Parsing Building Data from Industry Foundation Classes through IFCModelParser

First A. Raninder Kaur Dhillon, Second B. Mayur Jethwa, Third C. Hardeep Singh Rai

Abstract— Information technology has made tremendous progress across disparate disciplines, one of which is AEC (Architecture, Engineering and Construction) industry. Enormous amount of data is generated throughout a project life cycle, but exchanging this data among participants is inconsistent. So major concern in AEC is interoperability for exchanging information throughout the project lifecycle among various disciplines and technical applications. in which Industry Foundation Classes has played a critical role. IFC define object oriented data model of buildings shared by all IFC-compliant applications. This paper proposes a way of extracting geometric data from IFC file created by any Building Information Modeling Software.

Keywords— AEC, Building Information Modelling, CAD, IFC

I. Introduction

Architecture, engineering, construction, and facilities management (AEC/FM) are information intensive industries, and are increasingly dependent upon effective information technologies (IT). Various computer tools are used to support almost all AEC/FM design and management tasks. Interoperability is one of the major research and development task in information technology for the architecture, engineering, construction, and facilities management industries[1]. Industry Foundation Classes (IFC) have made substantial progress in recent years, and many design software companies like Archicad, Revit, Tekla now provide export capabilities of IFC-based product models. The Industry Foundation Classes (IFC), an open standard data model developed by the International Alliance for Interoperability (IAI), which was formed in 1995 by an international consortium of organisations in the AEC/FM industry. The IFC data model allows the building geometry and materials property information to be exported from a BIM authoring tool to a standard format such as the IFC compliant STEP (Standard for Exchange of Product Model Data) physical data file [2]. The technology for exchanging information using Industry Foundation Classes has now been established, but many areas require additional development before comprehensive interoperability solutions are reached.

II. Industry Foundation Classes

It is a platform neutral, open file format specification that is not controlled by a single vendor or group of vendors. It is an object-based file format with a data model developed by BuildingSMART (formerly the International Alliance for Interoperability, IAI) to facilitate interoperability in the architecture, engineering and construction (AEC) industry, and is a commonly used collaboration format in Building information modeling (BIM) based projects [3]. The IFC model specification is open and available. It is registered by ISO and is an official International Standard ISO 16739:2013.

Because of its focus on ease of interoperability between software platforms, the Danish government has made the use of IFC format(s) compulsory for publicly aided building projects. Also Finnish state owned Facility Management Company Senate Properties demands use of IFC compatible software and BIM in all their projects. the IFC specification is now developed and maintained by BuildingSMART.

A. *Technologies used in IFC development*

The IFC schema is specified using the EXPRESS data definition language, as defined in ISO10303-11:1994. IFC data files are clear text files following the STEP physical file format, as defined in ISO10303-21:1994. IFC files following the STEP physical file format are governed by the IFC schema in EXPRESS.

B. *Structure of the STEP physical file*

The ".ifc" file is a STEP physical file, SPF, and thereby a structured ASCII text file. The basic SPF structure divides each file into a header and a data section. The header section has information about:

- the IFC version used
- the application that exported the file
- the date and time when the export was done
- (often optionally) the name, company and authorizing person of the file

Example

```
HEADER;
FILE_DESCRIPTION(('IFC 2x platform'),'2;1');
FILE_NAME( 'Example.dwg', '2005-09-02T14:48:42', ('The User'), ('The Company'), 'The name and version of the IFC preprocessor', 'The name of the originating software system', 'The name of the authorizing person');
FILE_SCHEMA(('IFC2X2_FINAL'));
ENDSEC;
```

The data section contains all instances for the entities of the IFC specification. These instances have a unique (within the scope of a file) STEP Id, the entity type name and a list of explicit attribute.

