World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:8, No:6, 2014

A Novel Algorithm for Parsing IFC Models

Raninder Kaur Dhillon, Mayur Jethwa, Hardeep Singh Rai

Open Science Index, Computer and Information Engineering Vol:8, No:6, 2014 publications.waset.org/9998462/pdf

***Abstract*—**Information technology has made a pivotal progress across disparate disciplines, one of which is AEC (Architecture, Engineering and Construction) industry. CAD is a form of computer-aided building modulation that architects, engineers and contractors use to create and view two- and three-dimensional models. The AEC industry also uses building information modeling (BIM), a newer computerized modeling system that can create four-dimensional models; this software can greatly increase productivity in the AEC industry. BIM models generate open source IFC (Industry Foundation Classes) files which aim for interoperability for exchanging information throughout the project lifecycle among various disciplines. The methods developed in previous studies require either an IFC schema or MVD and software applications, such as an IFC model server or a Building Information Modeling (BIM) authoring tool, to extract a partial or complete IFC instance model. This paper proposes an efficient algorithm for extracting a partial and total model from an Industry Foundation Classes (IFC) instance model without an IFC schema or a complete IFC model view definition (MVD).

***Keywords*—** BIM, CAD, IFC, MVD.

I. Introduction

BUILDING Information Modeling is now being increasingly used as an emerging technology to assist in creating, designing, constructing and operating the buildings in many countries. In basic terms 'Building Information Model' refers to a digital model (as opposed to a physical model) that is capable of holding all information - both 'as-designed' and 'as-built' - relating to a building. BIM technologies refer to a set of hardware and software tools that support the production process of BIM models [1]. Industry Foundation Classes (IFC) [2] represents BIM for sharing construction and facility management data across various applications used in the Architecture, Engineering, Construction and Facility Management (AEC-FM) sector [3]. Additionally, IFC provides a neutral data format which is an ISO global standard and enabling interoperability between systems [2]. IFC captures geometry and properties of 'intelligent' building objects and their relationships within building information models.

This facilitates the coordination of information across incompatible applications, which is a prerequisite for improving building workflows using (BIM) methods. A major problem facing the implementation of IFC in university research projects is the process of parsing STEP files and the instantiation of the IFC Model, which is defined in EXPRESS ISO-10303-P11 [4]. An important objective of the undergoing research work is to extract information from the IFC model which is not explicitly available.

Raninder Kaur Dhillon is Assistant Professor in the Department of Information Technology, Guru Nanak Dev Engineering College, Ludhiana, India (phone: 9815433335; e-mail: rgill@ gndec.ac.in).

Mayur Jethwa is M.Tech. Student in CSE Department, Guru Nanak Dev Engineering College, Ludhiana, India (e-mail: veenajethwa@gmail.com).

Hardeep Singh Rai is Professor in Department of Civil Engineering, Guru Nanak Dev Engineering College, Ludhiana, India (email:hsrai@gmail.com).

II. Industry Foundation Classes

The IFCs are a high-level, object-oriented data model for the AEC/FM industry that model all types of AEC/FM project information such as parts of a building, the geometry and material properties of building products, project costs, schedules, and organizations, etc [5].

The information from almost any type of computer application that works with structured data about AEC building projects can be mapped into IFC data files. The main architecture diagram of the **IFC** model is divided into four separate layers: *Resource Layer, Core Layer, Interoperability*

