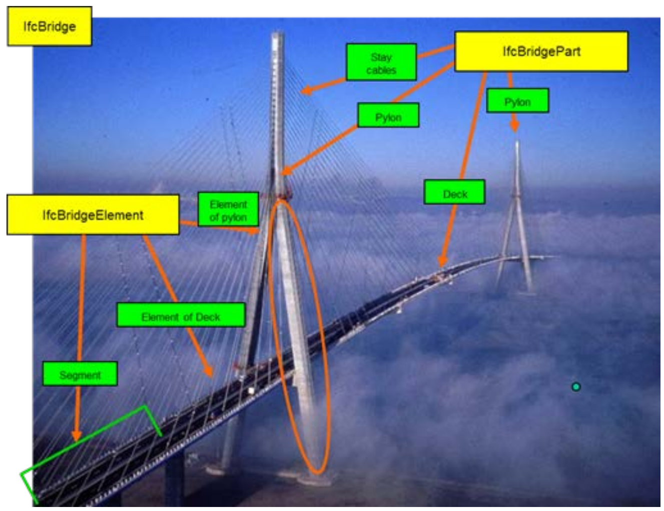
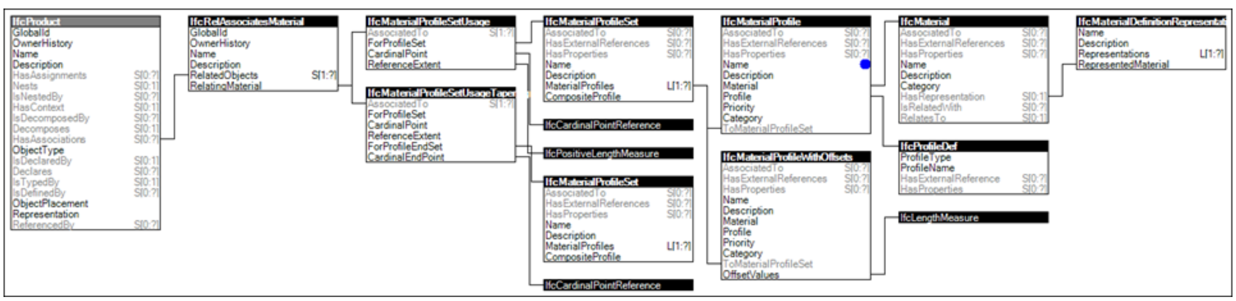
Bridge Information Model Standardization

VOLUME II: SCHEMA ANALYSIS

April 2016





U.S. Department of Transportation
Federal Highway Administration

FHWA-HIF-16-011

Foreword

Advancing the capability of computer modeling and analysis tools and techniques is clearly in the best interest of the U.S. bridge engineering practice. Without industry consensus standards for Bridge Information Modeling (BrIM) and related data exchange protocols, there is no common way to integrate the various phases of a bridge design and construction project and benefit from that information in the inspection, maintenance, and operational phases associated with its asset management. This work seeks to develop, validate, identify gaps, implement, and build consensus for standards for BrIM for highway bridge engineering.

The contributions and constructive review comments received from many professionals across the country are greatly appreciated. In particular, I would like to recognize Scot Becker of Wisconsin DOT, Christopher Garrell of National Steel Bridge Alliance, Danielle Kleinhans of Concrete Reinforcing Steel Institute, Josh Sletten of Utah DOT, Steven Austin of Texas DOT, Brad Wagner of Michigan DOT, Todd Thomson of South Dakota DOT, Ahmad Abu-Hawash of Iowa DOT, Mike Keever of Caltrans, Ali Koc of Red Equation Corporation, Hanjin Hu of Michael Baker International, and all those who participated in our workshops described in the Report.

Joseph L. Hartmann, PhD, P.E.
Director, Office of Bridges and Structures

Notice

Quality Assurance Statement

information. FHWA periodically reviews quality issues and adjusts its programs and processes to ensure continuous quality improvement.

<table>
<tr><td>1. Report No.
FHWA-HIF-16-011</td><td>2. Government Accession No.</td><td>3. Recipient's Catalog No.</td></tr>
<tr><td colspan="2" rowspan="2">4. Title and Subtitle
Bridge Information Modeling Standardization Report Volume II – Schema Analysis</td><td>5. Report Date
2016-04</td></tr>
<tr><td>6. Performing Organization Code</td></tr>
<tr><td colspan="2">7. Author(s)
Primary - Tim Chipman
Contributing – Charles Eastman, Thomas Liebich, Donghoon Yang
Editor – Roger Grant</td><td>8. Performing Organization Report No.</td></tr>
<tr><td colspan="2" rowspan="2">9. Performing Organization Name and Address
National Institute of Building Sciences
1090 Vermont Avenue, NW, 7th Floor
Washington, DC 20005</td><td>10. Work Unit No.</td></tr>
<tr><td>11. Contract or Grant No.
DTFH6114C00047</td></tr>
<tr><td colspan="2" rowspan="2">12. Sponsoring Agency Name and Address
Office of Bridges and Structures
Federal Highway Administration
6300 Georgetown Pike
McLean, VA 22101-2296</td><td>13. Type of Report and Period Covered</td></tr>
<tr><td>14. Sponsoring Agency Code</td></tr>
<tr><td colspan="3">15. Supplementary Notes
This document is the third of a multi-volume set of documents on Bridge Information Modeling Standardization. The volumes can be read individually, or sequentially as parts of the set. Reading the Introduction first is recommended to provide context and a summary of the work and its findings.</td></tr>
<tr><td colspan="3">16. Abstract
Bridge Information Modeling Standardization is a multi-volume report that analyzes options for standardized approaches for modeling bridges across their lifecycle. The goal of the report is to identify and evaluate candidate open standards that can be used to document all aspects of bridges to identify viable standards that can be used by bridge owners to specify information delivery requirements and by software providers to meet those requirements. After evaluation of the viable available options, the Report goes on to provide an in-depth analysis based on test cases of real bridge projects of the viable alternative. Accompanying the Report is a comprehensive exchange specification to assist software developers to implement the recommended alternative to the benefit of bridge owners. This volume, Schema Analysis, provides background and evaluation of standardization efforts underway nationally and internationally, analyzes the viable schemas produced by these efforts, recommends one of those schema, the buildingSMART Industry Foundation Class (IFC) model for further evaluation and identifies its applicability and potential shortcomings.</td></tr>
<tr><td>17. Key Words
bridge, design, construction, modeling, models, open, standards,</td><td colspan="2">18. Distribution Statement
No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22161.</td></tr>
</table>