Example

```
DATA;
#7=IFCCARTESIANPOINT((0.,0.,0.));
#8=IFCDIRECTION((0.,0.,1.));
#9=IFCDIRECTION((1.,0.,0.));
#10=IFCAXIS2PLACEMENT3D(#7,#8,#9);
ENDSEC;
```

C. Information contained in an IFC model

- Building hierarchy (Project, Site, Building, Story, Element)
- Element type (wall, slab, column, beam, roof, stair, zone etc.)
- Geometry
- Layer-system
- Standard and custom (application-dependent) IFC properties (material, color, cross-sections, fire rating etc.)
- Connections

D. IFC/ifcXML Specifications[4]

- IFC4 (March 2013)
- ifcXML2x3 (June 2007)
- IFC2x3 (February 2006)
- ifcXML2 for IFC2x2 add1 (RC2)
- IFC2x2 Addendum 1 (July 2004)
- ifcXML2 for IFC2x2 (RC1)
- IFC 2x2
- IFC 2x Addendum 1
- ifcXML1 for IFC2x and IFC2x Addendum 1
- IFC 2x
- IFC 2.0
- IFC 1.5.1
- IFC 1.5

E. Architecture of the IFC Model

The IFC model represents not just tangible building components such as walls, doors, beams, ceilings, furniture, etc., but also more abstract concepts such as schedules, activities, spaces, organization, construction costs, etc. in the form of entities [5]. All entities can have a number of properties such as name, geometry, materials, finishes, relationships, and so on. The latest release of the IFC has a total of 623 entity definitions i.e., 623 different kinds of components or concepts.

The IFC Model Architecture for IFC 2x as shown in Fig. 1 consists of the following layers [2].

- Resource Layer
- Core Layer

-Kernel

-Extensions

- Interoperability Layer
- Domain Layer

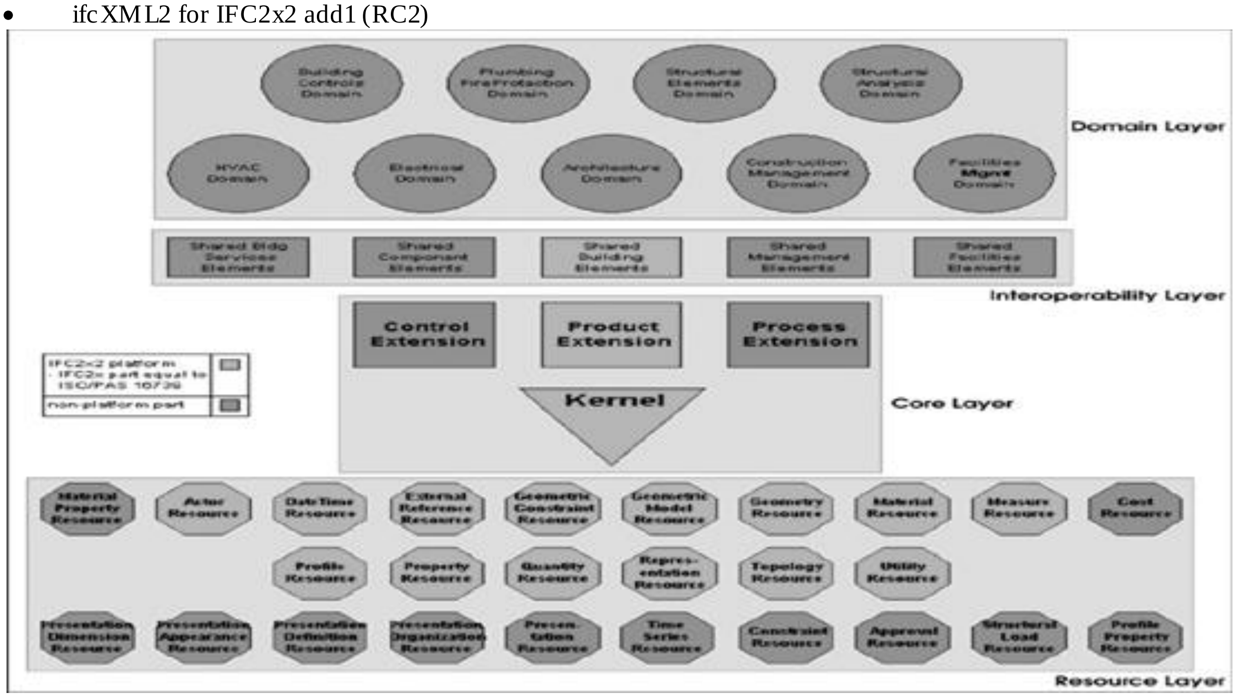

Fig. 1 IFC layered Architecture (Source: IFC2x2 Final Online Documentation)

1. Resource Layer: This layer contains the fundamental concepts expressed as entity types such as geometry (point, line and curve) topology (vertex, edge, face and shell), geometric model (CSG, B-Rep, Geometric Set). The elements of this layer can be referenced by elements of all other layers.

