Extracting Load Analysis Data from IFC

Prof. Bikramjit Singh [1], Prof. Hardeep Singh Rai [2], Rahul Kumar Nanda [1], Pawan Kumar [1]

[1] *Khalsa College of Engg. and Technology,Amritsar*

[2] *Guru Nanak Dev Engineering College, Ludhiana*

Abstract

IFC Format is being used to share the data between different Building Information Modeling Platforms. The IFC file is structured on the basis of various domains such as Architectural domain, Structural Domain etc. Architectural Domain Comprised of architectural data of building entities (BEAM, COLUMN, WALL, SLAB etc.).But the Structural Domain contracts with the Structural data of the analytical model of Building which describes load, type of Load, Bending Moment, Moment of Inertia, Shear Modulus, Buckling etc. This Paper describes the fetching of Structural data from the ifc File of .ifc Format. The fetching process is carried by File handling concepts of java Technology. In which the Structural Constraints such as Load Applied linearly or Pointly on Beam are performed. Calculations are manipulated to extract the bending moment with the use of data fetched from the .ifc file.

Keywords:
BIM , IFC , JAVA TECHNOLOGY

1. Introduction

BIM has totally changed the way of planning, building and management of an Infrastructure. Earlier the building designs were mostly dependent upon the 2-d drawings which caused damage on large extent on information and coordination[1]. With the advancement in BIM technology the problem has been solved mostly. It basically provides a platform to all engineers so that they can work together on the information of the infrastructure[2, 3, 4].

Earlier engineers worked on different software and developed the model in different software which made it very difficult to connect and relate the information with each other[2, 3]. Then with introduction of BIM, many BIM softwares came into existence which stored the information of an infrastructure in the standard form known as IFC. Hence, a software is created in java technology which can fetch the values of structural elements of Building to apply checks according to the Indian Standard Codes(IS) [2, 5, 3, 6, 4, 7, 8, 9, 10, 11].

Email addresses: bikramjitkritjit@gmail.com (Prof. Bikramjit Singh [1]), abc (Prof. Hardeep Singh Rai [2]), nanda49rahul@gmail.com (Rahul Kumar Nanda [1]), pawanbaldhotra94@gmail.com (Pawan Kumar [1])

1.1. Building Information Modeling(BIM)

Conventional Building plan was extremely susceptible upon the 2- D Drawings resulting in poor grouping and loss of processed data[1, 12]. But with the initiation of BIM, this problem has been solved up to maximum extent because BIM use 3- D Models instead of 2 D Drawings and BIM has alternated the way of Planning, Building and Management of an Infrastructure[1, 12].so, It's basically a platform on which all engineers can work together on the processed data of the infrastructure[1, 12].

It was quite difficult to job synchronously by engineers before the initiation of the BIM and one engineer develops a model in different software and other in the other software. They were not able to broadcast with each other. Hence lead to compatibility issues[1]. Then BIM softwares came into presence and proved to be a remedy to these compatibility complications . All the BIM softwares collect the information of an infrastructure in the standard format known as Industry Foundation Classes(IFC) [2, 5, 3, 6, 4, 7, 13, 14, 15, 16].

1.2. Industry Foundation Classes(IFC)

IFC format is used to describe the model of buildings[17, 12]. It describes the sharing and exchange of information among IFC compatible softwares. It

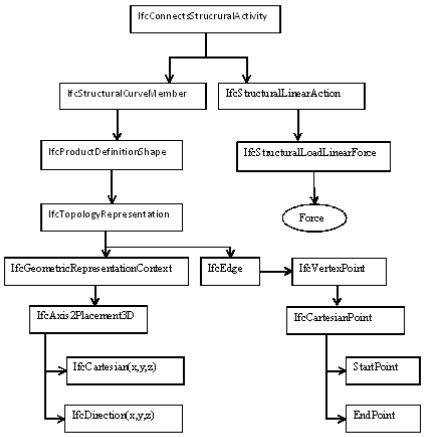


Figure 1: Flow chart of Structural Ananlysis.

stores each entity of the infrastructure that are known as objects. It describes how to show the individual properties of the objects[12].IFC holds data like geometry, quantities, calculations, etc. And the is data is further used by professionals such as Architectural Engineer, Structural Engineer, Plumbic Engineer, Electrical Engineer[2, 3].

