ABSTRACT

MCCARN, SLOAN ALEXIS. Integrated Approach to Design and Construction using Building Information Modeling. (Under the direction of Dr. Han and Dr. Gupta).

An integrated approach to design and construction within the AEC industry is necessary to reduce cost and time overruns. The use of Industry Foundation Classes (IFC) was developed with the interoperability issue in mind to provide conversion options between BIM models and analysis tools. While many BIM products have adopted IFC, its use in the industry is lacking. In order to combat the issue of interoperability, this paper presents an idea of version control in which a main BIM model serves as a data hub for information extraction with the use of IFC. The information extracted is analyzed for missing elements and important data to be converted and analyzed in an finite element (FE) software. A change is made in the FE software and converted back to the original BIM model, keeping old copies (without changes) to form a process of version control and bidirectional conversion of BIM to an analysis tool.

Integrated Approach to Design and Construction using Building Information Modeling.

by
Sloan Alexis McCarn

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Master of Science

Civil Engineering

Raleigh, North Carolina
2020

APPROVED BY:

Dr. Abhinav Gupta
Committee Co-Chair

Dr. Kevin Han
Committee Co-Chair

Dr. Giorgio Proestos

DEDICATION

This is in dedication to my parents, Todd and Donna McCarn for all their support throughout my career and education.

BIOGRAPHY

Alexis McCarn was born in Winston-Salem on January 17, 1996 to Donna and Todd McCarn. She later pursued a bachelor's degree in Civil Engineering and will receive her Master of Science in Civil Engineering in December 2020.

ACKNOWLEDGEMENTS

I am very grateful to the Department of Energy and Idaho National Laboratory for funding and contributing to this research. This research would not have been possible without them. A special thank you goes to my advisors Dr. Kevin Han and Dr. Abhinav Gupta. Thank you for your patience and teaching me to think critically and for believing in me. You all have been such great mentors and my master's experience has allowed me to learn so much. Also, thank you to Dr. Giorgio Proestos for your support and for being on my committee. Lastly, thank you to my family, whom without their support I would not have made it as far as I have. I am continuously full of gratitude for you all and the love and kindness you offer me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 Cost and Schedule Challenges in Nuclear Energy Industry

Increased construction costs and schedule delays have rendered nuclear energy commercially unattractive [1]. One of the reasons is an incomplete design before starting construction, leading to many changes during construction and therefore, causing delays and cost overruns [2]. There have also been delays caused by compatibility and quality issues of modular components and construction practices resulting in additional field changes to design. Despite these complications, there are no digital solutions that can readily port site changes into advanced design and simulation models to assess the implications of these changes. The work/change orders that communicate the changes are confusing and, therefore, cause delays in timely response [3].

Furthermore, there have been discrepancies and errors between as-built models and associated design and construction drawings and specifications. For example, Fermi nuclear power plant had over 7300 discrepancies and errors [4]. In a study on Fermi 2, a nuclear power plant in Michigan, "plant engineering" and design issues were found to have the greatest potential for improvement, right behind productivity [5]. In the failed VC Summers nuclear expansion project, of the eight significant issues facing the project, two are related to incomplete detailed design drawings and designs which were not constructible, thereby resulting in changes and delays [6]. The Harmon Hotel project in Las Vegas, Nevada was an $8.5 billion development and had several parties involved in its creation. There were numerous construction defects and the most important of those were improper rebar installation, causing a domino effect and the need for eventual drastic design changes. The miscommunication, human errors and poor change management led to the downfall of the project, as it was postponed indefinitely [7]. As can be seen from the examples above, a key reason for escalation in cost and schedule is identified as the lack of interoperability between modeling and simulation (M&S) tools. The lack of interoperability can also lead to errors due to incomplete information transfer between as-built construction models and structural design models. Many M&S tools are not compatible with each other, which creates the need to re-establish or redraw information from the original structural models. Examples of such changes can range from something as simple as boundary conditions to large-scale geometry.

A nuclear power plant requires close coordination between many different disciplines for design and construction of not only buildings but also equipment and piping. Discrepancies in design and construction drawings for nuclear power plant equipment and piping can also be quite significant with respect to cost, schedule, safety, and regulatory approvals. For example, the Oyster Bay Nuclear Power plant had more than 100,000 gallons drain from the condensate storage tank due to an incorrectly positioned valve from a "typographical error" [8]. Different disciplines use different design tools and therefore information exchange between them is a challenge. In fact, the current state of practice relies primarily on manual process.

For general buildings, however, the use of Building Information Modeling (BIM) is being widely adopted and implemented in practice. The use of modern BIM tools provides an efficient way to communicate changes through enhanced visualization. BIM models the life cycle of a building while providing information on objects within the model. For this reason, it is often used in design [9]. In this study, the concepts of interoperability between BIM and design tools are explored with the use of a data schema called Industry Foundation Classes (IFC).

1.2 Problem Statement – Interoperability

As identified in the discussion above, there is a significant need for improvement in communication between contractors and designers as well as for digital information exchange between the respective tools for modeling and simulation (M&S, which can encompass design and analysis tools). Lack of interoperability between these tools restricts the use of 3D geometric models provided by architects and plant layouts by designers and engineers. In most situations, the designers recreate the same models in their respective tools. This leads to incompatibility between models as well as human errors when creating a complex geometry and layout. Moreover, incomplete design before the start of construction further complicates this issue. During construction, any on-site change due to unexpected site conditions, flaws, construction imprecision, etc. must be captured accurately and rapidly incorporated into an "as-built" construction model. The new model must then be communicated to design engineers to assess the implication of the change. This is supported through the findings of National Institutes of Standards and Technology (NIST). They estimate that information copying and re-creation costs the industry about 15.8 billion dollars each year [10]. **Figure 1** - (a) Current Practice (Manual Process) and (b) Proposed Solution (Automated Process) of software interoperability in the AEC industry. **a)** illustrates the current practice of digital information transfer among contractors and designers. As seen in this figure, there is no "main model" from which information can be extracted in order to analyze a design change. Consequently, on-site changes and errors need to be communicated directly to design engineers in different disciplines. However, this communication is not always properly relayed which leads to additional construction delays and rework. During this phase, the need for change often causes a domino effect.

This paper presents a study that addresses the challenges in interoperability and proposes an automated approach to overcome the challenges. The main contributions of this paper are: 1) investigate and illustrate the current state of interoperability through a series of case studies, 2) propose an automated approach for interoperability between BIM software (Revit [11]) and M&S design tools such as ANSYS (commercial off-the-shelf package [12]) and MASTODON (open source - data framework [13]), and 3) provide guidelines and recommendations to apply the automated approach to other M&S tools. The proposed solution is presented through a detailed discussion of the development process in order to elaborate on the problems faced and the solution reached. The purpose is to help the research community and professionals with future integration of additional advanced M&S tools with BIM.

CHAPTER 2 CURRENT STATE OF INTEROPERABILITY

Fortunately, the industry has already seen advancement in areas of interoperability. Many of these advancements consider interoperability in regards to building information modeling (BIM) and M&S tools within the structural and construction industry. BIM is advantageous because of its visualization and design capabilities in which properties such as material information and cross-sectional characteristics are attached to the geometry. In an effort to overcome the challenge of interoperability using BIM, companies have developed product-specific software development kits (SDKs) and application programming interfaces (APIs). For example, Autodesk has developed APIs for their products. The COBie extension [14] developed by Autodesk enables products such as Revit and Navisworks [15] (also Autodesk products) to exchange information. Similarly, the use of APIs have allowed information exchange between Revit and Inventor [16]. Though companies are addressing solutions, these are mainly product–specific and M&S tools outside the scope of the product are not included in the interoperability challenge. To address this, BuildingSMART [17] was developed in an effort to create a data schema, industry foundation classes (IFC), that could be read by a large number of software and imported into M&S tools. IFC is a standardized description of the built environment. Many BIM applications have adopted IFC, including Revit, Tekla Structures [18], and SketchUp [19]. However, the research and development for the implementation and use of IFC continues to make progress towards expanding its capabilities. Currently, the adoption of IFC by commercial software only includes the IFC architectural domain. More information of the hierarchy and domains of IFC are found in **Appendix G**. An architectural IFC domain contains geometry and its property information, but unlike a structural IFC domain, does not contain analytical information such as loads or boundary conditions. Other solutions have displayed similar executions as IFC; however, these have lacked adoption in practice. Other contributions are being made with open-source software and tools in which the public has access and can contribute to the expansion of these projects. However, many of these open-source software and tools remain under development and often do not include import and export options with interoperable file types such as IFC. A true interoperability should be bidirectional. Much of the development has focused on one-way transfers although some research has attempted bidirectional conversion.

As IFC continues to develop, researchers have improved upon the data schema and proposed solutions to make IFC more efficient and beneficial. IFC continues to improve upon bug fixes and capabilities through its different versions. IFC has seen adopted by many BIM tools through the translation of architectural information. Because conversion to a structural IFC model has not been adopted by many BIM applications, the capabilities of IFC have not reached full potential and thus has been acknowledged in other studies.

Many studies have highlighted the importance of a structural IFC model and elaborated on how to achieve it. Wang et. al [20] explored the components of an IFC structural model and, using a structural analysis model, transferred data into a structural IFC file. They developed an IFC interpreter software that could read input from an IFC file, analyze the model, and export data as an IFC file. Ling et. al [21] developed a process in which an IFC architectural file was extracted from a BIM model. In this approach, appropriate information from the IFC file (such as geometry, materials) was read and converted to an XML based unified FE model that could export input files to common M&S tools. The unified model could also read input files from tools and create IFC files in a way that demonstrated bidirectional conversion. LIU et. al. [22] designed a one-way integration tool that converted an architectural based IFC model into a structural model. Wan et. al. [23] presented a concept of IFC bidirectional conversion using a web-based model server and identified gaps and areas of improvement in the use of the IFC model using SAP2000 and similar M&S tools. Other studies used IFC to increase data transfer for scheduling and updating in real time. Golparvar-Fard et al. [24] created an as-built point cloud of an existing structure. The as-planned model was joined with the as-built model. To automate progress monitoring, a coloring algorithm was expanded to the IFC model to indicate the progress of construction. Hameladari [25] similarly used IFC in correlation with a BIM software to identify task-object relationships and their schedule. Progress information was updated in BIM and an IFC file was formed to show items that were on schedule or delayed. The lack of adoption of IFC in M&S tools has been seen by the industry as well. IfcOpenShell [26] was developed as an open-source data schema to convert an IFC file into a file format through its executable, IfcConvert, which most tools can read such as WaveFront OBJ (.obj), Collada (.dae), STEP (.stp), IGES (.iges), XML (.xml), and SVG (.svg). However, with the conversion, not all the information is translated and the lack of consistency in translation leads to inaccurate models and thereby leading to inaccurate analysis results. The expansion of the IFC file and its use in the industry has shown the capabilities of IFC and data transfer. Although IFC is making progress, there are still limitations within the data schema.

Studies have recognized the limitations of IFC and suggested ways to reduce these limitations. Lee et al. [27] demonstrated the loss of information between BIM platforms. Although information is lost between transference, the geometry appeared to remain in-tact. Venugopal et al. [28] elaborates on this limitation and concludes that due to the difference in IFC files, the adoption by the industry is lacking. They proposed a domain-specific Model View Definition (MVD) to define precisely how building information should be expressed using IFC.

They define an MVD as “a subset of building product model schema that provides a complete representation of the information concepts needed for a particular information exchange in an AEC workflow.” They suggested an MVD of a specific task or analysis captured by IFC could cut the length of the IFC file, as well as reduce the dissimilarities produced by different IFC files. Ramaji and Memari, [29] elaborating on the use of an MVD, used interpreted information exchange to extract only the information needed by a specific discipline. This allowed the IFC file size to decrease. For example, they converted a wall with reinforcement to be considered as a shear wall with a particular thickness in analysis. However, MVD for IFC files would need to be considered across all software adopting IFC, meaning all commercial software would have to adhere to strict rules on defining import and export options of IFC files, as well as correct definitions of IFC instances. MVD across all platforms would also assume that there are no “bugs” with the file conversion, as has been seen in current conversions. If there were any kind of bug or discrepancies in comparing two IFC files, the comparison would become challenging. Although IFC could present a solution to the interoperability challenge, the developing schema is not without limitations. The industry has also recognized IFCs limitations and presented ways in which the schema can improve, other studies, however, have addressed the interoperability challenge without the use of IFC.