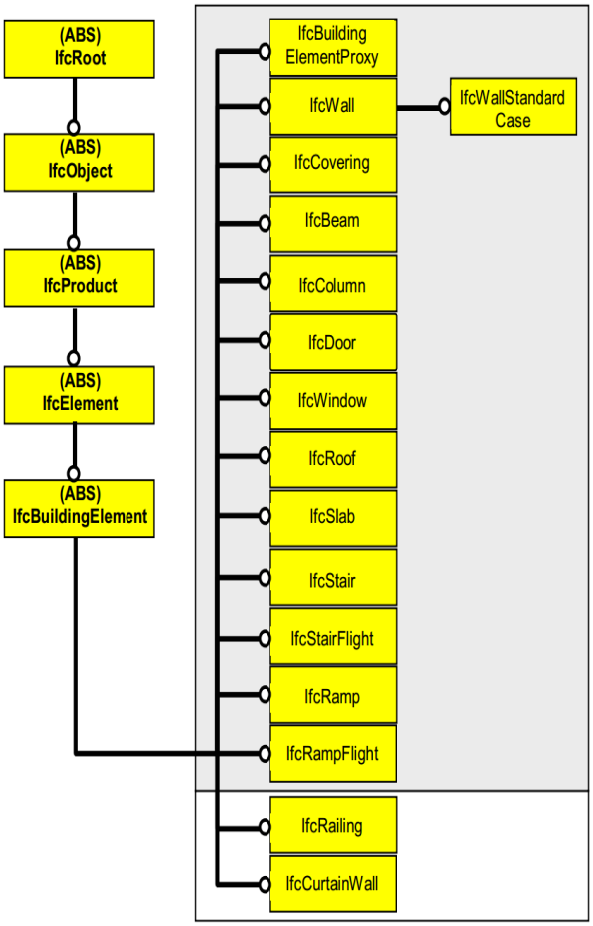


Fig. 1 IFC Hierarchy

Open Science Index, Computer and Information Engineering Vol:8, No:6, 2014 publications.waset.org/9998462/pdf

Layer and domain Layer, representing four different levels. Each layer comprises several diverse categories, and it is within each category or *schema* that the individual entities are defined [6]. Fig. 1 shows the hierarchy of various IFC elements [7].

Moreover each ifc instance contains reference to other instances as shown below:

```
#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));
```

III. Proposed Work

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. The methods developed in previous studies require either an IFC schema or MVD and software applications, such as an IFC model server or a building information modeling (BIM) authoring tool, to extract a partial or complete IFC instance model.

The algorithm proposed in this paper generates a partial and total model by traversing IFC data instances as Graph structure directly from an IFC instance model file, and it relies solely on the internal data structure of an IFC instance model, without an IFC schema or a MVD. The algorithm extracts physical and nonphysical data instances relevant to the user's selection of building elements by iterating through data instances based on the rules specified in the algorithm and claims numerous assertion and also verifies the correctness of the algorithm. For extracting partial model a set of required building elements should be predefined as part of an information delivery manual (IDM) and can be used as input or the algorithm will produce total model. The algorithm was tested by extracting partial models and complete model from 10 IFC 2x3 test cases—initially developed to evaluate the IFC compliancy of software applications in the construction industry—and from an actual BIM project. The integrity of the extracted models was first validated by checking the syntax against an IFC schema and by checking the semantics of partial models using two IFC validators, namely, the Solibri and the Ifc Object Counter.

IV. Algorithm

There are two ways of selecting the model- partial and complete. Thus two algorithms are created respectively as shown below. Fig. 2 shows the flowchart representing selection of different algorithms for extracting model data.