2. Core Layer: The Core layer provides the basic structure of the IFC object model and defines most general concepts that will be specialized by higher layers of the IFC object model.

The Core includes two levels of generalization:
1. The Kernel
2. Core Extensions

The Kernel provides all the basic concepts required for IFC models within the scope of the current IFC Release. It also determines the model structure and decomposition. Its constructs are very general and are not AEC/FM specific, Core Extensions, provide extension or specialization of concepts defined in the Kernel.

Each Core Extension is a specialization of classes defined in the Kernel and develops further specialization of classes rooted in the IfcKernel. Additionally, primary relationships and roles are also defined within the Core Extensions.

3. Interoperability Layer: This layer defines basic concepts for interoperability between different domain extensions. Shared building elements like beam, door, roof, window or ramp are defined in this layer.

4. Domain Layer: Domain Models provide further model detail within the scope requirements for an AEC/FM domain process or a type of application. Examples of Domain Models are Architecture, HVAC, FM, Structural Engineering etc.

F. IFC Limitations

While *Industry Foundation Classess (IFC)* and *Revit* provide rich information about a building and its components, some important construction-specific information is either not explicitly represented or missing [5]. For instance, the designations of a 'full height wall' or 'ceiling height wall', and 'wall shape' are not explicitly defined. *IFC* largely leaves the definition of component types (e.g., wall type) completely to the modeling application, and to other model extension software, through type definitions (*IfcDefinesByType*) and through the definition of properties.

III. PROPOSED WORK: IFCModelParser

There are various commercial as well as open source tools available for working with IFC data like IFC-SDK, Ifc-OpenShell, XBIM toolkit, Ifc-dotnet, IfcPlusPlus etc. but they do not provide all the information in detail like that which is needed for building code compliance checking. Compliance Checking involves comparison of known and derived information about a design with a set of constraints obtained from a building regulation such as IBC(International Building Code) or NBC(National Building code) of any country.

IFC Model Parser is GUI based application to parse IFC Models and to export them to Excel SpreadSheets and Text File [17] as shown in Fig. 2. The results can also be displayed also on Console Application if required. The Project is capable of determining the minute details of the model. And can create the Excel sheet on the fly. The test has showed that the project can handle creation of hundreds of worksheets per model with no problem. The project has two variations:

1) IFC 2x4 which can parse .ifc files of latest version of ifc and
2) IFC 2x3 parser

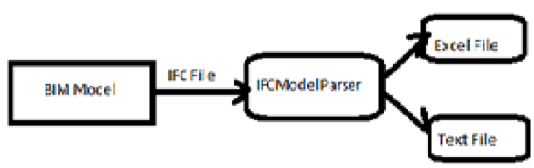

Fig. 2 Framework of IFCModel parser

A. Features of IFCModelParser [16]

- Dynamic EXPRESS parser: On -the-fly parsing of any Industry Foundation Classes (IFC) schema (IFC2x3, IFC4 are supported by default).
- Saves to Text file and Excel : The output of parsed model is stored in Excel SpreadSheet and Text file. And can write upto hundreds of page of worksheet in few minutes..
- Console Output : Have the capability to output the result in the console.
- Hyperlinks: For navigation with the excel files.

B. Class structure of IFCModel parser

Various classes created during the project are shown using UML class diagrams in Fig. 3.