19. Security Classif. (of this report)-Unclassified	20. Security Classif. (of this page)-Unclassified	21. No. of Pages - 142	22. Price

Form DOT F 1700.7 (8-72) Reproduction of completed pages authorized

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
		NOTE: volumes greater than 1000 L shall be shown in m^3		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
		TEMPERAT URE (exact degrees)		
°F	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	°C
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3 m^3
	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
		TEMPERAT URE (exact degrees)		
°C	Celsius	1.8C+32	Fahrenheit	°F
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela/m^2	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	0.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380. (Revised March 2003)

A 508 compliant version of this table is available at http://www.fhwa.dot.gov/publications/convtabl.cfm.

Contents

Table of Figures

Table of Tables

1 Background on Standardization Efforts

International Standards bring technological, economic and societal benefits. They help to harmonize technical specifications of products and services making industry more efficient and breaking down barriers to international trade. (International Standards Organization, 2016). In the construction industry (buildingSmart International, 2016), information standards facilitate:

- transparent, open workflow, allowing project members to participate regardless of the software tools they use;
- a common language for widely referenced processes, allowing industry and government to procure projects with transparent commercial engagement, comparable service evaluation and assured data quality;
- enduring project data for use throughout the asset life-cycle, avoiding multiple input of the same data and consequential error;
- small and large (platform) software vendors can participate and compete on system-independent, 'best-of-breed' solutions;
- energizes the online product supply side with more exact user demand searches, and delivers the product data directly into information models.

This volume of the report describes standardization efforts related to bridge information modelling, and performs a cursory review regarding the specific technical structure and functionality resulting from current standardization efforts.

The specific data structures defined by each standardization effort are collectively called a schema. This evaluation of schemas is focused specifically on the technical content regarding how information is represented. There are many other criteria to consider regarding suitability of a standard for implementation, some of which may be of greater importance than the scope or technical merits.

The International Standards Organization (ISO) uses the following criteria in deciding whether to approve standards:

- Validation: is there market interest demonstrated by widespread adoption by software vendors?
- Verification: do software applications comply with the specification?
- Conformance: are there test files and testing tools to check that software complies?
- Interoperability: can multiple software platforms import and export data according to the standard?

Much of the criteria above is only satisfied over time based on the usage of a standard, and repeats with every iteration of extending a standard. Some standards reviewed have demonstrated meeting all criteria for dozens of iterations over many years, while other standards efforts are attempting to achieve the criteria for their first iteration.

Leveraging existing standards with active communities and supporting software products can accelerate achievement of these criteria. Of note, the standards reviewed having the widest adoption were not created from scratch, but evolved from existing user bases – for example, IFC was a derivative of manufacturing schemas already in widespread use from STEP (STandard for the Exchange of Product model data) and was initially developed and supported by Autodesk.

Nevertheless, this review is focused specifically on technical content, and analyzes and discusses the identified candidate schemas based on what is documented.

2 Developments in infrastructure BIM data standards

Compared with buildings, the construction sector where “Building” Information Modeling has been introduced about one decade ago, the use of BIM in the infrastructure sector is a relatively new method. There the need to develop open standards to share infrastructure BIM data has just recently been expressed on a larger scale.

Besides several national initiatives, the development of international BIM data standards is driven by:

- buildingSMART International, a non-for-profit industry association to develop, promote and proliferate open standards within the construction industry
- ISO/TC 59/SC 13, the committee for “Organization of information about construction works”, the subcommittee for BIM standards within the International Standardization Organization

For the future, another important committee that deals with BIM data standards is the newly created CEN Technical Committee on Building Information Modelling. It has been officially accepted on 17 April 2015.

- CEN/TC 422 “Building Information Modeling (BIM)”

How do these committees interact? The mission of buildingSMART International is to engage with and support international communities to define the need for standards, and once established, to create high quality standards through a well-defined standards program, including worldwide reviews through expert panels. Once an industry standard has reached a sufficient level of robustness, it will be promoted to ISO for consideration as a formal international standard. Currently, the following buildingSMART standards that would also be relevant for future use in infrastructure projects, are also ISO standards:

- ISO 16739, Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries; the buildingSMART standard “IFC”
- ISO 29481-1, Building Information Models — Information Delivery Manual — Part 1: Methodology and format; the buildingSMART standard “IDM”
- ISO 12006-3, Building construction — Organization of information about construction works — Part 3: Framework for object-oriented information; the buildingSMART standard “IFD”

2.1 Infrastructure standard developments within buildingSMART

BIM data standard development within buildingSMART, at that time still under the name of “International Alliance for Interoperability”, started around 1996 with the publication of an early version of IFC. It focused mainly on the architectural aspects of buildings. Shortly after, it was enhanced by structural considerations and later mechanical systems (heating, cooling, ventilation and plumbing) and electrical systems. After 2000, with the publication of the first IFC platform IFC2x, facility management, cost calculation, and scheduling were added to the scope. With the new IFC4 platform for BIM data, the data needs for the building construction use cases are broadly covered.

