DETAILS OF THE MAPPING BETWEEN THE CIS/2 AND IFC PRODUCT DATA MODELS FOR STRUCTURAL STEEL

SUBMITTED: August 2008
REVISED: December 2008
PUBLISHED: January 2009 at http://www.itcon.org/2009/01
EDITOR: C. Eastman

Robert R. Lipman
National Institute of Standards and Technology, Gaithersburg, Maryland, USA
email: robert.lipman@nist.gov, http://cic.nist.gov/

SUMMARY*: A mapping between the CIMsteel Integration Standards (CIS/2) and the Industry Foundation Classes (IFC) product data models for structural steel has been developed. The development of the mapping takes a pragmatic approach through a manual inspection of both schemas to see which entities and attributes correspond to each other. In some cases there is a direct one-to-one mapping between CIS/2 and IFC entities and concepts, while in other cases there is a one-to-many or one-to-none mapping. The mapping has been implemented as a translator from CIS/2 to IFC files. Many examples are shown of partial CIS/2 files and the corresponding mapped IFC entities generated by the translator. The mapping examples and IFC test files generated by the translator have identified several deficiencies in the IFC schema for modeling structural steel and for general structural analysis models.*

KEYWORDS*: product data model, mapping, CIS/2, IFC, structural steel, interoperability*

1. INTRODUCTION

1.1 Background

The CIMsteel Integration Standards (CIS/2) (Crowley and Watson, 2000, Eastman et al, 2005) are the product model and electronic data exchange format for structural steel project information. CIS/2 is intended to create a seamless and integrated flow of information among all parties of the steel supply chain involved in the construction of steel framed structures. The Industry Foundation Classes (IFC) (Liebich et al, 2006) are the product model developed by the International Alliance for Interoperability to facilitate interoperability in the building industry. While the CIS/2 and IFC product models have different views of modeling structural steel, a useful mapping from the CIS/2 product model to the IFC product model has been developed. The mapping permits structural steel models from CIS/2-aware design, structural analysis, detailing and fabrication software packages to be imported into IFC-aware software packages to perform model coordination between the structural steel and the other parts of the building such as floors, walls, windows, doors, ductwork, and mechanical systems.

Conceptually, the relationship between CIS/2 and IFC is shown in Fig. 1. IFC provides integration of building information across multiple subsystems and ultimately across a building's entire lifecycle. CIS/2 provides vertical integration of structural steel information across design, structural analysis, detailing, and fabrication. Where CIS/2 and IFC intersect in the structural area is where the mapping between CIS/2 and IFC is important.

This paper provides additional details for the paper "Mapping Between the CIMsteel Integration Standards and Industry Foundation Classes Product Data Models for Structural Steel" (Lipman, 2006) that was published in the proceedings of the 11th International Conference on Computing in Civil and Building Engineering held in Montreal, Canada on 14-16 June 2006. That paper is a general overview of the mapping issues between CIS/2 and IFC, however, there are very few detailed examples. For this paper to be more useful, it is helpful if the reader has read the original paper and has some knowledge about the basics of CIS/2 and IFC.

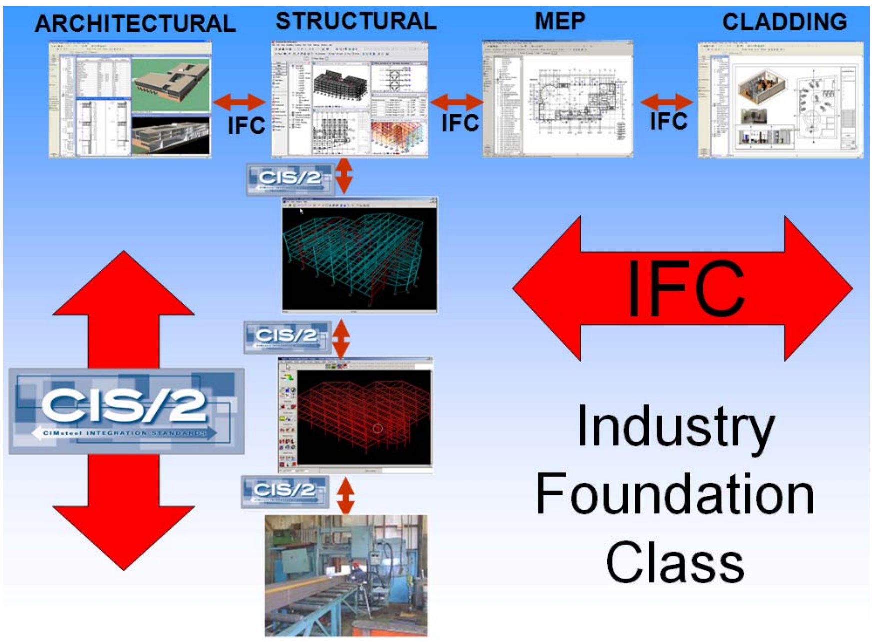


FIG. 1: Conceptual relationship between CIS/2 and IFC (Khemlani, 2007)

1.2 Motivation for the Mapping

The intent in developing of the mapping between CIS/2 (version LPM6) and IFC2x3 was to: (1) provide a practical tool to translate CIS/2 files to IFC files so that structural steel models could be imported to IFC applications that do not import CIS/2 files; (2) to identify deficiencies in the IFC schema for structural steel; and (3) generate IFC test files that use entities that are needed to model structural steel efficiently and that use parts of the IFC schema that have not been commonly implemented. Some of the concepts shown by the IFC test files are detailed in the subsequent IFC examples in the appendices.

The software package that was developed that translates CIS/2 files to IFC files allows structural steel models from design, structural analysis, detailing and fabrication software packages that export CIS/2 files to be imported into computer-aided design (CAD) software packages that support IFC. After the IFC file is imported to the CAD software, model coordination and clash detection can be performed between the structural steel and the other parts of the building such as floors, walls, windows, doors, ductwork, mechanical systems, and concrete structures. Typically CAD software that is used to do model coordination and clash detection import and export only IFC files and not CIS/2 files. Most software specific to structural steel design, structural analysis, detailing, and fabrication only supports CIS/2 file import and/or export and not IFC files.

A typical workflow involves using steel detailing software for detailing the connections between structural steel and the embeds in a concrete structure and then importing the model into CAD software where the concrete structure has already been designed and detailed. The steel model from the detailing software is exported as a CIS/2 file, then converted to IFC format and imported to the CAD software to see if the connection material at the interface between the steel and concrete structures is aligned properly and that there are no interferences. Another workflow involves taking the physical representation of an analysis model used in structural steel analysis software as a CIS/2 file, converting it to an IFC file, and importing the IFC into a CAD package to serve as the basis for a conceptual design.

Traditionally, IFC files have been used to exchange architectural models that contain walls, floors, doors, windows, stairs, beams, and columns. The use of IFC files to exchange structural steel information is a more recent development. The definition of structural steel in CIS/2 is very broad and detailed, whereas its definition in IFC is more generic and not as detailed. In the process of developing the mapping between CIS/2 and IFC, many deficiencies in being able to model structural steel were identified.

For example, CIS/2 can define a pattern or layout of bolts or holes that is located relative to a part or assembly. In IFC2x3, the geometry of the bolts and holes can be modeled by specifying the geometry of each bolt and hole; however, there is no concept of specifying them in a pattern which would be much more efficient.

Another example of a deficiency for modeling structural steel in IFC is the definition of where the longitudinal axis of a part is located relative to its cross section profile. In CIS/2 that location is known as a cardinal point and is typically specified as a point at the corners or mid-sides of the bounding box around the section profile. In IFC2x3, given the cardinal point from CIS/2, the correct geometry of a part and its section profile can be generated; however, there is no way to specify which cardinal point is being used other than deriving it from the cross section geometry placement. An IFC generic property set could be used to specify the cardinal point, however, there is no specification for this definition.

There are also issues related to constructs necessary to efficiently model structural steel in IFC that have not been implemented in some CAD software with IFC import and export capabilities. The most common IFC construct that is not implemented in some CAD software packages is the parametric dimensions of cross section profiles for I-beams, T-beams, channels, angles, and zee sections. The basic dimensions used to define those cross section profiles include depth, width, flange thickness, and web thickness. The CIS/2 to IFC translator, described below, can generate test files that use the parametric definition of cross section profiles which can be used by software vendors to test their IFC implementations of those concepts.

1.3 Mapping Development

The mapping between CIS/2 and IFC was not developed based on any information science research related to the comparison and mapping between different schemas, information domains, or ontologies. The development of the mapping takes a pragmatic approach by a manual inspection of both schemas to see which entities and attributes correspond to each other. The mapping was developed in the context of creating a translator from CIS/2 files to IFC files.

Some of the mappings are obvious such as the CIS/2 entity Cartesian_point mapping to the IFC entity IfcCartesianPoint. Other CIS/2 to IFC mappings are not as direct or have multiple ways of modeling in IFC a single concept from CIS/2. For example, most geometric shape definition is implicitly defined in CIS/2 while in IFC it has to be explicitly defined by extrusions or faceted boundary representations.

There is not necessarily a direct mapping between the semantic meanings of entities in each product model. The name of IFC entities IfcBeam and IfcColumn imply their semantic meaning whereas in CIS/2 the specification of part being a beam or a column is defined by an attribute. Therefore to determine if a CIS/2 part is mapped to an IfcBeam or IfcColumn depends on the orientation of the part or, if available, the attribute. However, the attribute can also indicate that the part is a brace for which there is no corresponding IFC entity. On the other hand, CIS/2 has entities such as Fastener_simple_bolt and Weld_mechanism_fillet whereas in IFC the definition of bolt or weld is an attribute of IfcMechanicalFastener or IfcFastener.

The CIS/2 to IFC mapping detailed by the examples in this paper can also serve as a basis for a reverse mapping between IFC and CIS/2. Independently, as part of a project to harmonize the CIS/2 and IFC product models (Eastman, 2004), a mapping between IFC and CIS/2 was developed for the use cases of structural analysis and steel detailing.

1.3.1 Other Schema Mapping Techniques

The mapping described in this paper has been used at Stanford to develop an information science approach to the mapping between the CIS/2 and IFC product models (Pan, et al 2008). The approach uses statistical semantic similarity techniques that are applied to ontology mappings. The semi-automated techniques used are an attribute-based approach, a corpus-based approach, and a named-based approach. The attribute-based approach looks at the attributes used to define CIS/2 and IFC entities. For example, if an entity from both CIS/2 and IFC reference the same type of attributes such as a cross section and length to define an extrusion, then the entities from each might be similar. The corpus-based approach uses text documents, such as building code documentation, to find similarities between entities names used in each product model. For example if the building code documentation frequently uses the words “beam”, “column”, and “part” in close proximity, then entities names in each product model that use those terms might be related such as IfcBeam and IfcColumn in IFC and Part_prismatic in CIS/2. The name-based approach looks at the obvious similarity between some entity names in CIS/2 and IFC such as Cartesian_point in CIS/2 and IfcCartesianPoint in IFC. However, that similarity does not guarantee that they refer to the same concept. For all three approaches statistical measures are applied to rank the similarities to determine if they can be used for the mapping. The mapping development described in this paper uses the attribute- and named-based approach based on a manual inspection of CIS/2 and IFC entities.

Other schema mapping techniques have been developed that compare differences between versions of the same schema such as IFC2x2 to IFC2x3. Comparing between different versions of the same schema reduces the scope of the mappings that need to be made. One method categorizes potential differences (identical, renamed, equivalent, modified, added, removed) between IFC schema entities and types and develops an automated system to make the comparison (Amor and Ge, 2002). About 65% of the mappings between different versions of the IFC schema can be generated automatically with this method. Similar research to Amor and Ge extends the classification system and develops a version matching approach that provides significantly better results when compared to manually matching between schema versions (Wang et al, 2006, Wang et al, 2007).

Mapping techniques have also been developed between schemas that have different modeling languages such as XML and EXPRESS. This technique uses domain specific constraints (Wang et al, 2008) in a semi-automated approach to map between IFC2x2 and the AEX (Automating Equipment Exchange, FIATECH 2004) XML schema for HVAC equipment such as fans, pumps, and dampers.

While they do not involve mapping between schemas, techniques have also been developed for automated methods to detect differences between IFC files of the same schema that represent similar structures (Ma, et al 2006, Arthaud and Lombardo 2006).

1.3.2 Mapping Categories

The development of the mapping described in this paper is grouped into five categories: (1) shape and geometry representation; (2) CIS/2 design model, (3) CIS/2 detailed model, (4) CIS/2 structural analysis model, and (5) other concepts. These mappings for categories are described in the following sections 1.4 through 1.8. Examples of partial CIS/2 and IFC files that give the details of the mapping are found in the corresponding appendices A-E.

In a CIS/2 or IFC file, entity names are always in upper case letters such as SECTION_PROFILE or IFCBEAM. To differentiate between entities from each type of file and to improve readability, in the text of the paper (not the figures), CIS/2 entities are written as Section_profile and IFC entities are written as IfcBeam. To save space, similar entities that can be grouped together are written as Ifc{Beam/Column/Member} or Section_profile_{I/T}_type.

1.4 Shape Representation

The shape representation refers to how the geometry of a part is defined. Fig. 2 shows some typical shapes that are used in CIS/2 including I-beams, T-beams, channels, angles, rectangle, circle, and zee sections. Also shown are plates, bent plates, corrugated decking, curved beams, and double sections.

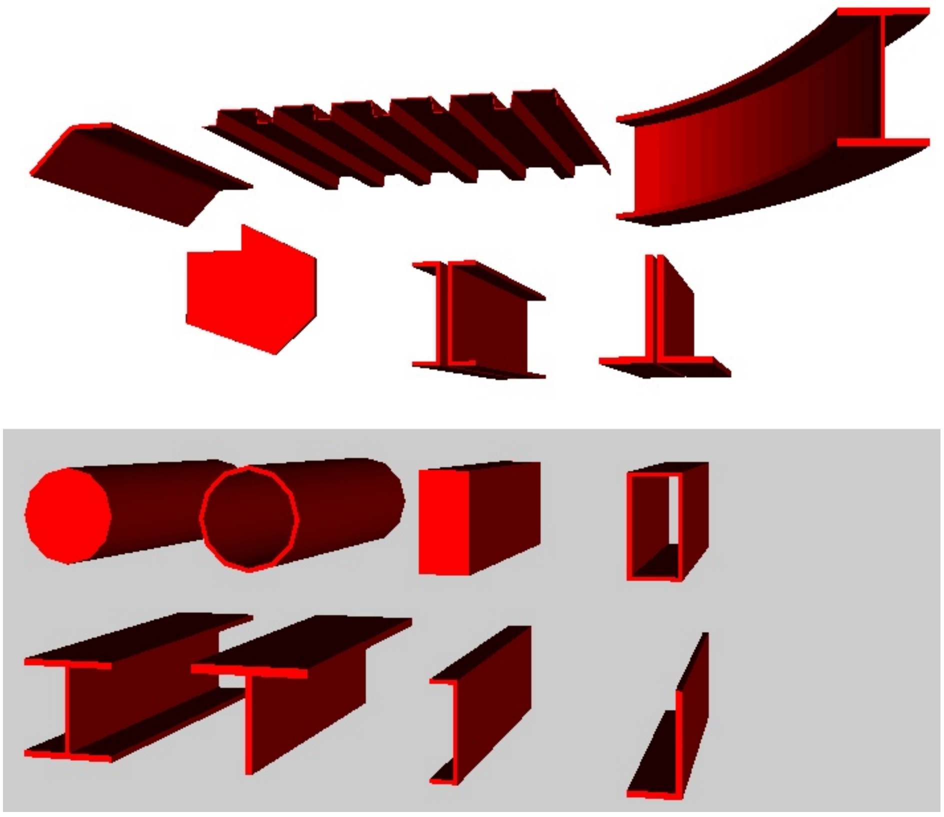


FIG. 2: Typical CIS/2 structural steel shapes

In CIS/2, a long and narrow (prismatic) part is defined by a cross section and length. The cross section is specified by a section designator, cardinal point, and boolean value to indicate if the section is mirrored. The dimensions of the cross section can be implied by the section designator or given explicitly by a section's depth, width, and web and flange thicknesses. If the dimensions are implied by the section designator, then software using the designator must have a lookup table of the section dimensions or parse the dimensions from the designator. The cardinal point defines the location of the origin of the cross section. The most commonly used cardinal points are the center, corners, and mid-sides of a bounding box around the cross section. For example, a cardinal point at the top of the web of an I-beam can be used to position the beam so that the top of the top flange is at a floor level. Mirroring involves flipping the cross section about its vertical axis. Mirroring a cross section is only relevant to sections that are non-symmetric about a vertical axis in the plane of the cross section such as angle, channel, and zee sections. For example, if the angle section in the lower right of Fig. 2 were to be mirrored, then the end of the cross section would appear as the letter 'L'.

Plates in CIS/2 are defined by their thickness and a set of points or line segments that define the edge of the plate. Bent plates and decking are defined their thickness, length, and points that define the profile of the bend or corrugation. Curved parts are defined by a cross section profile and points that define the curve of the part.

For the most part, all geometry in CIS/2 is implicitly defined. The information used to generate the geometry of parts is found on entities related to the definition of a part and not on entities related to the definition of geometry. Conversely, all geometry in IFC is explicitly defined. Information used to generate geometry in IFC is found strictly on entities related to the definition of geometry. Parts, such as beams and columns, must refer to the geometric shape representation entities if they are to have a physical representation.

Several different methods can be used for shape representation in IFC. The most general is a faceted boundary representation. Geometry is defined by points that are used to defined edges. The edges define faces that are used to define a closed set of faces representing the boundary of a geometric shape. With this type of representation any arbitrary geometry can be defined.

The other method to generate shapes in IFC involves extruding a cross section profile along an axis or curve. The profile can be defined by an arbitrary set of closed points or line segments or parametrically defined for an "I" shape, angle, channel, or other similar sections. The parametric values are typically the depth, width, and thickness of the section. The arbitrary closed set of points can also define the edge of a plate. An arbitrary coordinate transformation can be applied to the profile and can be used to mirror the cross section. In IFC2x3 there is no explicit definition of a cardinal point although that information can be implied by the cross section geometry.

Detailed examples of the mapping between CIS/2 and IFC for shape representation are in Appendix A. Generally, the mapping between CIS/2 and IFC for shape representation is straightforward. The most significant issue is the lack of a cardinal point in IFC. The correct geometry can be generated in IFC accounting for a cardinal point, however, there is no IFC2x3 entity to specify what the cardinal point is for design intent.

1.5 Design Model

A CIS/2 design model is characterized by individual parts, defined by the geometric shapes described in the previous section, and associates a position and orientation with them to place them in a steel structure. In practice, details such as assemblies, cutouts, connectivity, connection materials (clip angles, gusset plates), and miscellaneous materials (handrails, steps, floor grating) are not written to most CIS/2 design model files. Also in practice, parts in a design model are not grouped into assemblies. Fig. 3 shows a typical CIS/2 design model. Note the lack of connection material such as clip angles, gusset plate, or bolts and the overlap of beams, columns, and braces.

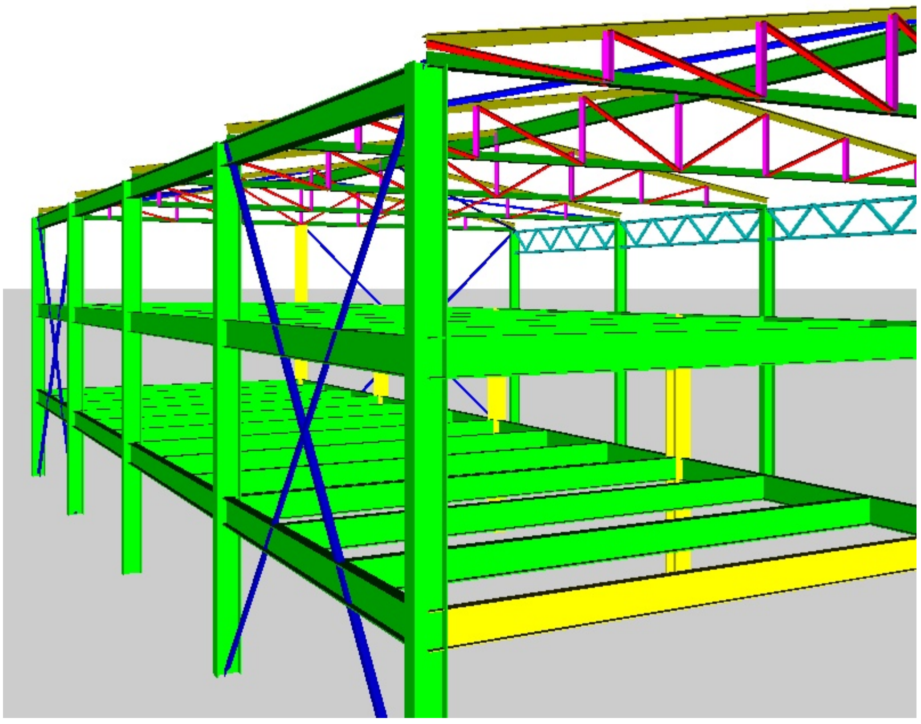


FIG. 3: Typical CIS/2 design model

The location of a design part is specified an origin and two direction vectors. The origin is given by a cartesian point. The orientation of a design part is given by direction vectors that define the longitudinal axis of the part and the orientation of the vertical axis of the part. There are many other features of a CIS/2 design model such as: the type of part (beam, column, brace, cable, cambered, etc.); the type of brace (horizontal, vertical, diagonal, etc.); and internal or external connections. The connections only imply that two or more parts are connected and do not necessarily indicate how they are connected. There can also be mapping between parts in a design model and elements in a structural analysis model (section 1.7). For example, a beam that is subdivided into smaller segments for analysis purposes is physically only one part in the design model.

Detailed examples of the mapping between the CIS/2 design model and IFC are in Appendix B. The mapping for the design model shows how entities such as Ifc{Beam/Column} are used for the design parts in the CIS/2 model. Because entities such as Ifc{Beam/Column/Member/Plate/Railing} have semantic meaning such as their orientation and function, their use has to be derived from the CIS/2 file. The use of Ifc{Beam/Column/Member} is determined by the orientation of the longitudinal axis of the CIS/2 design part. IfcMember can be used for parts with any orientation such as braces, but also beams and columns. IfcPlate is used depending on the shape of the part and IfcRailing is used when there is information in the CIS/2 indicating the function of the part. Other features of the CIS/2 design model that capture the design intent of the structure generally do not have an IFC equivalent.

1.6 Detailed Model

A detailed model is also known as a physical, manufacturing, or fabrication model. Detailed models are characterized by parts grouped into assemblies that make up the structure. In CIS/2 practice, an assembly is comprised of one main member such as a beam, column, or brace and associated connection material such as clip angles, gusset plates, bolts, holes, and welds. Items such as handrails, stairs, floor gratings, ladders, and surface treatments can be in a detailed model. Parts in a detailed model can have cutouts (copes) such as miter cuts, notches, and chamfers. Cutouts are defined parametrically by their length, width, depth, angle, and the location on the part where they are applied such as the start or end face, left or right side, or the top or bottom.

Fig. 4 shows a typical detailed model and a close-up of a connection. The structure is a braced frame with horizontal and vertical bracing and corrugated decking on the top of the structure. The connections contain bolts, holes, and welds, however, the welds, holes, nuts, and washers are not visible in the figure.

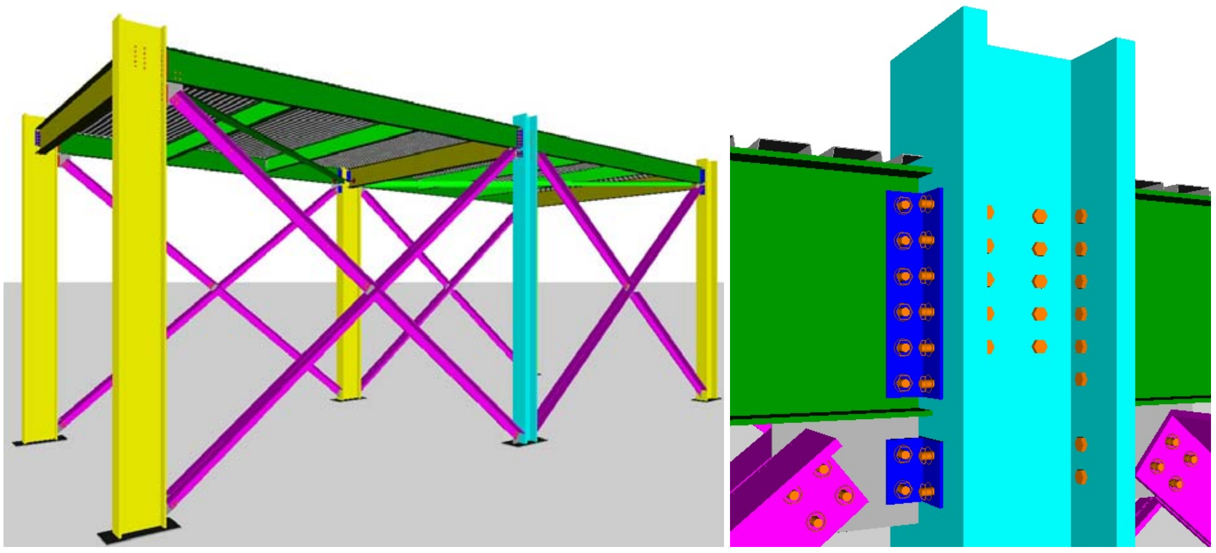


FIG. 4: Typical CIS/2 detailed model and close-up of a connection

In practice, parts in an assembly are located relative to the origin of the assembly while the assembly is located relative to the origin of the structure. Thus all parts in a detailed CIS/2 model are located relative to two nested coordinate systems. However, CIS/2 does allow for an assembly to be located relative to another assembly.