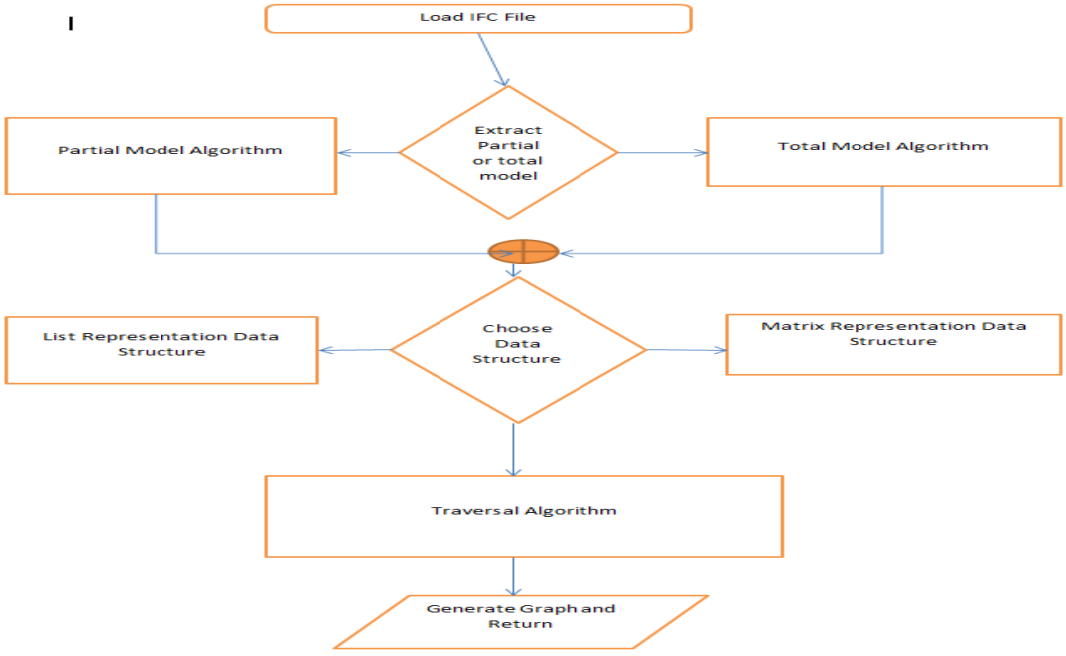


Fig. 2 Flowchart for selecting traversal algorithm

A. Total Model Traversal Algorithm

This algorithm let us select total model from give input file by calling traversal algorithm.

Open Science Index, Computer and Information Engineering Vol:8, No:6, 2014 publications.waset.org/9998462/pdf

Total_Traversal_Model_Algorithm(File(x))
f = Load(File(x))
while(f.hasNextLine())
line := in.nxtLine();
visited[v]:=false;
If(toLowerCase(subStr(line)="ifcrel")) then
Foreach v ∈ line(x), x ∈ {starts_with('#')}
if(visited[v] = false) then
Excel(G(V,E),x)=Traversal_Algo(v,line(x))
visited[v]:= true;
endForeach
end If
end while

B. Total Model Traversal Algorithm

This algorithm let us select partial model from give input file by calling traversal algorithm.

Partial_Traversal_Model_Algorithm(File(x))
f = Load(File(x))
while(f.hasNextLine())
line := in.nxtLine();
visited[v]:=false;
If(toLowerCase(subStr(line)=str∈ {InverseRelations}))
then
Foreach v ∈ line(x), x ∈ {starts_with('#')}
if(visited[v] = false) then
Traversal_Algo(v,line(x))
visited[v]:= true;
endForeach
end If
end while

C. Base Traversal Algorithm

This algorithm is based on BFS (Breadth first Search) which uses queue for storing IFC nodes.

Traversal_algo(s,G)
Initialise (Q=Empty,V=subStr(starts_with('#')));
For each v ∈ V do
σ[v]:= "Unvisited";
π[v] := nil;
d[v]:= ∞;
d[s]:= 0;
end for each
σ[s] := In-q;
Enqueue(Q, s);
while(Q ≠ 0) do
μ := Dequeue(Q);
For each v ∈ Adj[u] do
if(σ[v] = "Unvisited") then
σ[v] := "In-q";
Enqueue(Q, v);
π[v] := u;
d[v]:= d[u]+1;
End if
End for
σ[u] := "Visited";
end while
For ∀ x ∈ V − {s}
Ti = (V, π[x], x)
End for
Return (V,T)

The data structure used for these algorithms is shown in Figs. 3 (a)-(c) below:

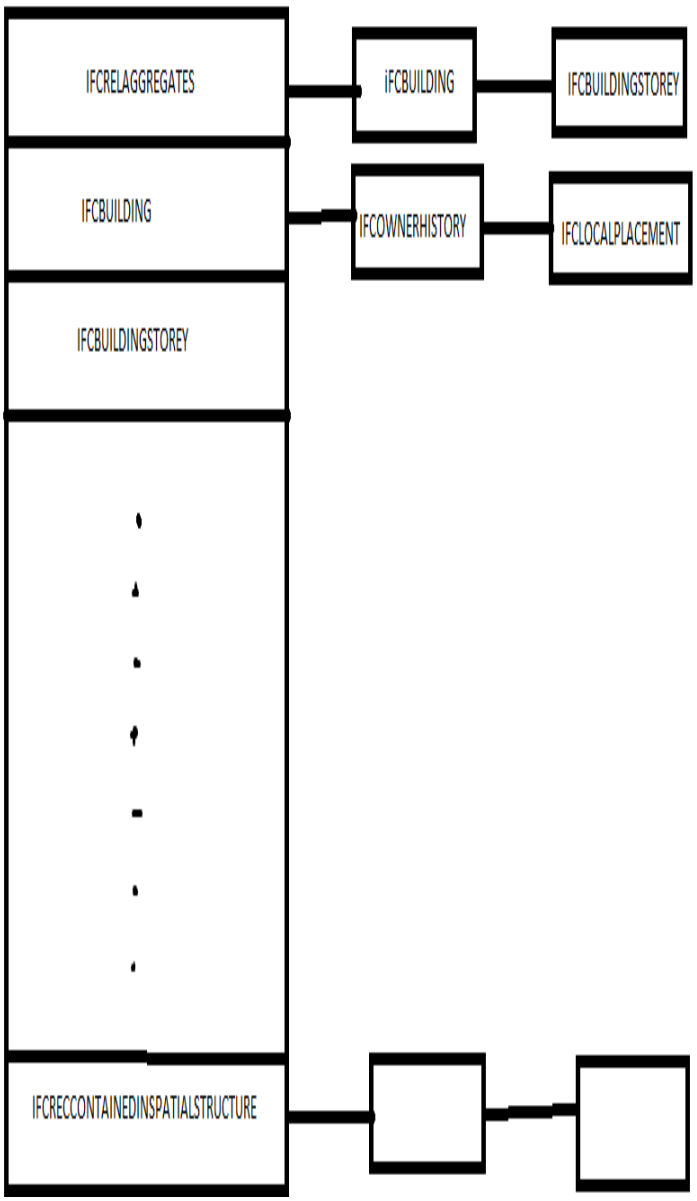


Fig. 3 (a) Data Structure List Representation

Open Science Index, Computer and Information Engineering Vol:8, No:6, 2014 publications.waset.org/9998462/pdf

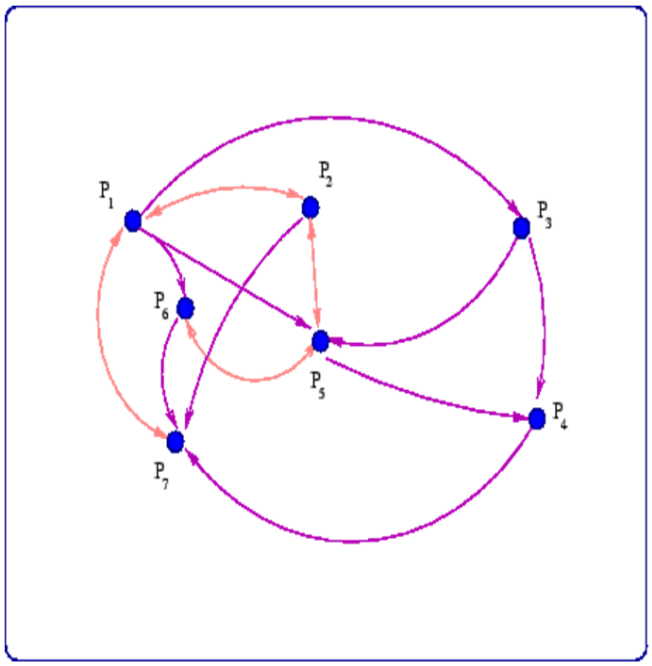


Fig. 3 (b) Graph representation

$$M = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 3 (c) Data Structure Matrix Representation of above graph