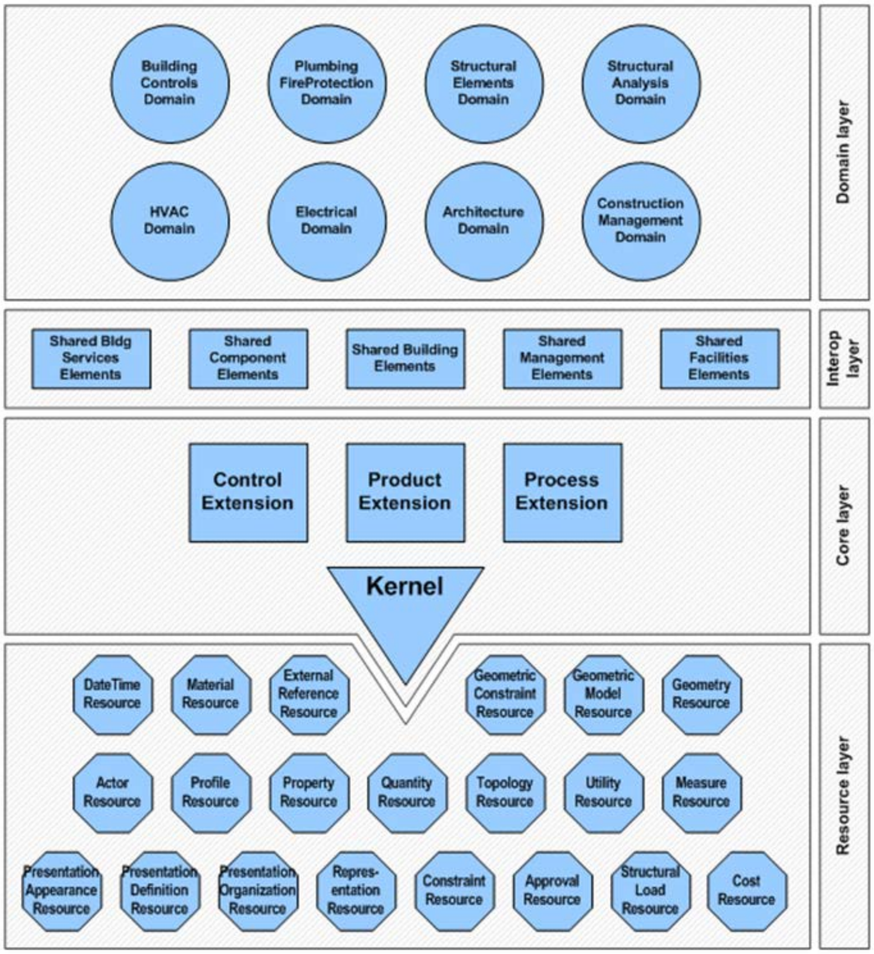


Figure 1: IFC4 Schema – building focused modules

Since 2010 interest has been raised to also include the various infrastructure works. As a result, a new committee, the "Infrastructure Room" had been founded. The first task of the infrastructure room was to analyze the various general use cases for BIM data[1] within the infrastructure world (see Figure 2). Next, the different existing national initiatives were invited and assessed. Finally, a road map had been created to identify the most important standard development projects to be carried out over the next several years.

[1] Since BIM had been seen for a longer time as being building centric, another term was coined "Virtual Design and Construction" VDC. Now, BIM is accepted as a single term to cover both buildings and infrastructure.

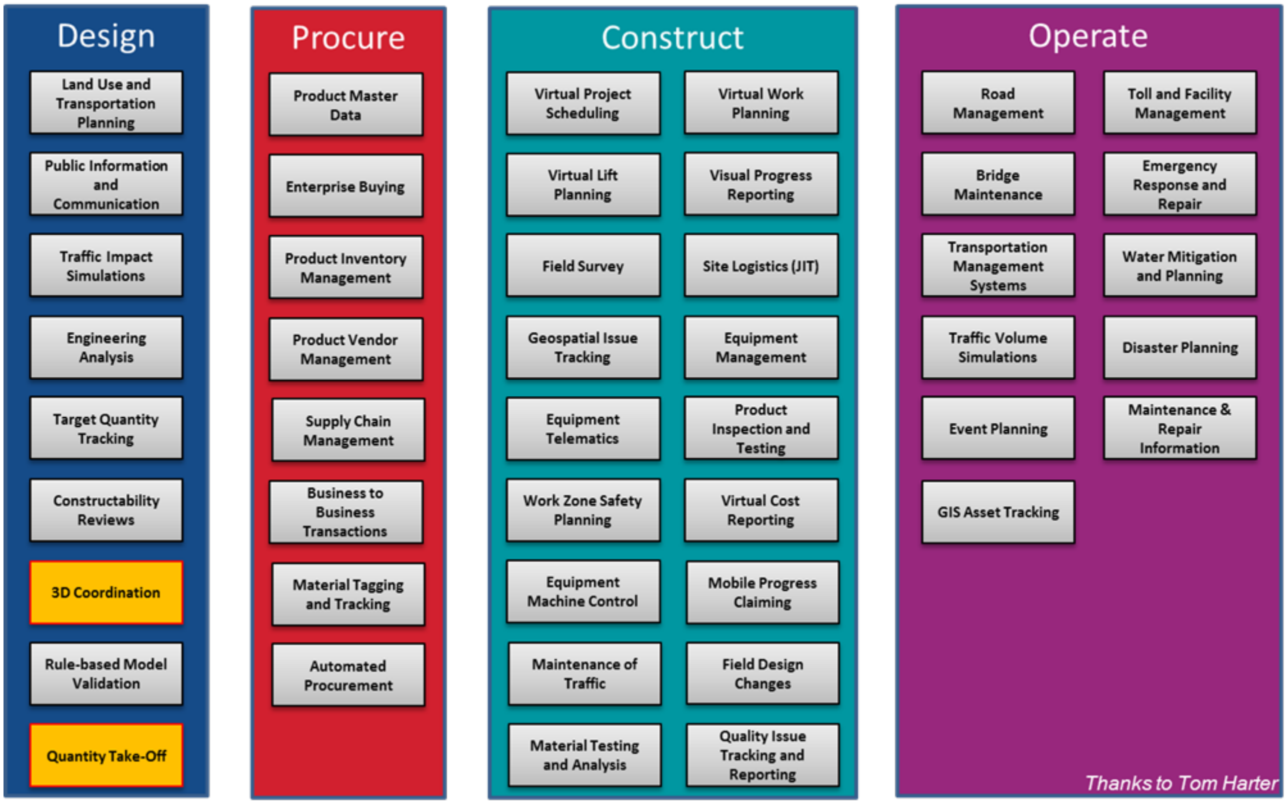


Figure 2: Initial use cases identified by the buildingSMART Infrastructure Room

The two most important vertical domains that had been identified by the Infrastructure Room had been Bridges and Roads. Bridges were identified for several reasons: (a) bridge structures are relatively close to building structures; (b) there had been also considerable work been done in France for the IFC-Bridge project; and (c) in many European and American countries the improvement of the current bridge infrastructure, that came into its age, is a high priority.