IFC schema is created by using graphical notation that is known as EXPRESS-G i.e. international data definition language[4, 17, 12]. It easily helps to make the development and reviewing of the model. IFC exists in another XML form which is known as ifcxml[8].

2. Research Methodology

- The .Ifc file is read simply as a simple text file by the concept of file handling in java technology.
- In file handling the file is being is pared character by character with help of fileInputStream class of java technology.
- After getting character by character all the character are joined to form a string until a semicolon occurs.
- Semicolon ; indicates the finishing of the line in .ifc format.

```
col=Splitting.read(str,'#');
        if(col.isEmpty())
        {
          str="";
          continue;
        }
        n=col.iterator();

        if(n.hasNext())
        {
          String s=n.next().toString();
          Element=n.next().toString();
        }
```

Figure 2: Splitting of the string with a delimeter .

```
col=Splitting.read(Element,'=');
                n=col.iterator();
                if(n.hasNext())
        {
          id=n.next().toString();
          text=n.next().toString();
        }
```

Figure 3: Fetching Data from Collection .

- Those line are processed for which # exists at the starting of the string.
- The commonly used function of strings i.e. split() is used to split the string in various subparts.
- The two parameters are being parsed in the split function. The first one is String itself and the other one is the character through is splitting is being performed.
- # is passed in the string in the split function. It eliminates the first occurrence of # and joins the other substring by #.
- Then further the result string is passed to split function with the splitting element as = and we get two results.

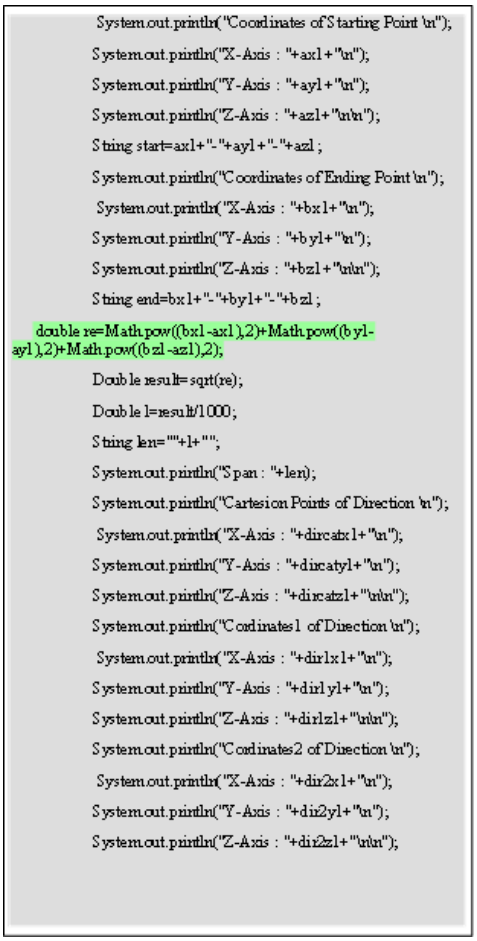
- The first result is id and the other one is the text related to that id.
- After splitting by = ,we get text related to id which further is split by the splitting element (.
- Then we get two results first is tag related to the id and the second one is data related to that tag.
- Then we compare that with any tag of the Ifc file for example IfcConnectsStructuralActivity.
- Then we decode IfcConnectsStrurctural activity to get the ref of IfcStructural LinearAction.
- Then,Further IfcStructuralLinearAction tag is being searched in the .ifc file and id is matched with the reference which got earlier.
- When the tag is matched with IfcStructuralLinearAction and id matches with reference the,we further decode if StructuralLinaerAction to get the reference of IfcStructuralLoadLinearForce.
- Then,further IfcStructuralLoadLinearForce is matched with the tag and id is matched with the reference got form earlier tag .
- When the tag gets matched and id also matches to reference the we decode it to get the value of load.
- Now again IfcConnectsStructuralActitvity tag is traversed and decoded to get the reference of IfcStructuralCurveMember.
- After matching the tag the id is matched with reference ,then it is further decoded to get the IfcProductDefinitonShape.
- The previous tag is searched for matching of id with reference we got.
- IfcTopologyRepresentation reference is searched for reference and the again scanned for its id.
- Now IfcTopologyRepresentation is decoded for two references .
  - The first reference is of the IfcGeometricRepresentationContext and will be searched in the .ifc file until id does not matches with the ref.
    * When tag and id gets matched the we decode that tag to get the reference of IfcAxis2Placement3D.

```
if(tag.equalsIgnoreCase("IFCAXIS2PLACEMENT3D"))
    {
        d=data.split(",");

        str1=str1.replace("#","");
        if(id.equalsIgnoreCase(str1))
        {
        ref=d[0];
        ref1=d[1];
        ref2=d[2];
        System.out.println("ref : "+ref);
        System.out.println("ref1 : "+ref1);
        System.out.println("ref2 : "+ref2);
        ref2=ref2.substring(0,ref2.length()-1);
        IFCCartesianPoint is=new IFCCartesianPoint();
        String g1=is.read(ref);
        IFCDirection im=new IFCDirection();
        String g2=im.read(ref1);
        String g3=im.read(ref2);
        String g=g1+","+g2+","+g3;
        return(g);
        }
    }
```