Increasing interoperability between M&S tools have also been acknowledged by other studies that do not rely on the use of IFC as a neutral conversion file. Hu et al. [30] created a unified information model very similar to IFC. They used a text-based conversion to convert four structural applications' input files (SAP 2000, ETABS, ANSYS and MIDAS) into a unified information model. They converted the original geometry into triangulated facets in an

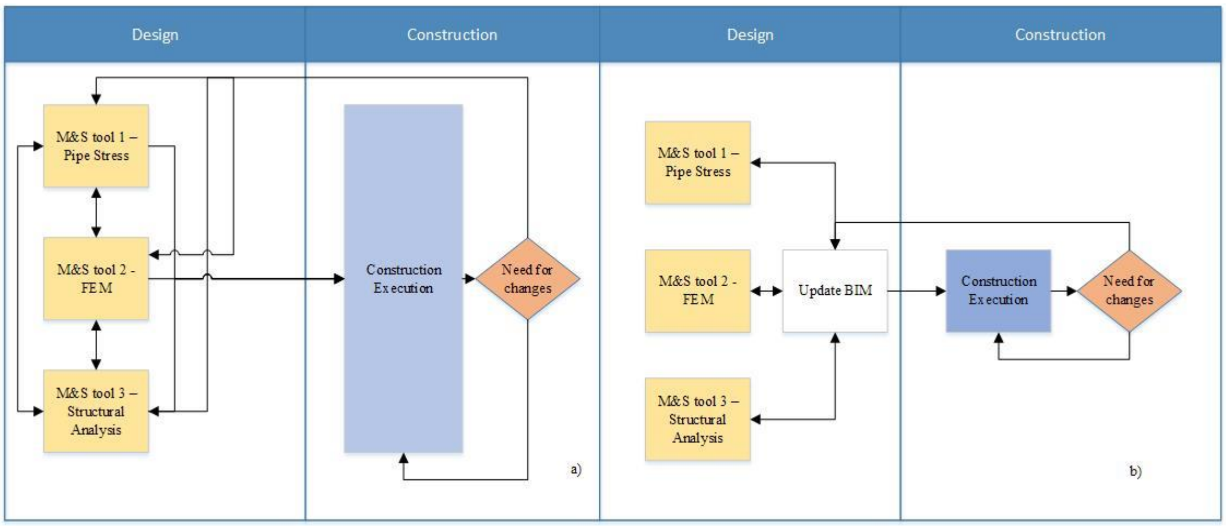


Figure 1 - (a) Current Practice (Manual Process) and (b) Proposed Solution (Automated Process) of software interoperability in the AEC industry.

OpenGL/Web-GL based platform. The conversion was conducted over the web in which users could edit their model. Ninic et. al. [31] addressed challenges to minimize user workload and computational input/output (I/O) time and enable real-time calculations. To do so, the authors used SATBIM [32], an open-source platform for structural analysis and visualization of the tunneling process and Kratos [33], an open-source finite element framework to create an analytical tunnel model for analysis. Their process included changing the levels of detail in which the tunneling model was reduced based upon the significance of the tunneling analysis. For example, a simpler analysis would result in a simpler model, allowing the user to determine the level of detail needed, which would cut back on the finite element run time and processing.

Their approach increased interoperability by using a BIM model to convert frameworks into a FE model. Many of the aforementioned studies have made progress toward interoperability. However, a more efficient, consistent, and accurate data transfer is needed in order for adoption by the industry. The use of IFC by BIM models is through transfer of architectural data models. There is a need to develop a structural model from IFC and, in addition, the transfer needs to map areas of improvement and lost information as well as solutions to overcome it. As of now, input is needed to visually check the model for missing elements and to provide additional input such as boundary conditions. For example, there is difficulty in determining whether end joints/nodes should contain a pin, fixed, or roller connection, which is difficult to determine without having a good understanding of detailed connections within the BIM model and the tranfserred data to an IFC model. Property information should also be included and assigned to geometry in order to have correct data transfer. When transferring to FE models, model simplifications are needed in order to reduce run time of the FE model. More complex finite elements need consideration, such as Q4 or Q8 elements within those models. In addition, many of the solutions presented use commercial software, however, the use of open-source software could be useful by allowing more flexibility to change software settings or include additions to improve upon the software. In areas of bidirectional transfer, many studies have suggested web applications for the transfer, however, this poses a security issue in regards to private information, especially within the nuclear industry. In addition, bidirectional models should keep copies of previous versions of models to lessen the risk of propogated errors. If the use of a data schema, such as IFC, is to be used, the need for consistency in model definitions among different programs should also be considered.

CHAPTER 3 ILLUSTRATION OF LIMITATIONS OF EXISTING INTEROPERABILITY USING CASE STUDIES

The following sections use case studies to discuss the current interoperability between BIM and commercial structural and finite element software as well as existing limitations in doing so. The case studies illustrate the limitations using a BIM software, Revit, and three commercial software SAP2000, Abaqus and ANSYS. These limitations are illustrated using two different building models, a simple building available modeled in Revit and an actual commercial building. Data transfer is shown using available import and export tools including IFC 2x3 (the most commonly used version of IFC).

3.1 Case Studies – Manual Information Transfer Using Commercially Available Tools

To illustrate current state of digital information exchange among commercially available software, a series of case studies are conducted using a BIM model (Revit 2020 [11]), a

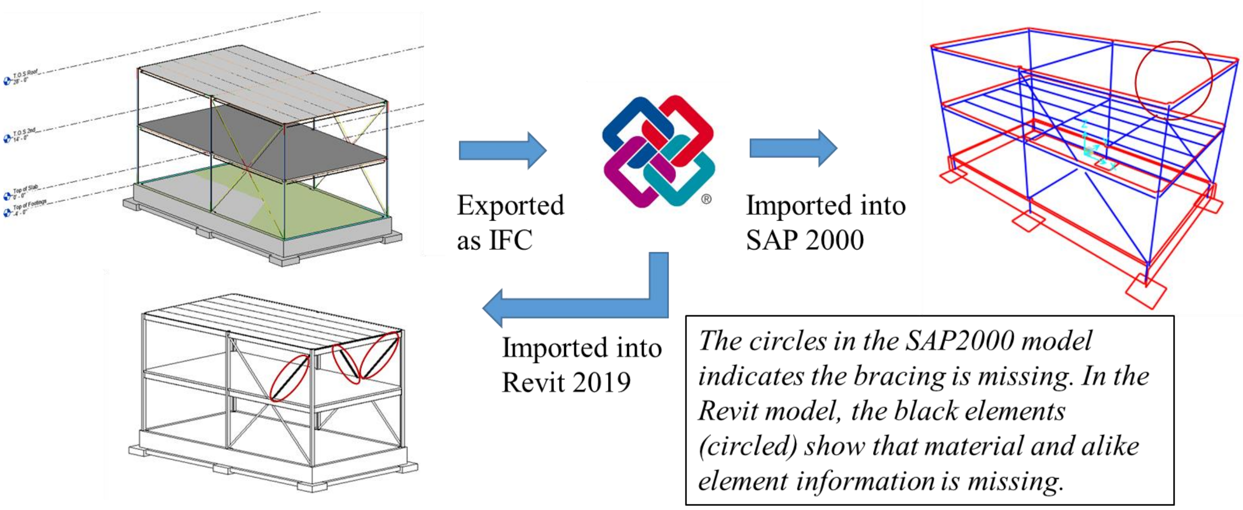


Figure 2 - Simple Revit structure exported as IFC and imported into SAP2000 and Revit to identify data loss.

structural analysis tool (SAP2000 [34]), and two finite element (FE) software (Abaqus [35] and ANSYS[12]). These data transfers are analyzed and lost data acknowledged. The performance capability of the transferred file used in analysis is also evaluated. The purpose of showing the data exchange is to illustrate the current state of interoperability.

A Revit structure, shown in **Figure 2 -** Simple Revit structure exported as IFC and imported into SAP2000 and Revit to identify data loss., is a simple building that is taken as the first case study application. It serves as a good example for comprehensive visualization of discrepancies. The Revit structure is exported as an IFC file version 2x3 coordination view, the most commonly adopted version of IFC, and then reintroduced back into Revit, as well as imported into SAP2000. As can be seen from the figure, not all elements are transferred. This is partly due to the definition of instances in the IFC file. For example, conversion errors occurred with instances *IfcArbitraryProfileDefWithVoids* and *IfcSweptSolid*, and therefore, the bracing that use Hollow Steel Structural (HSS) columns did not transfer. In addition, the joists and metal roof deck did not transfer. In the SAP2000 model, these items did not appear at all (as circled in **Figure 2 -** Simple Revit structure exported as IFC and imported into SAP2000 and Revit to identify data loss.). Upon importing back into Revit, these items are visualized but contain no

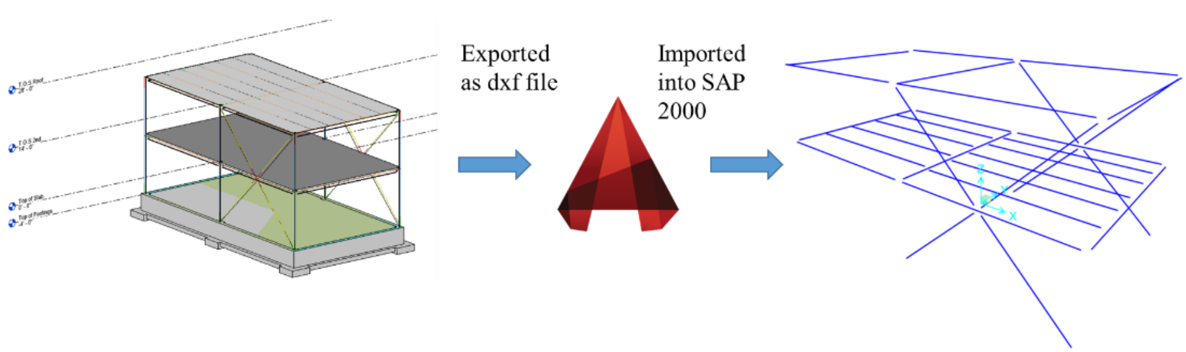


Figure 3 - Simple Revit structure exported as DXF file and imported into SAP2000 and illustrated data loss.

attributable property (as indicated by bold, black objects in **Figure 2 -** Simple Revit structure exported as IFC and imported into SAP2000 and Revit to identify data loss.). For example, instead of an HSS column, the object is defined as “structural framing.”

Alternatively, as shown in **Figure 3 -** Simple Revit structure exported as DXF file and imported into SAP2000 and illustrated data loss. , the Revit file is converted as a DXF (AutoCAD) file and imported into SAP2000. Here, there are more missing elements, and the only elements that transferred are steel beams and bracing.

Next, an actual commercial building is taken. The data for Fitts-Woolard Hall at North Carolina State University is exported from Revit using IFC 2X3 Coordination View and imported back into Revit. **Figure 4 –** Fitts-Woolard Hall conversion using Revit and IFC to map

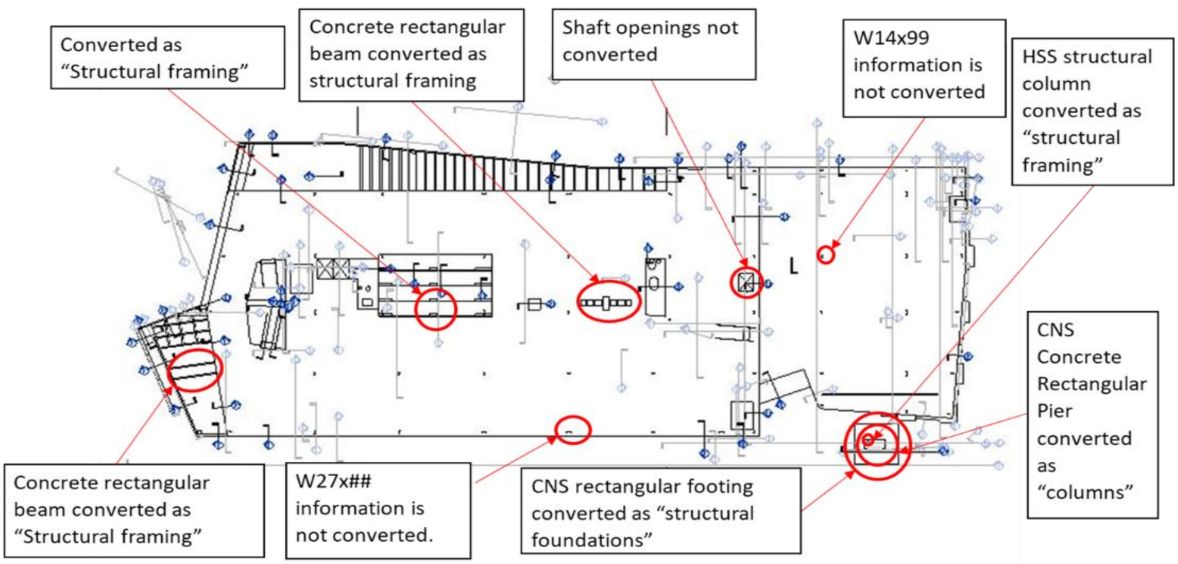


Figure 4 – Fitts-Woolard Hall conversion using Revit and IFC to map data loss.

data loss. shows most of the geometry converted, but there is still information loss. Similar to the simple building model considered earlier, information loss occurs. For example, a W10x30 beam is considered a "structural element," meaning it does not house information pertaining to that object (such as geometrical properties).