Roads are seen as an important part of countries' investment into infrastructure. Close to that are railways, which are considered as well. Tunnels, river canals, and dams had been added as additional scope. In order to proceed further, commonalities between these areas had been analyzed and, last but not least, a project and business plan needed to written and stakeholders needed to be addressed. Without considerable support by the international stakeholders, road/rail authorities, infrastructure and environmental agencies, large engineers, contractors and software providers, those standard developments cannot be achieved.

The alignment was identified as the most important common denominator of infrastructure assets. Strongly interlinked with alignment are linear referencing and connecting to GIS databases. Roads, rails, bridges, tunnels, canals, dams, power lines and other infrastructure assets are all designed, constructed and maintained using alignment and linear referencing. Other important common definitions are the terrain modeling, eventually including strata information, a common local coordinate system with mapping to the relevant geospatial coordinate reference system and earth works (cut and fill), as shown in Figure 3.

In 2014, with funding sponsored by Rijkswaterstaat (NL), Trafikverket (S), and the V-Con project (EU), buildingSMART started the first infrastructure data standard project, the IFC Alignment. One year later, the work has been accomplished and a Version 1.0 – that will be analyzed in more detail later – was published. It is expected to be released as a "buildingSMART Final Standard" according to the new standards program in May 2015.

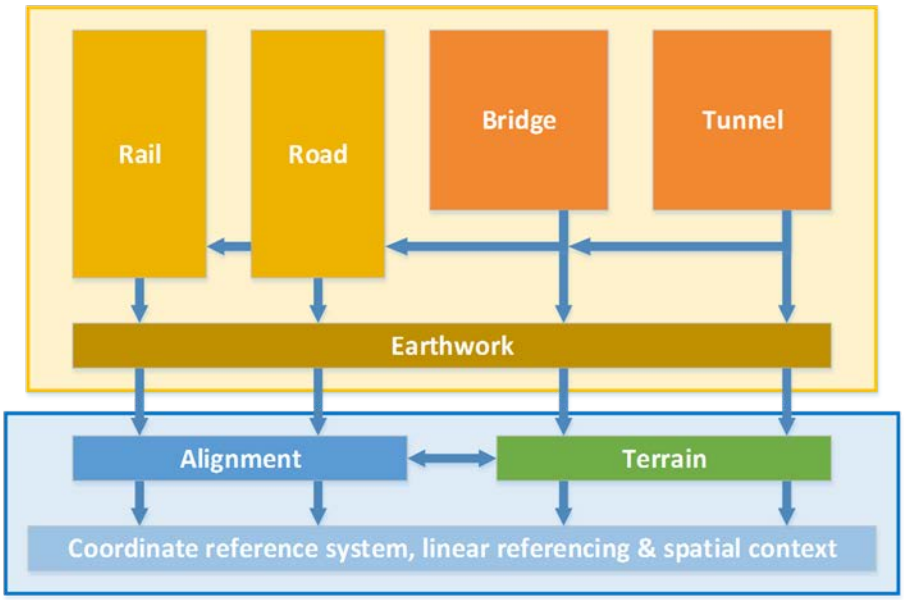


Figure 3: Main dependencies between selected infrastructure areas

2.1.1 IFC Alignment

The IFC Alignment Project had been carried out by a project team, led by the Chair of the buildingSMART Model Support Group (MSG) and had been supervised by a Project steering group established by the Infrastructure Room. The project had been set up as a joint project between the buildingSMART Infrastructure Room and the Land&Infra Domain and Specification Working Group, DWG/SWG, in OGC. The resulting conceptual model for alignment will be a standard within buildingSMART and OGC (BuildingSmart International, 2015).

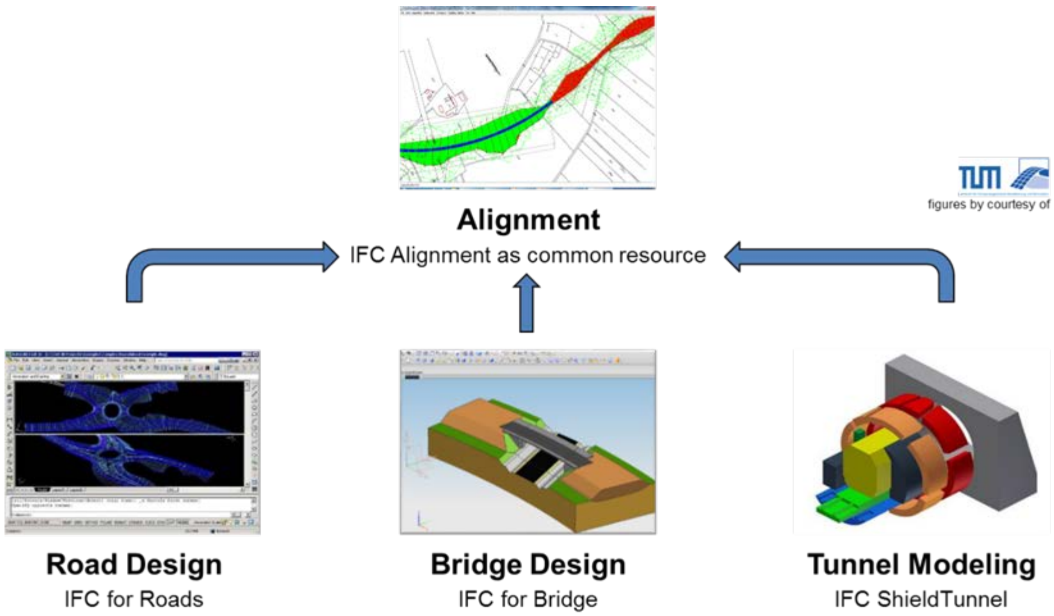


Figure 4: Alignment as central resource for infrastructure

The IFC Alignment 1.0 Project was initiated with the following objectives to be achieved:

- Ability to exchange alignment information from planning to construction phases of projects and from construction to asset management phases
- Ability to link alignment information to other project information such as super elevations, cross sections and full 3D geometry of construction elements
- Open data access of alignment information from asset management databases
- Lasting archiving of infrastructure information models, including alignment information
- Ability to map alignment information between InfraGML (developed by OGC), LandXML (latest InfraBIM version from Finland) and new versions of IFC

These objectives must be realized within the current architecture of the IFC schema and lead to a downward compatible extension. The baseline of the IFC Alignment extension is the IFC4 platform including the IFC4 Addendum 1 improvements.