Individual bolts and holes are located in a pattern or layout. For example, in Fig. 4, each brace shown in magenta has a 2x2 pattern of bolts and holes on each end. Hole patterns for a part are located relative to the origin of a part and bolt patterns are located relative to the origin of an assembly. Welds can be defined by a weld path and characteristics such as its dimensions and type.

Detailed examples of the mapping between the CIS/2 detailed model and IFC are in Appendix C. The geometry of the parts of detailed CIS/2 models can be represented in a manner similar to design models. However, there are many deficiencies in IFC2x3 that are needed for the efficient modeling of detailed models and to capture the design intent of those features. For example, the geometry of a bolt can be modeled; however, there is no method to specify a pattern of bolts. Each bolt has to be located individually in IFC. The geometry of features such as holes and cutouts that remove material from parts can be modeled in IFC as boolean operations. The boolean operations only facilitate modeling the geometry correctly; however, there are no IFC2x3 entities to describe the parametric dimensions of cutouts or the layout of holes. Parts can be grouped together in assemblies in IFC, however, some of the meaning that parts are located relative to assemblies is lost when mapping from CIS/2 to IFC.

Similar to a design model, several different rules can be applied to determine if IFC detailed parts should be Ifc{Beam/Column/Member/Plate}. Certainly IfcMember can be used for all parts except plates which can use IfcPlate. Ifc{Beam/Column} can be assigned depending on the orientation of the part or assembly. If Ifc{Beam/Column} is assigned based on the orientation of parts, then the dark blue angles in Fig. 4 would be IfcColumn. However, if Ifc{Beam/Column} is assigned based on the orientation of assemblies, then the dark blue angles in Fig. 4 would be IfcBeam because they are part of the assembly with the green beam. CIS/2 can also identify the main part in an assembly which is usually a beam or column. The non-main parts are typically connection material such as clip angles and gusset plates. Therefore IfcMember could be used for all non-main parts in an assembly and Ifc{Beam/Column} for all the main parts.

1.7 Structural Analysis Model

The basic features of a structural analysis model are nodes, elements (linear, planar, and solid) that connect the nodes, loads on elements and nodes, and reactions due to the loading. A structural analysis model is sometimes referred to as a finite element model, wireframe model, or stick model. Fig. 5 shows a typical CIS/2 analysis model. On the left are the wireframe analysis elements and nodes and on the right is the physical representation of the elements.

Detailed examples of the mapping between the CIS/2 structural analysis model and IFC are in Appendix D. The mapping between CIS/2 and IFC for the structural analysis model is straightforward. For most CIS/2 entities there is an equivalent IFC entity for the structural analysis model.

Currently, the IFC structural analysis model has not been widely implemented. The IFC examples in Appendix D are based on a thorough interpretation of the IFC specification. However, the IFC structural analysis examples have not been generally tested by importing them into applications that supports the IFC structural analysis model. The IFC examples have been confirmed that they conform to the IFC schema. Currently, there are also discussions amongst the IFC software developers and model developers to improve some aspects of the IFC structural analysis model for the next version of the IFC specifications.

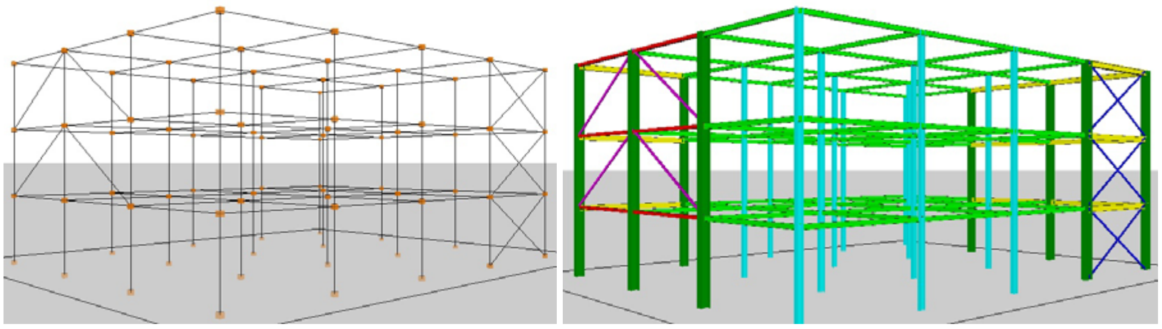


FIG. 5: Typical CIS/2 analysis model (left) and corresponding physical representation (right)

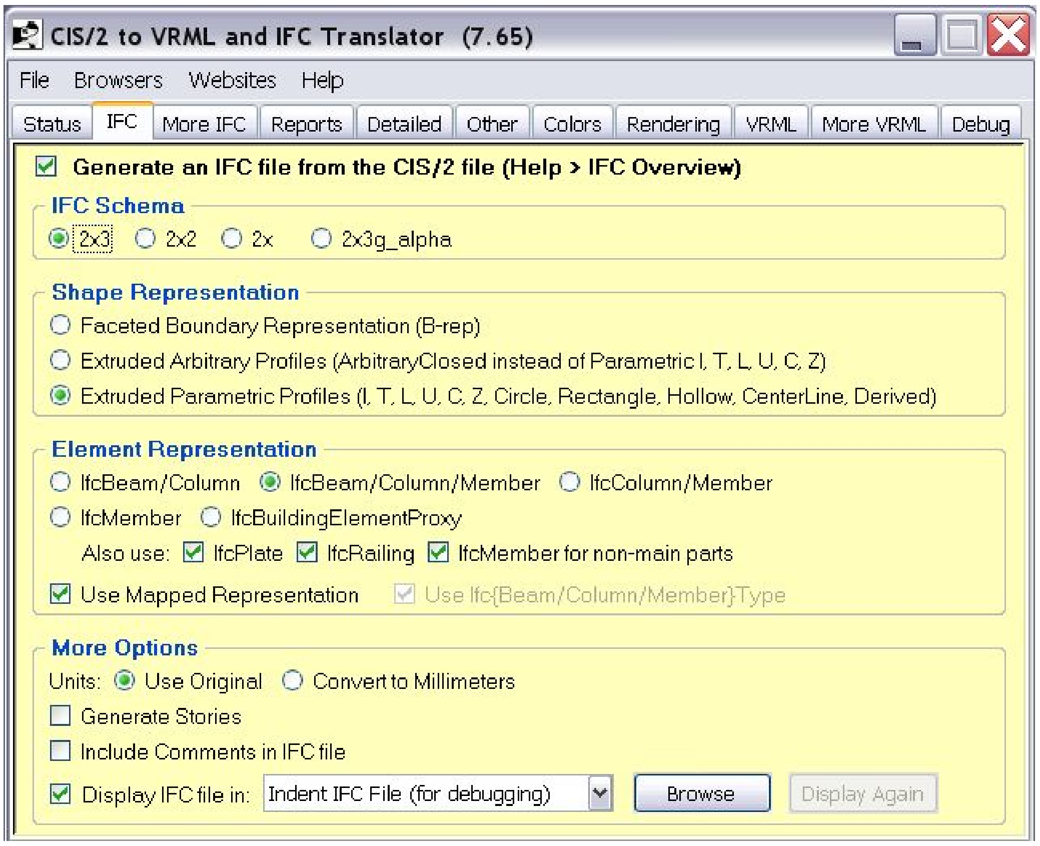
1.8 Other Model Concepts

Detailed examples of the mapping between the CIS/2 and IFC for some other concepts are in Appendix E. Some of those concepts are common to all CIS/2 and IFC models such as units, globally unique identifiers, and material, section, and generic properties. Other CIS/2 concepts such as surface treatments, grid planes and levels, camber, and document references do not necessarily have an IFC equivalent.

1.9 Mapping Implementation

The CIS/2 to IFC mapping has been implemented as a free software program available from NIST at http://cic.nist.gov/vrml/cis2.html. The translator between CIS/2 and IFC was developed as part of the existing CIS/2 to VRML (Virtual Reality Modeling Language) translator (Lipman and Reed, 2003). The translator does not use any software development toolkit to facilitate parsing the CIS/2 files and generating the IFC files based on their associated product model schemas. The translator simply parses the CIS/2 file line-by-line and generates some IFC entities as the CIS/2 file is being read. The bulk of the IFC entities are generated from CIS/2 information that has been stored by the translator in various data structures.

Fig. 6 and Fig. 7 show the user interface from the translator and the options available for generating IFC files. These options allow the generation of IFC files with many of the different representations described in this paper. The CIS/2 to IFC translator has been tested with over 500 different CIS/2 files for most combinations of shape and element representations, using mapped and non-mapped representation, and for the IFC2x3, IFC2x2, and IFC2x schemas. All of the IFC files that are generated by the translator were tested that they conform to the schema. In general, no unexpected errors and warnings resulted from the conformance checking of the IFC files. Some of the IFC files were also tested in a variety of IFC viewers, model checkers, and CAD applications that import IFC files. This served to show the capabilities and limitations of those programs.



[✓ Generate an IFC file from the CIS/2 file (Help > IFC Overview)]

IFC Schema

[✓ 2x3] 2x2 2x 2x3g_alpha

Shape Representation

Faceted Boundary Representation (B-rep)
Extruded Arbitrary Profiles (ArbitraryClosed instead of Parametric I, T, L, U, C, Z)
[✓ Extruded Parametric Profiles (I, T, L, U, C, Z, Circle, Rectangle, Hollow, CenterLine, Derived)]

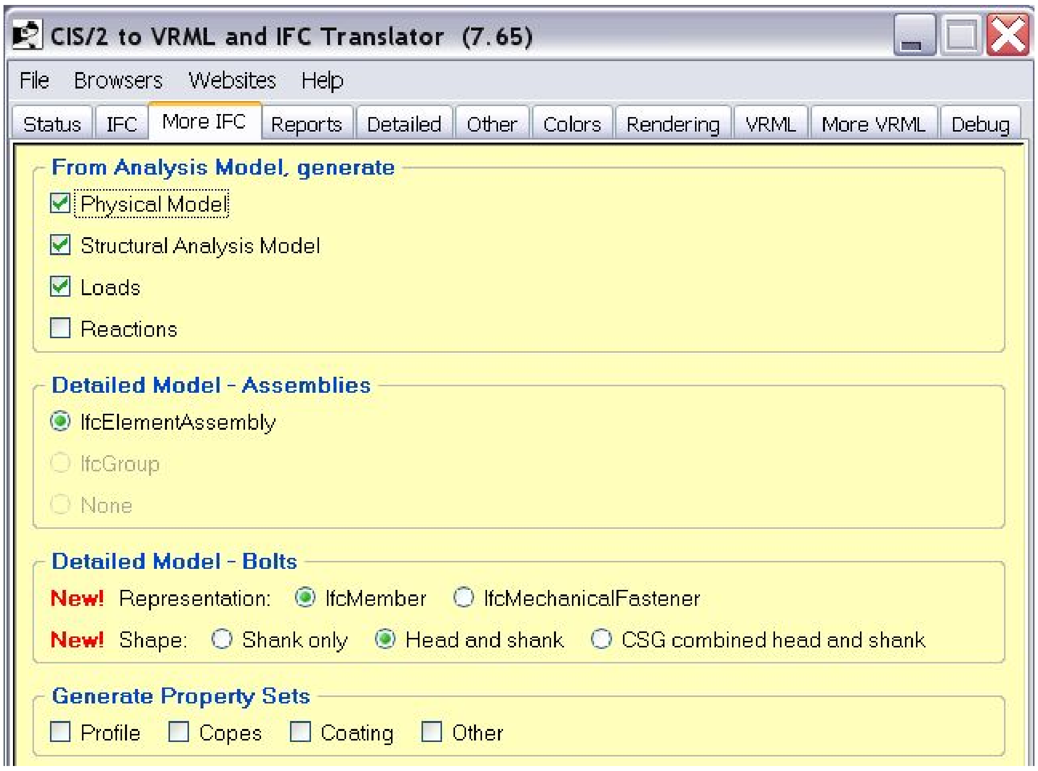
Element Representation

IfcBeam/Column [✓ IfcBeam/Column/Member] IfcColumn/Member
IfcMember IfcBuildingElementProxy
Also use: [✓ IfcPlate] [✓ IfcRailing] [✓ IfcMember for non-main parts]
[✓ Use Mapped Representation] [✓ Use Ifc{Beam/Column/Member}Type]

More Options

Units: [✓ Use Original] Convert to Millimeters
Generate Stories
Include Comments in IFC file
[✓ Display IFC file in:] Indent IFC File (for debugging)

FIG. 6: CIS/2 to IFC translator options


CIS/2 to VRML and IFC Translator (7.65)

File Browsers Websites Help

Status | IFC | More IFC | Reports | Detailed | Other | Colors | Rendering | VRML | More VRML | Debug

From Analysis Model, generate
- ✓ Physical Model
- ✓ Structural Analysis Model
- ✓ Loads
- ☐ Reactions

Detailed Model - Assemblies
- ● IfcElementAssembly
- ○ IfcGroup
- ○ None

Detailed Model - Bolts
- New! Representation: ● IfcMember ○ IfcMechanicalFastener
- New! Shape: ○ Shank only ● Head and shank ○ CSG combined head and shank

Generate Property Sets
- ☐ Profile ☐ Copes ☐ Coating ☐ Other

FIG. 7: More CIS/2 to IFC translator options

2. RECOMMENDATIONS

In the course of developing the mapping between CIS/2 and IFC, many insights have emerged into how structural steel should be modeled in IFC. The recommendations fall into two categories: (1) how the existing IFC specifications should be implemented by IFC applications and (2) what improvements to the IFC specifications are needed to model structural steel. Some of the recommendations are very specific relating to particular text strings while other recommendations are more general. With an increased focus on building information models that can be exchanged via IFC it is important to consider these recommendations so that the end-user can have a seamless and accurate exchange of all structural steel model information.

2.1 Implementing the existing IFC specification

- An agreement is needed to specify which text fields are used for steel information such as the section designator (i.e. W14x89) and the part and assembly piecemarks. The section designator would normally go on the entity that defines the section profile. However, if a boundary representation is used for the geometry, then there is not necessarily a section profile definition. In this case, the section designator, along with the piecemark, could go on Ifc{Beam/Column/Member/ElementAssembly}.

- The IFC applications should support several entities that are necessary to maintain the parametric definition (depth, width, thickness, curve, profile) of structural steel. The parametric profiles are Ifc{I/T/L/U/C/Z}ShapeProfileDef. IfcDerivedProfileDef is necessary to mirror non-symmetric parametric profiles. IfcCompositeProfileDef is useful for composite sections such as double angles and channels. IfcCenterLineProfileDef is necessary for bent plates and corrugated decking. IfcSurfaceCurveSweptAreaSolid is necessary for extruded curved parts. IFC applications should export structural steel using the parametric profiles as opposed to using a boundary representation. Applications should also import the parametric profiles, retain the parametric definition in their internal representation, have the parametric definition be accessible to the user, and be able to re-export the parametric definition.

- The IFC applications should support as needed: (1) IfcElementAssembly for element assemblies and preserve the hierarchy between parts and assemblies, (2) IfcFastener and IfcMechanicalFastener for bolted or welded connections, (3) objects comprising multiple shapes such a bolt with separate geometry for the head and shank, and (4) the structural analysis model.

- An agreement is needed to specify what type of Ifc{Beam/Column/Member} should be used. For example, is a vertically oriented clip angle in a horizontal beam assembly an IfcBeam or an IfcColumn? Is a horizontal brace an IfcBeam or an IfcMember? IfcBeam and IfcColumn are defined in architectural terms as members that are essentially horizontal and vertical, respectively. The orientation of an IfcMember is not relevant to its definition. The use of Ifc{Beam/Column/Member} differs between different CAD applications and whether it is an architectural or structural view.

- An agreement would be useful to indicate which type of structural steel model (design, analysis, detailed) is in an IFC file. This would be particularly useful for detailed and design models which appear very similar in an IFC file.

- There needs to be more implementations of the structural analysis model in CAD applications. The structural analysis model has not been implemented or tested. In developing the CIS/2 to IFC mapping some of the limitations of the IFC structural analysis model became apparent. A better IFC structural analysis model will come about only after software vendors start trying to implement it and end-users start to exchange structural analysis models via IFC.

2.2 Improvements to the IFC specification

Some of the recommendations below are already planned for the next version of IFC, IFC2x4.

- A cardinal point should be able to be associated with a section profile and generate the section with the correct cardinal point offset.

- As seen from some of the examples in the report, some structural steel information can only be specified in IFC with generic property sets. More structural steel information should be predefined with existing property sets, such as PSet_BeamCommon and others, with new property sets, or with specific attributes in other entities.

- The specification needs to have a better implementation of structural steel connections, particularly for specifying layouts or patterns of bolts and holes.

- The specification needs to have parametric definitions of cutouts rather than just being able to create the geometry of a part that has a cutout. At least there should be a specific property set with the parametric definition of a cutout.

- The structural analysis model needs a method to associate analysis results with element connectivity rather than just analysis nodes. This will allow results to be associated with the ends of an analysis element.

3. CONCLUSION

The conceptual scopes of the CIS/2 and IFC specifications overlap sufficiently to allow mapping of information from a CIS/2 (LPM6) representation to an IFC2x3 representation. For basic geometric resources such as coordinate systems, cartesian points, and others there is a one-to-one mapping. There is a one-to-many mapping of geometric shape representations between CIS/2 and IFC because of the multiple methods to explicitly represent geometry in IFC. The basic geometry of CIS/2 design, analysis, and detailed models can be represented in IFC. However, there are areas where the modeling of specific features such as bolts, holes, and welds needs to be improved in the IFC specification. There are also cases where the semantic meaning of a concept in CIS/2, such as detailed model assemblies, is not as strong as in IFC. The lack of robust implementations of the IFC structural analysis model makes it hard to evaluate the details of the mapping from the CIS/2 analysis model.

The mapping examples provide a basis for developing a more rigorous and formal computer science based mapping between CIS/2 and IFC. The examples also provide insight into a possible IFC to CIS/2 mapping. A significant issue for an IFC to CIS/2 mapping is how to determine which members in an IFC model are structural steel as opposed to other parts of a building such as walls, floors, doors, and windows. Other issues include: mapping from a boundary representation of structural member to a parametric representation; and determining what type of CIS/2 model, either design, analysis, or detailing, the IFC model should be mapped to.

4. REFERENCES

Amor, R.W. and Ge, C.W. (2002). Mapping IFC Versions, *Proceedings of the EC-PPM Conference on eWork and eBusiness in AEC*, Portoroz, Slovenia, 9-11 September 2002, pp. 373-377. http://www.cs.auckland.ac.nz/~trebor/papers/AMOR02A.pdf

Arthaud G. and Lombardo J-C. (2006). Automatic Semantic Comparison of STEP Model Instances, Application to IFC Models, *Proceedings of Design and Decision Support Systems Conference 2006*, Heeze, The Netherlands. http://www.springerlink.com/content/rr27752197800v81/

Crowley A.. and Watson A. (2000). CIMsteel Integration Standards Release 2, SCI-P-268, The Steel Construction Institute, Berkshire, England. http://www.steelbiz.org/Discovery/AllResults.aspx?q=%22P268%22

Eastman C., Wang F., You S.-J., and Yang D. (2005). Deployment of an AEC industry sector product model, *Computer-Aided Design*, Vol. 37-12, October 2005, 1214-1228.

Eastman C. (2004). Harmonization of CIS/2 and IFC, Georgia Institute of Technology. http://www.coa.gatech.edu/~aisc/cisifc/

FIATECH (2004). Automating Equipment Information Exchange (AEX). http://www.fiatech.org/projects/idim/aex.htm

Khemlani L. (2007). BIM Fundemantals Seminar for Structural Engineers http://www.aecbytes.com/buildingthefuture/2007/BIMFundamentalsSeminar.html

Liebich T., Adachi Y., Forester J., Hyvarinen J., Karstila K., Wix J. (2006). IFC 2x Edition 3. International Alliance for Interoperability. http://www.iai-tech.org/.

Lipman R., and Reed K.A., (2003). Visualization of Structural Steel Product Models, *Journal of Information Technology in Construction*, Vol. 8, 51-64, http://www.itcon.org/2003/5

Lipman R. (2006). Mapping Between the CIMsteel Integration Standards and Industry Foundation Classes Product Data Models for Structural Steel, *Proceedings of the International Conference on Computing in Civil and Building Engineering*, Montreal, Canada, June 2006, http://www.fire.nist.gov/bfrlpubs/build06/art025.html

Ma H., Ha K.M.E., Chung C.K., Amor R. (2006). Testing Semantic Interoperability, *Proceedings of the International Conference on Computing in Civil and Building Engineering*, Montreal, Canada, June 2006.

Pan J., Cheng C.-P., Lau G., Law K. (2008). Utilizing Statistical Semantic Similarity Techniques for Ontology Mapping – with Application to AEC Standard Model, *Proceedings of the 12th International Conference on Computing in Civil and Building Engineering (ICCCBE XII)*, Beijing, China, October 16-18, 2008, http://eil.stanford.edu/publications/jack_cheng/icccbe08_ontology_mapping.pdf

Wang H., Akinci B., Garrett J. (2006). A Semi-Automated Schema Matching Approach Based on Automated Detection of Version Differences, *Proceedings of the International Conference on Computing in Civil and Building Engineering*, Montreal, Canada, June 2006.

Wang H., Akinci B., Garrett J. (2007). Formalism for Detecting Version Differences in Data Models, *ASCE Journal of Computing in Civil Engineering*, Vol. 21, No. 5, September 2007.

Wang H., Akinci B., Garrett J., Reed K. (2008). Formalism for Applying Domain Constraints in Domain-Oriented Schema Matching, *ASCE Journal of Computing in Civil Engineering*, Vol. 22, No. 3, May 2008.

APPENDIX A - SHAPE REPRESENTATION EXAMPLES

The examples in Appendix A show how geometry related to parts in CIS/2 is mapped to shape representations in IFC. A discussion about geometry and shape representation is in section 1.4. The geometry examples are independent of how it is used in design, analysis, or detailed models. Its use in those models is shown in subsequent appendices.

Each example or concept contains at least two figures. In most cases, the first figure is the CIS/2 example and the subsequent figures are the corresponding IFC examples.

In the examples all of the IFC samples were generated directly from CIS/2 files with the CIS/2 to IFC translator described in section 1.9. In the examples the order of the entity instances has been changed from the original file and indentation added to show the hierarchy and relationship between the various entities. If possible, the translator uses the same entity IDs from the CIS/2 file in the IFC file to maintain the correspondence between equivalent entities.

For many of the examples, the relevant CIS/2 and IFC entities for that example have been highlighted. Some repetitive and common CIS/2 and IFC entities, such as Dimensional_exponents, IfcOwnerHistory, and IfcGeometricRepresentationContext, which are not important to the examples, are not shown to save space. One of the attributes on many IFC entities is a globally unique identifier known as a GUID. The GUID is a 22 character string such as '0bY52t0r7xHf8OxWOsY$t_'. In the examples, the GUID is shortened to the string ‘guid’ to save space and improve readability.

There are many text string attributes on the CIS/2 and IFC entities. Some text strings have specific rules about what can be used. However, there are many optional text strings that can be used to identify the specific types of information, such as a steel piecemark or section designator. Recommended practices and software vendor implementer’s agreements have defined what should be used for some of the optional strings, however, in practice there are widely different implementations of what should go in those strings, particularly with IFC.

A.1 Prismatic Parts

Fig. 8a is an example of an angle section that is mirrored (.T.) and has a cardinal point value of ‘1’. The cardinal point defines how the cross section is positioned relative to the longitudinal axis of the part. A value of ‘1’ means that the cardinal point is located in the lower left corner of the angle cross section. Other cardinal points are at the center, corners, and mid-points of the sides of a bounding box encompassing the cross section.

```
#1023=PART_PRISMATIC_SIMPLE(1,'A36',$,$,.UNDEFINED.,$,#1025,#1006,$,$);
  #1025=SECTION_PROFILE(101,'L10X6X1/2',$,$,1,.T.);
  #1006=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(36.),#1002);
    #1002=(CONTEXT_DEPENDENT_UNIT('INCH')LENGTH_UNIT()NAMED_UNIT(#1003));
      #1003=DIMENSIONAL_EXPONENTS(1.,0.,0.,0.,0.,0.,0.);
```

FIG. 8a: CIS/2 prismatic part – implied section dimensions, cardinal point, mirrored

In IFC, the cross section of a prismatic part can be specified by parametric profiles such as Ifc{I/T/L/U/C/Z}ShapeProfileDef, Ifc{Rectangle/Circle}ProfileDef, IfcCenterLineProfileDef, and others. Some of the parameters for those entities are the depth, width, and thickness of the cross section. The length of the prismatic part is specified on IfcExtrudedAreaSolid. Fig. 8b is an example of an angle section.