Lastly, a study is conducted to illustrate the transfer of a BIM model (Revit) to FE software Abaqus and ANSYS (Abaqus is illustrated). Currently, neither Revit export files nor IFC files can be imported directly into commercial FE software Abaqus or ANSYS. Therefore, as seen in **Figure 5 -** Revit file to SAP2000 file to IGES and SAT file formats to illustrate data loss using finite element software. , the Revit model is exported to IFC, imported into SAP2000 as an IFC file, and exported as an IGES file and an SAT file. The files Abaqus can import and SAP2000 can export are the SAT and IGES file type. The file ANSYS can import is an IGES file. The choice of exporting Revit explicitly as an IFC file is due to the incapability of directly transferring to file types that Abaqus and ANSYS can import. Alternatively, SAP2000, has the ability to not only import IFC files but also the ability to export data into file formats which can be imported into Abaqus and ANSYS. Therefore, SAP2000 is used as an intermediary. At first, it appears that IGES file is able to show most of the geometry in both Abaqus and ANSYS. Upon further evaluation, it is not rendered useful in analysis meaning loads, boundary conditions, and other analysis inputs cannot be applied to the model. The SAT files shows similar characteristics in Abaqus and suffers from missing information that renders it incomplete for analysis purposes. These case studies illustrate that information loss is very common during the digital information exchange. Data loss is more likely to occur when data transformation occurs multiple times (for example, Revit to IFC to SAP2000, versus Revit to IFC to SAP2000 to IGES to Abaqus).

The case studies summarized above are used to develop a detailed assessment of information that is converted through IFC architectural model and the information that requires

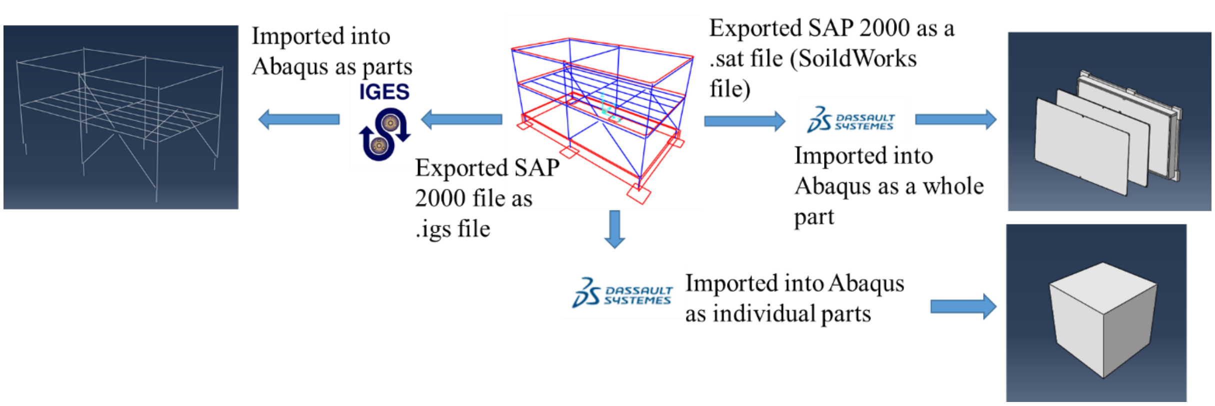


Figure 5 - Revit file to SAP2000 file to IGES and SAT file formats to illustrate data loss using finite element software.

manual involvement to ensure certain completeness. The details of the information type and the manual steps needed in this process are provided in **Table 1**. As can be seen in the table, IFC mainly converts geometry. However, IFC cannot yet be imported appropriately into FE software for analysis. For instance, IFC cannot convert loading, boundary conditions, etc. needed in analysis and is not an accepted file type for FE software. IFC also does not include information on “non-standard” beams, columns, etc. Therefore, this information will need manual input. The case studies presented above highlight that not only is the information missing, there is also an inconsistency in the missing elements/information. Therefore, this paper identifies areas of data loss and also the uses of IFC to transfer data from a BIM model (Revit) to a chosen FE commercial software, ANSYS [12] and an FE open-source software MASTODON, conduct a simple analysis, make changes, and import those changes back into a BIM model.

Table 1 – Conversion of Revit structure findings and mapping of losses.

IFC Conversion			
	SAP 2000	**Abaqus/ANSYS**	**Revit**
Converted	• Basic geometry for analysis from steel beams, columns, and footings, foundation wall, slabs, some steel bracing • Section property information for standard beams • Material information is loaded into materials	• Basic geometry lines for steel beams, steel bracing and columns • 3D geometry (based on the input file)	• Basic geometry (3D) • Information regarding standard steel structures • Grid lines and spacing
Manual Steps to Ensure Proper Conversion	• Boundary conditions and loading conditions need to be reestablished for each element (the import from a 3D Revit model to a mesh model establishes gaps that once withheld boundary condition information) • Check different materials to -make sure they are applied appropriately • Re-establish connections (fixing the gaps created from convergence) • Simplify model for analysis (i.e., removing footings and replacing with fixed boundary conditions)	• These software cannot directly take in IFC files, it uses geometry specific files (i.e., IGES and SAT). • Consequently, a lot of information is lost when importing. • The parts imported in Abaqus should be used as a baseline for recreating the structure or certain parts of the structure (and, therefore, assigning materials, etc.) for analyzing parts of the system.	• Re-establish information for steel members that was lost (i.e. for this example, bracing with HSS columns and K3 joists) • A review of elevations and floor plans to make sure everything is appearing properly as the conversion may have caused changes in the interface. To do this, re-establish grid lines while in the 3D view. • The analytical model may not show everything and, therefore, needs to be updated. This can be done with recreating a new analytical model. This can also be done by assigning Revit categories to IFC classes and types, such as IfcColumn to Column for Revit.

3.2 Current State of IFC for Structural Analysis

IFC adoption by BIM tools are limited by architectural data conversion. Loading, boundary conditions and other structural analysis information are not supplied by the IFC file in its adoption. However, the conversion of geometry is optimistic for IFC. Limitations arise when IFC is not adopted by analysis and FE modeling tools such as ANSYS and ABAQUS. Therefore, IfcOpenShell, a counterpart to IFC has the ability to convert IFC files into a readable file format by those analysis tools. IfcOpenShell is an open-source software library that helps users with the IFC file format[26]. Due to IfcOpenShell's open-source capability, the amount of converted data files can grow. IfcOpenShell is useable in the computer's terminal. It contains an executable, *IfcConvert,* that when called upon in the converter can convert an IFC file into an acceptable file type. Although IFC is widely accepted in the architectural format, it's schema and definitions, as described in IFC's addendum [36] has the capability to transfer more than geometry and its attributable information. **Table 2** lists the capabilities demonstrating its usefulness to various tools such as those used for structural analyses. The cells left blank indicate that the offered component of IFC would not apply to that tool, "✓" shows that the component would apply and is offered by IFC and "X" shows the component would apply but is not offered by IFC at this time.

For many BIM models, such as Revit, the only existing option for IFC export is an architectural model. For the information that IFC cannot store, such as geometrical properties, a connection to an additional database to acquire building information (e.g., the American Institute of Steel Construction (AISC) database) is needed to achieve the degree of automation in the conversion process. This is presented in the following section.

Table 2 - IFC capabilities expanded to different tools.

IFC Capabilities Expanded to Different types of Tools				
			Hazard and Soil Analysis for Earthquake Design	
	Frame Analysis	**FEM**	**Structural Model**	**Soil Model**
Geometries (3D, 2D) (Tessellated, curved, meshed, swept-solid, and boundary representation) • Spatial awareness of geometries and their corresponding hierarchies • Material properties of geometries • Displays voids Displays architectural elements (i.e. door, window,)	✓	✓	✓	✓
Monitor of behavior of flow elements				✓
Survey points				✓
Material profile sets (to standard beams, columns, and other longitudinal elements)	✓	✓	✓	✓
Thermal boundaries	✓	✓		✓
Support reactions	✓		✓	✓
Structural analysis: • Point, curve and surface connections and supports • Specifications of loading including point, curve, surface loads, temperature loads, assignment to load groups, load cases and load combinations, • Type of loads and corresponding safety factors (i.e. wind, dead, live, rain, etc.)			✓	✓

Table 2 (Continued)

Definition of eccentricity	✓		✓	
Footing and pile type information			✓	✓
Static and thermal calculations and load combinations	✓	✓	✓	✓
Analysis of segments, fittings and connections that constitute duct and piping distribution systems, equipment (HVAC), terminal and flow control devices			X	
Finite element topology		X		
Detailed results in finite element meshes as well as stresses and strains in structural elements		X		
Description of prestressed loads	X			X
Boundary conditions (i.e. fixed, roller, pin, etc.	X	X		
Public utility water and waste services			X	X
Industrial and institutional specialty equipment such as that for power production	X			
Provisions dealing with hazardous materials			X	

CHAPTER 4 AUTOMATED APPROACH

For illustration of the proposed automated approach, the data schema IFC is chosen as the tool for integration of the BIM format with two M&S tools. The use of this data schema is determined based on its open-source format. The automated approach exchanges digital information between IFC and FE analysis tools. The two tools considered in this study are: (i) MASTODON [37] which is an open-source software for soil-structure interaction analysis, and (ii) ANSYS Mechanical APDL [12] which is a widely used commercial FE software.

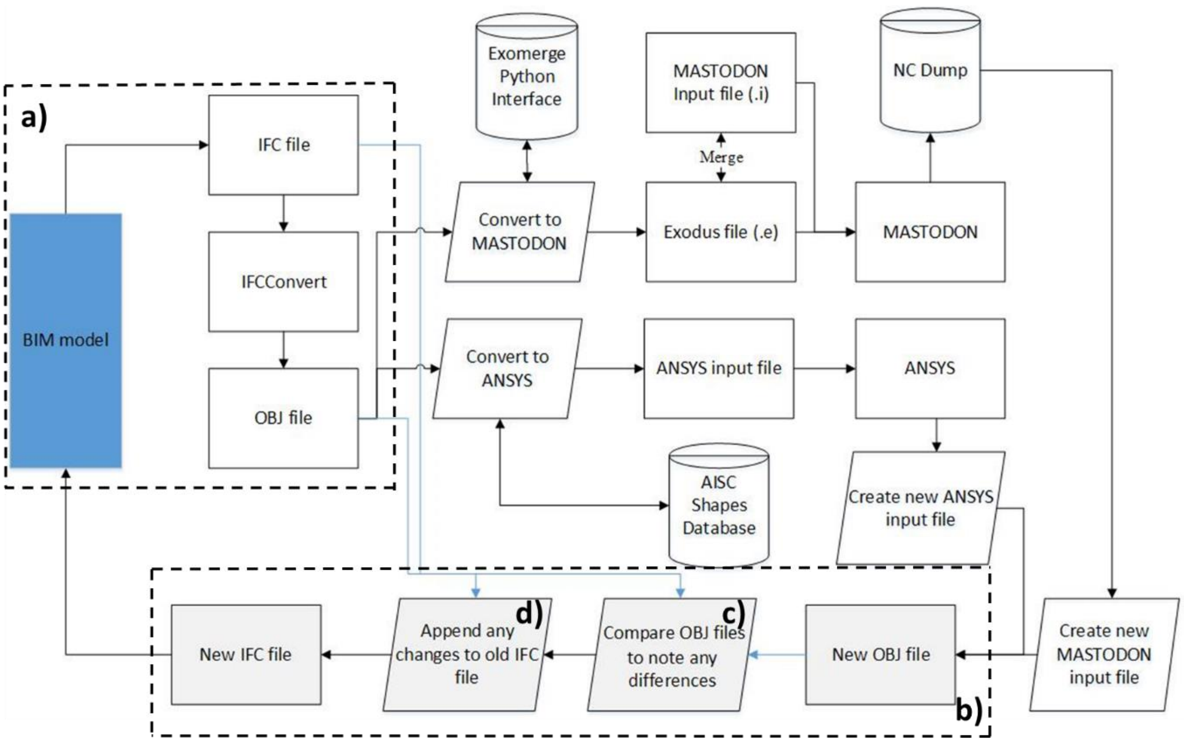


Figure 6 - a) Unidirectional conversion of an IFC file to an M&S tool; b) bidirectional conversion of changes made to an M&S tool to be introduced into a new IFC file; and c&d) these processes are illustrated further in Figure 9 and 10, respectively.

To begin the process of conversion, *IfcOpenShell* [38] is used to parse IFC files. Each IFC file splits a BIM object into a series of entities and attributes in a tree-like format, which references entities and attributes to the appropriate object. From the IFC data transfer, a framework can be developed to expand to other tools. **Figure 6 a)** shows this proposed framework in a unidirectional format. The framework for the reverse conversion process back to IFC is discussed in ***Section 4.2.2***, completing bidirectional conversion. As shown in **Figure 6 a)**, an IFC file is converted to a WaveFront OBJ file, and from there, expanded to MASTODON and ANSYS. Three case studies are considered to provide a detailed discussion of the proposed solution and illustrate its application to specific M&S tools. Case Studies 1 and 2 show the conversion of the geometry data into M&S tools that currently cannot import IFC files. Case Study 1 is specific to MASTODON and Case Study 2 to ANSYS. After the illustration for conversation of geometry, Case Study 3 shows the conversion process for analysis of converted geometries in both MASTODON and ANSYS. The following subsections describes these conversion processes and the three case studies in detail.