In order to achieve the objectives the following main deliveries were developed during the project

- A clear definition of the scope for alignment information in this standard and references to other standards covering other scopes relevant for alignment information
- A clear definition (ontology) of alignment in the scope of this standard as the conceptual schema jointly developed by buildingSMART P6 project and the OGC InfraGML SWG
- The extension to the buildingSMART IFC4 schema comprising the alignment model, delivered as

- EXPRESS schema for implementation (including extended data in property sets or dictionary)
- ifcXML XSD schema for implementation
- mvdXML for alignment information – a subset of the extended IFC schema focusing on alignment
- A freely available viewer capable to display alignments and used throughout the project for validating the selected data modelling solution for alignments
- Facilitation of two review panels
- From client and domain expert side to state the requirements and assess the solution
- From software vendor side to assess the implementability of the solution

The IFC Alignment 1.0 project has been accomplished. As of today, the final acceptance process as a final buildingSMART International Standard is about to close with an anticipated announcement in mid-May.

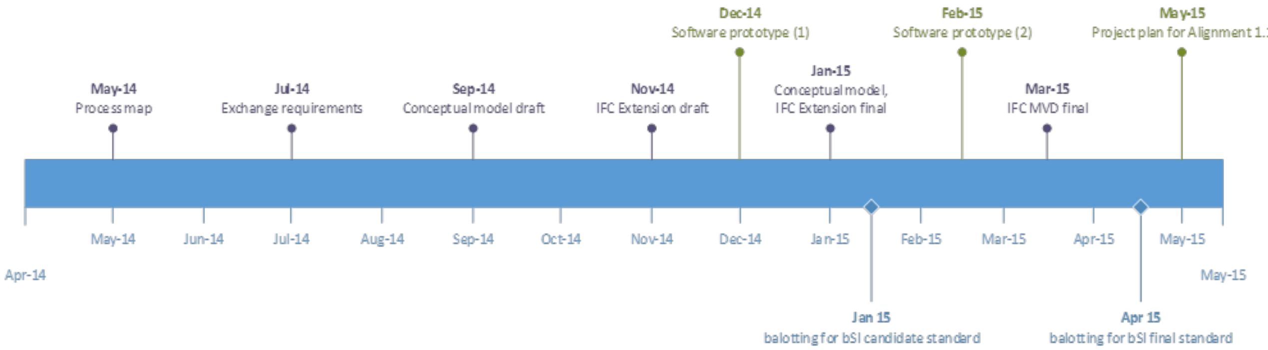


Figure 5: Project schedule of the IFC Alignment project

A follow-on project, IFC Alignment 1.1 has been drafted to handle additional scope that had not been covered in IFC Alignment 1.0. This includes (1) positioning along the alignment, (2) offsets between alignments, (3) rail specific alignment spirals, and (4) support the deployment of Alignment 1.0 including full software support.

The following results are achieved within the scope of IFC Alignment 1.0.

2.1.1.1 Scope definition

At the beginning the scope of the alignment project had to be clarified and fixed. Two particular decisions had been made:

- the support if alignment shall include both, the traditional 2D horizontal and vertical alignment definitions, and the full 3D alignment definition,
- the alignment is seen completely as “a reference system to position elements mainly for linear construction works, such as roads, rails, bridges, and others”, all other information

is regarded as information related to the alignment, like super elevation, cant, cross sections, line of sight, or other

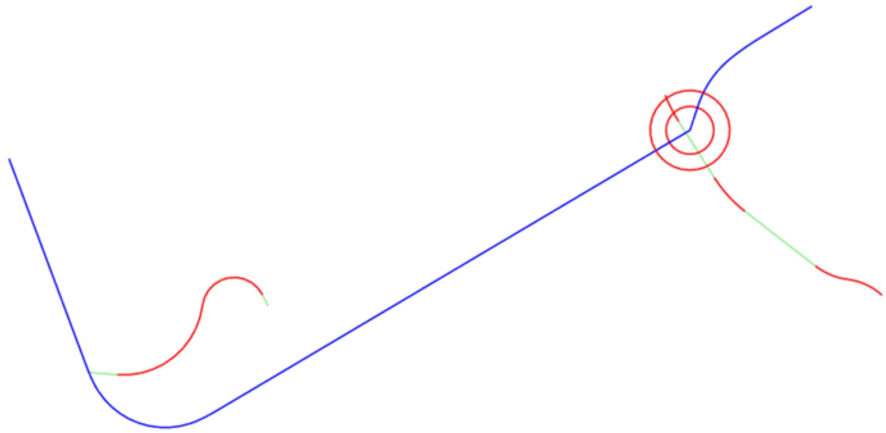


Figure 6: Horizontal alignments

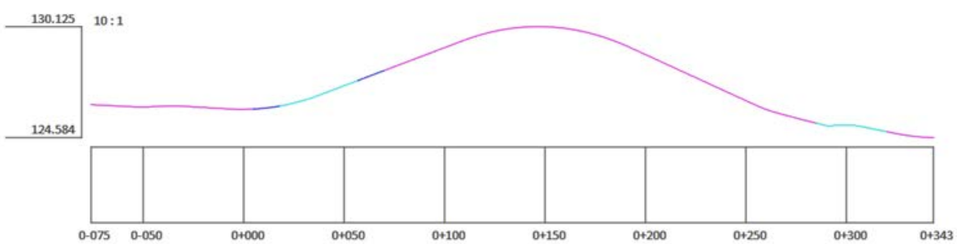


Figure 7: Vertical alignment

2.1.1.2 Process map

In order to determine the main information exchanges of alignment data within the overall life-cycle phases of infrastructure works two process maps had been created with help of the domain expert group. Since the scope had been international, the team attempted a very general definition of main processes and roles. Two scenarios were included:

- The traditional design-bid-build project execution, and
- The design-build project execution

The main task of the process maps is to identify the different levels of exchange that can later be used to determine different Levels of Detail of the alignment elements.