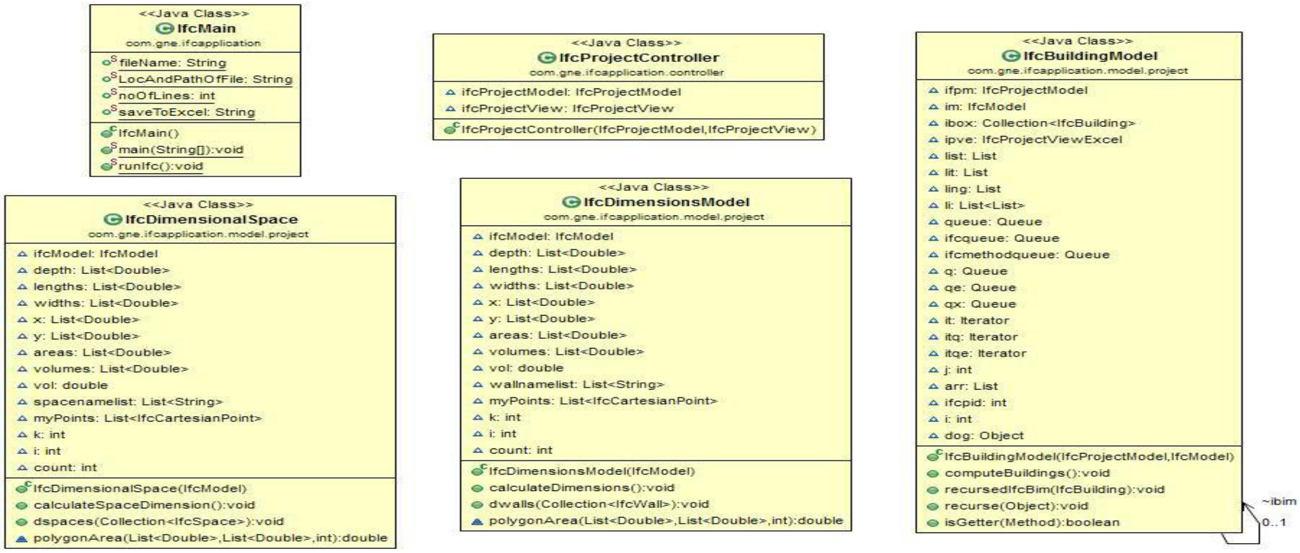

Fig. 3 class structure

IV. Results And Discussion

All the coding is done in Java eclipse. Input to the parser is IFC file. Various parts of the project are:

Packages

com.gne.ifcapplication
com.gne.ifcapplication.controller
com.gne.ifcapplication.model.project
com.gne.ifcapplication.view
com.gne.ifcapplication.viewexcel

All Classes

FileChooser, IfcBuildingModel IfcDimensionalSpace IfcDimensionalViewExcel, IfcDimensionsModel IfcDoorModel, IfcFileFilter, IfcFileProgres IfcMain IfcProjectController, IfcProjectModel, IfcProjectView IfcProjectViewExcel, IfcRoofModel IfcSiteModel IfcSlabModel IfcSpaceModel, IfcStoreyModel IfcWallMaterial. IfcWallModel ifcwalls, IfcWindowModel, Inverses PrintToFile, saveFiles, savefilter SpaceDimensionsExcel, Utils

Excel file generated from IFCModelParser contains all the information contained in building elements,even which is not explicitly specified.

e.g. If an ifc file doesn't contain information about wall length, height, area etc. then this parser computes it from co-ordinates and display it.

Part of ifc file and its view are shown in Fig.4:

```
#45 = IFCWALL('3Ep4r0uuX5ywPYOUG2H2A4', #2, 'Wall
xyz', 'Description of Wall', $, #46, #51, $);
#46 = IFCLOCALPLACEMENT(#36, #47);
#47 = IFCAXIS2PLACEMENT3D(#48, #49, #50);
#48 = IFCCARTESIANPOINT((0., 0., 0.));
#49 = IFCDIRECTION((0., 0., 1.));
#50 = IFCDIRECTION((1., 0., 0.));
#51 = IFCPRODUCTDEFINITIONSHAPE($, $, (#52, #81));
#52 = IFCSHAPEREPRESENTATION(#20, 'Body', 'Brep',
(#80));
#53 = IFCCLOSEDSHELL((#60, #67, #70, #73, #76, #79));
#54 = IFCPOLYLOOP((#55, #56, #57, #58));
#55 = IFCCARTESIANPOINT((0., 0., 0.));
#56 = IFCCARTESIANPOINT((0., 200., 0.));
#57 = IFCCARTESIANPOINT((5400., 200., 0.));
#58 = IFCCARTESIANPOINT((5400., 0., 0.));
```

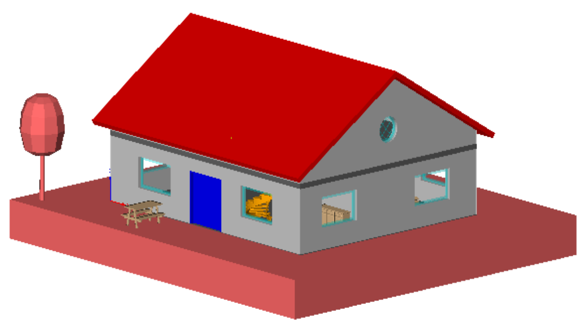

Fig 4 View of the ifc file in IFCViewer

Fig. 5 and Fig. 6 represents the excel files generated from this ifc file. Fig. 5 contains hyperlinks which contain links to different parts of the ifc entities e.g. when clicked on link units, it describes the units (SI) used for different entities like wall length, area etc.