Figure 4: Sending parameters to child tag.

    * The this tag is also searched in .ifc file till id matched .then we further decode it to get two referneces.
      · The first reference is IFCCartesianPoint.
      · The after searching and matching of tag and id we decoded to get the x,y,z coordinates.
      · The second reference got by decoding IFCAXIS2PLACEMENT3D is the the reference of IFCDIRECTION.
      · After searching and matching the tag with IFCDIRECTION and id qith the ref,we get coordinates of direction. The second reference is of IfcEdge.
  - The second reference is of IfcEdge.
    * From here we again get two further references i.e. of the IFCVertxPoint.
    * Again the tag is matched with the IfcVertexPoint and id with reference.the it is



```
System.out.println("Coordinates of Starting Point \n");
System.out.println("X-Axis : "+ax1+"\n");
System.out.println("Y-Axis : "+ay1+"\n");
System.out.println("Z-Axis : "+az1+"\n\n");
String start=ax1+"-"+ay1+"-"+az1;
System.out.println("Coordinates of Ending Point \n");
System.out.println("X-Axis : "+bx1+"\n");
System.out.println("Y-Axis : "+by1+"\n");
System.out.println("Z-Axis : "+bz1+"\n\n");
String end=bx1+"-"+by1+"-"+bz1;
double re=Math.pow((bx1-ax1),2)+Math.pow((by1-
ay1),2)+Math.pow((bz1-az1),2);
Double result=sqrt(re);
Double l=result/1000;
String len=""+l+"";
System.out.println("Span : "+len);
System.out.println("Cartesion Points of Direction \n");
System.out.println("X-Axis : "+dircatx1+"\n");
System.out.println("Y-Axis : "+dircaty1+"\n");
System.out.println("Z-Axis : "+dircatz1+"\n\n");
System.out.println("Cordinates1 of Direction \n");
System.out.println("X-Axis : "+dir1x1+"\n");
System.out.println("Y-Axis : "+dir1y1+"\n");
System.out.println("Z-Axis : "+dir1z1+"\n\n");
System.out.println("Cordinates2 of Direction \n");
System.out.println("X-Axis : "+dir2x1+"\n");
System.out.println("Y-Axis : "+dir2y1+"\n");
System.out.println("Z-Axis : "+dir2z1+"\n\n");
```

Figure 5: Finding moment of inertia and applied load

decoded to get two reference i.e. IfcCrtesianPoint.

* At the end,we matches the tag to IifcCartesianPoint and id with ref we decode to get the coordinates of two points that are Starting Point and Ending Point of the structural entity on which the load is applied.

- By collecting all the data in the digitized form elated to structural entities of building we can perform the load analysis on the building entities.