There is no equivalent IFC entity or attribute to specify a cardinal point, however, the offset of the cross section defined by the CIS/2 cardinal point is specified explicitly with the IfcAxis2Placement2d (#3051). The value of the offset is half the depth and width of the angle cross section. Although the CIS/2 cross section is mirrored, the IFC cross section is not mirrored. Non-symmetric parametric profiles in IFC are mirrored from their CIS/2 non-mirrored equivalents.

```
#3142= IFCPRODUCTDEFINITIONSHAPE($,$,(#3143));
  #3143= IFCSHAPEREPRESENTATION(#60011,'Body','SweptSolid',(#3144));
    #3144= IFCEXTRUDEDAREASOLID(#1025,#60049,#60044,36.);
      #1025= IFCLSHAPEPROFILEDEF(.AREA.,'L10X6X1/2',#3051,10.,6.,0.5,$,$,$,$,$);
        #3051= IFCAXIS2PLACEMENT2D(#3050,#60052);
          #3050= IFCCARTESIANPOINT((3.0,5.0));
          #60052= IFCDIRECTION((1.,0.));
      #60049= IFCAXIS2PLACEMENT3D(#60041,#60042,#60043);
        #60041= IFCCARTESIANPOINT((0.,0.,0.));
        #60042= IFCDIRECTION((1.,0.,0.));
        #60043= IFCDIRECTION((0.,1.,0.));
      #60044= IFCDIRECTION((0.,0.,1.));
```

FIG. 8b: IFC part – extruded parametric profile

In IFC, the cross section of a prismatic part can be also be specified by an arbitrary cross section with IfcArbitraryClosedProfileDef. The outline of the section profile is defined by a closed curve. Fig. 8c is an example of the same angle section using IfcArbitraryClosedProfileDef where an IfcPolyline defines the curve.

```
#3233= IFCPRODUCTDEFINITIONSHAPE($,$,(#3234));
  #3234= IFCSHAPEREPRESENTATION(#60011,'Body','SweptSolid',(#3235));
    #3235= IFCEXTRUDEDAREASOLID(#1025,#60049,#60044,36.);
      #1025= IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,'L10X6X1/2',#3126);
        #3126= IFCPOLYLINE((#3119,#3120,#3121,#3122,#3123,#3124,#3119));
          #3119= IFCCARTESIANPOINT((0.0,10.0));
          #3120= IFCCARTESIANPOINT((0.5,10.0));
          #3121= IFCCARTESIANPOINT((0.5,0.5));
          #3122= IFCCARTESIANPOINT((6.0,0.5));
          #3123= IFCCARTESIANPOINT((6.0,0.0));
          #3124= IFCCARTESIANPOINT((0.0,0.0));
          #3119= IFCCARTESIANPOINT((0.0,10.0));
      #60049= IFCAXIS2PLACEMENT3D(#60041,#60042,#60043);
      #60044= IFCDIRECTION((0.,0.,1.));
```

FIG. 8c: IFC part – extruded arbitrary closed profile

In IFC, any part can also be specified by a faceted boundary representation also known as a B-rep. The B-rep is constructed from points, faces that connect the points, and faces that make up the boundary. Fig. 8d shows a partial example, excluding some of the faces, of the boundary representation of the angle section from above.

```
#4365= IFCPRODUCTDEFINITIONSHAPE($,$,(#4366));
  #4366= IFCSHAPEREPRESENTATION(#60011,'Body','Brep',(#4368));
    #4368= IFCFACETEDBREP(#4422);
      #4422= IFCCLOSEDSHELL((#4386,#4389,#4392,#4395,#4398,#4401,#4404,#4407));
        #4386= IFCFACE((#4387));
          #4387= IFCFACEOUTERBOUND(#4388,.T.);
            #4388= IFCPOLYLOOP((#4369,#4370,#4371,#4372));
              #4369= IFCCARTESIANPOINT((0.,0.,10.));
              #4370= IFCCARTESIANPOINT((0.,0.5,10.));
              #4371= IFCCARTESIANPOINT((0.,0.5,0.5));
              #4372= IFCCARTESIANPOINT((0.,0.,0.5));
```

FIG. 8d: IFC part – faceted boundary representation (B-rep)

Fig. 9a shows the same CIS/2 example as in Fig. 8a except that the section profile is not mirrored (.F.).

```
#1023=PART_PRISMATIC_SIMPLE(1,'A36',$,$,.UNDEFINED.,$,#1025,#1006,$,$);
  #1025=SECTION_PROFILE(101,'L10X6X1/2',$,$,1,.F.);
  #1006=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(36.),#1002);
```

FIG. 9a: CIS/2 part – implied section dimensions, cardinal point, not mirrored

If the CIS/2 section is not mirrored, then the IFC section has to be mirrored. Mirrored sections are modeled with IfcDerivedProfileDef and IfcCartesianTransformationOperator2D for parametric profiles. The IfcDirection referred to by IfcCartesianTransformationOperator2D indicates how the section is mirrored. If an extruded IfcArbitraryClosedProfileDef or a B-rep is used for the shape representation, then the mirrored section can be modeled explicitly and IfcDerivedProfileDef is not necessary. Fig. 9b shows an example of a mirrored angle section.

```
#3106= IFCPRODUCTDEFINITIONSHAPE($,$,(#3107));
  #3107= IFCSHAPEREPRESENTATION(#60011,'Body','SweptSolid',(#3108));
    #3108= IFCEXTRUDEDAREASOLID(#425,#60049,#60044,36.);
      #1025= IFCDERIVEDPROFILEDEF(.AREA.,'L10X6X1/2',#3024,#3025, 'Mirror Y axis');
        #3024= IFCLSHAPEPROFILEDEF(.AREA.,'L10X6X1/2',#3023,10.,6.,0.5,$,$,$,$,$);
          #3023= IFCAXIS2PLACEMENT2D(#3022,#60052);
            #3022= IFCCARTESIANPOINT((3.0,5.0));
            #60052= IFCDIRECTION((1.,0.));
        #3025= IFCCARTESIANTRANSFORMATIONOPERATOR2D(#60054,$,#60051,$);
          #60054= IFCDIRECTION((-1.,0.));
          #60051= IFCCARTESIANPOINT((0.,0.));
      #60049= IFCAXIS2PLACEMENT3D(#60041,#60042,#60043);
      #60044= IFCDIRECTION((0.,0.,1.));
```

FIG. 9b: IFC part – extruded parametric profile, mirrored

A.2 Plates

In CIS/2, an arbitrarily shaped flat plate is defined by Part_sheet_bounded_complex which refers to a bounding curve that defines the shape of the plate. The bounding curve can be defined by a polyline or set of curve segments. The 3-dimensional points defining the polyline are in the YZ plane. Fig. 10a shows a plate whose shape is defined by a polyline with seven points.

```
#40=PART_SHEET_BOUNDED_COMPLEX(1,’Plate’,$,$,.ROLLED.,$,#4029,#4019);
  #4029=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(0.5),#53);
  #4019=CURVE_BOUNDED_SURFACE('P1',#4027,(#4021),.F.);
    #4027=PLANE('Plane',#4048);
      #4048=AXIS2_PLACEMENT_3D('Axis3d',#4062,#4054,#4055);
        #4062=CARTESIAN_POINT('Origin',(0.,0.,0.))
        #4054=DIRECTION('Direction',(0.,0.,1.));
        #4055=DIRECTION('Direction',(1.,0.,0.));
    #4021=BOUNDARY_CURVE('Boundary Curve',(#4023),.F.);
      #4023=COMPOSITE_CURVE_SEGMENT(.CONTINUOUS.,.T.,#4025);
        #4025=BOUNDED_SURFACE_CURVE('Surface Curve',#4039,(#4027),.CURVE_3D.);
          #4039=POLYLINE('PolyLine',(#4062,#40114,#40115,#40116,#40117,#40118,#40119));
            #4062=CARTESIAN_POINT('Origin',(0.,0.,0.));
            #40114=CARTESIAN_POINT('Point',(0.,0.,5.73036063577701));
            #40115=CARTESIAN_POINT('Point',(0.,7.67762750484352,9.36095759948835));
            #40116=CARTESIAN_POINT('Point',(0.,7.67762750484352,6.69109049467188));
            #40117=CARTESIAN_POINT('Point',(0.,13.470159914151,6.69109049467188));
            #40118=CARTESIAN_POINT('Point',(0.,13.6167563265736,-0.464185494476179));
            #40119=CARTESIAN_POINT('Point',(0.,4.17372019285461,-3.75494051077219));
          #4027=PLANE('Plane',#4048);
            #4048=AXIS2_PLACEMENT_3D('Axis3d',#4062,#4054,#4055);
              #4062=CARTESIAN_POINT('Origin',(0.,0.,0.));
              #4054=DIRECTION('Direction',(0.,0.,1.));
              #4055=DIRECTION('Direction',(1.,0.,0.));
```

FIG. 10a: CIS/2 plate

In IFC, an arbitrarily shaped flat plate is defined by IfcArbitraryClosedProfileDef in Fig. 10b similar to how other arbitrary closed profiles are defined above in Fig. 8c. The profile is defined by an IfcPolyline. The points defining the polyline are 2-dimensional. Since the profile must be closed the last point in the polyline is the same as the first point.

```
#41358= IFCPRODUCTDEFINITIONSHAPE($,$,(#41359));
  #41359= IFCSHAPEREPRESENTATION(#820011,'Body','SweptSolid',(#41360));
    #41360= IFCEXTRUDEDAREASOLID(#4019,#41361,#820044,0.5);
      #4019= IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,'Plate (Complex)',#41268);
        #41268= IFCPOLYLINE((#41260,#41261,#41262,#41263,#41264,#41265,#41266,#41260));
          #41260= IFCCARTESIANPOINT((0.0,0.));
          #41261= IFCCARTESIANPOINT((0.0,5.730361));
          #41262= IFCCARTESIANPOINT((7.677628,9.360958));
          #41263= IFCCARTESIANPOINT((7.677628,6.69109));
          #41264= IFCCARTESIANPOINT((13.47016,6.69109));
          #41265= IFCCARTESIANPOINT((13.616756,-0.464185));
          #41266= IFCCARTESIANPOINT((4.17372,-3.754941));
      #41361= IFCAXIS2PLACEMENT3D(#41362,#820042,#820043);
        #41362= IFCCARTESIANPOINT((-0.25,0.,0.));
        #820042= IFCDIRECTION((1.,0.,0.));
        #820043= IFCDIRECTION((0.,1.,0.));
      #820044= IFCDIRECTION((0.,0.,1.));
```

FIG. 7b: IFC plate

A.3 Bent and Corrugated Parts

In CIS/2, a bent plate is modeled with Section_profile_centreline where the bend of the plate is defined by a curve, usually a Polyline as in Fig. 11a. The polyline points are 3-dimensional.

```
#37=PART_PRISMATIC_SIMPLE(22,'A36',$,$,.UNDEFINED.,$,#160,#4000,$,$);
  #160=SECTION_PROFILE_CENTRELINE(0,'BPL1x11 3/16','ASTM specification A6',
                                'Bent plate',8,.F.,#10019,#10015);
    #10019=COMPOSITE_CURVE('bent plate bend Path',(#10018),.F.);
      #10018=COMPOSITE_CURVE_SEGMENT(.CONTINUOUS.,.T.,#10017);
        #10017=POLYLINE('bent plate',(#10028,#10029,#10030,#10031,#10032,#10033));
          #10028=CARTESIAN_POINT('bent plate bend point',(0.,0.,0.));
          #10029=CARTESIAN_POINT('bent plate bend point',(0.,4.,0.));
          #10030=CARTESIAN_POINT('bent plate bend point',(0.,4.388,-0.05111));
          #10031=CARTESIAN_POINT('bent plate bend point',(0.,4.75,-0.2));
          #10032=CARTESIAN_POINT('bent plate bend point',(0.,5.06,-0.43933));
          #10033=CARTESIAN_POINT('bent plate bend point',(0.,9.3033,-4.68198));
    #10015=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(0.5),#53);
  #4000=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(36.),#53);
```

FIG. 11a: CIS/2 bent plate

In CIS/2, corrugated decking is modeled with Part_sheet_profiled where the profile of the decking is defined by a curve, usually a Polyline as in Fig. 11b. The polyline points are 3-dimensional. The profile is defined as the minimum curve that is needed so that when reproduced along the length of the part, creates the entire corrugation profile.

```
#38=(PART(.UNDEFINED.,'Corrugated decking')PART_SHEET(#3827)PART_SHEET_BOUNDED()
      PART_SHEET_BOUNDED_SIMPLE(#3828,#3826,$,$,$,$)PART_SHEET_PROFILED(#3814,$);
  #3827=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(0.125),#53);
  #3828=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(36.),#53);
  #3826=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(36.),#53);
  #3814=POLYLINE('decking profile',(#3841,#3842,#3843,#3844,#3845));
    #3841=CARTESIAN_POINT('pt1',(-0.75,0.,0.));
    #3842=CARTESIAN_POINT('pt2',(-0.75,1.5,0.));
    #3843=CARTESIAN_POINT('pt3',(0.75,1.75,0.));
    #3844=CARTESIAN_POINT('pt4',(0.75,5.75,0.));
    #3845=CARTESIAN_POINT('pt5',(-0.75,6.,0.));
```

FIG. 11b: CIS/2 corrugated decking

In IFC, parts such as bent plates or corrugated decking are modeled with IfcCenterLineProfileDef where the bend of the plate is defined by a curve, usually an IfcPolyline as in Fig. 11c. The polyline points are 2-dimensional. If corrugated decking is modeled, then the curve must reflect the complete profile over its entire length. There is no IFC equivalent of the CIS/2 Part_sheet_profiled entity. The IFC example also does not map the CIS/2 entities Composite_curve and Composite_curve_segment in Fig. 11a. In IFC, bent plates and corrugated decking can also be modeled as a B-rep or by extruding an IfcArbitraryClosedProfileDef along an appropriate path.

```
#41140= IFCPRODUCTDEFINITIONSHAPE($,$,(#41141));
  #41141= IFCSHAPEREPRESENTATION(#820011,'Body','SweptSolid',(#41142));
    #41142= IFCEXTRUDEDAREASOLID(#160,#820049,#820044,36.);
      #160= IFCCENTERLINEPROFILEDEF(.AREA.,'BPL1x11-3/16',#41139,0.5);
        #41139= IFCPOLYLINE((#41132,#41133,#41134,#41135,#41136,#41137));
          #41132= IFCCARTESIANPOINT((0.,0.));
          #41133= IFCCARTESIANPOINT((4.,0.));
          #41134= IFCCARTESIANPOINT((4.388,-0.05111));
          #41135= IFCCARTESIANPOINT((4.75,-0.2));
          #41136= IFCCARTESIANPOINT((5.06,-0.43933));
          #41137= IFCCARTESIANPOINT((9.3033,-4.68198));
      #820049= IFCAXIS2PLACEMENT3D(#820041,#820042,#820043);
      #820044= IFCDIRECTION((0.,0.,1.));
```

FIG. 11c: IFC bent plate

A.4 Curved Parts

In CIS/2, a curved part is modeled with Part_prismatic_simple_curved where the curve can be defined by a Polyline as in Fig. 12a. The polyline points are 3-dimensional.

```
#39=PART_PRISMATIC_SIMPLE_CURVED(1,’A36’,$,$,.UNDEFINED,$,#3925,#3924,$,$,#3936);
  #3925=SECTION_PROFILE(1,'W12X50',$,$,5,.F.);
  #3934=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(184.508341),#53);
  #3936=POLYLINE('PolyLine',(#3937,#3940,#3945,#3950,#3955,#3960,#3965,#3970,#3975,
               #3980,#3985,#3990,#3995,#39100,#39105,#39110,#39115,#39120,#39125));
    #3937=CARTESIAN_POINT('Point',(0.,0.,0.));
    #3940=CARTESIAN_POINT('Point',(2.3,0.55,0.));
    #3945=CARTESIAN_POINT('Point',(4.8,1.4,0.));
    #3950=CARTESIAN_POINT('Point',(7.825,2.525,0.));
    #3955=CARTESIAN_POINT('Point',(10.275,3.525,0.));
    #3960=CARTESIAN_POINT('Point',(12.925,4.675,0.));
    #3965=CARTESIAN_POINT('Point',(15.7,6.,0.));
    #3970=CARTESIAN_POINT('Point',(18.875,7.65,0.));
    #3975=CARTESIAN_POINT('Point',(20.725,8.7,0.));
    #3980=CARTESIAN_POINT('Point',(22.85,10.,0.));
    #3985=CARTESIAN_POINT('Point',(24.925,11.4,0.));
    #3990=CARTESIAN_POINT('Point',(27.225,13.05,0.));
    #3995=CARTESIAN_POINT('Point',(29.125,14.575,0.));
    #39100=CARTESIAN_POINT('Point',(30.874,16.125,0.));
    #39105=CARTESIAN_POINT('Point',(32.9,18.1,0.));
    #39110=CARTESIAN_POINT('Point',(34.4,19.775,0.));
    #39115=CARTESIAN_POINT('Point',(35.9,21.675,0.));
    #39120=CARTESIAN_POINT('Point',(36.95,23.225,0.));
    #39125=CARTESIAN_POINT('Point',(38.6,26.3,0.));
```

FIG. 12a: CIS/2 curved part

In IFC, a curved part is modeled with IfcSurfaceCurveSweptAreaSolid where the curve can be defined by an IfcPolyline as in Fig. 12b. The polyline points are 2-dimensional and lie in a plane defined by IfcPlane. The actual length of the part is defined by the length of the curve. In IFC, curved parts can also be modeled as a B-rep.

```
#41196= IFCPRODUCTDEFINITIONSHAPE($,$,(#41197));
  #41197= IFCSHAPEREPRESENTATION(#820011,'Body','SweptSolid',(#41198));
    #41198= IFCSURFACECURVESWEPTAREASOLID(#3925,#820040,#41195,0.,1.,#41199);
      #3925= IFCISHAPEPROFILEDEF(.AREA.,'W12X50',#820050,8.08,12.2,0.37,0.64,$);
      #820040= IFCAXIS2PLACEMENT3D(#820041,#820044,#820042);
        #820041= IFCCARTESIANPOINT((0.,0.,0.));
        #820044= IFCDIRECTION((0.,0.,1.));
        #820042= IFCDIRECTION((1.,0.,0.));
      #41195= IFCPOLYLINE((#41175,#41176,#41177,#41178,#41179,#41180,#41181,
                          #41182,#41183,#41184,#41185,#41186,#41187,#41188,
                          #41189,#41190,#41191,#41192,#41193));
        #41175= IFCCARTESIANPOINT((0.,0.));
        #41176= IFCCARTESIANPOINT((2.3,-0.55));
        #41177= IFCCARTESIANPOINT((4.8,-1.4));
        #41178= IFCCARTESIANPOINT((7.825,-2.525));
        #41179= IFCCARTESIANPOINT((10.275,-3.525));
        #41180= IFCCARTESIANPOINT((12.925,-4.675));
        #41181= IFCCARTESIANPOINT((15.7,-6.));
        #41182= IFCCARTESIANPOINT((18.875,-7.65));
        #41183= IFCCARTESIANPOINT((20.725,-8.7));
        #41184= IFCCARTESIANPOINT((22.85,-10.));
        #41185= IFCCARTESIANPOINT((24.925,-11.4));
        #41186= IFCCARTESIANPOINT((27.225,-13.05));
        #41187= IFCCARTESIANPOINT((29.125,-14.575));
        #41188= IFCCARTESIANPOINT((30.874,-16.125));
        #41189= IFCCARTESIANPOINT((32.9,-18.1));
        #41190= IFCCARTESIANPOINT((34.4,-19.775));
        #41191= IFCCARTESIANPOINT((35.9,-21.675));
        #41192= IFCCARTESIANPOINT((36.95,-23.225));
        #41193= IFCCARTESIANPOINT((38.6,-26.3));
      #41199= IFCPLANE(#41200);
        #41200= IFCAXIS2PLACEMENT3D(#820041,#820043,#820044);
          #820041= IFCCARTESIANPOINT((0.,0.,0.));
          #820043= IFCDIRECTION((0.,1.,0.));
          #820044= IFCDIRECTION((0.,0.,1.));
```

FIG. 12b: IFC curved part

A.5 Compound Parts

In CIS/2, Section_profile_compound is used to model compound sections such as double angles and channels or the individual pieces of a joist. Double sections are made of two identical cross sections that are back to back and separated by a small offset distance. Fig. 13a shows the CIS/2 for a double channel where Section_profile_compound refers to two channel sections, one of which is mirrored. The offset for each section (#100, #101) is relative to cardinal point ‘4’ (mid-depth left-side) of the channel. Neither of the cross sections is rotated.

```
#32=PART_PRISMATIC_SIMPLE(18,'A36',$,$,.UNDEFINED.,$,#155,#4000,$,$);
  #155=SECTION_PROFILE_COMPOUND(18,'2C10X20','Double channel',$,5,.F.,(#171,#172),
                                (#100,#101),(#1,#1));
      #171=SECTION_PROFILE(36,'C10X20','Channel section',$,4,.F.);
      #172=SECTION_PROFILE(37,'C10X20','Channel section',$,4,.T.);
      #100=CARTESIAN_POINT('angle offset 1',(0.,-0.375,0.0));
      #101=CARTESIAN_POINT('angle offset 2',(0.,0.375,0.0));
      #1=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.0),#56);
        #56=(CONTEXT_DEPENDENT_UNIT('DEGREE')NAMED_UNIT(#55)PLANE_ANGLE_UNIT());
    #4000=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(36.),#53);
```

FIG. 13a: CIS/2 double channel

In IFC, a compound section such as a double angle or channel is specified with IfcCompositeProfileDef which refers to two profiles. In Fig. 13b, the two sections of the composite section are a channel section defined by IfcUShapeProfileDef and a mirrored channel section defined by IfcDerivedProfileDef. The offsets of each section (#41063, #41068) are relative to the center of the channel cross section.

```
#41117= IFCPRODUCTDEFINITIONSHAPE($,$,(#41118));
  #41118= IFCSHAPEREPRESENTATION(#820011,'Body','SweptSolid',(#41119));
    #41119= IFCEXTRUDEDAREASOLID(#155,#820049,#820044,36.);
      #155= IFCCOMPOSITEPROFILEDEF(.AREA.,'2C10X20',(#171,#172),$);
        #171= IFCDERIVEDPROFILEDEF(.AREA.,'C10X20',#41065,#41066, 'Mirror Y axis');
          #41065= IFCUSHAPEPROFILEDEF(.AREA.,'C10X20',#41064,
                                   10.,2.74,0.379,0.436,$,$,$,$);
            #41064= IFCAXIS2PLACEMENT2D(#41063,#820052);
              #41063= IFCCARTESIANPOINT((1.745,0.0));
              #820052= IFCDIRECTION((1.,0.));
          #41066= IFCCARTESIANTRANSFORMATIONOPERATOR2D(#820054,$,#820051,$);
            #820054= IFCDIRECTION((-1.,0.));
            #820051= IFCCARTESIANPOINT((0.,0.));
        #172= IFCUSHAPEPROFILEDEF(.AREA.,'C10X20',#41069,10.,2.74,0.379,0.436,$,$,$,$);
          #41069= IFCAXIS2PLACEMENT2D(#41068,#820052);
            #41068= IFCCARTESIANPOINT((1.745,0.0));
            #820052= IFCDIRECTION((1.,0.));
      #820049= IFCAXIS2PLACEMENT3D(#820041,#820042,#820043);
      #820044= IFCDIRECTION((0.,0.,1.));
```

FIG. 13b: IFC double channel

A.6 Edge Defined Parts

In CIS/2, Section_profile_edge_defined is used for cross sections defined by an arbitrary closed curve that is defined by a polyline or a set of curve segments as shown in Fig. 14a. This method of defining a section profile is not commonly used.

```
#90=PART_PRISMATIC_SIMPLE(5,'172407','','',.UNDEFINED.,'',#180,#110,$,$);
  #180=SECTION_PROFILE_EDGE_DEFINED(7,'W10X33',$,$,5,.F.,#160,());
    #160=COMPOSITE_CURVE($,(#140),.F.);
      #140=COMPOSITE_CURVE_SEGMENT($,.F.,#120);
        #120=POLYLINE('Polyline',(#330,#340,#350,#360,#370,#380,#390,#400,#410,
                                  #420,#430,#440,#450,#460,#470,#480));
```

FIG. 14a: CIS/2 edge defined profile

In IFC, an edge defined cross section is modeled with IfcArbitraryClosedProfileDef as shown in Fig. 14b.

```
#821= IFCPRODUCTDEFINITIONSHAPE($,$,(#822));
  #822= IFCSHAPEREPRESENTATION(#16011,'Body','SweptSolid',(#823));
    #823= IFCEXTRUDEDAREASOLID(#180,#16049,#16044,12.);
      #180= IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,'Edge Defined',#818);
        #818= IFCPOLYLINE((#801,#802,#803,#804,#805,#806,#807,#808,#809,#810,
                           #811,#812,#813,#814,#815,#816,#801));
```

FIG. 14b: IFC edge defined profile

APPENDIX B - DESIGN MODEL EXAMPLES

The examples in Appendix B show how CIS/2 design model entities are mapped to IFC. A discussion about the CIS/2 design model is in section 1.5.

