantic changes when one value of the element in the schema changed. Ideally, it has the ability to maintain semantic integrity automatically. Howerver, it is very complex to program and be implemented in software. The “bottom-up” approach has less layer of complexity. This approach has proven successful in offering web-based simulation service (notably Green Building Studio) for the industry.

PROPOSED LIGHTING SCHEMA AND ITS APPLICATION

Proposed lighting schema

Based on the analysis of pros and cons of IFC and gbXML, the latter is relatively simple to develop building application prototypes without understanding all the elements in the whole schema. Hence, gbXML is chosen for extension to support lighting simulation using Radiance (Radiance, 2007). By extending the common data schema to support Radiance, information can be shared across the concurrent engineering design framework with enegy and lighting in this instance. This will reduce the time and effort in capturing project related information such as operation schedules and product specifications. The most significant advantage will be the acquisition of building geometry and material properties, currently a time-consuming and error prone activity that is duplicated in every simulation of the same building in different domains. Since the geometry semantics between the energy and lighting domains are very similar, the interoperability afforded by using a common data schema will potentially and drastically alleviate the difficulties under current situation; the geometry model only has to be defined once in either domain and only slight modifications are necessary in the other. The approach builds upon the concepts of shared object model (SOM) and domain object model (DOM) from Madavi(2002).

This extension will also yield other advantages such as reduce the effort in ensuring that a common set of assumptions and specifications are adopted across the engineering team during concurrent activities. Currently, assumptions and specifications in the mutually exclusive activities have to be coordinated manually or run the risk of being incorrect and requiring extraneous efforts in correcting the inconsistencies. A global information model built upon the proposed common data schema will allow the coordination to be automated and transparent to a large extent. The use of a common protocol will also allow generalized transfer of data between different software across the concurrent engineering design framework.

The proposed schema is shown in Figure 4. The geometry information represented by “Surface” is considered as SOM, which will be inherited from original gbXML. Other information such as “LGTSkyInfor”, “LGTMat” and “LGTFunction” are under the extended “Light” element which is a typical DOM application. “Light” is in the same level of “Campus”, where all individual SOM and DOM element start from. In dealing with material representation, the current gbXML schema has no explicit way to show how material is attached to the surface. Hence, a “LGTMatID” element is added under “Surface” element or “Opening” either as shown in Figure 2. This id is a reference of the lighting material of the specified surface. In addition, “Lamp” which defines the lighting source is added under “Campus” as shown in Figure 2. “Lamp” also has geometry information which is defined under “Surface”. Currently, this schema is implemented with a java-based interface for Radiance simulation.

Software Application

The main objective in developing a lighting simulation application with extended gbXML schema is to reduce the overall effort required to perform lighting simulations and analyses. This is achieved by (1) providing a friendly-user interface, automating data import and processing where information is available but in unsuitable formats, (2) externalizing tacit knowledge by use of data libraries with defaults and preferences, and (3) design identifying and providing features that meet overall goals more efficiently.

The design of the interface reinforces the process flow of conducting simulations by the use of hierarchical tabs, guiding the user through sequential steps though an expert user can still navigate freely between different stages (Figure 5). To reduce the turn-over time in filling all the variables necessary for lighting simulations, the application populates the BIM with default values where necessary so that the model is immediately ready for simulation. In principle, the default values will be based on typical conditions taking into consideration the building type and location (similar in concept with the GBS energy simulation web services).

The application also maintains libraries for material properties and sky information. The libraries are kept separately in an XML format consistent with the extended gbXML schema. This allows the libraries to be shared among the different domain applications within the IBECS framework, following the principles of information sharing and consistency.

The current version of the application generates a series of Radiance input files as well as a batch file that launches Radiance to process the input files. A Radiance Picture File Format (.pic) image is produced at the end of the simulation, which contains luminance values of surfaces as seen from the defined viewpoint. The option to generate .pic images containing illuminance values can also be provided now.

CONCLUSION

A basic comparison between IFC and gbXML is conducted in terms of their data structure, representation and application. With a comprehensive “top-down” data schema, IFC shows potential benenifts in its highly organized and relational data representation. In contrast, the “buttom-up” gbXML schema is simpler and easier to understand which facilitates quicker implementation of schema extension for different design purposes. The proposed lighting schema and its applications demonstrate the benefits of gbXML. It reduces both time and effort for model preparation to perform a lighting simulation. In addition, it reuses information from energy simulation and shares libraries with other application. The future research in the schema development will focus on identifying and resolving information redundancy in the current gbXML schema. A complete sensor schema will also be developed for building controls in the future.

ACKNOWLEDGMENT

The authors acknowledge the support from United Technology Research Center (UTRC) for this research.