4.1 IFC Geometry and Discrepancies using IFC

Both case studies below demonstrate a unidirectional conversion using IFC, version reference 4, exported from Revit, and IfcOpenShell to convert the geometry and information from Revit into an understandable process. Currently, IFC is a very large file that maps elements to different attributes and properties. Because of this, parsing of the IFC file can be tedious. Therefore, the intermediary, IfcOpenShell, is used to convert the IFC file into an OBJ file that can directly define an element's name and geometry. IfcOpenShell contains the executable *IfcConvert* that changes the file format of an IFC file through terminal commands: *IfcConvert Name_of_file.ifc Name_of_wanted_file.<ext>* When just running this command, the output gives an OBJ file that splits each element by its name and a set of triangulated faces. However, because of the large file output of the 3D rendering, simplification of the model is necessary. To decrease the model size, the IFC export options are changed in Revit to convert only the structural elements (i.e., columns and beams, and bracing) to extract specific structural elements.

In addition, because of IfcOpenShell's open-source capability, the settings are changed to meet these simplification needs. The settings are changed to convert the 3D renderings to centerline information in which the updated OBJ file contains the name of the element, the two centerline coordinates, their line connectivity and material property information (see **Appendix A**).

However, upon observing the information gathered from the generated IFC file, there were missing element information. It appeared that either all, or the majority, of column geometry did not appear. This is due to entity definitions in the IFC file. Because of the missing instances in the IFC file, the conclusion can be drawn that the missing information comes from the Revit to IFC conversion. This discrepancy removes the columns from visualization, as they

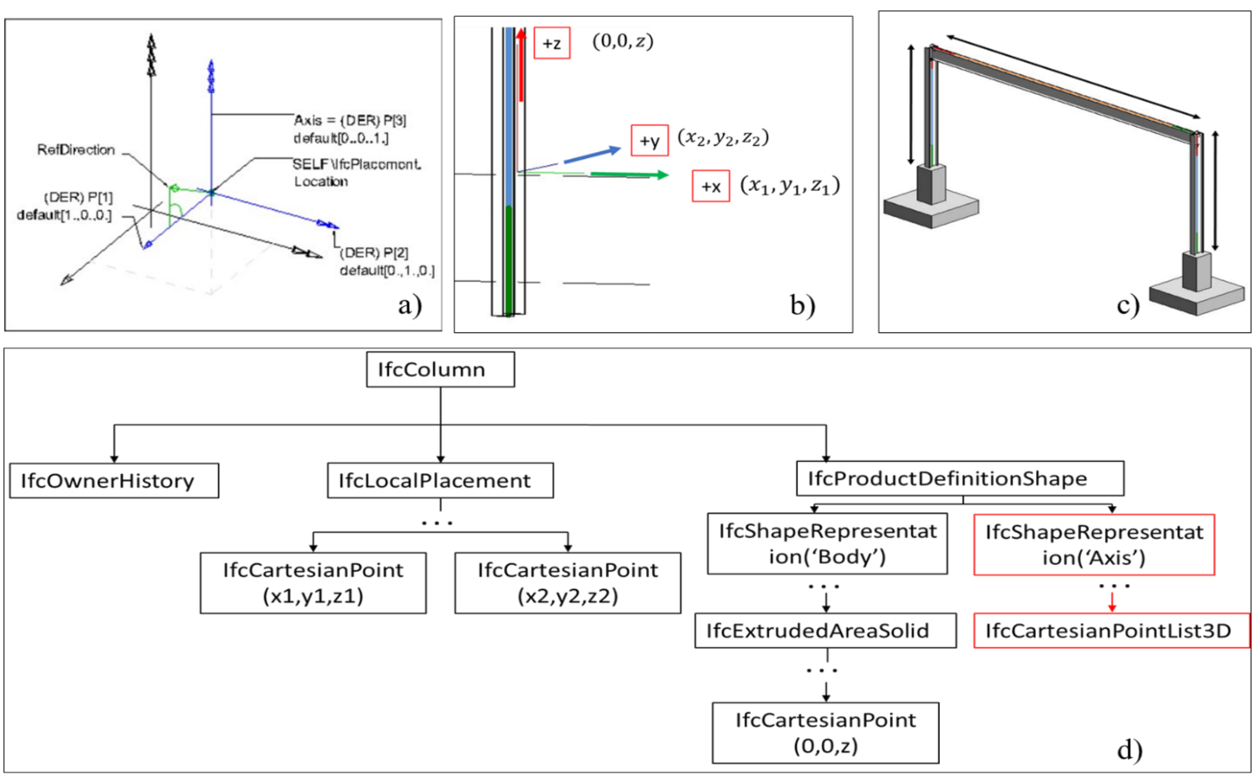


Figure 7 - Process of Retrieving centerline information. a) *IfcAxis2Placement3D* used in coordination with *IfcCartesianPointList3D* to create centerlines; b) Illustration of local Cartesian points; c) shear building created in Revit; 4) IFC mapping of *IfcColumn*.

are represented only as 3D parameterized elements and do not include axis geometry.

In order to fix this error, a python script is developed to create a new IFC file that includes the missing entities of the *IfcColumn.* To identify the missing instances, the mapping of *IfcBeam* was compared with *IfcColumn.* Each of the instances are mapped to three different definitions: *IfcOwnerHistory, IfcLocalPlacement,* and *IfcProductDefinitionShape.* It is found that *IfcBeam's* definition of *IfcProductDefinitionShape* contains the attributes *IfcShapeRepresentation* of "Mapped Body" and "Mapped Axis" whereas *IfcColumn* only contains *IfcShapeRepresentation* of "Mapped Body." The instance *IfcShapeRepresentation* of the axis contains *IfcCartesianPointList3D,* which defines the mapping of local to global coordinates of the element. This instance, in addition with other instances, named *IfcCartesianPoint,*or *IfcAxis2Placement3D* add up to create the global and final coordinates of the element. Without this instance, *IfcConvert* cannot convert the geometry into centerlines. However, *IfcCartesianPointList3D* can be created and mapped into the IFC file and is done so with the use of a Python script. The *IfcCartesianPointList3D* consists of two Cartesian points. The first Cartesian point is drawn from *IfcLocalPlacement.* The second Cartesian point comes from the addition of the first Cartesian point and the length of the column, given in *IfcExtrudedAreaSolid.* (seen in **Figure 7 d)**). Once this instance is created, the element's coordinates can be defined globally. For instance, as seen in **Figure 7 b)**, to obtain the global coordinates of the element, (x_1, y_1, z_1) ,$(0,0,z)$ and (x_2, y_2, z_2) from *IfcLocalPlacement* and the vertices from *IfcCartesianPointList3D* from *IfcProductDefinitionShape* are added to obtain the final centerline coordinates. Once this instance is mapped into the IFC file, it becomes the "new IFC file" for conversion into an OBJ file, where all the centerlines of the structure are included.

4.2 IFC to MASTODON

Case Study 1 explores the file conversion from the BIM model, Revit to MASTODON. MASTODON is a MOOSE-based [13] open-source software developed by Idaho National Lab (INL). Analysis is done through MOOSE in which an input file (.i file format) is created to run a simulation and produce output. Currently, the input files can only define very simple meshes. The program uses EXODUS (.e) files for meshes that are more complex. The EXODUS file, developed by Sandia National Laboratory, is a finite element data model used to store and retrieve data for analysis [39]. EXODUS is the most commonly used file format for creating mesh models in which the input file (.i) retrieves data from the EXODUS file to form the analysis. EXODUS files are binary; meaning the EXODUS file itself is not easily readable and does not have its own graphical user interface (GUI) to help with visualization. Therefore, tools such as CUBIT[40] and TRELIS[41] are used to visualize the EXODUS files and assign properties to the geometry created. To extract information in a readable format from an EXODUS file, NCdump, a command that converts the EXODUS file into a readable format [42] was created to parse the EXODUS file and output a readable format. Generally, EXODUS files are produced from scratch in either CUBIT or TRELIS, in which different elements are assigned an element group and number that is later called and used in the input (.i) file. However, this process uses an IFC model transformed into an OBJ file to create the geometry, which can be assigned properties in TRELIS or CUBIT. The OBJ file is parsed and coordinate and line connectivity extracted. Through the python interface Exomerge [43], a Python interface that creates an EXODUS model from the given finite element information such as nodal coordinates, is used to convert the geometry.

The original 3D OBJ model is used to extract vertices, faces, and lines. However, due to the large file size (using many elements), the entire structure is considered one "element block," which allows for visual conversion, but does not allow for correct or appropriate analysis models. For example, generally with EXODUS files, an element block houses a collection of elements of the same type. Therefore, in this example, proper analysis is not achieved, as columns and beams are both contained within the same "element block." To compensate for this, the conversion gives the user the option to individually analyze one element (i.e., a single beam), which can be visualized alone (without the rest of the structure) for the user to analyze. The OBJ file containing geometry of a structure is then used for conversion. However, manual input is needed by CUBIT or TRELIS to locate the element's name and number, assign conditions, and coordinate its information to the input file, so analysis can be conducted. The model is updated and analyzed using modal analysis. Similar results are seen in **Figure 8**.

4.2.1 IFC to ANSYS

Case Study 2 explores the conversion from the Revit model (the simple building in **Figure 2 -** Simple Revit structure exported as IFC and imported into SAP2000 and Revit to identify data loss.) to an IFC file and then to a text (.txt) file. The IFC file is converted into a wireframe (centerline) OBJ file in which the ANSYS input file is created. The AISC database [44] is used to add geometry to the wireframe model. A general name (such as W-flange) and ID reference number is used to extract geometrical information associated with each centerline, which enables adding geometrical properties and element weights that are not provided in the conversion to the OBJ file.

To perform analysis and confirm geometry and properties are converted properly, a simpler model is tested with the conversion script. This allows a better way to capture discrepancies, missing pieces, and data transfer mishaps. To begin, a simple shear model is created in Revit. From there, the same model is created in SAP2000 to compare hand calculations. Modal analysis is then performed (results can be seen in **Figure 8**). The method to calculate modal analysis is seen in **Appendix B**.

Upon analyzation, with a quick check on the lengths of the elements, it is found that the lengths are incorrect due to improper units (a bug of *IfcConvert*, as the IFC file had US customary units, but, once the IFC file was converted to an OBJ file using *IfcConvert*, it was changed to SI units). The units are thus accounted for in the script files converting OBJ files into input files. In both models, user input is needed for analysis purposes.

The model however, requires user input for boundary conditions. For the simple shear case, the boundary conditions connecting the beams to the columns are all fixed except in the “x” direction. This gave the result seen in **Figure 8**. However, for future cases, in order to achieve a full automated approach, the models will have to be more deeply analyzed in order to determine boundary conditions and avoid human input.

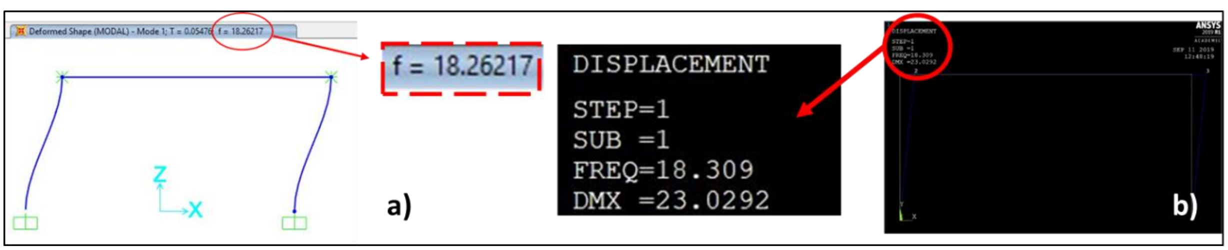


Figure 8 - (a) SAP2000 results of simple shear building and (b) ANSYS results from conversion.

4.2.2. Structural Model to BIM model

To complete the idea of a BIM data hub and a version-control-like systematic approach, the newly converted file should be returned to its original IFC file format which can be imported into a BIM program (such as Revit). The completed bidirectional version is illustrated using

ANSYS, however, the architecture of the approach for both cases is shown in **Figure 6 b)** and detailed in **Figures 9 and 10**.

For ANSYS, similar to version control processes, the copies of the original IFC and OBJ files are kept. Once a change is made in the input file of ANSYS, this is converted to a new OBJ file. The two OBJ files are compared to identify disparities, and the disparities are inputted into a new IFC file (based upon the old) to account for changes. The process is illustrated and executed using the simple shear building.