B.1 Design Parts

Fig. 15a shows a design model for a column and beam. A Design_part refers to a shape representation (Part_prismatic_simple) and at least one parent assembly (Assembly_design_structural_member_linear) and coordinate system (Coord_system_cartesian_3d). The coordinate system defines the beam or column position and orientation with an Axis2_placement_3d. The Assembly_design_structural_member_linear is a conceptual decomposition of the design model into assemblies. A design part can be part of multiple design assemblies and there can be multiple design parts in a design assembly, however, in practice there is usually only a one-to-one relationship between design parts and design assemblies.

```
#15=DESIGN_PART('C-2',#24,(#44),(#18));
  #24=PART_PRISMATIC_SIMPLE(2,'A36',$,$,.UNDEFINED.,$,#47,#41,$,$);
    #47=SECTION_PROFILE(2,'W10X45',$,$,8,.F.);
    #41=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(12.),#53);
  #44=ASSEMBLY_DESIGN_STRUCTURAL_MEMBER_LINEAR(0,'C-1',$,$,0,.LOW.,.T.,.F.,(),(),
            .T.,.UNDEFINED_ROLE.,.PRIMARY_MEMBER.,.COLUMN.);
  #18=COORD_SYSTEM_CARTESIAN_3D('Design Part','Design Part CS’,$,3,#21);
    #21=AXIS2_PLACEMENT_3D('Axis3d',#37,#33,#32);
      #37=CARTESIAN_POINT('Origin',(-44.9137841525084,37.8979091266392,0.));
      #33=DIRECTION('Direction',(1.,0.,0.));
      #32=DIRECTION('Orientation',(0.,0.,1.));

#16=DESIGN_PART('B-1',#25,(#45),(#19));
  #25=PART_PRISMATIC_SIMPLE(3,'A36',$,$,.UNDEFINED.,$,#48,#42,$,$);
    #48=SECTION_PROFILE(3,'W8X24',$,$,8,.F.);
    #42=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(15.38),#53);
  #45=ASSEMBLY_DESIGN_STRUCTURAL_MEMBER_LINEAR(0,'B-1',$,$,0,.LOW.,.T.,.F.,(),(),
            .T.,.UNDEFINED_ROLE.,.PRIMARY_MEMBER.,.BEAM.);
  #19=COORD_SYSTEM_CARTESIAN_3D('Design Part','Design Part CS’,$,3,#22);
    #22=AXIS2_PLACEMENT_3D('Axis3d',#39,#32,#33);
      #39=CARTESIAN_POINT('Origin',(-44.9137841525084,37.8979091266392,12.));
      #32=DIRECTION('Orientation',(0.,0.,1.));
      #33=DIRECTION('Direction',(1.,0.,0.));
```

FIG. 15a: CIS/2 design model, one beam and one column

In IFC, parts such as beams, columns, and braces use Ifc{Beam/Column/Member}. According to the IFC specifications, IfcBeam and IfcColumn are nearly horizontal and vertical members, respectively, that may or may not carry loads. The orientation of an IfcMember is not relevant to its definition. IfcPlate and IfcRailing (handrails) can also be used for other structural members.

Fig. 15b shows how IfcColumn and IfcBeam are used. Each refers to an IfcLocalPlacement to define its position and orientation with an IfcAxis2Placement3D. The physical representation of any Ifc{Beam/Column/Member} is given with an IfcProductDefinitionShape which with IfcShapeRepresentation refers to a shape representation as described in Appendix A.

There is no IFC equivalent of the CIS/2 Assembly_design_structural_member, although IfcElementAssembly could be used to aggregate the Ifc{Beam/Column/Member} into the appropriate assemblies.

```
#15= IFCCOLUMN('guid',#4005,'C-2','W10X45','Column',#18,#219,'C-2');
  #18= IFCLOCALPLACEMENT($,#21);
    #21= IFCAXIS2PLACEMENT3D(#37,#33,#32);
      #37= IFCCARTESIANPOINT((-44.9137841525084,37.8979091266392,0.));
      #33= IFCDIRECTION((1.,0.,0.));
      #32= IFCDIRECTION((0.,0.,1.));
  #219= IFCPRODUCTDEFINITIONSHAPE($,$,(#220));
    #220= IFCSHAPEREPRESENTATION(#4011,'Body','SweptSolid',(#221));
      #221= IFCEXTRUDEDAREASOLID(#47,#4049,#4044,12.);
        #47= IFCISHAPEPROFILEDEF(.AREA.,'W10X45',#207,
                                 0.668066,0.84133,0.029155,0.051646,$);
          #207= IFCAXIS2PLACEMENT2D(#206,#4052);
            #206= IFCCARTESIANPOINT((0.0,-0.420665));
            #4052= IFCDIRECTION((1.,0.));
        #4049= IFCAXIS2PLACEMENT3D(#4041,#4042,#4043);
        #4044= IFCDIRECTION((0.,0.,1.));

#16= IFCBEAM('guid',#4005,'B-1','W8X24','Beam',#19,#222,'B-1');
  #19= IFCLOCALPLACEMENT($,#22);
    #22= IFCAXIS2PLACEMENT3D(#39,#32,#33);
      #39= IFCCARTESIANPOINT((-44.9137841525084,37.8979091266392,12.));
      #32= IFCDIRECTION((0.,0.,1.));
      #33= IFCDIRECTION((1.,0.,0.));
  #222= IFCPRODUCTDEFINITIONSHAPE($,$,(#223));
    #223= IFCSHAPEREPRESENTATION(#4011,'Body','SweptSolid',(#224));
      #224= IFCEXTRUDEDAREASOLID(#48,#4049,#4044,15.3874369724483);
        #48= IFCISHAPEPROFILEDEF(.AREA.,'W8X24',#211,
                                 0.54145,0.660569,0.0204085,0.03332,$);
          #211= IFCAXIS2PLACEMENT2D(#210,#4052);
            #210= IFCCARTESIANPOINT((0.0,-0.3302845));
            #4052= IFCDIRECTION((1.,0.));
        #4049= IFCAXIS2PLACEMENT3D(#4041,#4042,#4043);
        #4044= IFCDIRECTION((0.,0.,1.));
```

FIG. 15b: IFC model, one beam and one column

Fig. 16a shows a different method to specify the position and orientation of a design part in CIS/2. The design part is located at the origin (0,0,0) defined by its coordinate system (#535). However, the Assembly_design_structural_member_linear is part of a Located_assembly which has a location defined by its coordinate system (#534).

```
#331=DESIGN_PART('B_7',#1679,(#1397),(#535));
  #1679=(PART(.UNDEFINED.,$)PART_PRISMATIC()PART_PRISMATIC_SIMPLE(#5878,#1540,$,$));
    #5878=SECTION_PROFILE(64,'W6X9',$,$,8,.F.);
    #1540=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(300.),#8249);
  #1397=ASSEMBLY_DESIGN_STRUCTURAL_MEMBER_LINEAR(65,'B_7',$,$,0,.LOW.,.F.,.F.,(),(),.F.,
          .UNDEFINED_ROLE.,.UNDEFINED_CLASS.,.BEAM.);
  #535=COORD_SYSTEM_CARTESIAN_3D('Design_Part','Design_Part CS',$,3,#929);
    #929=AXIS2_PLACEMENT_3D('Axis3d',#7519,#6470,#6471);
      #7519=CARTESIAN_POINT('Origin',(0.,0.,0.));
      #6470=DIRECTION('Direction',(0.,0.,1.));
      #6471=DIRECTION('Direction',(1.,0.,0.));

#1258=LOCATED_ASSEMBLY(65,'B_7',$,#534,$,#1397,#1754);
  #534=COORD_SYSTEM_CARTESIAN_3D('Assembly','Assembly CS',$,3,#928);
    #928=AXIS2_PLACEMENT_3D('Axis3d',#7518,#6468,#6469);
      #7518=CARTESIAN_POINT('Origin',(300.,360.,756.));
      #6468=DIRECTION('Direction',(0.,0.,1.));
      #6469=DIRECTION('Direction',(1.,0.,0.));
  #1397=ASSEMBLY_DESIGN_STRUCTURAL_MEMBER_LINEAR(65,'B_7',$,$,0,.LOW.,.F.,.F.,(),(),.F.,
          .UNDEFINED_ROLE.,.UNDEFINED_CLASS.,.BEAM.);
  #1754=STRUCTURE(1,'Design Model','');
```

FIG. 16a: CIS/2 design part, position and orientation with Located_assembly

Fig. 16b shows the equivalent IfcBeam where the location is derived from the location of the CIS/2 Located_assembly, however, no equivalent IfcElementAssembly is generated.

```
#331= IFCBEAM('1TVFxl3WpcIAxAFdoz9WES',#180005,'B_7','W6X9',$,#534,#9459,'B_7');
  #534= IFCLOCALPLACEMENT($,#928);
    #928= IFCAXIS2PLACEMENT3D(#7518,#6468,#6469);
      #7518= IFCCARTESIANPOINT((300.,360.,756.));
      #6468= IFCDIRECTION((0.,0.,1.));
      #6469= IFCDIRECTION((1.,0.,0.));
  #9459= IFCPRODUCTDEFINITIONSHAPE($,$,(#9460));
    #9460= IFCSHAPEREPRESENTATION(#180011,'Body','SweptSolid',(#9461));
      #9461= IFCEXTRUDEDAREASOLID(#5878,#180049,#180044,300.);
        #5878= IFCISHAPEPROFILEDEF(.AREA.,'W6X9',#9007,3.94,5.9,0.17,0.215,$);
```

FIG. 16b: Equivalent IFC design part

A CIS/2 Assembly_design_structural_connection_internal groups together multiple Assembly_design_structural_member, and thus their associated Design_part, that are connected together at a common location. For example, for the structure in Fig. 3, an Assembly_design_structural_connection_internal could be used wherever there would be a physical connection between a beam, column, or brace. The closest IFC equivalent is IfcRelConnectsElements; however, this is only a one-to-one relationship and would not be able to handle more than two parts being connected. A CIS/2 Design_joint_system is another method to specify which Design_part are connected together and with what type of connector.

APPENDIX C - DETAILED MODEL EXAMPLES

The examples in Appendix C show how CIS/2 detailed model entities are mapped to IFC. A discussion about the CIS/2 detailed model is in section 1.6.

C.1 Parts and Assemblies

In a detailed model, parts are located relative to assemblies and assemblies are located relative to a structure. Fig. 17 shows a typical assembly consisting of a beam with two clip angles at one end. The associated CIS/2 is shown in Fig. 18a. The part is located within an assembly with Located_part. The Located_part refers to the physical member defined by Part_prismatic_simple and to the Located_assembly it is part of. Located_assembly does not indicate which parts are contained in the assembly; rather, Located_part refers to which Located_assembly it is part of. For the two clip angles, the instances of Located_part are unique; however, they both refer to the same instance of Part_prismatic_simple (#57) that defines their geometry.

The coordinate system referred to by the Located_part (Coord_system) refers to the part coordinate system (Coord_system_cartesian_3d) and its parent coordinate system (Coord_system_child) for the assembly coordinate system. This is the method used locating parts within an assembly in CIS/2. In Fig. 18a all three Located_part refer to the same Located_assembly (#273) and same parent coordinate system (#270) thus associating the beam and two clip angles into one assembly.

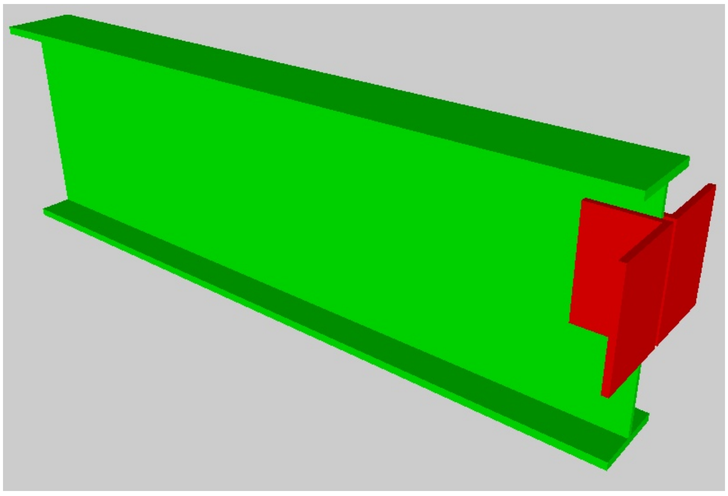


FIG. 17: Beam with two clip angles

```
#146=LOCATED_PART(12,'w4[12]','W flange',#121,#58,#273);
  #121=(COORD_SYSTEM('Local','Part CS',$,3)
        COORD_SYSTEM_CARTESIAN_3D(#243)COORD_SYSTEM_CHILD(#270));
    #243=AXIS2_PLACEMENT_3D('Part CS',#173,#218,#217);
      #173=CARTESIAN_POINT('axis point',(6.35,0.,0.));
      #218=DIRECTION('local z',(0.,0.,1.));
      #217=DIRECTION('local x',(1.,0.,0.));
    #270=COORD_SYSTEM_CARTESIAN_3D('Global','Assembly CS',$,3,#239);
      #239=AXIS2_PLACEMENT_3D('Assembly CS',#150,#151,#217);
        #150=CARTESIAN_POINT('axis point',(-1500.,500.,0.));
        #151=DIRECTION('local z',(-0.173,0.,0.985));
        #217=DIRECTION('local x',(1.,0.,0.));
  #58=PART_PRISMATIC_SIMPLE(12,'w4[12]',$,$,.ROLLED.,$,#51,#99,$,$);
    #51=SECTION_PROFILE_I_TYPE(0,'W12x22',$,'W flange',8,.T.,#94,#95,#96,#97,#98,$,$,$);
    #99=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(1000.),#87);
  #273=LOCATED_ASSEMBLY(1,'B_1[1]','beam',#270,$,#161,#276);
    #270=COORD_SYSTEM_CARTESIAN_3D('Global','Assembly CS',$,3,#239);
    #161=ASSEMBLY_MANUFACTURING(1,'sub material',$,$,0,.LOW.,$,$,$,.SHOP_PROCESS.);
    #276=STRUCTURE(0,'1 assembly - 3 parts',$);
```

FIG. 18a: CIS/2 assembly - beam with two clip angles (continued on next page)

```
#148=LOCATED_PART(1,'a2[1]','Clip Angle NS',#119,#57,#273);
  #119=(COORD_SYSTEM('Local','Part CS',$,3)
        COORD_SYSTEM_CARTESIAN_3D(#241)COORD_SYSTEM_CHILD(#270));
    #241=AXIS2_PLACEMENT_3D('Part CS',#170,#221,#220);
      #170=CARTESIAN_POINT('axis point',(1019.05,-3.30199987888336,-44.45));
      #221=DIRECTION('local z',(0.,-1.,0.));
      #220=DIRECTION('local x',(0.,0.,-1.));
    #270=COORD_SYSTEM_CARTESIAN_3D('Global','Assembly CS',$,3,#239);
  #57=PART_PRISMATIC_SIMPLE(1,'a2[1]',$,$,.ROLLED.,$,#54,#93,$,$);
    #54=SECTION_PROFILE(0,'L4x3 1/2x5/16',$,'Angle',1,.F.);
    #93=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(139.7),#87);
  #273=LOCATED_ASSEMBLY(1,'B_1[1]','beam',#270,$,#161,#276);

#149=LOCATED_PART(1,'a2[1]','Clip Angle FS',#120,#57,#273);
  #120=(COORD_SYSTEM('Local','Part CS',$,3)
        COORD_SYSTEM_CARTESIAN_3D(#242)COORD_SYSTEM_CHILD(#270));
    #242=AXIS2_PLACEMENT_3D('Part CS',#172,#223,#222);
      #172=CARTESIAN_POINT('axis point',(1019.05,3.30199987888336,-184.15));
      #223=DIRECTION('local z',(0.,1.,0.));
      #222=DIRECTION('local x',(0.,0.,1.));
    #270=COORD_SYSTEM_CARTESIAN_3D('Global','Assembly CS',$,3,#239);
  #57=PART_PRISMATIC_SIMPLE(1,'a2[1]',$,$,.ROLLED.,$,#54,#93,$,$);
    #54=SECTION_PROFILE(0,'L4x3 1/2x5/16',$,'Angle',1,.F.);
    #93=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(139.7),#87);
  #273=LOCATED_ASSEMBLY(1,'B_1[1]','beam',#270,$,#161,#276);
```

FIG. 18a (continued): CIS/2 assembly - beam with two clip angles

The IFC representation of the beam and clip angles is shown in Fig. 18b. The parts are modeled similarly to the design model beam and column example in Fig. 15b of Appendix B. IfcBeam is used for all three parts because the entire assembly is a horizontal assembly. However, the longitudinal axis of the clip angles is vertical so they could alternatively be an IfcColumn. Another possibility is to have all three parts be IfcMember. In this example, each clip angle has a unique IfcShapeRepresentation although the shape of both clip angles is identical.

To locate all three members in an assembly, IfcLocalPlacement is used for the part coordinate systems that refer to a relative IfcLocalPlacement (#270) for the assembly coordinate system. The IDs of the IfcLocalPlacement are the same as the IDs of the CIS/2 Coord_system entities in Fig. 18a.

IfcElementAssembly is used to group together the three parts in an assembly with IfcRelAggregates. The assembly is placed in an IfcBuilding with IfcRelContainedInSpatialStructure.

```
#146= IFCBEAM('guid',#6005,'w4[12]','W12x22',$,#121,#332,'w4[12]');
  #121= IFCLOCALPLACEMENT(#270,#243);
    #270= IFCLOCALPLACEMENT($,#239);
      #239= IFCAXIS2PLACEMENT3D(#150,#151,#217);
        #150= IFCCARTESIANPOINT((-1500.,500.,0.));
        #151= IFCDIRECTION((0.,0.,1.));
        #217= IFCDIRECTION((1.,0.,0.));
    #243= IFCAXIS2PLACEMENT3D(#173,#218,#217);
      #173= IFCCARTESIANPOINT((6.35,0.,0.));
      #218= IFCDIRECTION((0.,0.,1.));
      #217= IFCDIRECTION((1.,0.,0.));
  #332= IFCPRODUCTDEFINITIONSHAPE($,$,(#333));
    #333= IFCSHAPEREPRESENTATION(#6011,'Body','SweptSolid',(#334));
      #334= IFCEXTRUDEDAREASOLID(#51,#6049,#6044,1000.);
        #51= IFCISHAPEPROFILEDEF(.AREA.,'W12X22',#302,102.4,312.7,6.6,10.8,$);
```

FIG. 18b: IFC assembly - beam with two clip angles (continued on next page)

```
#148= IFCBEAM('guid',#6005,'a2[1]','L4x3-1/2x5/16',$,#119,#335,'a2[1]');
  #119= IFCLOCALPLACEMENT(#270,#241);
    #270= IFCLOCALPLACEMENT($,#239);
      #239= IFCAXIS2PLACEMENT3D(#150,#151,#217);
        #150= IFCCARTESIANPOINT((-1500.,500.,0.));
        #151= IFCDIRECTION((0.,0.,1.));
        #217= IFCDIRECTION((1.,0.,0.));
    #241= IFCAXIS2PLACEMENT3D(#170,#221,#220);
      #170= IFCCARTESIANPOINT((1019.05,-3.30199987888336,-44.45));
      #221= IFCDIRECTION((0.,-1.,0.));
      #220= IFCDIRECTION((0.,0.,-1.));
  #335= IFCPRODUCTDEFINITIONSHAPE($,$,(#336));
    #336= IFCSHAPEREPRESENTATION(#6011,'Body','SweptSolid',(#337));
      #337= IFCEXTRUDEDAREASOLID(#54,#6049,#6044,139.7);
        #54= IFCDERIVEDPROFILEDEF(.AREA.,'L4X3-1/2X5/16',#305,#306,'Mirror Y axis');
          #305= IFCLSHAPEPROFILEDEF(.AREA.,'L4X3-1/2X5/16',
                                    #304,101.6,88.9,7.9375,$,$,$,$,$);

#149= IFCBEAM('guid',#6005,'a2[1]','L4x3-1/2x5/16',$,#120,#338,'a2[1]');
  #120= IFCLOCALPLACEMENT(#270,#242);
    #270= IFCLOCALPLACEMENT($,#239);
      #239= IFCAXIS2PLACEMENT3D(#150,#151,#217);
        #150= IFCCARTESIANPOINT((-1500.,500.,0.));
        #151= IFCDIRECTION((0.,0.,1.));
        #217= IFCDIRECTION((1.,0.,0.));
    #242= IFCAXIS2PLACEMENT3D(#172,#223,#222);
      #172= IFCCARTESIANPOINT((1019.05,3.30199987888336,-184.15));
      #223= IFCDIRECTION((0.,1.,0.));
      #222= IFCDIRECTION((0.,0.,1.));
  #338= IFCPRODUCTDEFINITIONSHAPE($,$,(#339));
    #339= IFCSHAPEREPRESENTATION(#6011,'Body','SweptSolid',(#340));
      #340= IFCEXTRUDEDAREASOLID(#54,#6049,#6044,139.7);
        #54= IFCDERIVEDPROFILEDEF(.AREA.,'L4X3-1/2X5/16',#305,#306,'Mirror Y axis');
          #305= IFCLSHAPEPROFILEDEF(.AREA.,'L4X3-1/2X5/16',
                                    #304,101.6,88.9,7.9375,$,$,$,$,$);

#273= IFCELEMENTASSEMBLY('guid',#6005,'B_1[1]',$,$,#270,$,'B_1[1]',$,.NOTDEFINED.);
  #270= IFCLOCALPLACEMENT($,#239);
#341= IFCRELAGGREGATES('guid',#6005,'B_1[1]','Assembly',#273,(#146,#148,#149));
  #273= IFCELEMENTASSEMBLY('guid',#6005,'B_1[1]',$,$,#270,$,'B_1[1]',$,.NOTDEFINED.);
  #146= IFCBEAM('guid',#6005,'w4[12]','W12x22',$,#121,#332,'w4[12]');
  #148= IFCBEAM('guid',#6005,'a2[1]','L4x3-1/2x5/16',$,#119,#335,'a2[1]');
  #149= IFCBEAM('guid',#6005,'a2[1]','L4x3-1/2x5/16',$,#120,#338,'a2[1]');

#342= IFCRELCONTAINEDINSPATIALSTRUCTURE('guid',#6005,'Physical model',$,(#273),#6023);
  #273= IFCELEMENTASSEMBLY('guid',#6005,'B_1[1]',$,$,#270,$,'B_1[1]',$,.NOTDEFINED.);
  #6023= IFCBUILDING('guid',#6005,'Building',$,$,#6025,$,$,.ELEMENT.,$,$,$);
```

FIG. 18b (continued): IFC assembly - beam with two clip angles

Comparing the IFC representation of a CIS/2 design model in Fig. 15b of Appendix B and the IFC detailed model above in Fig. 18b; there is not much that differentiates the models from each other. All of them use IfcLocalPlacement for the position and orientation of Ifc{Beam/Column}. Generally, none of the IFC entities used positively identifies any of the physical models as a design, analysis, or detailed model.

C.2 Mapped Representation

Instead of using unique shape representations for both clip angles in Fig. 18b, in IFC a mapped representation can be used so that both clip angles share a single shape representation as shown in Fig. 18c. This is a more efficient and compact method to represent identical shapes and is comparable to the way it is modeled with CIS/2 in Fig. 18a. For an IFC mapped representation, the geometric shape representation (#344) is referred to by a type IfcBeamType and IfcRepresentationMap (#347). The IfcBeam refer to the geometric shape representation through IfcMappedItem. The IfcBeamType is associated with the occurrence of each IfcBeam with IfcRelDefinesByType.

```
#346= IFCBEAMTYPE('guid',#6005,'L4x3-1/2x5/16',$,$,$,(#347),$,$,.NOTDEFINED.);
  #347= IFCREPRESENTATIONMAP(#6040,#344);
    #6040= IFCAXIS2PLACEMENT3D(#6041,#6044,#6042);
      #6041= IFCCARTESIANPOINT((0.,0.,0.));
      #6044= IFCDIRECTION((0.,0.,1.));
      #6042= IFCDIRECTION((1.,0.,0.));
    #344= IFCSHAPEREPRESENTATION(#6011,'Body','SweptSolid',(#345));
      #345= IFCEXTRUDEDAREASOLID(#54,#6049,#6044,139.7);
        #54= IFCDERIVEDPROFILEDEF(.AREA.,'L4X3-1/2X5/16',#305,#306,'Mirror Y axis');
          #305= IFCLSHAPEPROFILEDEF(.AREA.,'L4X3-1/2X5/16',
                                    #304,101.6,88.9,7.9375,$,$,$,$,$);

#148= IFCBEAM('guid',#6005,'a2[1]','L4x3-1/2x5/16',$,#119,#340,'a2[1]');
  #119= IFCLOCALPLACEMENT(#270,#241);
    #270= IFCLOCALPLACEMENT($,#239);
      #239= IFCAXIS2PLACEMENT3D(#150,#151,#217);
    #241= IFCAXIS2PLACEMENT3D(#170,#221,#220);
  #340= IFCPRODUCTDEFINITIONSHAPE($,$,(#341));
    #341= IFCSHAPEREPRESENTATION(#6011,'Body','MappedRepresentation',(#342));
      #342= IFCMAPPEDITEM(#347,#6059);
        #347= IFCREPRESENTATIONMAP(#6040,#344);
        #6059= IFCCARTESIANTRANSFORMATIONOPERATOR3D($,$,#6041,1.,$);
          #344= IFCSHAPEREPRESENTATION(#6011,'Body','SweptSolid',(#345));

#149= IFCBEAM('guid',#6005,'a2[1]','L4x3-1/2x5/16',$,#120,#348,'a2[1]');
  #120= IFCLOCALPLACEMENT(#270,#242);
    #270= IFCLOCALPLACEMENT($,#239);
      #239= IFCAXIS2PLACEMENT3D(#150,#151,#217);
    #242= IFCAXIS2PLACEMENT3D(#172,#223,#222);
  #348= IFCPRODUCTDEFINITIONSHAPE($,$,(#349));
    #349= IFCSHAPEREPRESENTATION(#6011,'Body','MappedRepresentation',(#350));
      #350= IFCMAPPEDITEM(#347,#6059);
        #347= IFCREPRESENTATIONMAP(#6040,#344);
          #344= IFCSHAPEREPRESENTATION(#6011,'Body','SweptSolid',(#345));

#354= IFCRELDEFINESBYTYPE('guid',#6005,'Beam',$,(#148,#149),#346);
```

FIG. 18c: IFC mapped representation for two clip angles

C.3 Cutouts

Cutouts, also known as copes, are features that remove material from a part. Fig. 19 shows the five cutouts most commonly implemented in CIS/2. They are miter cuts, notches, chamfers, flange notches, and flange chamfers. Other CIS/2 cutouts include edge chamfers and web penetrations.