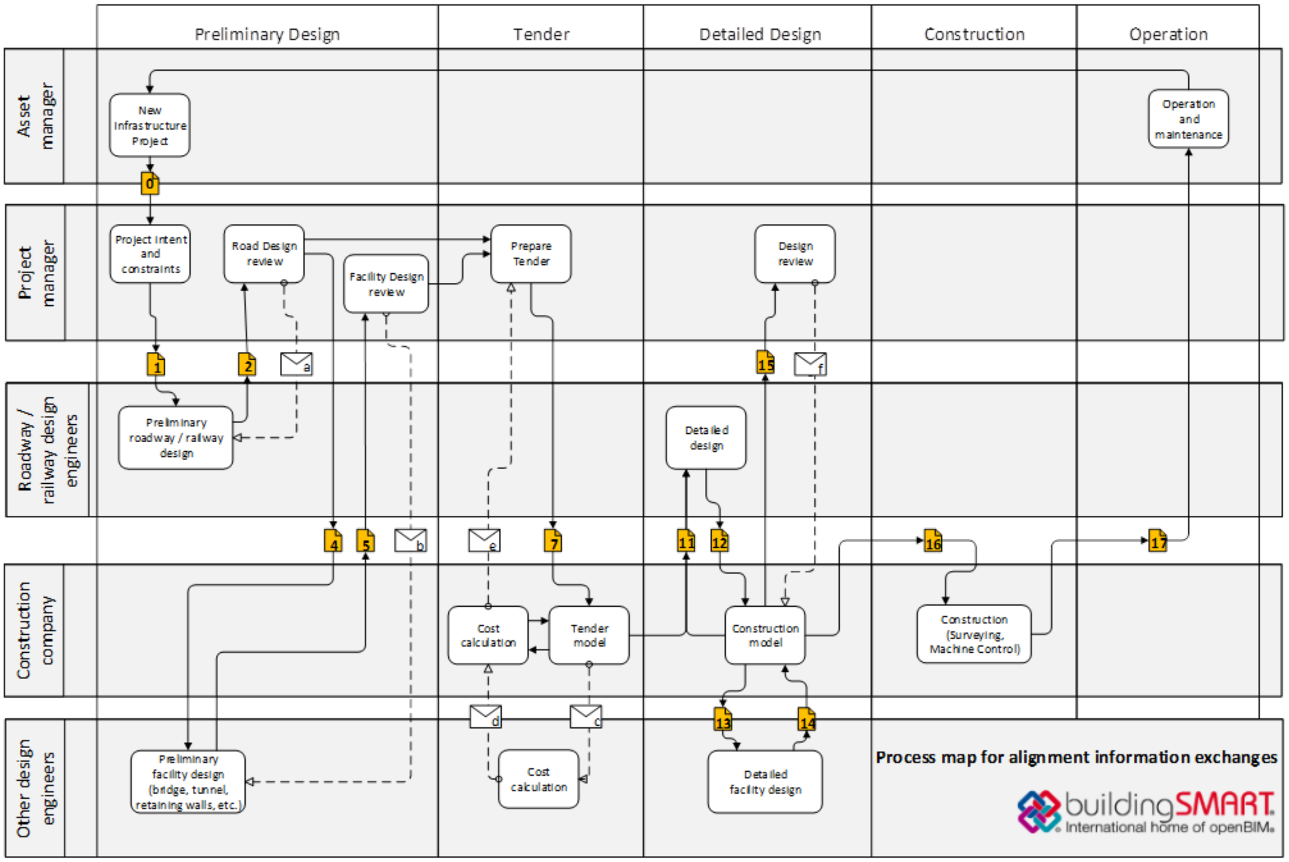


Figure 8: Process map for the design-build process of roadways and railways

2.1.1.3 Exchange scenarios and Levels of Detail

Each numbered document symbol within the process map represents a point of information exchange. Those are analyzed in regard to the sender, receiver, the purpose and the required details of the alignment information. Then the various levels of alignment definitions had been rationalized into three main Levels of Detail.

No.	Sender	Receiver	Purpose	Content	Level of Definition	Remarks / Issues / Open Questions
0	Asset Manager	Project Manager	Handover of as-built information of existing infrastructure as basis for new design activities	Detailed Alignment + additional information	Detailed design alignment with associated 3D elements	*If available electronically from asset database*
1	Project Manager	Road- or Railway design engineers	Request for new infrastructure project	Definition of boundary conditions	Simple horizontal alignment	*Early phase for alignment as special level of definition*
2	Road- or Railway design engineers	Project manager	Review of preliminary design (in cycles)	Preliminary Alignment	Preliminary design alignment	
4	Project manager	Facility design engineers	Handover of alignment information to facility designer (bridges, tunnels, retaining walls)	Preliminary Alignment + additional information	Preliminary design alignment	*Additional information is required (road width etc.)*
5	Facility	Project	Handover of preliminary	Models of bridges, tunnels	Preliminary design	*this may include alignment*

Figure 9: Analysis of the information exchanges

- Simple horizontal alignment
- Preliminary design alignment
- Detailed design alignment

The conceptual model and IFC extension do support all three levels.

2.1.1.4 Conceptual model

The jointly developed conceptual model together with OGC is an intermediate abstraction without taking the specific IFC Schema architecture into account. It reflects a new type of buildingSMART and IFC deliveries using the Unified Modelling Language (UML) for publishing an intermediate conceptual model.

When working on the conceptual model, an important design decision had to be made. There are two different ways to represent the horizontal and vertical sections of the alignment:

- either strictly by segments, each having a start point, a start direction, a segment length and curve parameter;
- or by a virtual point of intersection and distances to and from the VPI with additional curve parameter.

Both are valid representations and can be transformed into each other. The final decision had been to use the segment based approach.