For this purpose, the beam length is changed, an element is added, and an element is deleted in ANSYS. The corresponding change is made to the IFC file. In order to do this, starting from the original IFC file, the following steps are taken:

1. The simple shear model is drawn in Revit. **(Appendix C)**
2. From the initial Revit model, the model is converted to an IFC model using IFC 4 Reference view. Revit allows the user to have control over the export options to indicate what should and should not be converted to the IFC file (however, there is a limited number of IFC entities Revit can convert to). Therefore, these settings are checked to confirm that a Revit beam would result in an *IfcBeam*. This can be seen in **Appendix D.** Once the settings are changed and confirmed by the user, the model is ready for export.
3. With the current IFC file, the new IFC file is run through a Python script to contain appropriate information for column centerlines. This is done by adding the entity *IfcCartesianPointList3D* to *IfcColumn.* Other entities containing color information are also added. Color information must come in the form of material, therefore, the material “paint” versus the default (A992 steel) is added in which

the associated colors are red, green or blue, to indicate to the user changes in the file. These entities are originally found in IFC version 2x3, however, they are not included in the IFC 4 file from Revit (therefore, the color change would not show in the BIM model), and is thus the reason for the addition of the entities.

4. Once this new IFC file is generated, IfcOpenShell is used to convert the IFC file into an OBJ format that puts the IFC file into a more simplistic format. However, the settings in IfcOpenShell are changed in order to only export centerlines that can be used as a frame model in ANSYS. This is seen in **Appendix A.** Once the settings are changed, the IFC file is converted to an OBJ file using the terminal.

5. Once the format is placed in an OBJ file, this is used to create an ANSYS input file. The OBJ file splits each element into four parts: group name, vertices, line connectivity and material information. From this information, for the ANSYS input file, using the group name, element properties are attached using the AISC shapes database (which gives moment of inertia, torsional constant, cross sectional area, etc.). If the name is not found in the database, the properties are taken to be equal to one. To get the mass, the weight is taken from the AISC shapes database and applied to the calculated length of the element, then placed at one of the two vertices' coordinates of that element. The total script creates an ANSYS input file containing geometric information and mass information. The final version is seen in **Appendix E**.

6. Once the ANSYS file is created, it is changed to increase the length of the beam, delete a column, and add an additional column. To simplify the scripting process the new element is given its name (i.e. W10x30) with an ID of '0' to indicate its

newness. The file is then converted back to an OBJ file, where the bidirectional process began.

7. To understand the backwards process, it is important to understand the IFC file itself. When observing the new element (in this case a column), there are two important entities to add to the original IFC file: *IfcColumn, IfcColumnType* and their corresponding supertypes: *IfcRelDefinesbyType, IfcRelDefinesbyProperties, IfcRelAssociatesClassification, IfcRelContainedInSpatialStructure,* and *IfcRelAssociatesMaterials.* These entities are all illustrated and further explained in **Error! Reference source not found.**

Similar information is found in the *IfcBeam* entity, although it varied slightly. Upon understanding the IFC file, it, in coordination with the old and new OBJ files to produce a final

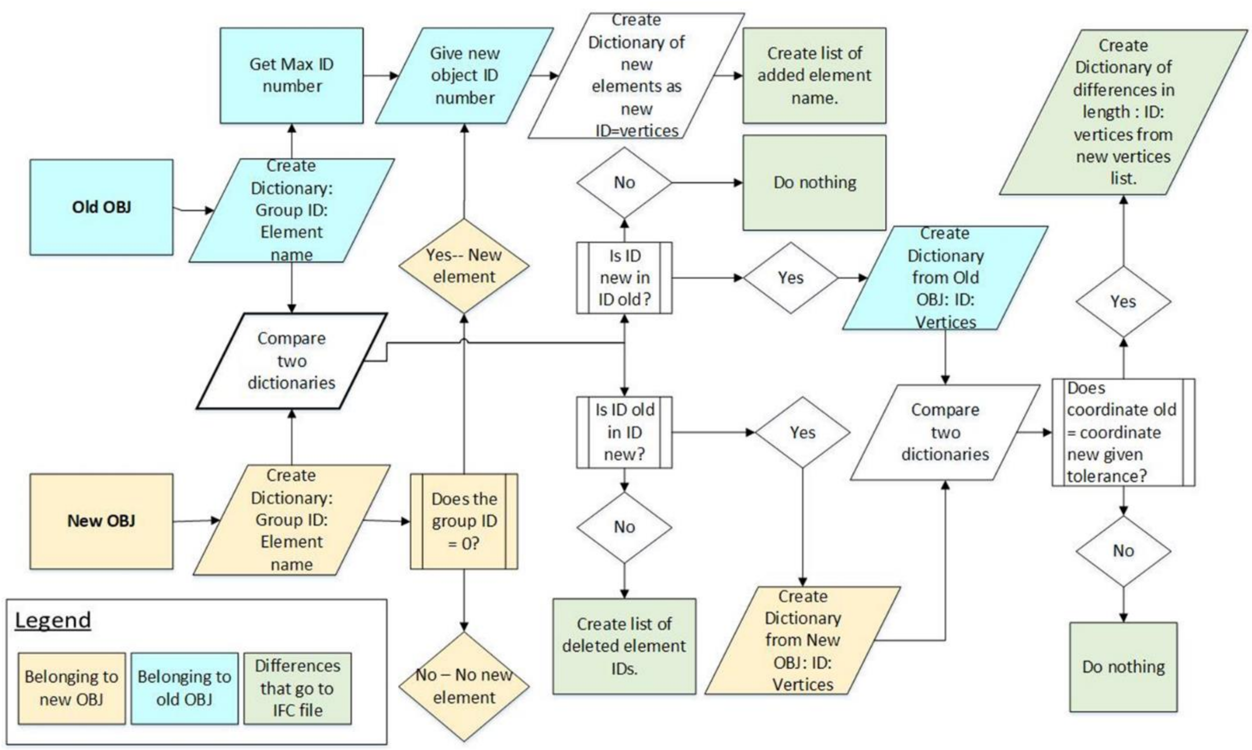


Figure 9 - Comparing of OBJ files to identify differences for bidirectional conversion.

IFC file with added changes. The flow chart seen in **Figure 9** and **Figure 10** shows the process of comparing these files.

As can be seen from the flow charts, the two OBJ files are compared to analyze whether a member has been added, deleted or edited. Once this is determined, the group name is associated to an IFC "key" (the number equaling an IFC entity). Keys in the old IFC file are then added or edited depending on the changes made. Once a change is made to an element, it is assigned a

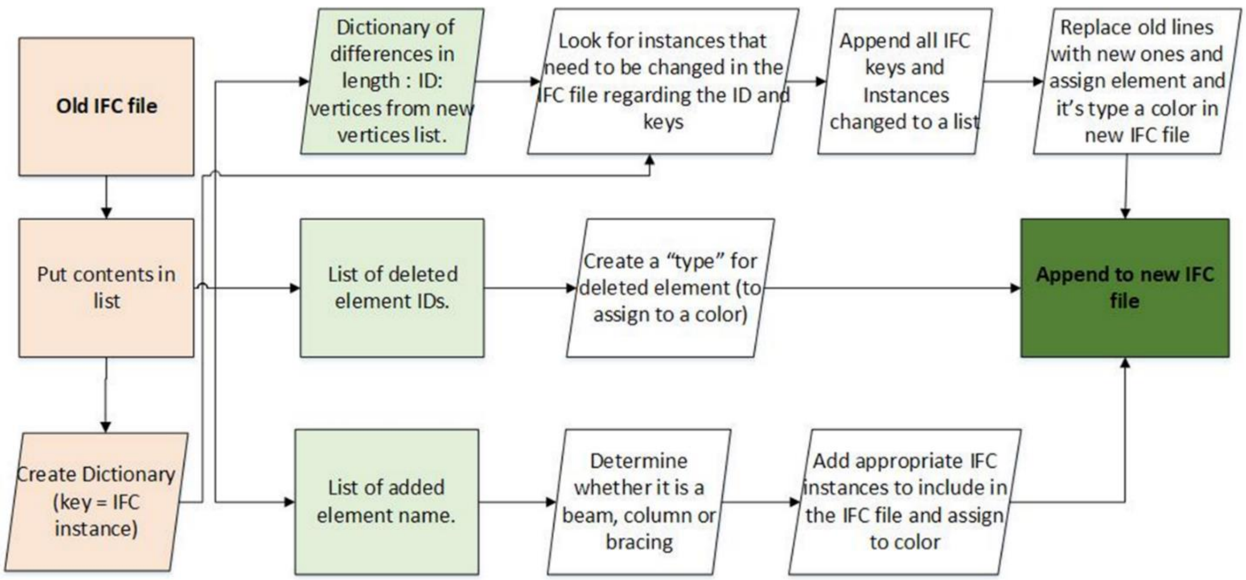


Figure 10 - Changes identified are communicated to the old IFC file in order to create the new IFC file for bidirectional conversion.

color – red for deleted, green for edited, and blue for added. Once the script is run, the new IFC

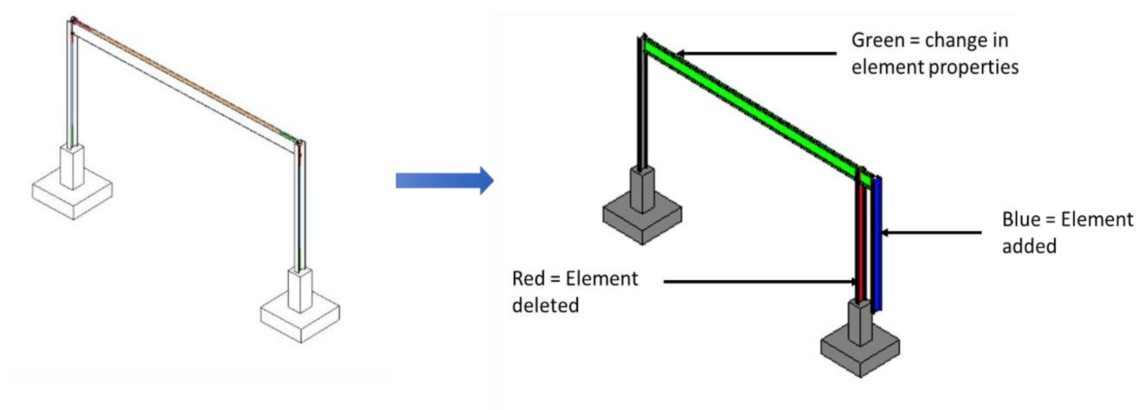


Figure 11 - Final Change with coloring algorithm.

file is created and the outcome can be seen in **Figure 11**. It is important to note that the new IFC file is based off the old IFC file. Therefore, if some geometric renderings were not exported with the original IFC file, they are not included in the new one either, so further evaluation of the BIM to IFC conversion will need to be researched to account for missing pieces.

For MASTODON, the idea is very similar to that proposed in the ANSYS bidirectional conversion. The slight difference is to put the EXODUS file format into a readable file. This is done using the terminal command NCdump (a command that converts the EXODUS file into a readable format) [42]. Using NCdump, a new OBJ file can be created.

CHAPTER 5 RECOMMENDATIONS AND GUIDELINES

The idea of data transfer from a BIM model to M&S tools requires some evaluation. There are ways in which the effort can be improved upon. For one, it is important to mention that so long as IFC or an alike data schema is used, a check must be put in place to notice missing pieces. If a significant amount of pieces are missing, or if a structure cannot be analyzed, a framework design to convert data should be considered. From the experience gathered through

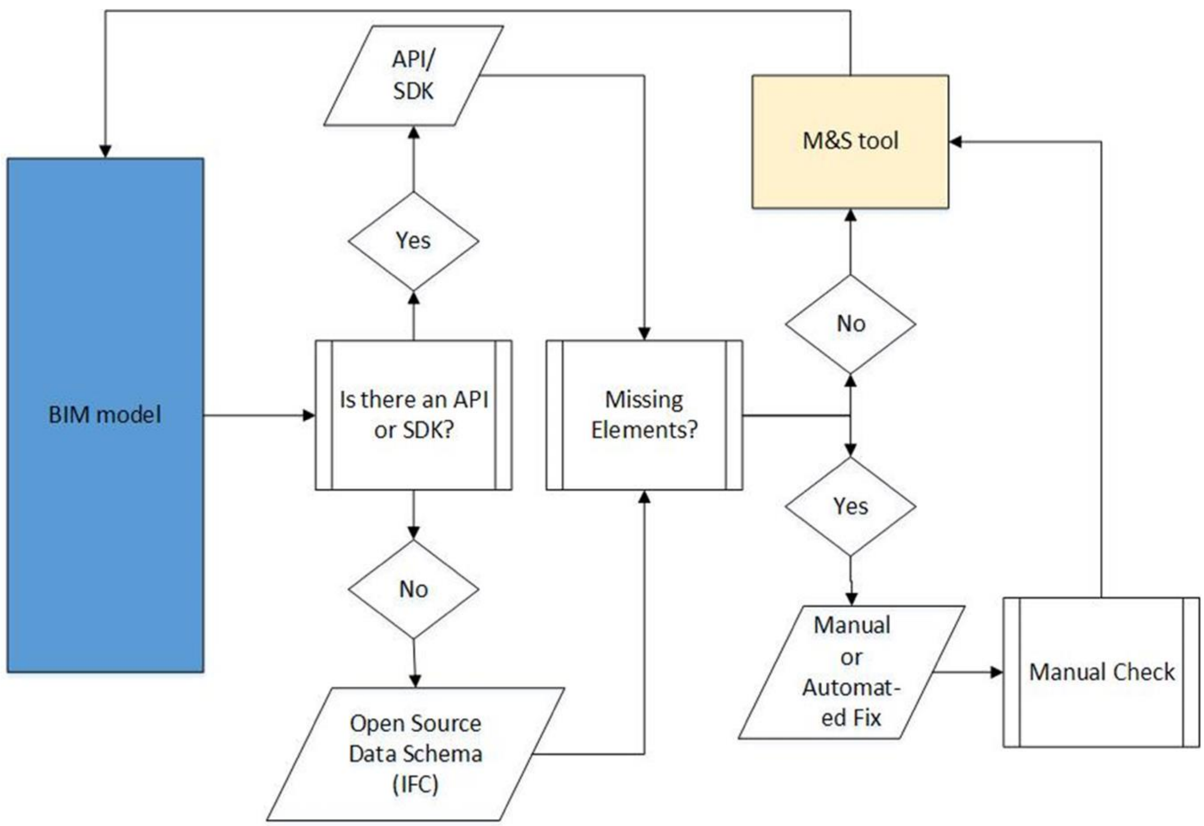


Figure 12 - Process for implementing integration between BIM and analysis tools.

this research, there are guidelines to discuss such data transfer.