Fig. 20a shows how the miter cut and notch shown in Fig. 19 is modeled in CIS/2. Cutouts (known as features in CIS/2) are located on a part with Located_feature_for_located_part. Similar to how parts are located relative to an assembly coordinate system in Fig. 17a, the feature is located relative to the part coordinate system. The features are parametrically defined by their dimensions (length, width, depth, and angle) and location on the part where they are applied. The location specifies which end, side, face, or edge the cutout is applied to. For example, Feature_volume_prismatic_skewed_end shows that the miter cut is applied to the bottom edge of the start face of the part. Feature_volume_prismatic_notch shows that the notch is applied in the same location. The location of the features is also specified with a feature coordinate system although this information is redundant because it is already specified parametrically.

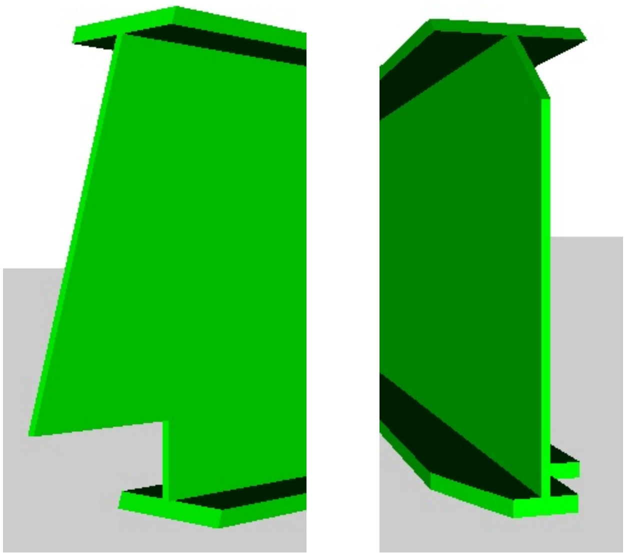


FIG. 19: Typical cutouts – miter cut, notch (left) – chamfer, flange chamfer, flange notch (right)

```
#149=LOCATED_PART(12,'w4[12]',$,#121,#58,#273);
  #121=(COORD_SYSTEM('Local','Part CS',$,3)COORD_SYSTEM_CARTESIAN_3D(#243)
        COORD_SYSTEM_CHILD(#270));
    #243=AXIS2_PLACEMENT_3D('Part CS',#173,#218,#217);
    #270=COORD_SYSTEM_CARTESIAN_3D('Global','Assembly CS',$,3,#239);
      #239=AXIS2_PLACEMENT_3D('Assembly CS',#164,#218,#217);
  #58=PART_PRISMATIC_SIMPLE(12,'w4[12]',$,$,.ROLLED.,$,#51,#99,$,$);
    #51=SECTION_PROFILE(0,'W12x22','ASTM spectification A6','W flange',5,.F.);
    #99=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(3028.95),#87);
  #273=LOCATED_ASSEMBLY(1,'B_1[1]','beam',#270,$,#161,#276);
    #161=ASSEMBLY_MANUFACTURING(1,'sub material',$,$,0,.LOW.,$,$,$,.SHOP_PROCESS.);
    #276=STRUCTURE(0,'Cutouts',$);

#1051=LOCATED_FEATURE_FOR_LOCATED_PART(0,'Miter',$,#2035,#1030,#149);
  #2035=(COORD_SYSTEM('Local','Feature CS',$,3)COORD_SYSTEM_CARTESIAN_3D(#3621)
         COORD_SYSTEM_CHILD(#121));
    #3621=AXIS2_PLACEMENT_3D('Feature CS',#2684,#3536,#3533);
    #121=(COORD_SYSTEM('Local','Part CS',$,3)COORD_SYSTEM_CARTESIAN_3D(#243)
         COORD_SYSTEM_CHILD(#270));
      #243=AXIS2_PLACEMENT_3D('Part CS',#173,#218,#217);
#270=COORD_SYSTEM_CARTESIAN_3D('Global','Assembly CS',$,3,#239);
        #239=AXIS2_PLACEMENT_3D('Assembly CS',#164,#218,#217);
  #1030=FEATURE_VOLUME_PRISMATIC_SKEWED_END(1,'Cope','Miter cut',.BOTTOM_EDGE.,
                                            .START_FACE.,.T.,#303,#304);
    #303=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.244978666305542),#302);
    #304=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.),#302);
  #149=LOCATED_PART(12,'w4[12]',$,#121,#58,#273);

#1047=LOCATED_FEATURE_FOR_LOCATED_PART(0,'Notch',$,#2035,#1026,#149);
  #2035=(COORD_SYSTEM('Local','Feature CS',$,3)COORD_SYSTEM_CARTESIAN_3D(#3621)
         COORD_SYSTEM_CHILD(#121));
    #121=(COORD_SYSTEM('Local','Part CS',$,3)COORD_SYSTEM_CARTESIAN_3D(#243)
          COORD_SYSTEM_CHILD(#270));
      #270=COORD_SYSTEM_CARTESIAN_3D('Global','Assembly CS',$,3,#239);
  #1026=FEATURE_VOLUME_PRISMATIC_NOTCH(1,'Cope','Notch',.BOTTOM_EDGE.,.START_FACE.,.T.,
                                       #1525,#1557,#1558);
    #1525=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(108.9),#87);
    #1557=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(61.75),#87);
    #1558=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(30.7),#87);
  #149=LOCATED_PART(12,'w4[12]',$,#121,#58,#273);
```

FIG. 20a: CIS/2 part with two cutouts, miter cut and notch

In IFC, the easiest method to model a part with cutouts is to use a boundary representation to explicitly model how the cutouts remove material from a part. However, the boundary representation does not capture the parametric information about the cutout. IfcPropertySet can be used to specify the information about a cutout and assign it to an IfcBeam as shown in Fig. 20b. The property set contains all of the same parametric information that was specified in the CIS/2 model. There is no property set for cutouts defined in the IFC specifications; the following example is a suggestion.

```
#5037= IFCRELDEFINESBYPROPERTIES('guid',#100005,'Cutout','Notch',(#149),#5014);
  #149= IFCBEAM('guid',#100005,'p2','W12x22',$,#121,#5032,'p2');
  #5014= IFCPROPERTYSET('guid',#100005,'PSet_Cutout','Notch',
                       (#5011,#5005,#5006,#5012,#5013,#5009));
    #5011= IFCPROPERTYSINGLEVALUE('Cutout',$,IFCLABEL('Notch'),$);
    #5005= IFCPROPERTYSINGLEVALUE('Length',$,IFCLENGTHMEASURE(108.9),$);
    #5006= IFCPROPERTYSINGLEVALUE('Depth',$,IFCLENGTHMEASURE(62.),$);
    #5012= IFCPROPERTYENUMERATEDVALUE('TopOrBottomEdge',$,
                                     (IFCTEXT('bottom_edge')),#5002);
      #5002= IFCPROPERTYENUMERATION('TopOrBottomEdgeEnum',(IFCTEXT('top_edge'),
                                 IFCTEXT('bottom_edge')),$);
    #5013= IFCPROPERTYENUMERATEDVALUE('StartOrEndFace',$,(IFCTEXT('start_face')),#5003);
      #5003= IFCPROPERTYENUMERATION('StartOrEndFaceEnum',(IFCTEXT('start_face'),
                                 IFCTEXT('end_face')),$);
    #5009= IFCPROPERTYENUMERATEDVALUE('OriginalFace',$,(IFCTEXT('T')),#5004);
      #5004= IFCPROPERTYENUMERATION('OriginalFaceEnum',(IFCTEXT('T'),IFCTEXT('F')),$);
```

FIG. 20b: IFC property set for a notch cutout

If extruded solids are used to model parts, then IFC boolean operations can be used to subtract material from a part for a cutout. Boolean operations can also be applied to boundary representation geometry. To model the miter cut, an IfcHalfSpaceSolid (#1030) defined by an IfcPlane is applied to the part with IfcBooleanClippingResult (#5036). To model the notch, another solid defined by IfcExtrudedAreaSolid (#1026) is used to define the volume that is subtracted from the part with IfcBooleanResult (#5035). Although this method will create the geometry of a part with cutouts, it does not specify the parametric information about the cutouts. IfcPropertySet as shown above in Fig. 20b could be used to specify that information.

```
#149= IFCBEAM('guid',#100005,'p2','W12x22',$,#121,#5032,'p2');
  #121= IFCLOCALPLACEMENT(#270,#243);
    #270= IFCLOCALPLACEMENT($,#239);
  #5032= IFCPRODUCTDEFINITIONSHAPE($,$,(#5033));
    #5033= IFCSHAPEREPRESENTATION(#100011,'Body','CSG',(#5035));
      #5035= IFCBOOLEANRESULT(.DIFFERENCE.,#5036,#1026);
        #5036= IFCBOOLEANCLIPPINGRESULT(.DIFFERENCE.,#5034,#1030);
          #5034= IFCEXTRUDEDAREASOLID(#51,#100049,#100044,3028.95);
            #51= IFCISHAPEPROFILEDEF(.AREA.,'W12X22',#100050,
                                    102.362,312.42,6.604,10.795,$);
          #1030= IFCHALFSPACESOLID(#5027,.F.);
            #5027= IFCPLANE(#5028);
              #5028= IFCAXIS2PLACEMENT3D(#5029,#5030,#5031);
                #5029= IFCCARTESIANPOINT((0.,0.,-156.20742));
                #5030= IFCDIRECTION((-0.97013,0.,0.2426));
                #5031= IFCDIRECTION((0.2426,0.,0.97013));
        #1026= IFCEXTRUDEDAREASOLID(#5019,#5020,#100044,3.77);
          #5019= IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,'NOTCH',#5022);
            #5022= IFCPOLYLINE((#5023,#5024,#5025,#5026,#5023));
          #5020= IFCAXIS2PLACEMENT3D(#5021,#100042,#100043);
            #5021= IFCCARTESIANPOINT((-1.,0.,-158.775));
            #100042= IFCDIRECTION((1.,0.,0.));
            #100043= IFCDIRECTION((0.,1.,0.));
          #100044= IFCDIRECTION((0.,0.,1.));
```

FIG. 20c: IFC with two cutouts (copes), chamfer and miter cut

C.4 Bolts

Connections in CIS/2 include bolts, nuts, washers, welds, shear studs, and holes. Fig. 21 shows a bolted connection with two bolts connecting clip angles to an I-beam. The beam and clip angles are transparent to show the bolts. The associated CIS/2 model is shown in Fig. 22a. The arrangement of bolts in a pattern or layout is specified with Joint_system_mechanical that contains a list of bolt locations. The bolt locations are specified relative to a joint coordinate system. The joint coordinate system is located relative to the assembly coordinate system similar to how parts are located relative to an assembly.

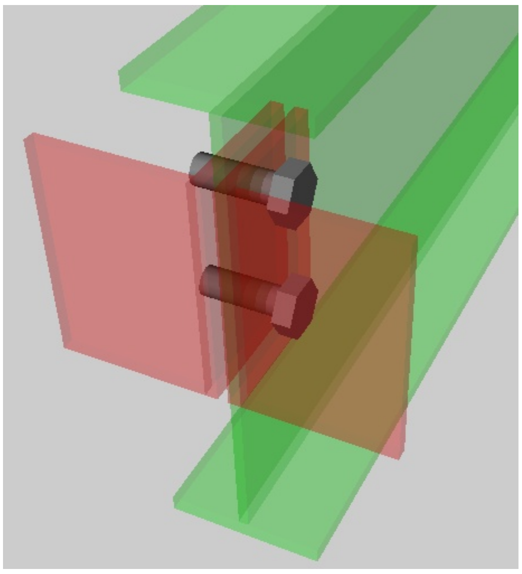


FIG. 21: Bolted connection

The joint system refers to a Fastener_mechanism that is comprised of Fastener_simple_bolt, Fastener_simple_washer, and Fastener_simple_nut. Each of those refers to the dimensions of a bolt, washer, and nut. In this example, the position of the nut and washer relative to the length of the bolt is not specified. Fastener_mechanism_with_position can be used to specify that information. The type of bolt head is not specified in this example but can be with entities such as Fastener_simple_bolt_{hexagonal/square/circular}_head.

```
#37=LOCATED_JOINT_SYSTEM(1,'2 bolts',$,#122,#31,#273);
  #122=(COORD_SYSTEM('Local','Bolt CS',$,3)
        COORD_SYSTEM_CARTESIAN_3D(#244)COORD_SYSTEM_CHILD(#270));
    #244=AXIS2_PLACEMENT_3D('Bolt CS',#176,#225,#224);
      #176=CARTESIAN_POINT('axis point',(2990.85,11.2394998788834,-76.2));
      #225=DIRECTION('local z',(0.,0.,-1.));
      #224=DIRECTION('local x',(0.,-1.,0.));
    #270=COORD_SYSTEM_CARTESIAN_3D('Global','Assembly CS',$,3,#239);
      #239=AXIS2_PLACEMENT_3D('Assembly CS',#150,#151,#217);
        #150=CARTESIAN_POINT('axis point',(-1500.,500.,0.));
        #151=DIRECTION('local z',(-0.173,0.,0.985));
        #217=DIRECTION('local x',(1.,0.,0.));
  #31=JOINT_SYSTEM_MECHANICAL(1,'2 bolts',$,.SHOP_PROCESS.,(#174,#175),#27);
    #174=CARTESIAN_POINT('bolt location',(0.,0.,0.));
    #175=CARTESIAN_POINT('bolt location',(0.,0.,76.2));
    #27=FASTENER_MECHANISM(1,'bolt with nut and washer',$,$,'0',(#23,#22,#21));
      #23=FASTENER_SIMPLE_BOLT(0,'Shop Bolt',$,$,'A325N',#100,#101,$,$,$,$);
        #100=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(19.05),#87);
        #101=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(50.8),#87);
      #22=FASTENER_SIMPLE_WASHER(0,'Hardened',$,$,$,#100,#102,$,$,$);
        #100=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(19.05),#87);
        #102=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(0.15625),#87);
      #21=FASTENER_SIMPLE_NUT(0,'Nut',$,$,$,#100,$);
        #100=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(19.05),#87);
  #273=LOCATED_ASSEMBLY(1,'B_1[1]','beam',#270,$,#161,#276);
```

FIG. 22a: CIS/2 bolt layout

In the current version of IFC2x3, there is no concept of a layout, pattern, or arrangement of items. Although the geometry of bolts and other fasteners can be specified, there is no efficient way to indicate the layout of the fasteners. The geometry of a bolt is modeled similar to how beams and columns are modeled. In the example in Fig. 22b, the bolt geometry is modeled by specifying the geometry through IfcMechanicalFastenerType and each instance of a bolt with IfcMechanicalFastener. This is similar to how the pair of clip angles is modeled in Fig. 18c. The head of the bolt is ignored and only a cylinder is used for the geometry of the bolt in this example. The cylinder for the bolt is modeled as an extruded solid referring to IfcCircleProfileDef.

Each IfcMechanicalFastener refers to three nested IfcLocalPlacement. The placements are for the bolt location in the layout, the layout location in the joint system coordinate, and the joint system in the assembly coordinate. To simulate the concept of a bolt layout in IFC, an IfcBuildingElementProxy is used represent the layout and the bolts are associated with it through IfcRelAssignsToProduct.

```
#338= IFCMECHANICALFASTENERTYPE('guid',#6005,'D=3/4 L=2 Shop Bolt',
                                $,$,$,(#339),$,'Bolt');
  #339= IFCREPRESENTATIONMAP(#6040,#332);
    #332= IFCSHAPEREPRESENTATION(#6011,'Body','SweptSolid',(#333));
      #333= IFCEXTRUDEDAREASOLID(#334,#6049,#6044,50.8);
        #334= IFCCIRCLEPROFILEDEF(.AREA.,'Bolt shank',#6050,9.525);

#359= IFCMECHANICALFASTENER('guid',#6005,'D=3/4 L=2 Shop A325N','Bolt','Bolt',
                            #360,#363,$,19.05,50.8);
  #360= IFCLOCALPLACEMENT(#122,#361);
    #122= IFCLOCALPLACEMENT(#270,#244);
      #270= IFCLOCALPLACEMENT($,#239);
        #239= IFCAXIS2PLACEMENT3D(#150,#151,#217);
          #150= IFCCARTESIANPOINT((-1500.,500.,0.));
          #151= IFCDIRECTION((0.,0.,1.));
          #217= IFCDIRECTION((1.,0.,0.));
      #244= IFCAXIS2PLACEMENT3D(#176,#225,#224);
        #176= IFCCARTESIANPOINT((962.8,11.2394998788834,-76.2));
        #225= IFCDIRECTION((0.,0.,-1.));
        #224= IFCDIRECTION((0.,-1.,0.));
    #361= IFCAXIS2PLACEMENT3D(#362,#6044,#6042);
      #362= IFCCARTESIANPOINT((0.,0.0,76.2));
      #6044= IFCDIRECTION((0.,0.,1.));
      #6042= IFCDIRECTION((1.,0.,0.));
  #363= IFCPRODUCTDEFINITIONSHAPE($,$,(#364));
    #364= IFCSHAPEREPRESENTATION(#6011,'Body','MappedRepresentation',(#365));
      #365= IFCMAPPEDITEM(#339,#6059);
        #339= IFCREPRESENTATIONMAP(#6040,#332);
          #332= IFCSHAPEREPRESENTATION(#6011,'Body','SweptSolid',(#333));

#366= IFCMECHANICALFASTENER('guid',#6005,'D=3/4 L=2 Shop A325N','Bolt','Bolt',
                            #367,#370,$,19.05,50.8);
  #367= IFCLOCALPLACEMENT(#122,#368);
    #122= IFCLOCALPLACEMENT(#270,#244);
      #270= IFCLOCALPLACEMENT($,#239);
        #239= IFCAXIS2PLACEMENT3D(#150,#151,#217);
      #244= IFCAXIS2PLACEMENT3D(#176,#225,#224);
    #368= IFCAXIS2PLACEMENT3D(#369,#6044,#6042);
      #379= IFCCARTESIANPOINT((0.,0.0,0.));
      #6044= IFCDIRECTION((0.,0.,1.));
      #6042= IFCDIRECTION((1.,0.,0.));
  #370= IFCPRODUCTDEFINITIONSHAPE($,$,(#371));
    #371= IFCSHAPEREPRESENTATION(#6011,'Body','MappedRepresentation',(#372));
      #372= IFCMAPPEDITEM(#339,#6059);
        #339= IFCREPRESENTATIONMAP(#6040,#332);
          #332= IFCSHAPEREPRESENTATION(#6011,'Body','SweptSolid',(#333));

#378= IFCRELDEFINESBYTYPE('guid',#6005,'Bolt',$,(#359,#366),#338);
```

FIG. 22b: IFC bolts (continued next page)

```
#37= IFCBUILDINGELEMENTPROXY('guid',#6005,'Bolt layout',$,$,#122,$,$,.COMPLEX.);
  #122= IFCLOCALPLACEMENT(#270,#244);
    #270= IFCLOCALPLACEMENT($,#239);
      #239= IFCAXIS2PLACEMENT3D(#150,#151,#217);
        #150= IFCCARTESIANPOINT((-1500.,500.,0.));
        #151= IFCDIRECTION((0.,0.,1.));
        #217= IFCDIRECTION((1.,0.,0.));
    #244= IFCAXIS2PLACEMENT3D(#176,#225,#224);
      #176= IFCCARTESIANPOINT((962.8,11.2394998788834,-76.2));
      #225= IFCDIRECTION((0.,0.,-1.));
      #224= IFCDIRECTION((0.,-1.,0.));

#373= IFCRELASSIGNSTOPRODUCT('guid',#6005,'Bolt layout',$,(#359,#366),.PRODUCT.,#37);
```

FIG. 22b (continued): IFC bolts

C.5 Welds

In CIS/2, welds are modeled similarly to bolts as shown in Fig. 23a. Welds are located relative to an assembly coordinate system with Located_joint_system. Welds are defined by Joint_system_welded_linear where the weld is specified by a Weld_mechanism and a weld path defined by a Polyline. The Weld_mechanism indicates that it is a fillet weld with full penetration. More information about welds can be specified with entities such as Weld_mechanism_{fillet/groove_beveled/groove_butt/spot_seam} and Weld_{arc/beam/gas/pressure/resistance/stud} although in practice they have not been implemented.

```
#236=LOCATED_JOINT_SYSTEM(1,'[1] weld:1/5','Weld connecting (0) to Member[1]',
                          #1605,#158,#4027);
  #1605=(COORD_SYSTEM('Local','Joint CS’,$,3)
         COORD_SYSTEM_CARTESIAN_3D(#3627)COORD_SYSTEM_CHILD(#3990));
    #3627=AXIS2_PLACEMENT_3D('Joint CS’,#2733,#3539,#3537);
      #2733=CARTESIAN_POINT('axis2 point',(25.4,178.371498062134,322.961003875732));
      #3539=DIRECTION('local z',(1.,0.,0.));
      #3537=DIRECTION('local x',(0.,-1.,0.));
    #3990=COORD_SYSTEM_CARTESIAN_3D('Global','Assembly CS’,$,3,#3618);
      #3618=AXIS2_PLACEMENT_3D('Assembly CS’,#2684,#3532,#3531);
        #2684=CARTESIAN_POINT('axis point',(0.,0.,0.));
        #3532=DIRECTION('local z',(-1.,0.,0.));
        #3531=DIRECTION('local x',(0.,0.,1.));
  #158=JOINT_SYSTEM_WELDED_LINEAR(1000,'Fillet','5/16 Fillet 5 15/16 long',
                                  .SHOP_PROCESS.,#157,#95);
    #157=WELD_MECHANISM(1,'item_name',$,$,.FILLET_WELD.,.FULL_PENETRATION.,$,$,$);
    #95=COMPOSITE_CURVE('Weld Path',(#1376),.F.);
      #1376=COMPOSITE_CURVE_SEGMENT(.CONTINUOUS.,.T.,#1271);
        #1271=POLYLINE('weld',(#2731,#2732));
          #2731=CARTESIAN_POINT('weld vertex',(0.,0.127,0.127));
          #2732=CARTESIAN_POINT('weld vertex',(150.622,0.127,0.127));
  #4027=LOCATED_ASSEMBLY_MARKED(1,'C_5[1]','column',#3990,$,#2657,#4072,
                                '[1]',$,$,'C_5',$);
```

FIG. 23a: CIS/2 weld

Similar to how bolts are modeled in IFC, IfcFastenerType and IfcFastener refer to the geometry of the weld path and the position of the weld as shown in Fig. 23b. The weld path is defined by an IfcPolyline. Other than the geometry of the weld path, there is no other information in IFC that can describe a weld.

```
#5627= IFCFASTENERTYPE('guid',#100005,'5/16 Fillet 5 15/16 long Shop',
                       $,$,$,(#5628),$,'Weld');
  #5628= IFCREPRESENTATIONMAP(#100040,#5625);
    #5625= IFCSHAPEREPRESENTATION(#100011,'Body','GeometricCurveSet',(#5626));
      #5626= IFCGEOMETRICSET((#1271));
        #1271= IFCPOLYLINE((#2731,#2732));
          #2731= IFCCARTESIANPOINT((0.,0.127,0.127));
          #2732= IFCCARTESIANPOINT((150.622,0.127,0.127));
```

FIG. 23b: IFC weld (continued next page)

```
#5707= IFCFASTENER('guid',#100005,'5/16 Fillet 5 15/16 long Shop',
                   'Weld','Weld',#1605,#5708,$);
  #1605= IFCLOCALPLACEMENT(#3990,#3627);
    #3990= IFCLOCALPLACEMENT($,#3618);
      #3618= IFCAXIS2PLACEMENT3D(#2684,#3532,#3531);
        #2684= IFCCARTESIANPOINT((0.,0.,0.));
        #3532= IFCDIRECTION((-1.,0.,0.));
        #3531= IFCDIRECTION((0.,0.,1.));
    #3627= IFCAXIS2PLACEMENT3D(#2733,#3539,#3537);
      #2733= IFCCARTESIANPOINT((25.4,178.371498062134,322.961003875732));
      #3539= IFCDIRECTION((1.,0.,0.));
      #3537= IFCDIRECTION((0.,-1.,0.));
  #5708= IFCPRODUCTDEFINITIONSHAPE($,$,(#5709));
    #5709= IFCSHAPEREPRESENTATION(#100011,'Body','MappedRepresentation',(#5710));
      #5710= IFCMAPPEDITEM(#5628,#100059);
        #5628= IFCREPRESENTATIONMAP(#100040,#5625);

#10573= IFCRELDEFINESBYTYPE('guid',#100005,'Weld',$,(#5707),#5627);
```

FIG. 23b: IFC weld

C.6 Holes

Holes in CIS/2 are applied to parts similar to how cutouts are applied. Fig. 24 shows how the hole depth (Feature_volume_curved), hole radius (Feature_volume_hole_circular), and the layout of holes (Feature_volume_with_layout) is specified. The layout of holes is located relative to the part coordinate system.