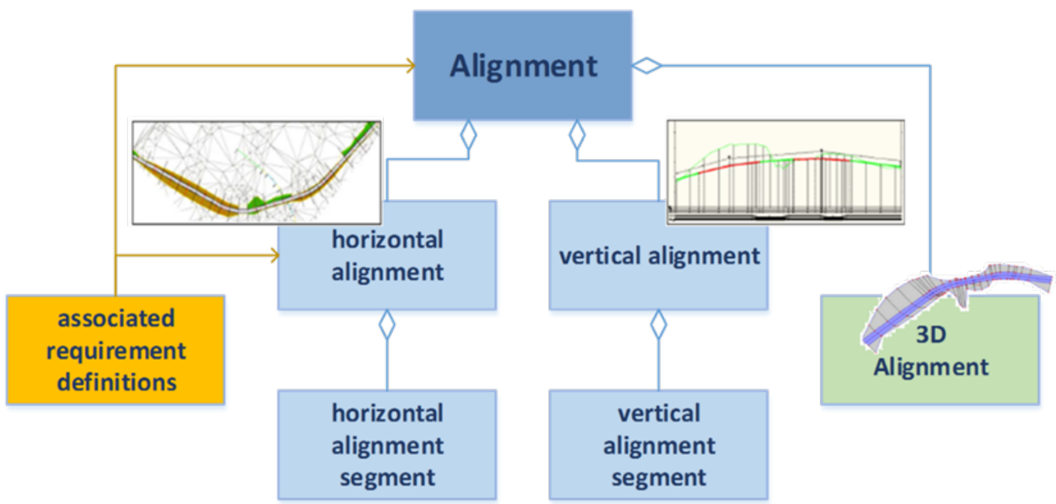


Figure 10: main structure of the alignment based on segments

The conceptual schema is reflecting this decision and is agreed among the buildingSMART and OGC groups as the international standard representation. Extensive work was performed to verify, that VPI can always be generated out of the segment representation.

In particular, this enables re-use of data stored in LandXML files, since the LandXML schema uses VPI based representation of alignments.

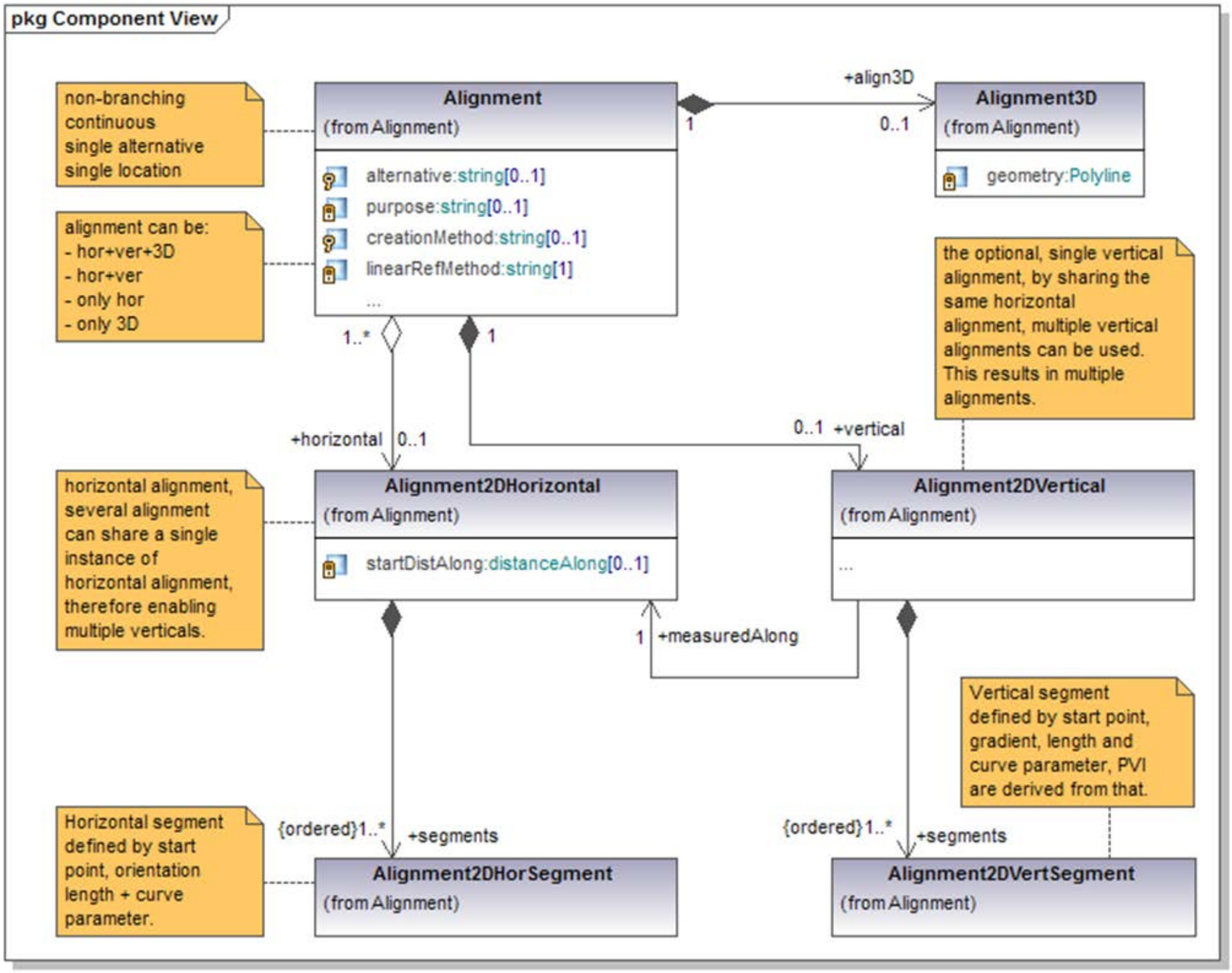


Figure 11: Conceptual schema using UML

2.1.1.5 IFC Extension

From the conceptual model, the IFC schema extension was derived. It had to be adapted to the existing IFC schema architecture and the existing IFC class hierarchy.

The alignment, as *IfcAlignment* class, is considered as a positioning element in a comparable way, as a building grid is seen as a positioning for building elements, such as columns. Hence the *IfcAlignment* class is introduced in parallel to the existing *IfcGrid* class.