REFERENCES

Akin, O., Turkaslan-Bulbul, T., Gursel, I., Garrett, J., Akinci, B., Wang, H. (2004) “Embedded Commissioning for Building Design.” ECPPM (European Conference on Product and Process Modeling in the Building Industry), 09/08-10/2004, Istanbul, Turkey.

Augenbroe,G. 1992. Integrated building performance evaluation in early design stages. *Building and Environment* 27(2) pp.149-161.

Bazjanac, V., Forester, J., Haves, P., Sucic, D., and Xu, P. 2002. HVAC componenet data modeling using industry foundation classes. *5th International Conference on System Simulation in Buildings*, Liege, Belgium.

Bazjanac, V. 2001. Acquisition of building geometry in the simulation of energy performance. *Proceeding of IBPSA 2001*. Rio de Janeiro, Brazil.

Bazjanac, V. & T. Maile. 2004. IFC HVAC Interface to EnergyPlus - A Case of Expanded Interoperability for Energy Simulation. Proceedings of the SimBuild 2004 Conference in Boulder, CO, 4-6 August 2004.

Bedell, J. and Kohler, N. 1993. CAAD Future, pp 423-435.

BLIS, http://www.blis-project.org/index2.html. last reviewed, 02/17/2007

DOE-2, http://doe2.com/, last reviewed, 02/14/2007

e-QUEST, http://doe2.com/equest/, ;ast reviewd, 02/14/2007.

gbXML. http://www.gbXML.org/. last reviewed, 02/17/2007.

Gauchel, J., Hovestadt, S., Van Wyk, and Bhat, R. 1992. Building modeling based on concepts of autonomy. Proceedings of AID 1992, Pittsburgh, PA.

Harfmann, A., and Majkowski, B. 1993. A componenet-based approach to building product representation and design development. CAAD Futures, pp 437-452.

International Alliance for Interoperability (IAI). 2006. http://www.iai-international.org. last reviewed: 11/22/2006.

ISO 1992. Product data representation and exchange – part 1; Overview and Fundamental Principles. STEP Document ISO TC184/SC4/PMAG.

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.

Kimber, W.E. 1999. XML Representation Methods for EXPRESS-Driven Data. NIST GCR 99-781. National Institute of Standards and Technology.

Lam K. P., et. al. (2004) SEMPER-II: an internet-based multi-domain building performance simulation environment for early design support. Automation in Construction Vol. 13, Issue 15, pp. 651-663.

Lam K.P., et. al. (2006) Integrated Concurrent Design of High Efficient Commercial Buildings Report #3. Report to Institute of Standards and Technology (NIST) Cooperative Agreement No. 70NANB4H3024 Technical Services Agreement No. 1119346. April 29, 2006. Pittsburgh, PA: Center for Building Performance and Diagnostics, Carnegie Mellon University.

Mahdavi, A.2000. Supporting collaborative design via integrated building performance computing. *12 International Conference on Systems Research*, Informatics and Cybernetics, Baden-Baden, Germany.

Pohi, J., and Rep, I. 1988. An integrated intelligent CAD environment. *Proceedings, 4th International Conference on Systems Research*, Informatics and Cybernetics, Baden-Baden, Germany.

Robert, H. 2002. Software interoperability for energy simulation. *LBNL Report: 51244.* LBNL

Radiance. http://radsite.lbl.gov/radiance/, last reviewd, 01/12/2007.

Rutherford, J. (1993). KNODES: knowledged-based design decision support. *CADD Future 1993*, pp 257-374.

SABLE, http://www.blis-project.org/~sable/, last reviewed, 02/17/2007.

Sanvido, V. 1992. Linking levels of abstraction of a building design. Building and Environment, 27(2), pp195-208.

W3C. 2006. The World Wide Web Consortium. http://www.w3.org/. last reviewed: 11/17/2006.