The following suggests guidelines to follow to use this automated approach:

1. Determine the available input files used for the M&S tool. The availability of imported files is not consistent for every M&S tool. For instance, ANSYS can import IGES, CATIA, and similar files; however, it cannot import IFC files.

SAP2000 can import IFC files. However, sometimes, these files only contain geometrical information and do not convert material properties. Therefore, as demonstrated in this paper, appropriate measures can be taken to attribute geometries to material properties. It is also important to notice if, of the items that are converted, they can be analyzed in the chosen tool. In ***Table 2***, the IFC data schema was evaluated on its capabilities to BIM interoperability in the AEC industry. Because of its popular use and potential, it was used in this project. However, it is important to determine what data schema or converter is most convenient and useful to the task at hand. Of the missing information, it can then be determined if there are capabilities to adjust for missing information. For example, if material names are given, but material information is missing, a data base can be found and used to attribute materials with their properties (as was done with the AISC database).

Some M&S tools contain APIs or SDKs that can import and export between their company's products. For instance, many Autodesk tools contain APIs that allow files to be shared between their products. Even with APIs or SDKs, all entities need to be carefully checked to avoid any loss when data is transferred. Although relatively easier to manage attributes, the complexity of data structure (factoring in human errors, for example) can easily lead to data loss between conversions.

2. Determine the available export files for the BIM model of choice and see if it matches that of the M&S tool. Again, there are some products that contain APIs or SDKs for this conversion. In some cases, direct conversion can occur. However, in other cases, the need for an intermediary step, such as a data schema

to convert to files that the M&S tool can read. If SDKs or APIs are available, or if direct transformation of data is available, it is important to check for missing information such as geometry and material properties. If there is no error, this can conclude the data transfer.

3. If there is not an exportable format and there needs to be a direct export to the M&S input file, convert the file of the data schema into an easily readable file (i.e., OBJ file) using the available export options or an open-source data schema (such as IfcOpenShell). As in the case with using a data schema for conversion, the simplest method is to convert the file into a most easily read format (OBJ file in this paper). The file can be converted without a "gateway" file, however, this was proven somewhat difficult. This "gateway" file is used as a step between conversion, so there is not direct conversion, but avoids confusing in the complexity of the IFC file. For example, in this paper, the OBJ file was used to put information into easily read group names, vertices, and line connectivity information. However, if there was direct transfer from the IFC file, the vertices information would be extracted from several separate instances, and there could be confusion among the mapping that the IFC file uses. Therefore, using the tools already available, the gateway file was used to put information into a more easily read file.

4. If only geometrical information is converted or shown (OBJ file in this paper), use the AISC database (or an alike database) to gather information to be correlated to the geometry. Other databases can be created through Excel or a similar tool to

extract information for material properties (i.e., creating a database with ASTM information).

5. Using the database chosen, create a script, switching between the M&S tool and the chosen gateway file (or the BIM file) as decided by the user. After creating script files, it is important to keep a record of old files, or determine how changes will be found to import back into the BIM model. Our process kept record of all old files to compare between each other in order to create a new import (IFC) file.
6. Perform checks through log files to determine if geometries correspond directly.

During this approach, as done with FE analysis, for simplification purposes, the models were not meshed. Therefore, one thing to consider, is if a model is meshed in an FE model, steps should be taken to merge this meshed model into a model that can be exported without the confusion of added objects, or understand the complexity of converting a meshed model.

CHAPTER 6 DISCUSSION

Presented in this paper was an idea of version control in the AEC industry. To implement this, the use of BIM was suggested for better visualization and record keeping. It was proposed that data transfer be used to convert from a BIM model to another M&S tool and back to the BIM model with changes made and older copies kept. For the purpose of this plan of work, IFC was the basis of data conversion and transfer. As can be seen from the examples, IFC does not convert everything, but steps can be taken to adjust the IFC file as needed. For example, the columns in the IFC file were not exported, so additions had to be made to the file to export coordinates. In other trials, other structural elements were not exported or could be visualized. In the same way, structural analysis components were not able to be converted (such as boundary conditions and forces). Although the data can be manipulated account for missing elements, a manual check should be done to find discrepancies to avoid inaccurate analysis results.

In using this process, there were some challenges faced. For example, when importing an IFC file back into a BIM model, there is an associated log file of the elements not properly imported back, however, this does not provide an appropriate explanation for all missing pieces. For example, if there are two HSS columns, and one is imported but the other is not, there is insufficient instruction as to why the piece is missing. For this reason, debugging the IFC file was timely but a necessary cost to obtain a correct IFC file. Similar challenges arose with missing pieces in exporting information, and having to debug the IFC file. This could be due to the application-programming interface (API) of the BIM model to convert to IFC or within IFC itself. Either way, it is difficult to know which was causing error.

Although this paper presents one way to create a data hub, geometries can be exported in other ways:

1. Manual input. The user can manually map out changes from one program to another. However, this is time consuming and is a risk for human error, especially as the design becomes more complex.
2. Use of API or SDK. The use of an application programming interface (API) or software development kit (SDK) can be used to convert geometries. Many software companies have API's that run between their programs. However, the majority are only company specific. Other API's and SDKs can be costly and do not always convert all of the geometry.
3. The use of IFC or an alike data schema. The use of an open-source data schema, such as IFC, is free to the public and can be used to convert geometries. However, not all information is converted, and there is potential lack of documentation.

In each of these cases, it is important to note that manual input will still be needed. It is important to double check models to recognize missed information. However, the use of a type of version control could be beneficial to the industry to both reduce error and therefore cost.

CHAPTER 7 CONCLUSIONS AND FUTURE WORK

Currently, interoperability remains a challenge with the nuclear industry. Because of the lack of interoperability, errors occur from data transfer and lack of communication of design changes. These errors cause time and cost increases during the construction process. Steps have been made to support the interoperability challenge, however, mapping of missing information of existing interoperability tools, consideration of bidirectional transfer without the use of a web-based platform, the lack of model view definitions by interoperability tools (such as IFC), finite element transfer, the lack of record keeping of old models, and interoperability tools adopted by the industry are still challenges among this issue. Thus, this paper has acknowledged a few of these challenges through the mapping of missing information of current interoperability tools, creating a version-control concept to allow bidirectional transfer, and mapping missing structural information and solutions to this missing information. Although this paper presents a proof of concept for version control in the AEC and nuclear industry, there are still challenges faced that should be addressed with future work. This process suggests an open-source data schema, which is to be used to achieve a "data hub." However, there are still challenges that need addressing:

1. Further evaluation into IFC to rule out "bugs" or more documentation on the data schema.
2. A need for a model view definition of the IFC model so the models can be simplified for analysis purposes. In addition, consistency amount IFC files of the same version across different BIM platforms is necessary for efficiency.
3. Conversion of more complex geometries within FE models, such as 3D structures versus a wireframe model. The inclusion of higher order elements (such as Q4 or Q8) should also be considered, in addition to processing time of complex models.

4. Integration of interoperability between different disciplines, such as piping and structural systems.

As further research is implemented, this process can reduce the time it takes to create a model. It can also reduce error by avoiding the need to recreate models in M&S tools. By the use of a BIM data hub, communication can be more effectively achieved.

REFERENCES

[1] K. Han and A. Gupta, "PERFORMANCE MONITORING OF MODULAR CONSTRUCTION THROUGH A VIRTUALLY CONNECTED PROJECT SITE AND OFF-," *SMiRT 25*, 2019.

[2] "News About What Went Wrong at VC Summer Gets Worse," *Neutron Bytes*, 2017. [Online]. Available: https://neutronbytes.com/2017/09/10/news-about-what-went-wrong-at-v-c-summer-gets-worse/. [Accessed: 13-Jul-2020].

[3] R. Korman, "Witness to the Origins of a Huge Nuclear Construction Flop: An inside account from 2010 of events that led to the Westinghouse bankruptcy," *ENR: Engineering News Record*, 2017. [Online]. Available: https://www.enr.com/articles/43325-witness-to-the-origins-of-a-huge-nuclear-construction-flop. [Accessed: 11-Sep-2019].

[4] "Information Notice No. 85-66:Discrepancies Between As-Built Construction Drawings and Equipment Installations," U.S. NRC, Washington, D.C, 2015.

[5] S. Alsharif and A. Karatas, "A Framework for Identifying Causal Factors of Delay in Nuclear Power Plant Projects," *Procedia Eng.*, vol. 145, no. 248, pp. 1486–1492, 2016.

[6] W. N. News, "US Governor Releases Report on VC Summer Flaws," *World Nuclear Association*, 2017. [Online]. Available: https://www.world-nuclear-news.org/C-US-governor-releases-report-on-VC-Summer-flaws-06091701.html. [Accessed: 09-Aug-2020].

[7] A. Way, "Deconstructing a $1 Billion Disaster," *Project Perfect*, 2011. [Online]. Available: http://www.projectperfect.com.au/white-paper-deconstructing-a-one-billion-disaster.php.

[8] D. Lochbaum, “Nuclear Pipe Nightmares,” *Union of Concerned Scientists*, 2015. [Online]. Available: https://allthingsnuclear.org/dlochbaum/nuclear-pipe-nightmares. [Accessed: 11-Sep-2019].

[9] R. Sacks, C. Eastman, G. Lee, and P. Teicholz, *BIM Handbook BIM Handbook Rafael Sacks*, 3rd ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2018.

[10] P. Michael, O. Connor, C. Alan, D. Jr, L. John, and T. Linda, *Cost Analysis of Inadequate Interoperability in the U . S . Capital Facilities Industry*. Raleigh, NC: U.S. Department of Commerce Technology Administration, 2004.

[11] Autodesk, “Revit,” *Revit*, 2019. [Online]. Available: https://www.autodesk.com/products/revit/overview. [Accessed: 05-Jul-2019].

[12] A. Inc., “ANSYS,” *ANSYS*, 2019. [Online]. Available: https://www.ansys.com/. [Accessed: 05-Jul-2019].

[13] I. N. Laboratory, “MOOSE,” *MOOSE*. [Online]. Available: https://mooseframework.org/. [Accessed: 05-Jun-2019].

[14] A. B. I. Tools, “COBie.” [Online]. Available: https://www.biminteroperabilitytools.com/cobieextensionrevit.php. [Accessed: 11-Mar-2018].

[15] “Navisworks.” [Online]. Available: https://www.autodesk.com/products/navisworks/. [Accessed: 03-May-2020].

[16] “Inventor.” [Online]. Available: https://www.autodesk.com/products/inventor/. [Accessed: 01-Jul-2020].

[17] BuildingSMART, “BuildingSMART,” *Industry Foundation Classes: Version 4 Addendum 2*, 2016. [Online]. Available: http://www.buildingsmart-

tech.org/ifc/IFC4/Add2/html/. [Accessed: 05-Apr-2019].

[18] Trimble, "Tekla Structures." [Online]. Available: https://www.tekla.com/us/products/tekla-structures. [Accessed: 01-May-2020].

[19] Trimble, "SketchUp." [Online]. Available: https://www.sketchup.com/. [Accessed: 04-Aug-2019].

[20] X. Wang, H. Yang, and Q. Zhang, "Research of the IFC-based Transformation Methods of Geometry Information for Structural Elements," *J. Intell. Robot. Syst.*, vol. 79, no. 3–4, pp. 465–473, 2015.

[21] Q. I. N. Ling, D. Xue-yuan, and X.-L. Liu, "Industry Foundation Classes Based Integration of Architectural Design and Structural Analysis," *J. Shanghai Jiaotong Univ.*, vol. 16, no. 1, pp. 83–90, 2011.

[22] L. I. U. Zhao-qiu, L. I. Yun-gui, and Z. Han-yi, "IFC-based integration tool for supporting information exchange from architectural model to structural model," *J. Cent. South Univ. Technol.*, vol. 17, pp. 1344–1350, 2010.