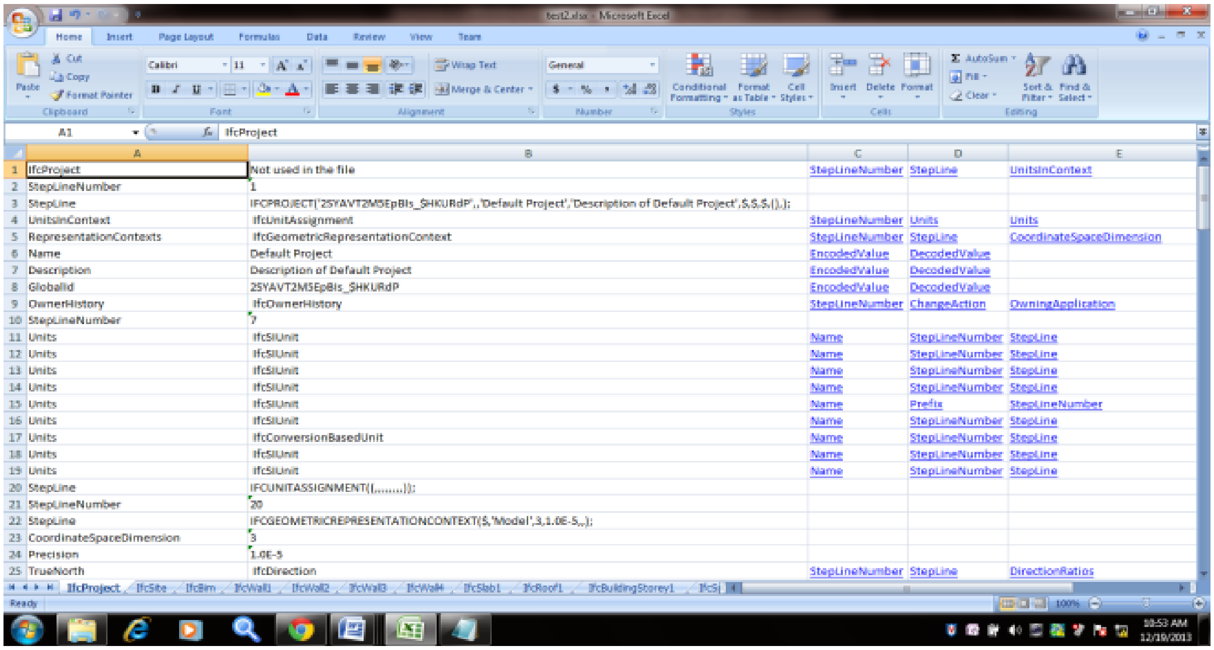

Fig. 5 IFC file in Excel format

Fig. 6 shows the values of parameters which are not provided explicitly like wall height, wall width, area etc. They are deduced from above ifc file with IFCModelParser.