V. Results and Discussion

The traversal algorithm take input from the total or partial model algorithm and generates graph with minimum spanning tree and returns it to the caller by iteratively generating children of the element.

Time Complexity: $C_1 n + C_2 \sum \deg(u) \ \forall (u \in V) = O(n+m)$,

where n=n(V), m=n(E)

Space Complexity: O(n+m), where is total element generated and m is the total connections generated by the algorithm. Output of this algorithm generates excel file as shown in Fig. 4. Here excel file contains various hyperlinks which contain link to other IfcEntities [8] e.g. Units contain link to units used for IFc entities like SI for wall length, area etc.

IfcProject	Not used in the file	StepLineNumber	RepresentationCon
StepLineNumber	53		
RepresentationC	IfcGeometricRepresent	StepLineNumber	Precision
UnitsInContext	IfcUnitAssignment	StepLineNumber	Units
StepLine	IFCPROJECT('344O7vICcv		
Name	Project	EncodedValue	DecodedValue
OwnerHistory	IfcOwnerHistory	CreationDate	StepLineNumber
GlobalId	344O7vICcwH8qAEnwJD	EncodedValue	DecodedValue
StepLineNumber	50		
Precision	1.0E-5		
TrueNorth	IfcDirection	StepLineNumber	StepLine
StepLine	IFCGEOMETRICREPRESEI		
CoordinateSpace	3		
WorldCoordinate	IfcAxis2Placement3D	Axis	StepLineNumber
ContextType	Model	EncodedValue	DecodedValue
StepLineNumber	29		
Units	IfcSIUnit	Name	StepLineNumber
Units	IfcSIUnit	Name	StepLineNumber
Units	IfcSIUnit	Name	StepLineNumber

Fig. 4 Output in tabular format in an Excel File

VI. Conclusion

This research work aims to parse the Industry Foundation Class (IFC) generated by the Building Information Modelling software. An algorithm is proposed based on breadth First Search technique which is capable of selecting partial or total IFC model. The algorithm extracts physical and nonphysical data instances relevant to the user's selection of building elements. Test results showed that the proposed algorithm successfully extracts the intended partial models from the IFC instance files without using a schema.

References

[1] Graham Brewer, Thayaparan Gajendran, Raichel Le Goff, "Building Information Modelling (BIM): an Introduction and International Perspectives", *Research Report*, CIBER, 2012.

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

[3] Howard, Rob and Björk, BoChrister., "Building Information Modelling – Experts´ Views on Standardisation and Industry Deployment", *Advanced Engineering Informatics*, Vol 22, No. 2, pp. 271-280, 2008.

[4] M. Nour, M., K. Beucke, "Manipulating IFC model data in conjunction with CAD", *Proceedings of CIB w78*, 22nd Conference on Information Technology in Construction, Dresden, Germany, 2005.

[5] Thomas Froese, "Future Directions for IFC-based interoperability", *ITcon* Vol. 8, pg. 231, 2008.

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

[7] T. Liebich , "IFC 2x Edition 3 Model Implementation Guide", Version 2.0 May 18, 2009.
[8] Raninder Kaur Dhillon, H. S. Rai, "Code Compliance with Building Information Modelling", *International Conference on Recent Trends in Information, communication and Technology*, 2013.

Open Science Index, Computer and Information Engineering Vol:8, No:6, 2014 publications.waset.org/9998462/pdf