In IFC, it is possible to generate the geometry of a part that shows holes penetrating the part; however, there is no method to specify a layout of holes. No IFC example of holes is shown.

```
#904=LOCATED_FEATURE_FOR_LOCATED_PART(0,'hole','1 1/16 Std Round',#1596,#834,#2484);
  #1596=(COORD_SYSTEM('Local','Feature CS',$,3)
         COORD_SYSTEM_CARTESIAN_3D(#3620)
        COORD_SYSTEM_CHILD(#1595));
    #3620=AXIS2_PLACEMENT_3D('Feature CS',#2685,#3536,#3535);
      #2685=CARTESIAN_POINT('axis2_placement_3d point',(12.7,0.,0.));
      #3536=DIRECTION('local z',(0.,0.,1.));
      #3535=DIRECTION('local x',(-1.,0.,0.));
    #1595=(COORD_SYSTEM('Local','Part CS,$,3)COORD_SYSTEM_CARTESIAN_3D(#3619)
          COORD_SYSTEM_CHILD(#3990));
      #3619=AXIS2_PLACEMENT_3D('Part CS',#2685,#3534,#3533);
        #2685=CARTESIAN_POINT('axis2_placement_3d point',(12.7,0.,0.));
        #3534=DIRECTION('local z',(0.,0.,-1.));
        #3533=DIRECTION('local x',(1.,0.,0.));
      #3990=COORD_SYSTEM_CARTESIAN_3D('Global','Assembly CS',$,3,#3618);
        #3618=AXIS2_PLACEMENT_3D('Assembly CS',#2684,#3532,#3531);
          #2684=CARTESIAN_POINT('axis2_placement_3d point',(0.,0.,0.));
          #3532=DIRECTION('local z',(-1.,0.,0.));
          #3531=DIRECTION('local x',(0.,0.,1.));
  #834=(FEATURE()FEATURE_VOLUME()FEATURE_VOLUME_CURVED(#1267)FEATURE_VOLUME_HOLE()
        FEATURE_VOLUME_HOLE_CIRCULAR(#1506)
        FEATURE_VOLUME_WITH_LAYOUT((#2690,#2691,#2692,#2693)));
    #1267=POLYLINE('hole depth',(#2694,#2695));
      #2694=CARTESIAN_POINT('hole depth pt1',(0.,0.,0.));
      #2695=CARTESIAN_POINT('hole depth pt2',(-25.4,0.,0.));
    #1506=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(13.49375),#1504);
    #2690=CARTESIAN_POINT('hole loc',(0.,228.6,-457.2));
    #2691=CARTESIAN_POINT('hole loc',(0.,-228.6,-457.2));
    #2692=CARTESIAN_POINT('hole loc',(0.,228.6,457.2));
    #2693=CARTESIAN_POINT('hole loc',(0.,-228.6,457.2));
  #2484=LOCATED_PART_MARKED(2100,'BP2','System connection material: Column Base Plate',
                            #1595,#1578,#4027,'BP2','ABM Page:1 Line:1',$,6,.F.);
```

FIG. 24: CIS/2 holes

APPENDIX D - STRUCTURAL ANALYSIS MODEL EXAMPLES

The examples in Appendix D show how CIS/2 structural analysis model entities are mapped to IFC entities. A discussion about the CIS/2 and IFC structural analysis model is in section 1.7.

D.1 Linear Elements

In CIS/2, an Element_curve_simple (#8) analysis element is referred to by two Element_node_connectivity (#13, #15) as shown in Fig. 25a. The connectivity defines the 'Start Node' and 'End Node' of the analysis element. The use of those specific strings is required on Element_node_connectivity. The Node (#4, #7) for each connectivity is defined by a 3-dimensional Cartesian_point. The position of the nodes at each end defines the location and length of the analysis element. Element_curve_simple refers to a Section_profile and a Direction vector which specifies the element orientation. The element orientation is relative to the longitudinal axis of the element defined by the start and end nodes and can also be specified by an angle instead of a direction. Each Node and Element is also part of an Analysis_model. The physical representation of an analysis model can be implied from the cross section, length, position, and orientation of the analysis elements.

The optional Boundary_condition_logical refers to the fixity of the six degrees-of-freedom of a node that can be free (.T.) or fixed (.F.). The optional Release_logical refers to the fixity of the six degrees-of-freedom at either end of the analysis element. The boundary conditions and releases in the following example are for illustration purposes only and do not necessarily make sense for a real analysis model. Instead of fixed or free boundary and release conditions, specific spring values can be specified with Boundary_condition_spring_linear and Release_spring_linear. The Element_with_material also refers to the name of a material defined by Material.

```
#13=ELEMENT_NODE_CONNECTIVITY(1,'Start Node',#4,#8,$,#12);
  #4=NODE('1',#2,#3,#1);
    #2=CARTESIAN_POINT('Node point',(0.,0.,0.));
    #3=BOUNDARY_CONDITION_LOGICAL($,$,.T.,.T.,.T.,.T.,.T.,.T.);
    #1=(ANALYSIS_MODEL('Analysis Model',$,.SPACE_FRAME.,$,3);
  #8=(ELEMENT('E1',$,#1,3)ELEMENT_CURVE($)ELEMENT_CURVE_SIMPLE(#9,#11)
      ELEMENT_WITH_MATERIAL(#10));
    #1=(ANALYSIS_MODEL('Analysis Model',$,.SPACE_FRAME.,$,3);
    #9=SECTION_PROFILE(1,'C10X15.3',$,$,5,.T.);
    #11=DIRECTION('Beam normal',(1.,0.,0.));
    #10=MATERIAL(2,'S235JRG2','beam material');
  #12=RELEASE_LOGICAL($,$,.F.,.F.,.F.,.T.,.T.,.T.);

#15=ELEMENT_NODE_CONNECTIVITY(2,'End Node',#7,#8,$,#14);
  #7=NODE('2',#5,#6,#1);
    #5=CARTESIAN_POINT('Node point',(1000.,0.,0.));
    #6=BOUNDARY_CONDITION_LOGICAL($,$,.F.,.F.,.F.,.T.,.T.,.T.);
    #1=(ANALYSIS_MODEL('Analysis Model',$,.SPACE_FRAME.,$,3);
  #8=(ELEMENT('E1',$,#1,3)ELEMENT_CURVE($)ELEMENT_CURVE_SIMPLE(#9,#11)
      ELEMENT_WITH_MATERIAL(#10));
    #1=(ANALYSIS_MODEL('Analysis Model',$,.SPACE_FRAME.,$,3);
    #9=SECTION_PROFILE(1,'C10X15.3',$,$,5,.T.);
    #11=DIRECTION('Beam normal',(1.,0.,0.));
    #10=MATERIAL(2,'S235JRG2','beam material');
  #14=RELEASE_LOGICAL($,$,.F.,.F.,.F.,.F.,.T.,.T.);
```

FIG. 25a: CIS/2 linear analysis element

In IFC, an IfcStructuralCurveMember (#8) analysis element is referred to by two IfcRelConnectsStructuralMember (#13, #15) that define the element connectivity as shown in Fig. 25b. The element connectivity also refers to two IfcStructuralPointConnection (#4, #7).

In IFC, the topology representation for analysis elements and nodes also has to be defined. The topology representation of a node is an IfcVertexPoint which refers to the location of the node defined by IfcCartesianPoint. The topology representation of an element is an IfcEdge which refers to each IfcVertexPoint at either end of the edge. The optional placement of the topology representation, for every IfcStructuralCurveMember and IfcStructuralPointConnection, is defined by an IfcLocalPlacement that is the world coordinate system.

IfcRelAssociatesProfileProperties, through IfcGeneralProfileProperties, associates the analysis element to a section profile. IfcRelAssociatesMaterial associates a material to the analysis element. IfcRelAssignsToGroup assigns the analysis element and nodes to the analysis model defined by IfcStructuralAnalysisModel.

```
#13= IFCRELCONNECTSSTRUCTURALMEMBER('guid',#2005,'E1','Start',#8,#4,#12,$,$,$);
  #8= IFCSTRUCTURALCURVEMEMBER('guid',#2005,'E1',$,'C10X15.3',#208,#119,.NOTDEFINED.);
    #208= IFCLOCALPLACEMENT(#4025,#4040);
      #4025= IFCLOCALPLACEMENT($,#4040);
        #4040= IFCAXIS2PLACEMENT3D(#4041,#4044,#4042);
          #4041= IFCCARTESIANPOINT((0.,0.,0.));
          #4044= IFCDIRECTION((0.,0.,1.));
          #4042= IFCDIRECTION((1.,0.,0.));
    #119= IFCPRODUCTREPRESENTATION($,$,(#120));
      #120= IFCTOPOLOGYREPRESENTATION(#2012,$,'Edge',(#121));
        #121= IFCEDGE(#115,#118);
          #115= IFCVERTEXPOINT(#2);
            #2= IFCCARTESIANPOINT((0.,0.,0.));
          #118= IFCVERTEXPOINT(#5);
            #5= IFCCARTESIANPOINT((1000.,0.,0.));
  #4= IFCSTRUCTURALPOINTCONNECTION('guid',#2005,'1',$,$,#215,#113,#3);
    #215= IFCLOCALPLACEMENT(#4025,#4040);
    #113= IFCPRODUCTREPRESENTATION($,$,(#114));
      #114= IFCTOPOLOGYREPRESENTATION(#2012,$,'Vertex',(#115));
        #115= IFCVERTEXPOINT(#2);
          #2= IFCCARTESIANPOINT((0.,0.,0.));
    #3= IFCBOUNDARYNODECONDITION('Node BC: TTTTTT',0.,0.,0.,0.,0.,0.);
  #12= IFCBOUNDARYNODECONDITION('Element release: FFFTTT',-1.,-1.,-1.,0.,0.,0.);

#15= IFCRELCONNECTSSTRUCTURALMEMBER('guid',#2005,'E1','End',#8,#7,#14,$,$,$);
  #8= IFCSTRUCTURALCURVEMEMBER('guid',#2005,'E1',$,'C10X15.3',#208,#119,.NOTDEFINED.);
    #208= IFCLOCALPLACEMENT(#4025,#4040);
    #119= IFCPRODUCTREPRESENTATION($,$,(#120));
      #120= IFCTOPOLOGYREPRESENTATION(#2012,$,'Edge',(#121));
        #121= IFCEDGE(#115,#118);
  #7= IFCSTRUCTURALPOINTCONNECTION('guid',#2005,'2',$,$,#219,#116,#6);
    #116= IFCPRODUCTREPRESENTATION($,$,(#117));
      #117= IFCTOPOLOGYREPRESENTATION(#2012,$,'Vertex',(#118));
        #118= IFCVERTEXPOINT(#5);
    #6= IFCBOUNDARYNODECONDITION('Node BC: FFFTTT',-1.,-1.,-1.,0.,0.,0.);
  #14= IFCBOUNDARYNODECONDITION('Element release: FFFFTT',-1.,-1.,-1.,-1.,0.,0.);

#229= IFCRELASSOCIATESPROFILEPROPERTIES('guid',#4005,$,$,(#8),#201,$,$);
  #8= IFCSTRUCTURALCURVEMEMBER('guid',#4005,'E1',$,'C10X15.3',$,#119,.NOTDEFINED.);
  #201= IFCGENERALPROFILEPROPERTIES('C10X15.3',#9,$,$,$,$,$);
    #9= IFCUSHAPEPROFILEDEF(.AREA.,'C10X15.3',#4050,254.0,66.04,6.096,11.0744,$,$,$,$);

#127= IFCRELASSOCIATESMATERIAL('guid',#2005,'S235JRG2 beam material',$,(#8),#10);
  #8= IFCSTRUCTURALCURVEMEMBER('guid',#2005,'E1',$,'C10X15.3',$,#119,.NOTDEFINED.);
  #10= IFCMATERIAL('S235JRG2 beam material');

#129= IFCRELASSIGNSTOGROUP('guid',#2005,'Analysis Model',$,(#4,#7,#8),.PRODUCT.,#1);
  #4= IFCSTRUCTURALPOINTCONNECTION('guid',#2005,'1',$,$,$,#113,#3);
  #7= IFCSTRUCTURALPOINTCONNECTION('guid',#2005,'2',$,$,$,#116,#6);
  #8= IFCSTRUCTURALCURVEMEMBER('guid',#2005,'E1',$,'C10X15.3',$,#119,.NOTDEFINED.);
  #1= IFCSTRUCTURALANALYSISMODEL('guid',#2005,'Analysis Model',$,$,.LOADING_3D.,$,$,$);
```

FIG. 25b: IFC linear analysis element

Different IfcBoundaryNodeCondition are used to define both the fixity of the degrees of freedom of the nodes and of the ends of the analysis element. The values for fixity can be free (0.), fixed (-1.), or a spring stiffness defined by a value greater than zero.

In IFC, the physical representation of an analysis element can also be explicitly defined. The physical representation considers the cross section dimensions and length of the analysis element. The physical element can be defined by an Ifc{Beam/Column/Member} similar to the design model example in Fig. 13b and is shown in Fig. 25c. Based on the coordinates of the nodes at the ends of an element and element orientation, a coordinate system defining the position and orientation of the element can be computed and defined by an IfcLocalPlacement. The element length is defined on IfcExtrudedAreaSolid.

An analysis model element (IfcStructuralCurveMember) can be associated with a physical element with IfcRelConnectsStructuralElement. Assembly_map is a CIS/2 equivalent of IfcRelConnectsStructuralElement.

```
#105= IFCBEAM('guid',#2005,'E1','C10X15.3',$,#106,#107,’E1’);
  #106= IFCLOCALPLACEMENT($,#112);
    #112= IFCAXIS2PLACEMENT3D(#2,#110,#111);
      #2= IFCCARTESIANPOINT((0.,0.,0.));
      #110= IFCDIRECTION((0.,0.,1.));
      #111= IFCDIRECTION((1.,0.,0.));
  #107= IFCPRODUCTDEFINITIONSHAPE($,$,(#108));
    #108= IFCSHAPEREPRESENTATION(#2011,'Body','SweptSolid',(#109));
      #109= IFCEXTRUDEDAREASOLID(#9,#2049,#2044,1000.);
        #9= IFCUSHAPEPROFILEDEF(.AREA.,'C10X15.3',#2050,254.0,66.04,6.096,11.0744,
                                $,$,$,$);

#122= IFCRELCONNECTSSTRUCTURALELEMENT('guid',#2005,$,$,#105,#8);
  #105= IFCBEAM('guid',#2005,'E1','C10X15.3',$,#106,#107,’E1’);
  #8= IFCSTRUCTURALCURVEMEMBER('guid',#2005,'E1',$,'C10X15.3',$,#119,.NOTDEFINED.);
```

FIG. 25c: IFC physical representation of an analysis element

D.2 Element Eccentricity

Fig. 26a shows how Element_eccentricity is used in CIS/2 to define the offset (eccentricity) of an analysis element from its connecting node. In this example, the offset is in the Z direction and the amount is defined by Length_measure_with_unit.

```
#13=ELEMENT_NODE_CONNECTIVITY(1,'Start Node',#4,#8,#50,#12);
  #4=NODE('1',#2,#3,#1);
    #2=CARTESIAN_POINT('Node point',(0.,0.,0.));
  #8=(ELEMENT('E1',$,#1,3)ELEMENT_CURVE($)ELEMENT_CURVE_SIMPLE(#9,#11)
      ELEMENT_WITH_MATERIAL(#10));
  #50=ELEMENT_ECCENTRICITY('1',$,$,#51);
    #51=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(50.),#42);
  #12=RELEASE_LOGICAL($,$,.F.,.F.,.F.,.F.,.F.,.F.);
```

FIG. 26a: CIS/2 analysis node with eccentricity of 50 units in the Z direction

Fig. 26b shows how IfcRelConnectsWithEccentricity is used, instead of IfcRelConnectsStructuralMember, to define an offset (eccentricity) of an analysis element from its connecting node. The value of the offset is defined on IfcConnectionPointEccentricity and is applied to the IfcVertexPoint. IfcRelConnectsWithEccentricity and IfcConnectionPointEccentricity are only available in IFC version 2x3 or higher. There is no way to explicitly assign element eccentricity with previous versions of IFC. Currently, IfcConnectionPointEccentricity does not define the coordinate system that the eccentricity is defined in.

```
#13= IFCRELCONNECTSWITHECCENTRICITY('guid',#2005,'E1','Start',#8,#4,#12,$,$,$,#119);
  #8= IFCSTRUCTURALCURVEMEMBER('guid',#2005,'E1',$,'C10X15.3',$,#115,.NOTDEFINED.);
  #4= IFCSTRUCTURALPOINTCONNECTION('guid',#2005,'1',$,$,#107,#108,#3);
  #12= IFCBOUNDARYNODECONDITION('Release FFFFFF',$,$,$,$,$,$);
  #119= IFCCONNECTIONPOINTECCENTRICITY(#110,$,0.0,0.0,50.0);
    #110= IFCVERTEXPOINT(#2);
      #2= IFCCARTESIANPOINT((0.,0.,0.));
```

FIG. 26b: IFC analysis node with eccentricity of 50 units in the Z direction

D.3 Element Orientation

Element orientation is the rotation of the section profile about the element's locating longitudinal axis. In Fig. 27a the element orientation is specified by an orientation vector defined by Direction. Alternatively, the element orientation can be specified by an angle with Plane_angle_measure_with_unit.

```
#8=(ELEMENT('E1',$,#1,3)ELEMENT_CURVE($)ELEMENT_CURVE_SIMPLE(#9,#11));
  #1=(ANALYSIS_MODEL('Analysis Model',$,.SPACE_FRAME.,$,3);
  #9=SECTION_PROFILE(1,'C10X15.3',$,$,5,.F.);
  #11=DIRECTION('Beam normal',(0.866,-0.5,0.));
```

FIG. 27a: CIS/2 analysis element orientation of 30 degrees about longitudinal axis

In Fig.27b, IfcRelAssociatesProfileProperties is used to associate an element orientation defined by IfcPlaneAngleMeasure to an analysis element defined by IfcStructuralCurveMember and its corresponding physical representation IfcBeam. The element orientation can also be defined by a vector with IfcDirection similar to how it is specified in CIS/2. Assigning the element orientation vector with IfcRelAssociatesProfileProperties is only available in IFC version 2x3 or higher. There is no way to explicitly assign an element orientation vector in previous versions of IFC.

```
#130= IFCRELASSOCIATESPROFILEPROPERTIES('guid',#2005,’Beta angle: 30.’,
                                      $,(#105,#8),#129,$,IFCPLANEANGLEMEASURE(30.));
  #105= IFCBEAM('guid',#2005,'E1',$,'C10X15.3',#106,#107,$);
  #8= IFCSTRUCTURALCURVEMEMBER('guid',#2005,'E1',$,'C10X15.3',$,#119,.NOTDEFINED.);
  #129= IFCGENERALPROFILEPROPERTIES('C10X15.3',$,$,$,$,$,$);
```

FIG. 27b: IFC analysis element orientation of 30 degrees about longitudinal axis

D.4 Surface Elements

Fig. 28a shows how in an analysis model a 3-noded surface element is modeled with Element_surface_simple similar to how a linear analysis element is modeled in Fig. 25a. Each Element_surface_simple is referred to by three Element_node_connectivity. In practice, surface elements in analysis models have not been implemented in CIS/2.

```
#20=ELEMENT_NODE_CONNECTIVITY(1,'Start Node',#7,#13,$,#34);
  #7=NODE('A1',#1,#32,#1642);
    #1=CARTESIAN_POINT('nodepoint1',(0.0,0.0,0.0));
    #32=BOUNDARY_CONDITION_LOGICAL('PINNED',$,.F.,.U.,.F.,.U.,.T.,.U.);
    #1642=ANALYSIS_MODEL('my model',$,.PLANE_FRAME.,$,2);
  #13=ELEMENT_SURFACE_SIMPLE('E1',$,#1642,2,#39,.TRIANGLE.,.PLANE_STRAIN.);
    #1642=ANALYSIS_MODEL('my model',$,.PLANE_FRAME.,$,2);
    #39=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(5.0),#1645);
  #34=RELEASE_LOGICAL('FIXED END',$,.F.,.U.,.F.,.U.,.F.,.U.);

#21=ELEMENT_NODE_CONNECTIVITY(2,'Second Node',#8,#13,$,#34);
  #8=NODE('A2',#2,#32,#1642);
    #2=CARTESIAN_POINT('nodepoint2',(50.,0.,0.));
    #32=BOUNDARY_CONDITION_LOGICAL('PINNED',$,.F.,.U.,.F.,.U.,.T.,.U.);
    #1642=ANALYSIS_MODEL('my model',$,.PLANE_FRAME.,$,2);
  #13=ELEMENT_SURFACE_SIMPLE('E1',$,#1642,2,#39,.TRIANGLE.,.PLANE_STRAIN.);

#22=ELEMENT_NODE_CONNECTIVITY(3,'Third Node',#9,#13,$,#34);
  #9=NODE('A3',#3,#32,#1642);
    #3=CARTESIAN_POINT('nodepoint3',(50.0,0.,100.0));
    #32=BOUNDARY_CONDITION_LOGICAL('PINNED',$,.F.,.U.,.F.,.U.,.T.,.U.);
    #1642=ANALYSIS_MODEL('my model',$,.PLANE_FRAME.,$,2);
  #13=ELEMENT_SURFACE_SIMPLE('E1',$,#1642,2,#39,.TRIANGLE.,.PLANE_STRAIN.);
```

FIG. 28a: CIS/2 analysis model surface element

Fig. 28b shows how, in an IFC analysis model, the equivalent 3-noded surface element is modeled with IfcStructuralSurfaceMember similar to how a linear analysis element is modeled in Fig. 25b. Each IfcStructuralSurfaceMember is referred to by three IfcRelConnectsStructuralMember to define its connectivity. The topological representation of a surface element is an IfcFace which eventually refers to IfcOrientedEdge. IfcOrientedEdge refers to IfcEdge which refers to IfcVertexPoint. IfcEdge and IfcVertexPoint are used for the topological representation of a linear analysis element. The physical representation of a surface element could be an IfcPlate, however, an example is not provided.

```
#20= IFCRELCONNECTSSTRUCTURALMEMBER('guid',#40005,'Start',$,#13,#7,#34,$,$,$);
  #13= IFCSTRUCTURALSURFACEMEMBER('guid',#40005,$,$,$,$,#2011,.SHELL.,5.0);
    #2011= IFCPRODUCTREPRESENTATION($,$,(#2012));
      #2012= IFCTOPOLOGYREPRESENTATION(#40012,$,'Face',(#2013));
        #2013= IFCFACE((#2014));
          #2014= IFCFACEBOUND(#2015,.T.);
            #2015= IFCEDGELOOP((#2016,#2017,#2018));
              #2016= IFCORIENTEDEDGE(*,*,#2019,.T.);
                #2019= IFCEDGE(#2004,#2007);
                  #2004= IFCVERTEXPOINT(#1);
                  #2007= IFCVERTEXPOINT(#2);
              #2017= IFCORIENTEDEDGE(*,*,#2020,.T.);
                #2020= IFCEDGE(#2007,#2010);
                  #2007= IFCVERTEXPOINT(#2);
                  #2010= IFCVERTEXPOINT(#3);
              #2018= IFCORIENTEDEDGE(*,*,#2021,.T.);
                #2021= IFCEDGE(#2010,#2004);
                  #2010= IFCVERTEXPOINT(#3);
                  #2004= IFCVERTEXPOINT(#1);

  #7= IFCSTRUCTURALPOINTCONNECTION('guid',#40005,'A1',$,$,$,#2002,#32);
    #2002= IFCPRODUCTREPRESENTATION($,$,(#2003));
      #2003= IFCTOPOLOGYREPRESENTATION(#40012,$,'Vertex',(#2004));
        #2004= IFCVERTEXPOINT(#1);
          #1= IFCCARTESIANPOINT((0.0,0.0,0.0));
    #32= IFCBOUNDARYNODECONDITION('Boundary condition PINNED FUFUTU',-1.,$,-1.,$,0.,$);
  #34= IFCBOUNDARYNODECONDITION('Release FIXED END FUFUFU',$,$,$,$,$,$);

#21= IFCRELCONNECTSSTRUCTURALMEMBER('guid',#40005,'Second',$,#13,#8,$,$,$,$);
  #13= IFCSTRUCTURALSURFACEMEMBER('guid',#40005,$,$,$,$,#2011,.SHELL.,5.0);
  #8= IFCSTRUCTURALPOINTCONNECTION('guid,#40005,'A2',$,$,$,#2005,#32);
    #2005= IFCPRODUCTREPRESENTATION($,$,(#2006));
      #2006= IFCTOPOLOGYREPRESENTATION(#40012,$,'Vertex',(#2007));
        #2007= IFCVERTEXPOINT(#2);
          #2= IFCCARTESIANPOINT((50.,0.,0.));

#22= IFCRELCONNECTSSTRUCTURALMEMBER('guid',#40005,'Third',$,#13,#9,$,$,$,$);
  #13= IFCSTRUCTURALSURFACEMEMBER('guid',#40005,$,$,$,$,#2011,.SHELL.,5.0);
  #9= IFCSTRUCTURALPOINTCONNECTION('guid',#40005,'A3',$,$,$,#2008,#32);
    #2008= IFCPRODUCTREPRESENTATION($,$,(#2009));
      #2009= IFCTOPOLOGYREPRESENTATION(#40012,$,'Vertex',(#2010));
        #2010= IFCVERTEXPOINT(#3);
          #3= IFCCARTESIANPOINT((50.0,0.,100.0));
```