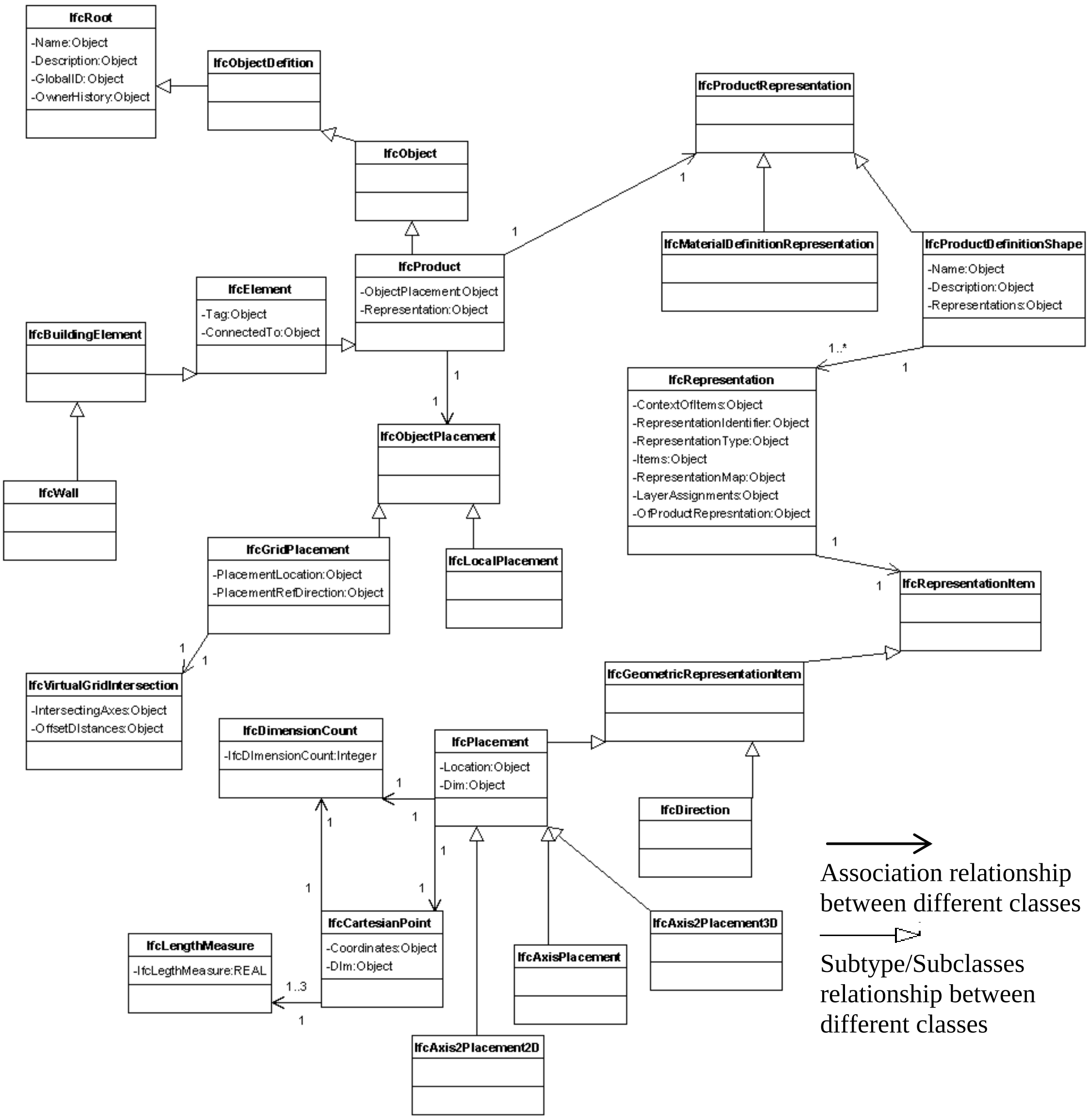

Figure 1 UML diagram of the relationship between wall and its dimension in IFC

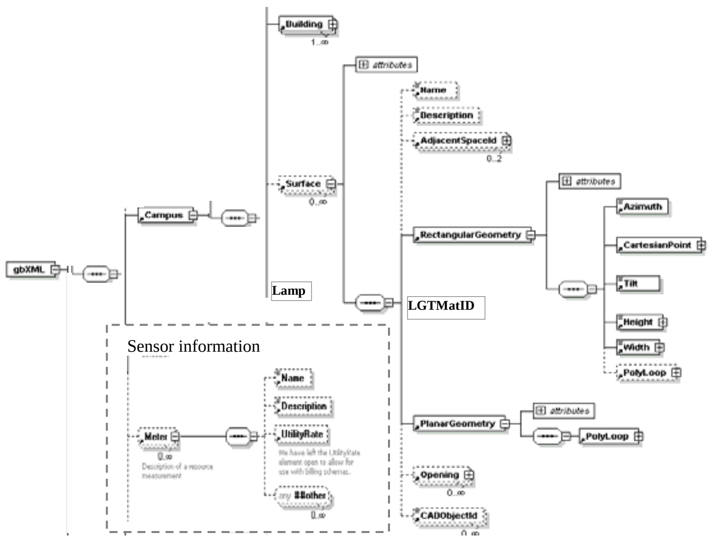

Figure 2 Geometry and sensor representations in gbXML

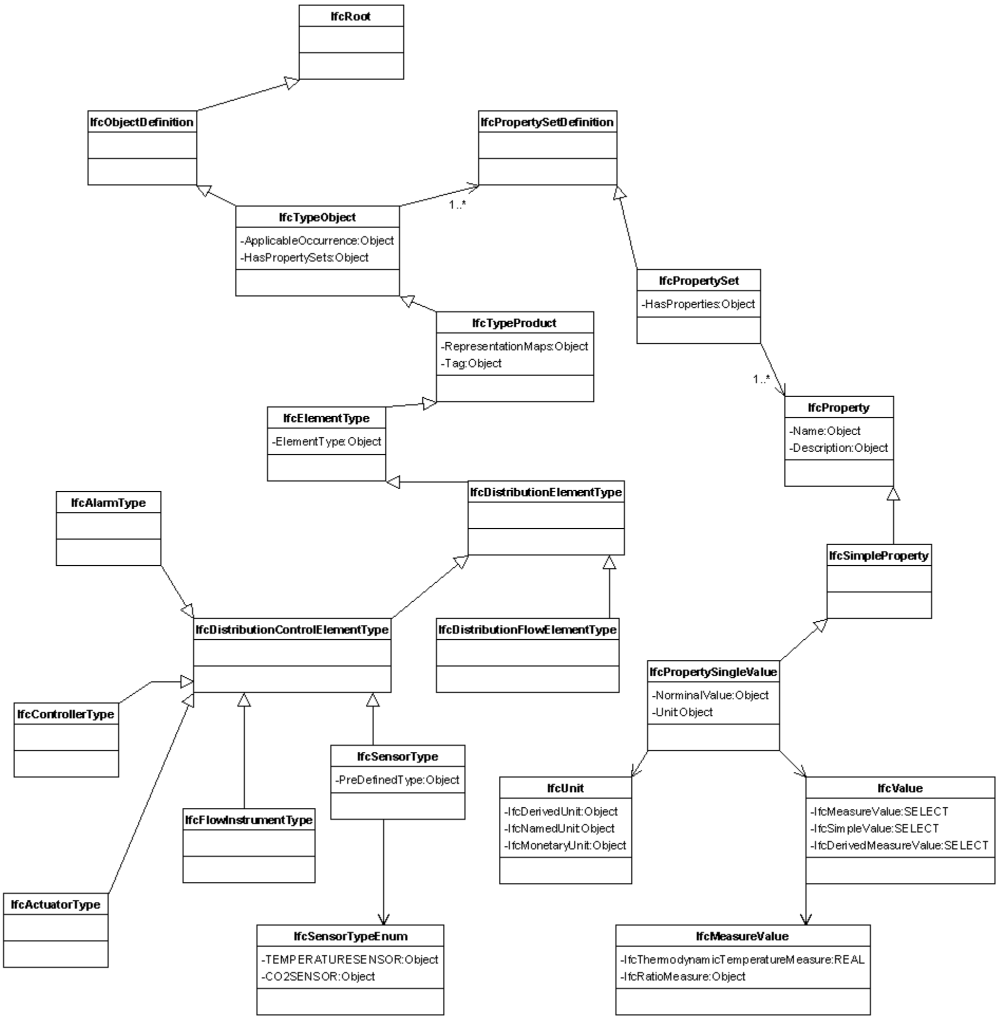