3. Result and Discussion

The SCRUTINIZER STRUCT is just been created to alter the conventional technique which is used to apply structural checks manually. The Conventional technique was very difficult and full of errors. SCRUTINIZER STRUCT is a desktop application to apply the structural checks on the entities which are being formulated after fetching data from the .ifc file produced by any 3-D Structural analysis software such as StadPRO , ETABS etc. The calculations are being performed to measure the quantities such as Moment of Inertia, Relative Stiffness and Bending Moment etc. The calculations are performed as the .ifc file is hot having the data related to the above entities. So the parameters as Load, Span etc. are being used to calculate these entities. The java Technology is being used to create the SCRUTINIZER STRUCT as it is fully object oriented programming language and there are various In-Built Libraries to perform calculations. The SCRUTINIZER STRUCT is fruitful desktop application for Structural Analysis and further changes could be made to make it more efficacious.

4. Conclusion and Future Aspects

In Conclusion, the research work describes how the IFC file of structural domain is being used to extract the structural data of an construction entity. The approach used is the automated code compliance and java technology is being used to for the extraction process. Automated Code Compliance helps in saving time , manual efforts and gives error free results.

But, still it is in the developing stage. The values extracted from the IFC of structural domain can be used to perform load analysis on the BIM model. Hence checks could be applied according to the Indian Standard Codes. The Reporting could be improved for the improved graphical representation of the results.

5. References

[1] M.Fischer, Linking cadand expert systems for constructability reasoning, Proceedings of the Fifth International Conference on Computing in Civil and Building Engineering 322 (10) (1993) 156370.

[2] P. Nawari. O. Nawari, Ph.D., A framework for automating codes conformance in structural domain, Journal of Computer and Information Technology 1 (1).

[3] Q.Z. Yang*,Xingjian Xu, Design knowledge modeling and software implementation for building code compliance checking, Building and Environment 39 (6) (June,2004) 689698.

[4] C. fu*, Ifc model viewer to support nd model application, Automation in Construction 15 (2) (2006) 178–185.

[5] zgn Balaban, Elif Sezen Yagmur Kilimci, Glen agdas3, Automated code compliance checking model for fire egress codes, Digital Applications in Construction 2 (2012) 117–125.

[6] David greenwood,Stephen Lockley,Sagar malsane,Jane methews, Automated compliance checking using building information models, The Construction, Building and Real Estate Research Conference of the Royal Institution of Chartered Surveyors.

[7] Susmita Sinha,Anil sawhney, Extracting information from building information models for energy code compliance of building envelope.

[8] Parsing of xml file.
URL Http://blog.sanaulla.info/2013/05/23 parsing-xml-using-dom-sax-and-stax-parser-in-java.

[9] Sharpe R, Oakes S., Advanced it processing of australian standards and regulations, The International Journal of Construction Information Technology 3 (1) (1995) 7389.

[10] Zarli A, Debras P., Integration of corba and web technologies in the vega dis., Proceedings of the European Conference on Integration in Manufacturing, Gothenburg, Sweden 3 (1) (1998) 184 97.

[11] S. M. Inc., J2ee overviewthe java 2 platform enterprise edition.
URL http://java.sun.com/j2ee/

[12] Building, C. A. of Singapore (BCA), e-plancheck system in corenet.
URL http://www.bca.gov.sg/

[13] C. D., Development andimplementation of automated code compliancechecking in the u.s., International Code Council,2007.

[14] Delis E.A.,Delis A., Automatic firecode checking using expert-system technology,journal of computing in civil engineering, ASCE 2 (1995) 141–156.

[15] Ding, L., Drogemuller, R., Rosenman, M., Marchant,Gero, D. J., Automating code checking for building designs: in: K. brown, k. hampson, p.brandon (eds.), clients driving construction innovation, Cooperative Research Centre (CRC) for Construction Innovation (2006) 113–126.

[16] Eastman, C. M., Jae-min Lee, Yeon-suk Jeong, Jinkook Lee, Review automatic rule-based checking of building designs, Automation in Construction 18 (2009) 10111033.

[17] E. Technology, Express data manager.
URL http://www.epmtech.jotne.com