FIG. 28b: IFC analysis model surface element

D.5 Analysis Loads

An analysis model can have a variety of applied loads on either elements or nodes. Fig. 29a shows a uniformly distributed load on an analysis element in CIS/2. Load_element_distributed_curve_line defines the load at both ends of the element. The load also refers to a Load_case. A non-uniform load can be defined by having different load values at each end of the element. A load on only a section of the element can be defined with a Line that does not start or end at either end of the element.

```
#462=LOAD_ELEMENT_DISTRIBUTED_CURVE_LINE(#457,'MbLd_1',$,#156,$,$,$,.GLOBAL_LOAD.,
                                         .TRUE_LENGTH.,#464,#464,#467);
  #457=LOAD_CASE('Dead load',$,(#21),#20);
    #21=ANALYSIS_METHOD_STATIC('1st order',$,.ELASTIC_1ST_ORDER.);
    #20=PHYSICAL_ACTION(.STATIC.,.FIXED_ACTION.,.DIRECT_ACTION.,$,$,(1.0),(' '));

  #156=ELEMENT_CURVE_SIMPLE('286','desc',#1,1,1,#157,$);
    #1=ANALYSIS_MODEL('Loads and Results','Exported from GTSTRUDL',.SPACE_FRAME.,$,3);
    #157=SECTION_PROFILE(0,'W27X194',$,$,10,.F.);

  #464=APPLIED_LOAD_STATIC_FORCE('For_Y',$,#466,$,$,$,$);
    #466=FORCE_MEASURE_WITH_UNIT(FORCE_MEASURE(-16.148100),#7);
      #7=(CONTEXT_DEPENDENT_UNIT('POUNDS')FORCE_UNIT()NAMED_UNIT(#8));
        #8=DIMENSIONAL_EXPONENTS(1.,1.,-2.,0.,0.,0.,0.);

  #464=APPLIED_LOAD_STATIC_FORCE('For_Y',$,#466,$,$,$,$);
    #466=FORCE_MEASURE_WITH_UNIT(FORCE_MEASURE(-16.148100),#7);

  #467=LINE('Member centroid',#468,#469);
    #468=CARTESIAN_POINT('X_local:start',(0.,0.0,0.0));
    #469=VECTOR('X_local:length',#15,300.);
      #15=DIRECTION('Local X',(1.0,0.0,0.0));
```

FIG. 29a: CIS/2 analysis model uniform element load

In IFC, IfcStructuralLinearAction is used to apply a constant linear action on an analysis element as shown in Fig. 29b. The value of the load is defined by IfcStructuralLoadLinearForce. Each IfcStructuralLinearAction also refers to a topology representation, in this case an IfcEdge. IfcRelConnectsStructuralActivity associates the load with the analysis element defined by IfcStructuralCurveMember. IfcRelAssignsToGroup associates multiple loads with the load case defined by IfcStructuralLoadGroup. IfcStructuralLinearActionVarying (not shown) can be used for loads that are non-uniform and vary along the element.

```
#462= IFCSTRUCTURALLINEARACTION('guid',#5,'MbLd_1','Load',$,$,
                                #6028,#6375,.GLOBAL_COORDS.,.F.,$,.TRUE_LENGTH.);
  #6028= IFCPRODUCTREPRESENTATION($,$,(#6029));
    #6029= IFCTOPOLOGYREPRESENTATION(#120012,$,'Edge',(#6030));
      #6030= IFCEDGE(#6024,#6027);
  #6375= IFCSTRUCTURALLOADLINEARFORCE('Load',$,-16.148,$,$,$,$);

#6376= IFCRELCONNECTSSTRUCTURALACTIVITY('guid',#5,'MbLd_1','Load line',#156,#462);
  #156= IFCSTRUCTURALCURVEMEMBER('guid',#5,'286',$,'W27X194',$,#6028,.NOTDEFINED.);
  #462= IFCSTRUCTURALLINEARACTION('guid',#5,'MbLd_1','Load',$,$,#6028,#6375,
                                  .GLOBAL_COORDS.,.F.,$,.TRUE_LENGTH.);

#6411= IFCRELASSIGNSTOGROUP('guid',#5,'Dead load','Load case',(#462),.PRODUCT.,#457);
  #462= IFCSTRUCTURALLINEARACTION('guid',#5,'MbLd_1','Load',$,$,#6028,#6375,
                                  .GLOBAL_COORDS.,.F.,$,.TRUE_LENGTH.);
  #457= IFCSTRUCTURALLOADGROUP('guid',#5,'Dead load',$,$,
                               .LOAD_CASE.,.PERMANENT_G.,.NOTDEFINED.,$,$);
```

FIG. 29b: IFC analysis model uniform element load

In CIS/2, a concentrated load applied to an analysis node is shown in Fig. 30a. The load is defined by Load_node and its value by Applied_load_static_force. The nodal load also refers to a load case.

```
#806=LOAD_NODE(#458,'JtLd_1',$,#100,#807);
  #458=LOAD_CASE('2::Wind from -Y as joint loads',$,(#21),#20);
    #21=ANALYSIS_METHOD_STATIC('1st order',$,.ELASTIC_1ST_ORDER.);
    #20=PHYSICAL_ACTION(.STATIC.,.FIXED_ACTION.,.DIRECT_ACTION.,$,$,(1.0),(' '));
  #100=NODE('node_1',#101,$,#1);
    #101=CARTESIAN_POINT('node_1',(180.,0.,150.));
    #1=ANALYSIS_MODEL('Loads and Results','Exported from GTSTRUDL',.SPACE_FRAME.,$,3);
  #807=APPLIED_LOAD_STATIC_FORCE('Joint load',$,#808,$,$,$,$);
    #808=FORCE_MEASURE_WITH_UNIT(FORCE_MEASURE(1000.),#7);
      #7=(CONTEXT_DEPENDENT_UNIT('POUNDS')FORCE_UNIT()NAMED_UNIT(#8));
        #8=DIMENSIONAL_EXPONENTS(1.,1.,-2.,0.,0.,0.,0.);
```

FIG. 30a: CIS/2 analysis model concentrated nodal load

In IFC, IfcStructuralPointAction is used to apply a point action on an analysis node as shown in Fig. 21b. The value of the load is defined by IfcStructuralLoadSingleForce. Each IfcStructuralPointAction also refers to a topology representation, in this case an IfcVertex. IfcRelConnectsStructuralActivity associates the load with the analysis node defined by IfcStructuralPointConnection. IfcRelAssignsToGroup associates multiple loads with the load case defined by IfcStructuralLoadGroup.

```
#806= IFCSTRUCTURALPOINTACTION('guid',#120005,'JtLd_1','Load',$,$,#6168,#807,
                             .GLOBAL_COORDS.,.F.,$);
  #6168= IFCPRODUCTREPRESENTATION($,$,(#6169));
    #6169= IFCTOPOLOGYREPRESENTATION(#120012,$,'Vertex',(#6170));
      #6170= IFCVERTEXPOINT(#101);
  #807= IFCSTRUCTURALLOADSINGLEFORCE('Load',$,1000.,$,$,$,$);

#6275= IFCRELCONNECTSSTRUCTURALACTIVITY('guid',#120005,'JtLd_1','Load point',#100,#806);
  #100= IFCSTRUCTURALPOINTCONNECTION('guid',#120005,'node_1',$,$,$,#6168,$);
  #806= IFCSTRUCTURALPOINTACTION('guid',#120005,'JtLd_1','Load',$,$,
                              #6168,#807,.GLOBAL_COORDS.,.F.,$);

#6412= IFCRELASSIGNSTOGROUP('guid',#120005,'2::Wind from -Y as joint loads',
                          'Load case',(#806,...),.PRODUCT.,#458);
  #806= IFCSTRUCTURALPOINTACTION('guid',#120005,'JtLd_1','Load',$,$,
                              #6168,#807,.GLOBAL_COORDS.,.F.,$);
  #458= IFCSTRUCTURALLOADGROUP('guid',#120005,'2::Wind from -Y as joint loads',
                             $,$,.LOAD_CASE.,.PERMANENT_G.,.WIND_W.,$,$);
```

FIG. 30b: IFC analysis model concentrated nodal load

D.6 Analysis Results

An analysis model can have analysis results consisting of forces, moments, and displacements. In CIS/2, analysis results can be associated with the analysis nodes or with the element connectivity, i.e. the ends of an analysis element.

Fig. 31a is a CIS/2 example showing displacements and rotations associated with a node. Analysis_result_node associates the reactions (Reaction_displacement) with the analysis Node. The Reaction_displacement refers to the three components of displacement (Length_measure_with_unit) and rotation (Plane_angle_measure_with_unit). Analysis_results_set_basic is used to associate the analysis results with the load case.

```
#1034=ANALYSIS_RESULT_NODE('jt disp: 0',$,#21,#100,#1035);
  #21=ANALYSIS_METHOD_STATIC('1st order',$,.ELASTIC_1ST_ORDER.);
  #100=NODE('node_1',#101,$,#1);
    #101=CARTESIAN_POINT('node_1',(180.,0.,150.));
    #1=ANALYSIS_MODEL('Loads and Results','Exported from GTSTRUDL',.SPACE_FRAME.,$,3);
  #1035=REACTION_DISPLACEMENT(#1036,#1037,#1038,#1039,#1040,#1041);
    #1036=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.000469),#3);
      #3=(CONTEXT_DEPENDENT_UNIT('INCH')LENGTH_UNIT()NAMED_UNIT(#4));
        #4=DIMENSIONAL_EXPONENTS(1.,0.,0.,0.,0.,0.,0.);
    #1037=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(-1.575543),#3);
    #1038=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.000225),#3);
    #1039=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(1.002279),#5);
      #5=(CONTEXT_DEPENDENT_UNIT('DEGREE')NAMED_UNIT(#6)PLANE_ANGLE_UNIT());
        #6=DIMENSIONAL_EXPONENTS(0.,0.,0.,0.,0.,0.,0.);
    #1040=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.000008),#5);
    #1041=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.713858),#5);

#5674=ANALYSIS_RESULTS_SET_BASIC('Dead load',(#1034,...),#457);
  #1034=ANALYSIS_RESULT_NODE('jt disp: 0',$,#21,#100,#1035);
  #457=LOAD_CASE('Dead load',$,(#21),#20);
    #21=ANALYSIS_METHOD_STATIC('1st order',$,.ELASTIC_1ST_ORDER.);
    #20=PHYSICAL_ACTION(.STATIC.,.FIXED_ACTION.,.DIRECT_ACTION.,$,$,(1.0),(' '));
```

FIG. 31a: CIS/2 analysis model nodal displacement

In IFC, IfcStructuralPointReaction defines the nodal displacements and rotations at a node as shown in Fig. 31b. The force and moment results are defined by IfcStructuralLoadSingleDisplacement. The displacements and rotations are associated with the topology representation of the node (IfcVertex) whereas IfcRelConnectsStructuralActivity associates the displacements and rotations with IfcStructuralPointConnection.

IfcRelAssignsToGroup associates the analysis results with a results group (IfcStructuralResultGroup) that refers to the load case (IfcStructuralLoadGroup).

```
#1034= IFCSTRUCTURALPOINTREACTION('guid',#120005,'jt disp: 0','Result node',$,$,
                                  #6168,#1035,.GLOBAL_COORDS.);
  #6168= IFCPRODUCTREPRESENTATION($,$,(#6169));
    #6169= IFCTOPOLOGYREPRESENTATION(#120012,$,'Vertex',(#6170));
      #120012= IFCREPRESENTATIONCONTEXT('Mechanical Structure','Design');
      #6170= IFCVERTEXPOINT(#101);
        #101= IFCCARTESIANPOINT((180.,0.,150.));
  #1035= IFCSTRUCTURALLOADSINGLEDISPLACEMENT('Result',-0.000469,-1.575543,
                                             -0.000225,1.002279,0.000008,0.713858);

#6546= IFCRELCONNECTSSTRUCTURALACTIVITY('guid',#120005,
                                        'jt disp: 0','Result node',#100,#1034);
  #100= IFCSTRUCTURALPOINTCONNECTION('guid',#120005,'node_1',$,$,$,#6168,$);
  #1034= IFCSTRUCTURALPOINTREACTION('guid',#120005,'jt disp: 0','Result node',$,$,
                                   #6168,#1035,.GLOBAL_COORDS.);

#6999= IFCRELASSIGNSTOGROUP('guid',#120005,'Dead load','Result set',
                            (#1034),.PRODUCT.,#5674);
  #1034= IFCSTRUCTURALPOINTREACTION('guid',#120005,'jt disp: 0','Result node',$,$,
                                   #6168,#1035,.GLOBAL_COORDS.);
  #5674= IFCSTRUCTURALRESULTGROUP('guid',#120005,'Dead load',
                                 $,$,.FIRST_ORDER_THEORY.,#457,.T.);
    #457= IFCSTRUCTURALLOADGROUP('guid',#120005,'Dead load',
                                $,$,.LOAD_CASE.,.PERMANENT_G.,.NOTDEFINED.,$,$);
```

FIG. 31b: IFC analysis model nodal displacement

Fig. 32a shows nodal forces and moments at one end of a CIS/2 analysis element. Analysis_result_element_node refers to the start node of the element (Element_node_connectivity) and the reactions at that node (Reaction_force). The Reaction_force refers to the three components of force (Force_measure_with_unit) and three components of moment (Moment_measure_with_unit). Analysis_results_set_basic is used to associate the analysis results with the load case.

```
#2234=ANALYSIS_RESULT_ELEMENT_NODE('mb_start_force: 0',$,#21,#161,#2235);
  #161=ELEMENT_NODE_CONNECTIVITY(1,'Start Node',#130,#156,$,$);
  #2235=REACTION_FORCE(#2236,#2237,#2238,#2239,#2240,#2241);
    #2236=FORCE_MEASURE_WITH_UNIT(FORCE_MEASURE(4338.),#7);
      #7=(CONTEXT_DEPENDENT_UNIT('POUNDS')FORCE_UNIT()NAMED_UNIT(#8));
        #8=DIMENSIONAL_EXPONENTS(1.,1.,-2.,0.,0.,0.,0.);
    #2237=FORCE_MEASURE_WITH_UNIT(FORCE_MEASURE(-29.761),#7);
    #2238=FORCE_MEASURE_WITH_UNIT(FORCE_MEASURE(-7623.),#7);
    #2239=MOMENT_MEASURE_WITH_UNIT(MOMENT_MEASURE(6.689),#9);
      #9=MOMENT_UNIT((#10,#11));
        #10=DERIVED_UNIT_ELEMENT(#7, 1.0);
          #7=(CONTEXT_DEPENDENT_UNIT('POUNDS')FORCE_UNIT()NAMED_UNIT(#8));
            #8=DIMENSIONAL_EXPONENTS(1.,1.,-2.,0.,0.,0.,0.);
        #11=DERIVED_UNIT_ELEMENT(#3, 1.0);
          #3=(CONTEXT_DEPENDENT_UNIT('INCH')LENGTH_UNIT()NAMED_UNIT(#4));
            #4=DIMENSIONAL_EXPONENTS(1.,0.,0.,0.,0.,0.,0.);
    #2240=MOMENT_MEASURE_WITH_UNIT(MOMENT_MEASURE(938838.),#9);
    #2241=MOMENT_MEASURE_WITH_UNIT(MOMENT_MEASURE(-8927.),#9);

#5674=ANALYSIS_RESULTS_SET_BASIC('Dead load',(#2234,...),#457);
  #2234=ANALYSIS_RESULT_ELEMENT_NODE('mb_start_force: 0',$,#21,#161,#2235);
  #457=LOAD_CASE('Dead load',$,(#21),#20);
    #21=ANALYSIS_METHOD_STATIC('1st order',$,.ELASTIC_1ST_ORDER.);
    #20=PHYSICAL_ACTION(.STATIC.,.FIXED_ACTION.,.DIRECT_ACTION.,$,$,(1.0),(' '));
```

FIG. 32a: CIS/2 analysis model force and moment results at an element node

In IFC, there is no exact equivalent of the CIS/2 Analysis_result_element_node where analysis results are associated with the ends of an element (Element_node_connectivity) rather than an analysis node. There is no equivalent because IfcRelConnectsStructuralActivity can associate reactions to structural member, connections, or building elements and not to the element connectivity (IfcRelConnectsStructuralMember).

In Fig. 32b, IfcStructuralPointReaction defines the analysis results at a vertex. The force and moment results are defined by IfcStructuralLoadSingleForce which can also used to define load values as shown in Fig. 30b. The forces and moments are applied to the topology representation of the node (IfcVertex) whereas IfcRelConnectsStructuralActivity associates the analysis results with IfcStructuralCurveMember. Given the relationships and associations, the end of the analysis member where the reactions are applied can be determined. IfcRelAssignsToGroup associates the analysis results with a results group (IfcStructuralResultGroup) that refers to the load case (IfcStructuralLoadGroup).

```
#2234= IFCSTRUCTURALPOINTREACTION('guid',#120005,'mb_start_force: 0','Element node',$,$,
                                  #6022,#2235,.GLOBAL_COORDS.);
  #6022= IFCPRODUCTREPRESENTATION($,$,(#6023));
    #6023= IFCTOPOLOGYREPRESENTATION(#120012,$,'Vertex',(#6024));
      #120012= IFCREPRESENTATIONCONTEXT('Mechanical Structure','Design');
      #6024= IFCVERTEXPOINT(#131);
        #131= IFCCARTESIANPOINT((0.,0.,0.));
  #2235= IFCSTRUCTURALLOADSINGLEFORCE('Result',4338.,-29.761,-7623.,
                                    6.689,938838.,-8927.);

#6652= IFCRELCONNECTSSTRUCTURALACTIVITY('guid',#120005,
          'mb_start_force: 0','Element node',#130,#2234);
  #156= IFCSTRUCTURALCURVEMEMBER('guid',#120005,'286 desc','Undefined',
                                'W27X194',#6037,#6038,.NOTDEFINED.);
  #2234= IFCSTRUCTURALPOINTREACTION('guid',#120005,'mb_start_force: 0',
                                  'Element node',$,$,#6022,#2235,.GLOBAL_COORDS.);

#6999= IFCRELASSIGNSTOGROUP('guid',#120005,'Dead load','Result set',
                            (#2234,...),.PRODUCT.,#5674);
  #2234= IFCSTRUCTURALPOINTREACTION('guid',#120005,'mb_start_force: 0',
                                  'Element node',$,$,#6022,#2235,.GLOBAL_COORDS.);
  #5674= IFCSTRUCTURALRESULTGROUP('guid',#120005,'Dead load',
                                 $,$,.FIRST_ORDER_THEORY.,#457,.T.);
    #457= IFCSTRUCTURALLOADGROUP('guid',#120005,'Dead load',
                                $,$,.LOAD_CASE.,.PERMANENT_G.,.NOTDEFINED.,$,$);
```

FIG. 32b: IFC analysis model force and moment results at an element node

In CIS/2, Load_combination_occurrence is used to create a new load case from a combination of other load cases as shown in Fig. 33a. The new load case is the Loading_combination created from the combination of three other Load_case. Load_combination_occurrence also provides for a load combination factor.

```
#1024=LOAD_COMBINATION_OCCURRENCE(1.,#461,#457);
  #461=LOADING_COMBINATION('4','all loads with factor = 1',#1);
    #1=ANALYSIS_MODEL('Loads and Results','Exported from GTSTRUDL',.SPACE_FRAME.,$,3);
  #457=LOAD_CASE('Dead load',$,(#21),#20);
    #21=ANALYSIS_METHOD_STATIC('1st order',$,.ELASTIC_1ST_ORDER.);
    #20=PHYSICAL_ACTION(.STATIC.,.FIXED_ACTION.,.DIRECT_ACTION.,$,$,(1.0),(' '));

#1025=LOAD_COMBINATION_OCCURRENCE(1.,#461,#458);
  #461=LOADING_COMBINATION('4','all loads with factor = 1',#1);
  #458=LOAD_CASE('2::Wind from -Y as joint loads',$,(#21),#20);

#1026=LOAD_COMBINATION_OCCURRENCE(1.,#461,#459);
  #461=LOADING_COMBINATION('4','all loads with factor = 1',#1);
  #459=LOAD_CASE('3::Uniform dead load',$,(#21),#20);
```

FIG. 33a: CIS/2 analysis model load combination

In IFC, the generic entity IfcRelAssignsToGroup is used to create a new load case from a combination of other load cases (IfcStructuralLoadGroup) as shown in Fig. 33b. While IfcStructuralLoadGroup provides for a load factor, there is no way to specify load combination factors when creating the new load case.

```
#6415= IFCRELASSIGNSTOGROUP('guid',#120005,'all loads with factor = 1',
                            'Load combination',(#457,#458,#459),.NOTDEFINED.,#461);
  #457= IFCSTRUCTURALLOADGROUP('guid',#120005,'Dead load',$,$,
                               .LOAD_CASE.,.PERMANENT_G.,.NOTDEFINED.,$,$);
  #458= IFCSTRUCTURALLOADGROUP('guid',#120005,'2::Wind from -Y as joint loads',$,$,
                               .LOAD_CASE.,.PERMANENT_G.,.WIND_W.,$,$);
  #459= IFCSTRUCTURALLOADGROUP('guid',#120005,'3::Uniform dead load',$,$,
                               .LOAD_CASE.,.PERMANENT_G.,.DEAD_LOAD_G.,$,$);
  #461= IFCSTRUCTURALLOADGROUP('guid',#120005,'all loads with factor = 1',$,$,
                               .LOAD_COMBINATION_GROUP.,.PERMANENT_G.,.NOTDEFINED.,$,$);
```

FIG. 33b: IFC analysis model load combination

D.7 Assembly Map

In CIS/2, an Assembly_map is used to provide a logical relationship between analysis elements and a physical design part. It is a many-to-one relationship. For example, a beam that is subdivided in an analysis model might be represented physically by a single beam in a design model.

The association between elements and parts is made indirectly through Assembly_design_structural_member_linear as shown in Fig. 34a. Assembly_map provides a many-to-one association between analysis elements (Element_curve_simple) and Assembly_design_structural_member_linear. Design_part also refers to Assembly_design_structural_member_linear and thus the relationship between analysis elements and design parts. In this example five analysis elements are mapped to one assembly design.

```
#74=ASSEMBLY_MAP(#1397,(#3836,#3837,#3838,#3839));
  #1397=ASSEMBLY_DESIGN_STRUCTURAL_MEMBER_LINEAR(65,'B_7',$,$,0,.LOW.,.F.,.F.,(),(),.F.,
                                              .UNDEFINED_ROLE.,.UNDEFINED_CLASS.,.BEAM.);
  #3836=(ELEMENT('E65',$,#8334,1)ELEMENT_CURVE(16)ELEMENT_CURVE_SIMPLE(#5727,#6017));
    #8334=ANALYSIS_MODEL('Analysis Model','',.SPACE_FRAME.,$,3);
    #5727=SECTION_PROFILE(65,'W6X9',$,$,8,.F.);
    #6017=DIRECTION('OV_E65',(0.,0.,1.));
  #3837=(ELEMENT('E66',$,#8334,1)ELEMENT_CURVE(16)ELEMENT_CURVE_SIMPLE(#5728,#6018));
    #8334=ANALYSIS_MODEL('Analysis Model','',.SPACE_FRAME.,$,3);
    #5728=SECTION_PROFILE(66,'W6X9',$,$,8,.F.);
    #6018=DIRECTION('OV_E66',(0.,0.,1.));
  #3838=(ELEMENT('E67',$,#8334,1)ELEMENT_CURVE(16)ELEMENT_CURVE_SIMPLE(#5729,#6019));
    #8334=ANALYSIS_MODEL('Analysis Model','',.SPACE_FRAME.,$,3);
    #5729=SECTION_PROFILE(67,'W6X9',$,$,8,.F.);
    #6019=DIRECTION('OV_E67',(0.,0.,1.));
  #3839=(ELEMENT('E68',$,#8334,1)ELEMENT_CURVE(16)ELEMENT_CURVE_SIMPLE(#5730,#6020));
    #8334=ANALYSIS_MODEL('Analysis Model','',.SPACE_FRAME.,$,3);
    #5730=SECTION_PROFILE(68,'W6X9',$,$,8,.F.);
    #6020=DIRECTION('OV_E68',(0.,0.,1.));

#331=DESIGN_PART('B_7',#1679,(#1397),(#535));
  #1679=(PART(.UNDEFINED.,$)PART_PRISMATIC()PART_PRISMATIC_SIMPLE(#5878,#1540,$,$));
    #5878=SECTION_PROFILE(64,'W6X9',$,$,8,.F.);
    #1540=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(300.),#8249);
  #1397=ASSEMBLY_DESIGN_STRUCTURAL_MEMBER_LINEAR(65,'B_7',$,$,0,.LOW.,.F.,.F.,(),(),.F.,
                                              .UNDEFINED_ROLE.,.UNDEFINED_CLASS.,.BEAM.);
  #535=COORD_SYSTEM_CARTESIAN_3D('Design_Part','Design_Part CS’,$,3,#929);
```

FIG. 34a: CIS/2 assembly map

In IFC, IfcRelConnectsStructuralElement is used to create an association between a physical member (IfcBeam) and an analysis element (IfcStructuralCurveMember) as shown in Fig. 34b. Since it is only a one-to-one relationship, multiple IfcRelConnectsStructuralElement are necessary to create the equivalent Assembly_map relationship shown in Fig. 34a. IfcRelConnectsStructuralElement is also shown in Fig. 25c.

```
#10254= IFCRELCONNECTSSTRUCTURALELEMENT('guid',#15,$,$,#331,#3836);
  #331= IFCBEAM('guid',#15,'B_7','Beam','W6X9',#534,#9459,'B_7');
  #3836=IFCSTRUCTURALCURVEMEMBER('guid',#15,'E65','Beam','W6X9',$,#10250,.NOTDEFINED.);
#10259= IFCRELCONNECTSSTRUCTURALELEMENT('guid',#15,$,$,#331,#3837);
  #331= IFCBEAM('guid',#15,'B_7','Beam','W6X9',#534,#9459,'B_7');
  #3837= IFCSTRUCTURALCURVEMEMBER('guid',#15,'E66','Beam','W6X9',$,#10255,.NOTDEFINED.);
#10264= IFCRELCONNECTSSTRUCTURALELEMENT('guid',#15,$,$,#331,#3838);
  #331= IFCBEAM('guid',#15,'B_7','Beam','W6X9',#534,#9459,'B_7');
  #3838= IFCSTRUCTURALCURVEMEMBER('guid',#15,'E67','Beam','W6X9',$,#10260,.NOTDEFINED.);
#10269= IFCRELCONNECTSSTRUCTURALELEMENT('guid',#15,$,$,#331,#3839);
  #331= IFCBEAM('guid',#15,'B_7','Beam','W6X9',#534,#9459,'B_7');
  #3839= IFCSTRUCTURALCURVEMEMBER('guid',#15,'E68','Beam','W6X9',$,#10265,.NOTDEFINED.);
#10274= IFCRELCONNECTSSTRUCTURALELEMENT('guid',#15,$,$,#331,#3840);
  #331= IFCBEAM('guid',#15,'B_7','Beam','W6X9',#534,#9459,'B_7');
  #3840= IFCSTRUCTURALCURVEMEMBER('guid',#15,'E69','Beam','W6X9',$,#10270,.NOTDEFINED.);
```

FIG. 34b: IFC assembly map

APPENDIX E - OTHER CONCEPT EXAMPLES

Some of the examples in Appendix E show how concepts common to all CIS/2 files are mapped to IFC entities. The other examples are of concepts that are not commonly implemented in CIS/2 or IFC files, yet there is a mapping between the required CIS/2 and IFC entities for those concepts.