[23] C. Wan, P. Chen, and R. L. K. Tiong, "ASSESSMENT OF IFCS FOR STRUCTURAL ANALYSIS DOMAIN," *ITcon*, vol. 9, no. June, pp. 75–95, 2004.

[24] M. Golparvar-fard, F. Pena-Mora, and S. Saverese, "Automated Progress Monitoring Using Unordered Daily Construction Photographs and IFC-Based Building Information Models," *J. Comput. Civ. Eng.*, vol. 29, no. 1, pp. 1–19, 2015.

[25] H. Hamledari, B. Mccabe, S. Davari, and A. Shahi, "Automated Schedule and Progress Updating of IFC-Based 4D BIMs," *J. Comput. Civ. Eng.*, vol. 31, no. 4, pp. 1–16, 2017.

[26] "IfcOpenShell," *BuildingSMART International Ltd.* [Online]. Available: https://github.com/IfcOpenShell/IfcOpenShell. [Accessed: 13-Jul-2020].

[27] G. Lee, J. Won, S. Ham, and Y. Shin, “Metrics for Quantifying the Similarities and Differences between IFC Files,” *J. Comput. Civ. Eng.*, vol. 25, no. 2, pp. 172–181, 2011.

[28] M. Venugopal, C. M. Eastman, R. Sacks, and J. Teizer, “Semantics of model views for information exchanges using the industry foundation class schema,” *Adv. Eng. Informatics*, vol. 26, no. 2, pp. 411–428, 2012.

[29] I. Ramaji and A. M. Memari, “Interpreted Information Exchange : Systematic Approach for BIM to Engineering Analysis Information Transformations,” *J. Comput. Civ. Eng.*, vol. 30, no. 6, pp. 1–17, 2016.

[30] Z. Hu, X. Zhang, H. Wang, and M. Kassem, “Improving interoperability between architectural and structural design models : An industry foundation classes-based approach with web-based tools,” *Autom. Constr.*, vol. 66, pp. 29–42, 2016.

[31] J. Ninic, H.-G. Bui, C. Koch, and G. Meschke, “Computationally Efficient Simulation in Urban Mechanized Tunneling Based on Multilevel BIM Models,” *J. Comput. Civ. Eng.*, vol. 33, no. 3, pp. 1–17, 2019.

[32] J. Ninic, “SATBIM,” *Github*. [Online]. Available: https://github.com/satbim. [Accessed: 09-Nov-2019].

[33] International Center for Numerical Methods in Engineering and Technical University of Munich, “Kratos,” *Github*. [Online]. Available: https://github.com/KratosMultiphysics/Kratos. [Accessed: 11-Dec-2019].

[34] Computers & Structures Inc, “SAP 2000,” *SAP 2000*, 2019. [Online]. Available: https://www.csiamerica.com/products/sap2000. [Accessed: 05-Jun-2019].

[35] Dassault Systemes, “Abaqus,” *Abaqus Unified FEA*, 2019. [Online]. Available: https://www.3ds.com/products-services/simulia/products/abaqus/. [Accessed: 05-Jul-

2019].

[36] T. Liebich *et al.*, “Industry Foundation Classes Release 4 (IFC4).”

[37] I. N. Laboratory, “MASTODON,” *Multi-Hazard Analyzis for Stocastic time-Domain Phenomena*. [Online]. Available: https://www.mooseframework.org/mastodon/. [Accessed: 02-Feb-2019].

[38] “IfcOpenShell,” *IfcOpenShell*, 2018. [Online]. Available: http://www.ifcopenshell.org/index.html. [Accessed: 28-Feb-2019].

[39] Sandia National Laboratory, “SEACAS,” *Sandia Engineering Analysis Code Access System*, 2019. [Online]. Available: https://github.com/gsjaardema/seacas. [Accessed: 05-May-2019].

[40] Sandia National Laboratory, “CUBIT,” 2015. [Online]. Available: https://cubit.sandia.gov/. [Accessed: 05-Aug-2019].

[41] csimsoft, “Trelis,” 2018. [Online]. Available: https://www.csimsoft.com/trelis. [Accessed: 08-Dec-2019].

[42] U. C. Programs, “Unidata NetCDF,” *Github*, 2019. [Online]. Available: https://github.com/Unidata/netcdf-c. [Accessed: 01-Mar-2019].

[43] T. D. Kostka, “Exomerge User ’ s Manual : A lightweight Python interface for manipulating Exodus files,” pp. 1–80, 2013.

[44] AISC, “AISC,” 2019. [Online]. Available: https://www.aisc.org/. [Accessed: 06-May-2019].

APPENDICES

Appendix A

IfcOpenShell settings are changed to export only centerline geometry and its information.

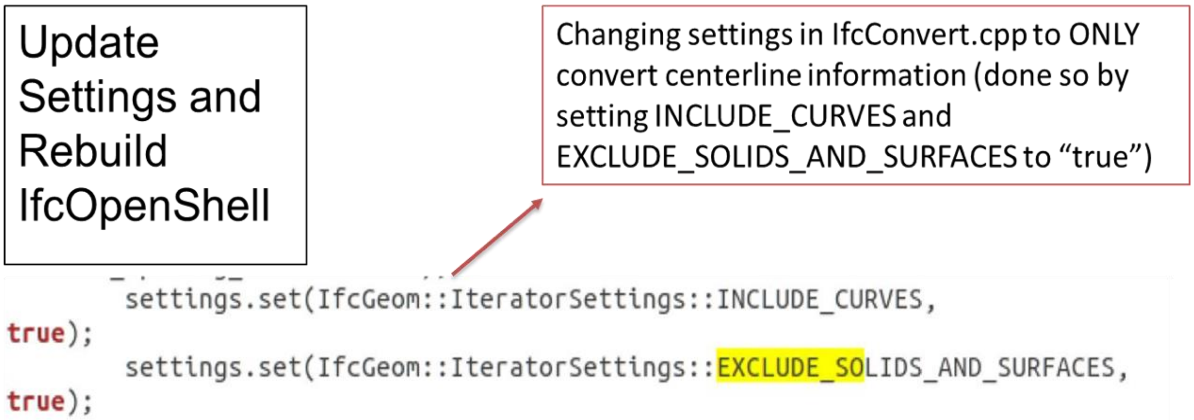


Figure 13 - IfcOpenShell settings for exporting centerlines only.

Appendix B

The following formulas were used to calculate the modal frequency:

$$m = W * L$$

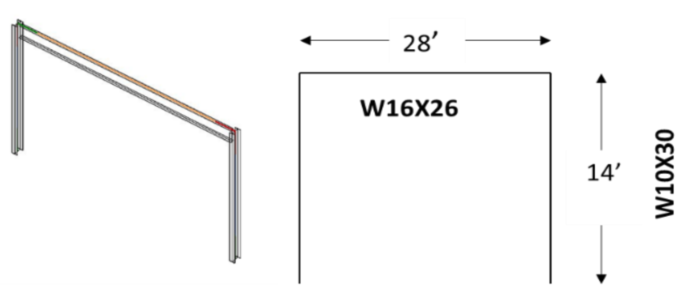
where *m* is the mass of the beam (W12X26) in *k-s*^{2}*/in*, *W* is the weight of the beam in *lb/ft*, and *L* is the length of the beam in *ft*.

$$k = \frac{12EI}{L^3}$$

where *k* is the shear stiffness in *k/in*, *E* is the Elasticity Modulus (taken to be 29,000 *ksi*), *I* is the moment of inertia for the W10x30 column (170 *in*^{4}) in this case, and *L* is the length of the column in *fi*.

$$\omega = \sqrt{\frac{2k}{m}}$$

where *ω* is the natural frequency of the system in *Hz*, *k* is the shear stiffness of an individual column in *k/in*, and *m* is the mass of the beam in *k-s*^{2}*/in*.



W16X26

$$\mathrm{m} = .001884 \left(\frac{\mathrm{k}-\mathrm{s}^2}{\mathrm{in}}\right)$$

W10x30

$$\mathrm{k}_{\text{single column}} = 12.477\,\frac{\mathrm{k}}{\mathrm{in}}$$

Modal Analysis:

$$\omega = 115.088\,\frac{\mathrm{rad}}{\mathrm{s}} = 18.31\ Hz$$

Figure 14 - Revit simple shear building and associated modal calculations.

Appendix C

Shown in **Figure 15** is the simple shear Revit model used for bi-directional conversion.

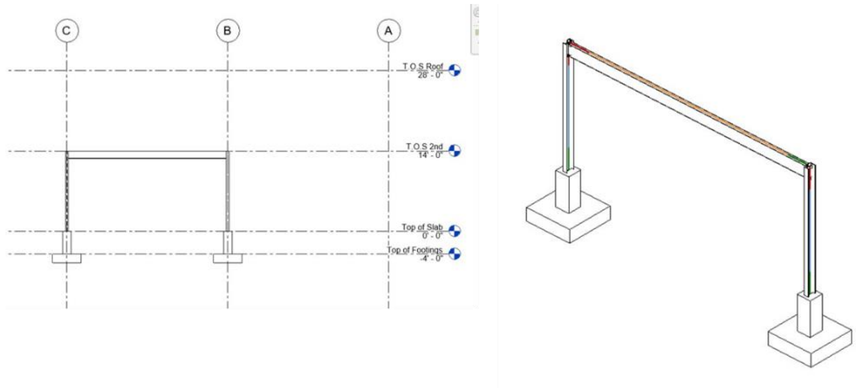


Figure 15 - Initial Revit model of a simple shear building.

Appendix D

The settings to export the IFC file in the ways mentioned in the case study are mentioned here.

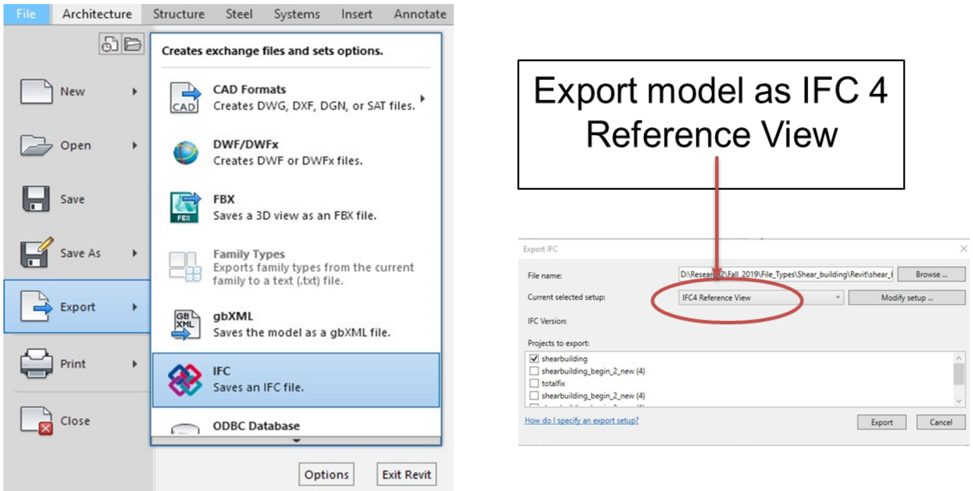


Figure 16 - Revit export to IFC version.

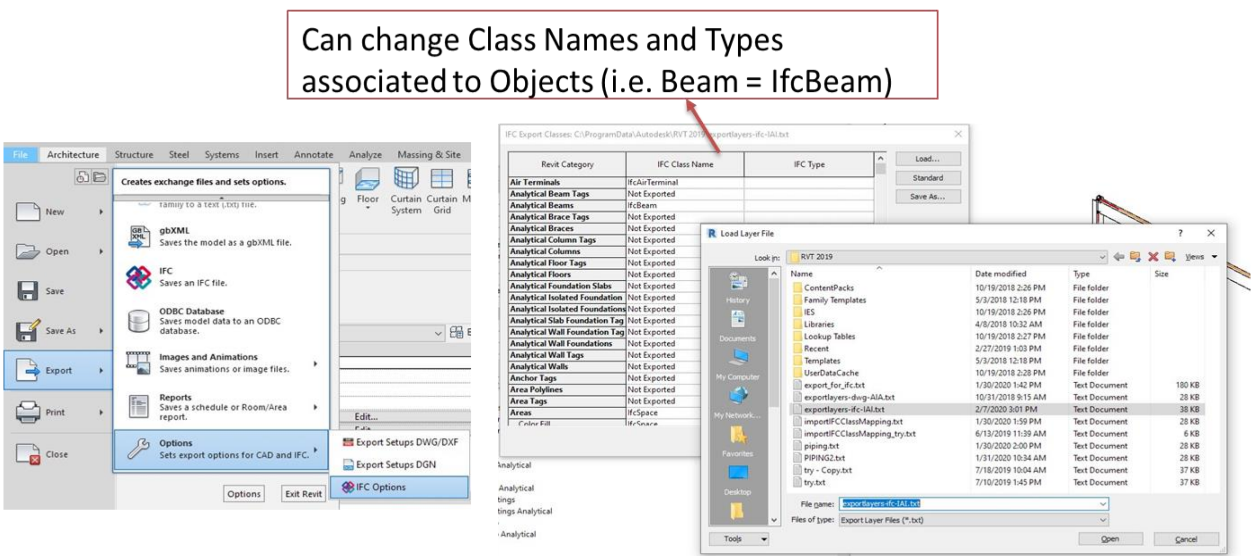


Figure 17 - Revit export of IFC classes.