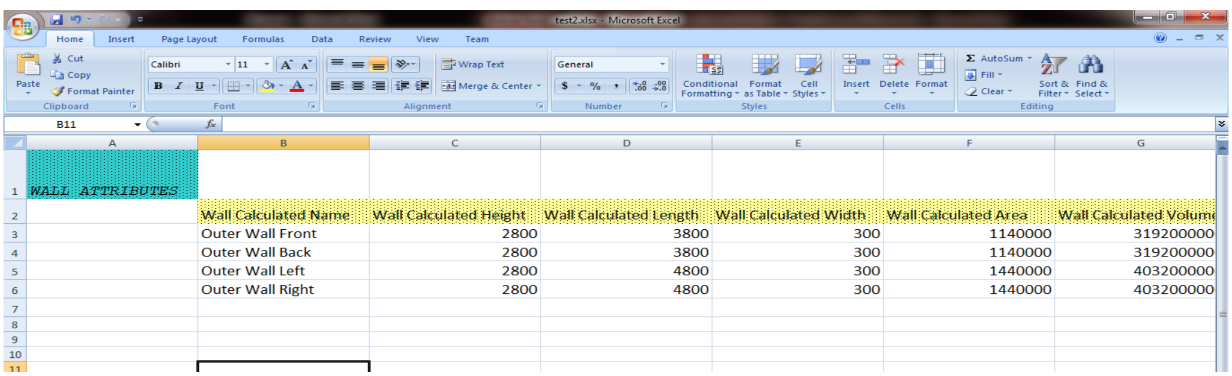

Fig. 6 Excel file from IFCModelparser displaying geometric quantities of wall

IV. Conclusion And Future Work

This research work aims to parse the Industry Foundation Class (IFC) generated by the Building Information Modelling software. It aims to smooth the process of sharing and getting information from the building models. It supports all IFC versions. It makes it ideal for FDS fire simulation, automated building code checking, energy performance evaluation applications. Future work is enhancing this poject by adding more data like deducing relationships such as door contained in a wall and using this data for specific application like checking regulations.

References

[1] Thomas Froese, "Future Directions for IFC-based Interoperability", *ITcon* Vol. 8 (2003); pg. 231. http://www.itcon.org/2003/17

[2] ISO 10303-21, "Industrial Automation System – Exchange of Product Model Data – Part 21: Implementation Methods; Clear Text Encoding of the Exchange Structure," *International Organization for Standardization*, 2002.

[3] T. Liebich, Y. Adachi, J. Forester, J. Hyvarinen, K.Karstila, and J. Wix, "Industry Foundation Classes IFC2x3", *International Alliance for Interoperability*,2006.

[4] www.wikipedia.com

[5] Khemlani Lachmi, "IFC: Look Under the Hood", *Article AECBytes*, 2004.

[6] M. P. Nepal, "Automated Extraction and Querying of Construction-Specific Design Features from a Building Information Model," *PhD Thesis*, The University of British Columbia, Vancouver, Canada, 2011.
[7] R. Vanlande, C. Nicolle and C. Cruz, "IFC and building lifecycle management", *Automation in Construction*, Vol. 18(1), pp. 70-78, 2008.
[8] ISO 10303-21, "Industrial Automation System – Exchange of Product Model Data – Part 21:Implementation Methods; Clear Text Encoding of the Exchange Structure," *International Organization for Standardization*, 2002.
[9] C. Clemen and L. Gründig , "The Industry Foundation Classes – Ready for Indoor Cadastre?",*Shaping the ChangeXXIII FIG Congress*, Munich, Germany, October 8-13, 2006
[10] T. Liebich ," IFC 2x Edition 3 Model Implementation Guide" ,Version 2.0 May 18, 2009.
[11] CHAPTER NINE, Industry Foundation Classes, BuildingSMART chapters - http://buildingsmart.com/organization/chapters
[12] G. Brewer, T. Gajendran, R. Le Goff, "Building Information Modelling (BIM): an Introduction and InternationalPerspectives", *Research Report* ,July 2012
[13] Robert J. Hitchcock , Justin Wong ,"Transforming IFC Architectural View BIMS for Energy Simulation: 2011" *Proceedings Building Simulation Conference*, Sydney, Australia,2011.
[14] J. Dimyadi., M. Spearpoint, R. Amor, "Sharing Building Information using the IFC Data Model for FDS", *Proceedings of 9th IAFSS Symposium on Fire Safety Science*, Karlsruhe, Germany, 21-26 September,2008.
[15] Q. Z. Yang, X. Xu, "Design knowledge Modelling and software implementation for building code compliance checking", *Building and Environment*, 39(6), 689-698. (2004). doi:10.1016/j.buildenv.2003.12.004
[16] D. Greenwood, D, S. Lockley, S. Malsane, J. Matthews, "Automated compliance checking using building information models", *COBRA* 2010 (pp. 266-274).
[17] Raninder Kaur Dhillon, H. S. Rai, "Code Compliance with Building Information Modelling", *International Conference on Recent trends in Information, Communicaton and Technology*,2013.