Figure 3 UML diagram of the relationship between sensor and sensed data in IFC

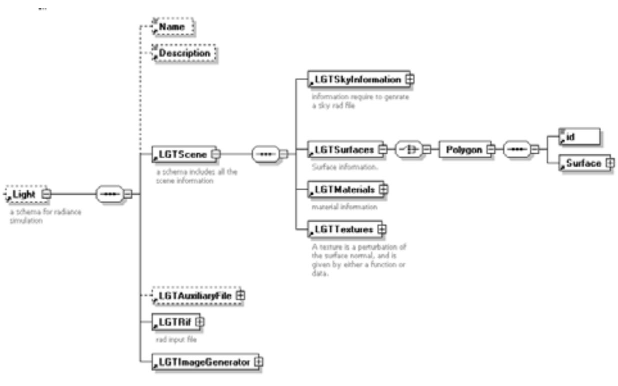

Figure 4 Proposed schema for lighting simulation (Lam et al.,2006)

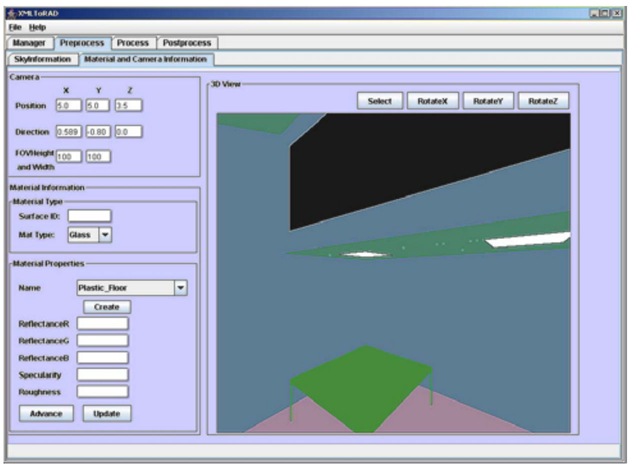

Figure 5. Lighting application GUI tabbed design (Lam et al.2006)