Appendix E

ANSYS input file for modal analysis after conversion

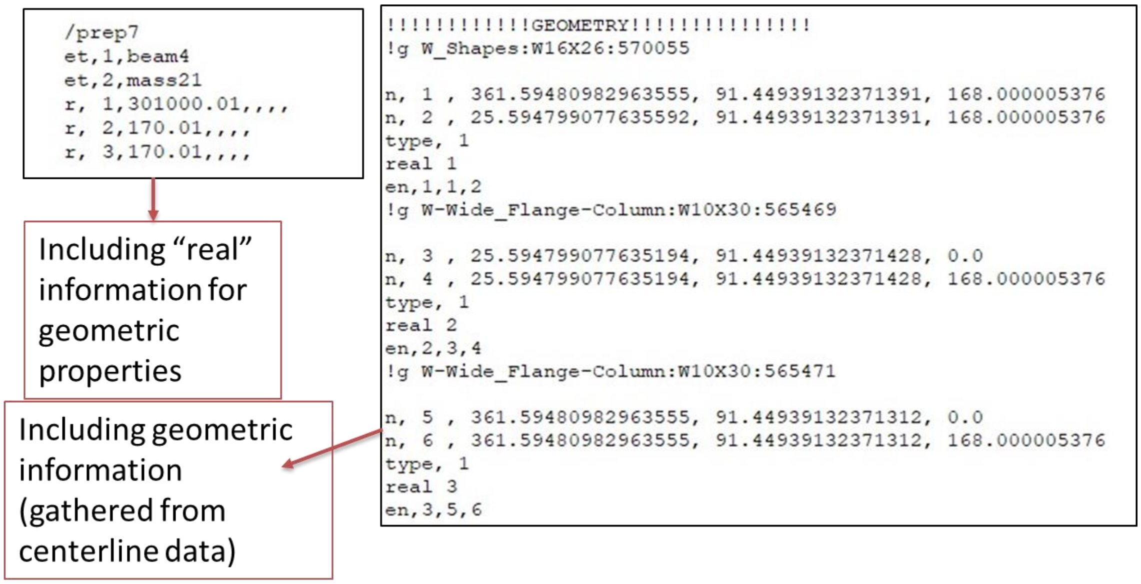


Figure 18 - ANSYS input file of converted geometry.

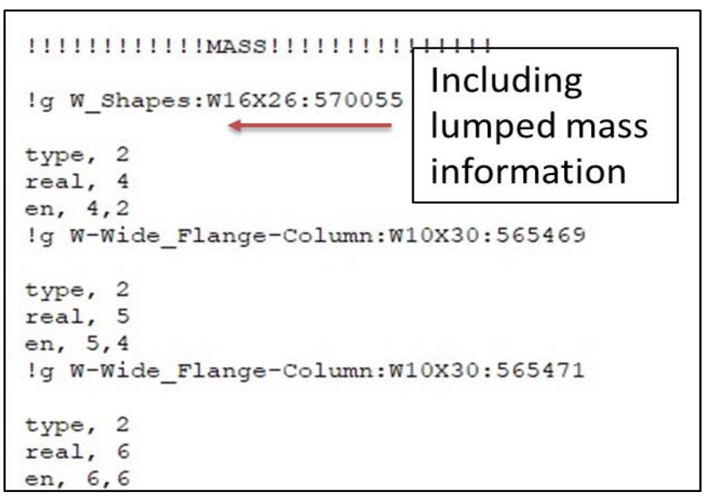


Figure 19 - ANSYS input file of converted masses

Appendix F

Figure 20 - IFC instances for creating a new element.

This shows the IFC instances for creating a new element in the IFC file. *IfcColumn* is the starting IFC instance for a column definition. It then carries its own subtypes (seen in brackets) that carry information about that member. (1) is the GUID for that IFC entity, and varies for each independent entity. (2) references the owner history (when the IFC file was created, who it was created by, etc.). (3) is the column's group name, attached to its ID. Its ID is specific to the member, which is an important piece when it comes to making changes. (4) is the local placement which houses the subtypes *IfcCartesianPoint*, needed for creating the centerline vertices of the column. (5) is the shape definition entity that contains three important subtypes *IfcCartesianPointList3D* (used for centerline vertices), *IfcExtrudedAreaSolid* (containing the height of the column) and *IfcCartesianPointList2D* which contains a group of 2D coordinates, that, when graphed, give the cross sectional shape of that element. (6) shows the ID once again in the entity. *IfcColumnType* is used to group alike entities (i.e. all W10x30's used as columns with the same height, material properties, etc.). It contains property sets for the entity (7,8,9), that houses information for reinforcement, environment impact indicators, etc. It also contains representation maps (10,11) that houses important geometry information such as *IfcCartesianPoint* and *IfcCartesianPointList3D. IfcRelDefinesByType* relates the column (12) to the column type (13). *IfcRelDefinesByProperties* relates the column to specific properties (14). *IfcRelAssociatesClassification* relates the column or column type to its classification reference (15). *IfcRelContainedInSpatialStructure* relates the column to where it is in the Revit space (i.e. its relation to the grids (16). Lastly, *IfcRelAssociatesMaterial* relates the column and column type to a material (17).

Appendix G

The IFC schema consists of different levels: the Domain in which entities are specific to that domain and are not shared between disciplines; The interoperability layer which constitutes

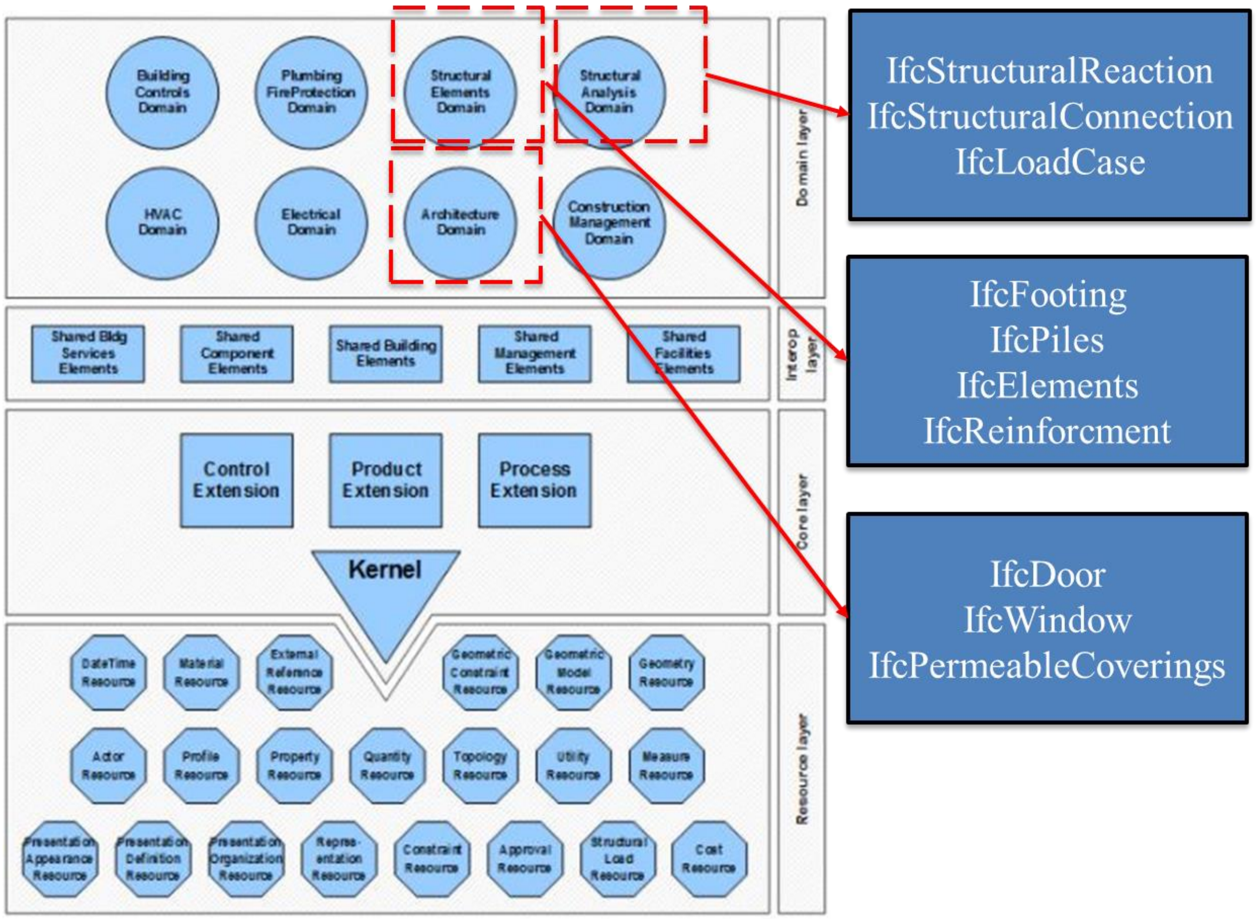


Figure 21 - IFC domain and hierarchy.

the architectural design of a building structure and is shared across all domains. The core, derived from IFC root, which can be referenced by or reference the elements in the hierarchy and the resource layer which is the supporting data structure. For example, looking at the domains, the architectural domain houses specific IFC entities of door, window and permeable coverings that is not shared amongst domains. Similarly, the structural elements domain contains entities relaitng to footings, piles and reinforcement, and the structral anlysis domain contains entities relating to reaction, connections and load cases.

An example of the IFC hierarchy can be seen in **Figure 22**. Starting with the core layer, *IfcRelDefinesByProperties* contains two attributes of the interoperability layer (*IfcBeam* and *IfcPropertySet). IfcPropertySet* contains information relating to *IfcBeam.* The interoperability layer, as well as the core layer is shared by all domains. Within the interoperability layer, for

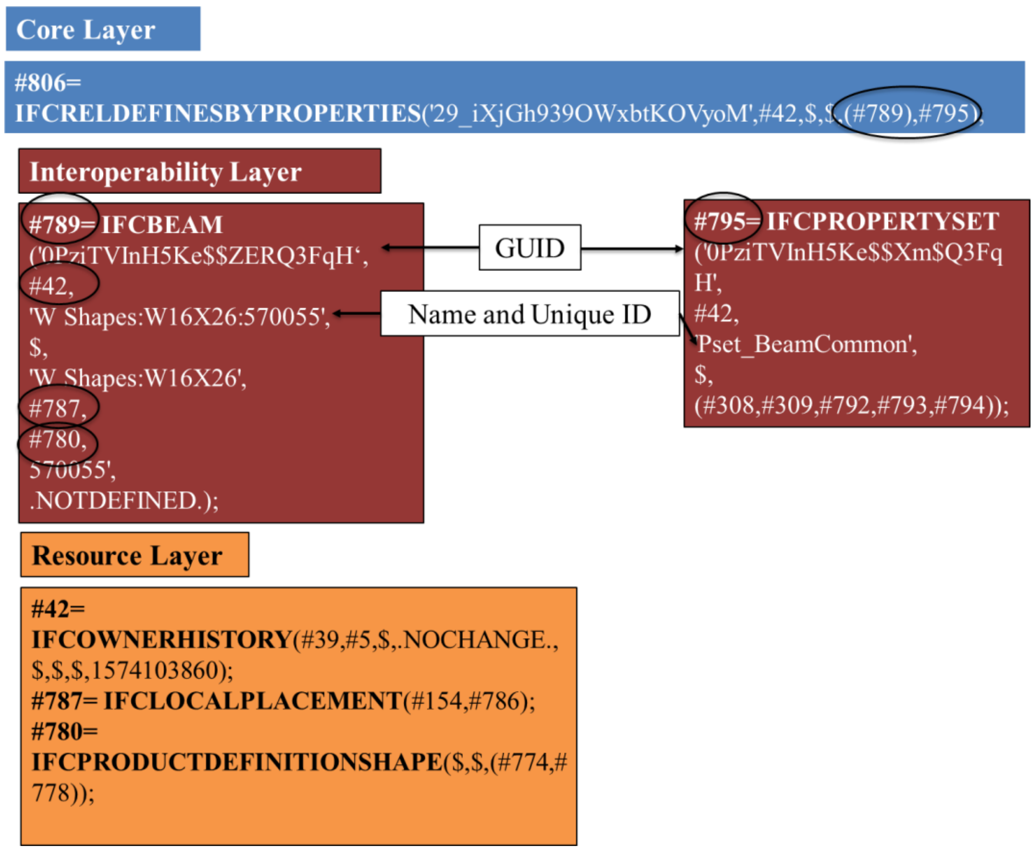


Figure 22 - Example of IFC hierarchy.

example, *IfcBeam,* information is attached in the IFC definition of the unique ID and or a further description/name of that definition. It also contains attributes relating to the resource layer, such as *IfcOwnerHistory, IfcLocalPlacement,* and *IfcProductDefinitionShape.*