E.1 Unit Assignment

In CIS/2, units for length and other properties have to be specifically assigned as shown in Fig. 35a with Representation. In this case the Representation is for Polyline which refers to Cartesian_point thus assigning the units of millimeters (#1504) to the coordinates. Representation could refer directly to Cartesian_point. Other non-SI units can be specified with Context_dependent_unit, Conversion_based_unit, or Derived_unit.

Unit assignments for lengths use Positive_length_measure_with_unit. Units for angle, force, mass, moment, pressure, stiffness, temperature, modulus, and others are assigned in a similar manner. Either method to specify units allows for mixed units (i.e. millimeters and inches) for the same measure in the CIS/2 model.

```
#4105=REPRESENTATION('polylines',(#1266,#1267),#17);
  #1267=POLYLINE('hole depth',(#2694,#2695));
    #2694=CARTESIAN_POINT('hole depth pt1',(0.,0.,0.));
    #2695=CARTESIAN_POINT('hole depth pt2',(-25.4,0.,0.));
  #1268=POLYLINE('mtrl',(#2696,#2697,#2698,#2699));
    #2696=CARTESIAN_POINT('desc',(0.,-356.997007751465,-106.278711509705));
    #2697=CARTESIAN_POINT('desc',(0.,-591.947007751465,-293.603711509705));
    #2698=CARTESIAN_POINT('desc',(0.,-591.947007751465,-490.453711509705));
    #2699=CARTESIAN_POINT('desc',(0.,-356.997007751465,-490.453711509705));
  #17=(GEOMETRIC_REPRESENTATION_CONTEXT(3)GLOBAL_UNIT_ASSIGNED_CONTEXT((#1504))
       REPRESENTATION_CONTEXT('polylines','Polylines'));
    #1504=(LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.));

#1505=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(25.4),#1504);
```

FIG. 35a: CIS/2 unit assignment

In IFC a much different approach is used to assign units to items as shown in Fig. 35b. Units are not assigned to each individual coordinate or length value, rather, the type of global units used for various measures are assigned to the IfcProject. This method does not allow for mixed units for the same measure. In addition to IfcSIUnit, units can be specified with IfcContextDependentUnit, IfcConversionBasedUnit, or IfcDerivedUnit.

```
#100010= IFCPROJECT('guid',#100005,'Design Data SDS/2 - Detailed Model',
                    $,$,$,$,(#100011),#100060);
  #100060= IFCUNITASSIGNMENT((#1504,#100062,#100063,#100064,#100065,
                              #100066,#100067));
    #1504= IFCSIUNIT(*,.LENGTHUNIT.,.MILLI.,.METRE.)
    #100062= IFCSIUNIT(*,.PLANEANGLEUNIT.,$,.RADIAN.);
    #100063= IFCSIUNIT(*,.MASSUNIT.,.KILO.,.GRAM.);
    #100064= IFCSIUNIT(*,.TIMEUNIT.,$,.SECOND.);
    #100065= IFCSIUNIT(*,.AREAUNIT.,$,.SQUARE_METRE.);
    #100066= IFCSIUNIT(*,.PRESSUREUNIT.,$,.PASCAL.);
    #100067= IFCSIUNIT(*,.FORCEUNIT.,$,.NEWTON.);
```

FIG. 35b: IFC unit assignment

E.2 Globally Unique Identifier

A Globally Unique Identifier, known as a GUID, is a unique identifier that is unique throughout the software world. The identifier is a unique 128-bit number and can be generated by the Microsoft Foundation Class function “CoCreateGuid”. A GUID is used to keep track of a data item when it is transferred from one software system to another. This provides a mechanism to identify and track a part created in one CAD system when it is transferred to another CAD system.

Fig. 36a shows how Managed_data_item is used to assign a GUID to a located part in CIS/2. The string encoding of the GUID is a 36 character alphanumeric string. Managed_data_item also refers to the application that created the item on Managed_application_installation and when the item was created on Managed_data_creation. CIS/2 can also keep track of the history of an item with Managed_data_item_with_history that refers to data management transactions such as Managed_data_creation, Managed_data_deleted, and Managed_data_modification.

```
#72=MANAGED_DATA_ITEM('b5d5f30c-0a2e-4138-a912-bee715f7e4d4',#282,#337,(#17),T.);
  #282=MANAGED_APPLICATION_INSTALLATION(103,'SDS/2','SDS/2 Version 7.025 on NT',
                                      'Steel Detailing System',$,100,'Design Data',$);
  #337=LOCATED_PART_MARKED(2800000,'a10','User material: Angle',#327,#325,#342,
                         'a10','4/6 : H_+3.68',$,20,.F.);
  #17=MANAGED_DATA_CREATION(#282,#14,#10,.T.,'Pre Fabrication','Created by SDS/2');
    #14=PERSON_AND_ORGANIZATION(#16,#15);
      #16=PERSON('Id','User','Lipman',$,$,$);
      #15=ORGANIZATION('ID','Unknown','SDS/2 User');
    #10=DATE_AND_TIME(#13,#12);
      #13=CALENDAR_DATE(2007,23,1);
      #12=LOCAL_TIME(18,26,38.,#11);
        #11=COORDINATED_UNIVERSAL_TIME_OFFSET(5,0,.BEHIND.);
```

FIG. 36a: CIS/2 GUID

In IFC, the GUID is an attribute of all entities whose top level abstract supertype is IfcRoot. This includes all objects, property sets, and relationships. Many of the previous figures show IFC entities that require a GUID. In IFC a compression algorithm is used to convert the 36 character GUID to a 22 character alphanumeric string as shown on IfcMember in Fig. 36b. All entities that have a GUID also refer to an IfcOwnerHistory which can keep track of the history of that item.

```
#337= IFCMEMBER('2rrVCC2Yv1EAaIlkSLz_JK',#17,'a10','L80x80x6','Beam',#327,#607,'a10');
  #17= IFCOWNERHISTORY(#14,#282,$,.NOCHANGE.,$,$,$,1197067060);
    #14= IFCPERSONANDORGANIZATION(#16,#15,$);
      #16= IFCPERSON('Id','User','Lipman',$,$,$,$,$);
      #15= IFCORGANIZATION('ID','Unknown','SDS/2 User',$,$);
    #282= IFCAPPLICATION(#12002,'SDS/2 Version 7.025 on NT','SDS/2',
                              'Steel Detailing System');
      #12002= IFCORGANIZATION($,'Unknown',$,$,$);
```

FIG. 36b: IFC GUID

E.3 Material and Section Properties

In CIS/2, material properties can be associated with parts and elements in design, detailed, and analysis models. In analysis models Element_with_material is used and in design and detailed models Structural_frame_product_with_material is used. Fig. 37a shows how different material properties are associated with a type of steel (Grade 50).

```
#38601=MATERIAL_ISOTROPIC(1,'Grade 50',’ASTM A572: 1994',#39501);
  #39501=MATERIAL_REPRESENTATION('Grade 50 Steel',(#38401,#38701,#39701,#39702,
                               #39703,#39801),#27201);
    #38401=MATERIAL_ELASTICITY('material elasticity',0.27,29000.0,$,$);
    #38701=MATERIAL_MASS_DENSITY('Average mass per unit volume',0.283);
    #39701=MATERIAL_STRENGTH('Minimum Yield Stength',50.0);
    #39702=MATERIAL_STRENGTH('Minimum Tensile Stength',70.0);
    #39703=MATERIAL_STRENGTH('Maximum Tensile Stength',100.0);
    #39801=MATERIAL_THERMAL_EXPANSION('Average coefficient',6.5E-6);
```

FIG. 37a: CIS/2 material properties

In IFC, material properties are specified on IfcMechanicalSteelMaterialProperties and associated with a material name on IfcMaterial as shown in Fig. 37b. The material is associated with parts with IfcRelAssociatesMaterial.

```
#496394= IFCMECHANICALSTEELMATERIALPROPERTIES(#38601,$,29000.0,$,0.27,6.5E-6,
                                              50.0,$,$,$,$,$,$);
  #38601= IFCMATERIAL('Grade 50 ASTM A572: 1994');

#4688= IFCRELASSOCIATESMATERIAL('guid',#80005,'A500_46',$,(#107),#963);
  #107= IFCBEAM('guid',#80005,'Design Part','HSS6X6X4','Column',
               #1119,#4012,'Design Part');
  #38601= IFCMATERIAL('Grade 50 ASTM A572: 1994');
```

FIG. 37b: IFC material properties

In CIS/2, properties of a Section_profile are specified with Section_properties as shown in Fig. 38a. The section properties include the moment of inertia, torsional constant, shear area, radius of gyration, plastic modulus, buckling parameter, and mass per length. In IFC, section properties are specified on IfcStructuralSteelProfileProperties as shown in Fig. 38b.

```
#1673=SECTION_PROPERTIES(#1687,(#1688,#1689),#1690,#1691,#1692,#1693,
                        #1694,#1695,$,$,$,$,$,$,$,$,$,$,$);
  #1687=SECTION_PROFILE(4,'AGravBm',$,$,10,.F.);
  #1688=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0000000),#1224);
  #1689=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0000000),#1224);
  #1690=INERTIA_MEASURE_WITH_UNIT(INERTIA_MEASURE(1.14),#1813);
  #1691=INERTIA_MEASURE_WITH_UNIT(INERTIA_MEASURE(24.9),#1813);
  #1692=INERTIA_MEASURE_WITH_UNIT(INERTIA_MEASURE(984.),#1813);
  #1693=AREA_MEASURE_WITH_UNIT(AREA_MEASURE(14.7),#1814);
  #1694=AREA_MEASURE_WITH_UNIT(AREA_MEASURE(7.904),#1814);
  #1695=AREA_MEASURE_WITH_UNIT(AREA_MEASURE(5.823),#1814);
```

FIG. 38a: CIS/2 section properties

```
#1673= IFCSTRUCTURALSTEELPROFILEPROPERTIES('AGravBm',$,$,$,$,$,14.7,1.14,$,24.9,984.,
                                           $,$,$,$,$,$,$,$,$,$,$,$,5.823,7.904,$,$);
```

FIG. 38b: IFC section properties

E.4 Generic Properties

In CIS/2, Item_property and Item_property_assigned can be used to associate generic properties with an item. In Fig. 39a, the advanced bill of material (ABM Mark) is associated with a plate. In IFC, an IfcPropertySet can be used to assign the property to IfcPlate as shown in Fig. 39b.

```
#485=ITEM_PROPERTY_ASSIGNED(#658,#2484);
  #658=ITEM_PROPERTY('ABM Mark','ABM Page:1 Line:1',#831);
  #2484=LOCATED_PART(2100,'BP2',’Base Plate',#1595,#1578,#4027);
```

FIG. 39a: CIS/2 item property ABM Mark

```
#485= IFCRELDEFINESBYPROPERTIES('guid',#100005,'ABM Mark',$,(#2484),#658);
  #2484= IFCPLATE('guid',#100005,'BP2','Column','Plate (3-8x1-10)',#1595,#5177,$);
  #658= IFCPROPERTYSET('guid',#100005,'PSet_ABM_Mark',$,(#5967));
    #5967= IFCPROPERTYSINGLEVALUE('ABM Mark',$,IFCLABEL('Page:1 Line:1'),$);
```

FIG. 39b: IFC property set for ABM Mark

E.5 Surface Treatment

Surface treatments in CIS/2 are modeled with Surface_treatment_coat and Coating as shown in Fig. 40a. The surface treatment is associated with a Part_prismatic_simple with Structural_frame_item_relationship.

```
#149=LOCATED_PART(12,'w4[12]',$,#121,#58,#273);
  #121=(COORD_SYSTEM('Local','Part CS',$,3)COORD_SYSTEM_CARTESIAN_3D(#243)
        COORD_SYSTEM_CHILD(#270));
    #243=AXIS2_PLACEMENT_3D('Part CS',#173,#218,#217);
    #270=COORD_SYSTEM_CARTESIAN_3D('Global','Assembly CS',$,3,#239);
      #239=AXIS2_PLACEMENT_3D('Assembly CS',#164,#218,#217);
  #58=PART_PRISMATIC_SIMPLE(12,'w4[12]',$,$,.ROLLED.,$,#51,#99,$,$);
    #51=SECTION_PROFILE(0,'L10X5X1/4','ASTM specification A6','W flange',1,.T.);
    #99=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(3028.95),#87);
  #273=LOCATED_ASSEMBLY(1,'B_1[1]','beam',#270,$,#161,#4076);

#1153=SURFACE_TREATMENT_COAT(3,'Coating','Galvanized',$,'AS PER BID',(.DIPPED.),
                             (#1559),(#1154));
  #1559=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(0.),#87);
  #1154=COATING(3,'Galvanized','$','$',.CORROSION_PROTECTION.);

#1155=STRUCTURAL_FRAME_ITEM_RELATIONSHIP('Coating','Material Surface Finish',#58,#1153);
  #58=PART_PRISMATIC_SIMPLE(12,'w4[12]',$,$,.ROLLED.,$,#51,#99,$,$);
  #1153=SURFACE_TREATMENT_COAT(3,'Coating','Galvanized',$,'AS PER BID',(.DIPPED.),
                               (#1559),(#1154));
```

FIG. 40a: CIS/2 surface treatment

In IFC, an IfcPropertySet can be used to associate a surface treatment with an IfcBeam as shown in Fig. 40b. The IFC specification does not indicate how surface treatments can be specified with an IfcPropertySet.

```
#149= IFCBEAM('guid',#100005,'w4[12]',$,'L10X5X1/4',#121,#5013,$);
  #121= IFCLOCALPLACEMENT(#270,#243);
    #270= IFCLOCALPLACEMENT($,#239);
  #5013= IFCPRODUCTDEFINITIONSHAPE($,$,(#5014));
  #5014= IFCSHAPEREPRESENTATION(#100011,'Body','SweptSolid',(#5015));
    #5015= IFCEXTRUDEDAREASOLID(#51,#100049,#100044,3028.95);
      #51= IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,'L10X5X1/4',#5008);
#5023= IFCRELDEFINESBYPROPERTIES('guid',#100005,'Coating galvanized',
                                 $,(#149),#1153);
  #149= IFCBEAM('guid',#100005,'w4[12]',$,'L10X5X1/4',#121,#5013,$);
  #1153= IFCPROPERTYSET('guid',#100005,'PSet_Coating','Coating galvanized',(#5022));
    #5022= IFCPROPERTYSINGLEVALUE('Surface treatment',$,
                                  IFCLABEL('Coating galvanized Dipped'),$);
```

FIG. 40b: IFC surface treatment property set

A proposed version IFC, IFC2x3g, has a specific property for surface properties, IfcShapeAspectSurfaceProperties, as shown in Fig. 40c.

```
#5022= IFCRELDEFINESBYPROPERTIES('guid',#100005,'Coating galvanized',$,(#149),#1153);
  #149= IFCBEAM('2zWaaTpdqpI9VYwEQ1KKN_',#100005,'w4[12]',$,'L10X5X1/4',#121,#5013,$);
  #1153= IFCSHAPEASPECTSURFACEPROPERTIES('guid',#100005,'Coating galvanized',
                                         'Surface treatment',$,$,$,$,$,'Dipped',.T.);
```

FIG. 40c: IFC surface treatment (IFC2x3g)

E.6 Grid Lines

In CIS/2, a grid can be orthogonal, skewed, or radial and is modeled with Gridline as shown in Fig. 41a. Gridline defines a vertical plane rather than an actual line and is specified by Axis2_placement_3d. For visualization purposes, a grid line, similar to what is on a CAD drawing, can be computed from the intersection of the vertical plane and a horizontal plane at the base of the structure.

```
#416=GRIDLINE('A',#374,#422,$);
  #374=AXIS2_PLACEMENT_3D('Axis3d',#332,#248,#249);
    #332=CARTESIAN_POINT('Origin',(0.,0.,0.));
    #248=DIRECTION('Direction',(1.,0.,0.));
    #249=DIRECTION('Direction',(0.,0.,-1.));
  #422=GRID('GridSystem_1',$,$);

#417=GRIDLINE('B',#375,#422,$);
  #375=AXIS2_PLACEMENT_3D('Axis3d',#333,#248,#249);
    #333=CARTESIAN_POINT('Origin',(6096.,0.,0.));
  #422=GRID('GridSystem_1',$,$);

#419=GRIDLINE('1',#377,#422,$);
  #377=AXIS2_PLACEMENT_3D('Axis3d',#332,#254,#249);
    #332=CARTESIAN_POINT('Origin',(0.,0.,0.));
    #254=DIRECTION('Direction',(0.,1.,0.));
  #422=GRID('GridSystem_1',$,$);

#420=GRIDLINE('2',#378,#422,$);
  #378=AXIS2_PLACEMENT_3D('Axis3d',#336,#254,#249);
    #336=CARTESIAN_POINT('Origin',(0.,4572.,0.));
  #422=GRID('GridSystem_1',$,$);
```

FIG. 41a: CIS/2 grid lines

In IFC, IfcGrid and IfcGridAxis are used to model grid lines as shown in Fig. 41b. IfcGridAxis refers to an IfcLine which corresponds to a grid line in a CAD drawing. The position and orientation of the line is derived from the intersection of the vertical plane defined by Gridline in CIS/2 and a horizontal plane at the base of the structure.

```
#1147= IFCGRID('guid',#20005,'GridSystem_1 422',$,$,#1148,$,
               (#1132,#1137),(#1117,#1122),$);
  #1148= IFCLOCALPLACEMENT($,#1149);
    #1149= IFCAXIS2PLACEMENT3D(#1150,#20044,#20042);
      #1150= IFCCARTESIANPOINT((0.,0.,0.));
      #20044= IFCDIRECTION((0.,0.,1.));
      #20042= IFCDIRECTION((1.,0.,0.));
  #1132= IFCGRIDAXIS('2',#1133,.T.);
    #1133= IFCLINE(#1134,#1135);
      #1134= IFCCARTESIANPOINT((-2438.4,4572.));
      #1135= IFCVECTOR(#1136,15120.0);
        #1136= IFCDIRECTION((1.0,0.0));
  #1137= IFCGRIDAXIS('1',#1138,.T.);
    #1138= IFCLINE(#1139,#1140);
      #1139= IFCCARTESIANPOINT((-2438.4,0.));
      #1140= IFCVECTOR(#1136,15120.0);
  #1117= IFCGRIDAXIS('A',#1118,.T.);
    #1118= IFCLINE(#1119,#1120);
      #1119= IFCCARTESIANPOINT((0.,-1828.8));
      #1120= IFCVECTOR(#1121,11880.0);
        #1121= IFCDIRECTION((0.0,1.0));
  #1122= IFCGRIDAXIS('B',#1123,.T.);
    #1123= IFCLINE(#1124,#1125);
      #1124= IFCCARTESIANPOINT((6096.,-1828.8));
      #1125= IFCVECTOR(#1121,11880.0);
```

FIG. 41b: IFC grid lines

E.7 Camber

In CIS/2, Part_prismatic_simple_cambered is used to specify the camber of a beam as shown in Fig. 42a. Camber can also be applied to design parts in a design model and analysis elements in a structural analysis model. IFC does not have a method to specify camber although an IfcPropertySet could be used as shown in Fig. 42b.

```
#101746=LOCATED_PART(50300,'w229',’W flange',#93799,#60353,#119649);
  #93799=(COORD_SYSTEM('Local','Part CS’,$,3)COORD_SYSTEM_CARTESIAN_3D(#113565)
         COORD_SYSTEM_CHILD(#118360));
  #60353=(PART(.ROLLED.,$)PART_PRISMATIC()PART_PRISMATIC_SIMPLE(#68038,#82247,$,$)
          PART_PRISMATIC_SIMPLE_CAMBERED('Camber UP 31.800000')
          PART_PRISMATIC_SIMPLE_CAMBERED_ABSOLUTE(#82351,#60351,#60352));
    #68038=SECTION_PROFILE (30,'W30x90','ASTM spectification A6','W flange',8,.T.);
    #82247=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(14376.4),#82019);
    #82351=POSITIVE_LENGTH_MEASURE_WITH_UNIT(POSITIVE_LENGTH_MEASURE(7188.2),#82019);
    #60351=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.),#82019);
    #60352=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(31.75),#82019);
  #119649=LOCATED_ASSEMBLY(738,'216B5[738]','beam',#118360,$,#103803,#120597);
```

FIG. 42a: CIS/2 camber

```
#133977= IFCRELDEFINESBYPROPERTIES('guid',#2420005,'Camber','Absolute',
                                   (#101746),#133973);
  #101746= IFCBEAM('guid',#2420005,'w229','Beam','W30x90',#93799,#133974,$);
  #133973= IFCPROPERTYSET('guid',#2420005,'PSet_Camber',$,(#133972));
    #133972= IFCPROPERTYSINGLEVALUE('Camber',$,IFCLENGTHMEASURE(31.75),$);
```

FIG. 42b: IFC camber as a property set

E.8 Document Reference

In CIS/2, Group_of_structural_data is used to associate an external reference to a drawing, specified by Media_file_drawing to a Located_part_marked as shown in Fig. 43a.

```
#83=GROUP_OF_STRUCTURAL_DATA(#23,(#2828));
  #23=MEDIA_FILE_DRAWING('G2','Part Drawing','GSheets/G2.pdf','pdf',#246,'drawing',
                         (#242),(),'G2',.PART_DRAWING.,'','0',$,$,$);
    #246=DATE_AND_TIME(#252,#250);
      #252=CALENDAR_DATE(2004,13,2);
      #250=LOCAL_TIME(10,26,55.,#248);
        #248=COORDINATED_UNIVERSAL_TIME_OFFSET(6,0,.BEHIND.);
    #242=PERSON_AND_ORGANIZATION(#244,#243);
      #244=PERSON('Id','User','barry',$,$,$);
      #243=ORGANIZATION('ID','','SDS/2 User');
  #2828=LOCATED_PART_MARKED(6500,'p15','User material: Plate',#2112,#2064,
                            #4984,'p15','',$,1,.F.);
```

FIG. 43a: CIS/2 document reference

In IFC, IfcRelAssociatesDocument is used to associate an external reference to a drawing, specified by IfcDocumentReference to an IfcPlate as shown in Fig. 43b.

```
#83= IFCRELASSOCIATESDOCUMENT('guid',#120005,'Part/assembly drawing',$,(#2828),#23);
  #2828= IFCPLATE('guid',#120005,'p15','Plate (0-6x0-6)','Member',#2112,#6516,'p15');
  #23= IFCDOCUMENTREFERENCE('GSheets/G2.pdf',$,$);
```

FIG. 43b: IFC document reference