Beside the *IfcAlignment* the following new definitions are included:

- *IfcAlignment2DHorizontal* as the end to start list of all horizontal alignment segments
- *IfcAlignment2DVertical* as the end to start list of all horizontal alignment segments
- *IfcAlignment2DHorizontalSegment* as the individual horizontal segment with the three geometric forms, line segment, circular arc segment and clothoidal arc segment

- *IfcAlignment2DVerticalSegment* as the individual vertical segment with the three geometric forms, line segment, circular arc segment and parabolic arch segment, each defined in distance along / Z coordinates

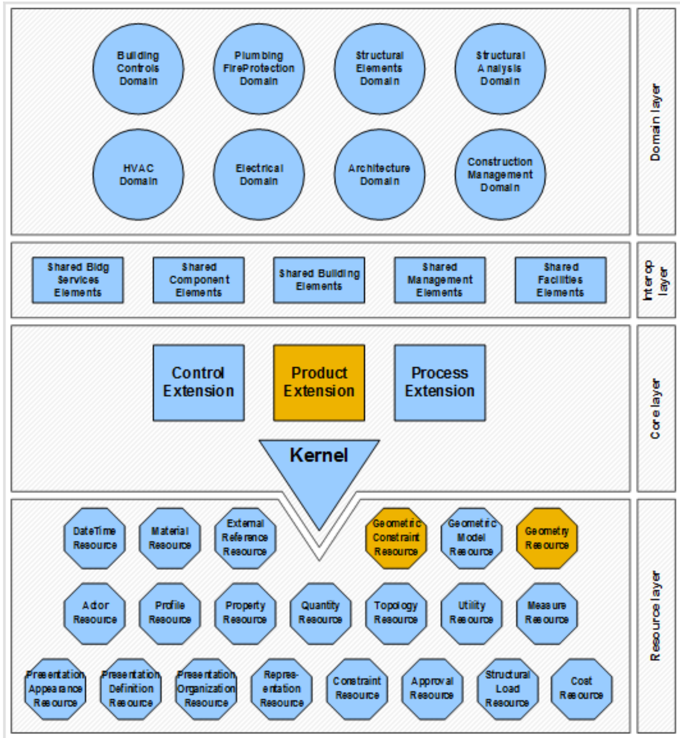


Figure 12: Extensions to the IFC schema architecture for alignment

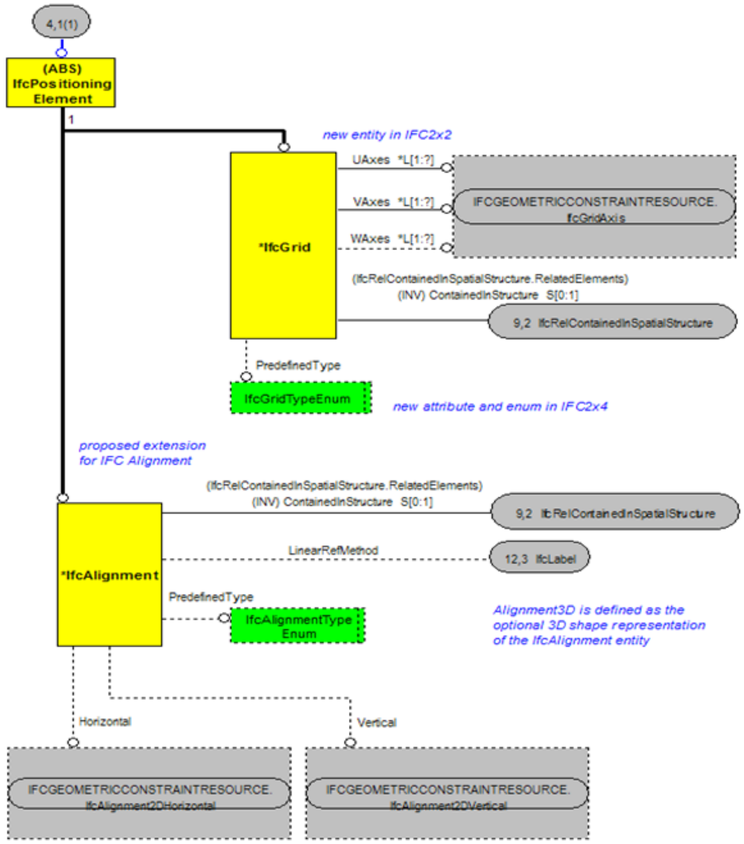


Figure 13: Excerpt from the IFC schema extension for alignment, defined in Express-G

The IFC schema extension was initially drafted as an Express-G visual diagram for review. The specification was produced using the ifcDoc software from buildingSMART, which generated combined documentation and formal computer-interpretable representations of the object model. The specification consists of:

- Formal representation in
  - the EXPRESS language, used by most IFC supporting software applications today, and defined in ISO 10303-11,
  - the XML Schema Definition (XSD), language representation, automatically translated using the configuration developed by buildingSMART from the ISO 10303-28
  - the mvdXML instance document, that defines the subset of the overall IFC schema which is required for a particular model exchange (the Model View Definition, MVD)
- Documentation, including
  - the overall purpose, scope and terms used within the alignment extension,
  - the semantic definition of each class, relationship and attribute of the alignment extension,
  - the concept templates and concepts explaining how to use the various alignment definitions,

- o various listings how to find, and diagrams explaining how to use the alignment definitions,
- o unit test cases with IFC data files with real cases of alignments

Figure 14: cover of the IFC alignment documentation

2.1.1.6 Software prototypes

An important part of the quality assurance during the development of the alignment extension has been the parallel development of prototype software, both for export and import. Many minor inconsistencies of the IFC schema extension had been found, and the document had been enhanced by the outcomes of the prototype development work.

During the development period, two software prototypes were used:

- the Open Infrastructure Platform, OIP, developed by the Technical University of Munich (Technical University of Munchen, 2016),
- the eveBIM software, developed by CSTB in France (CSTB, 2016)

The OIP, as the main development prototype, has been used to:

- read LandXML files (for alignment, terrain and cross sections)
- export IFC Alignment files (for alignment, terrain and geo-referencing)
- export an excel control sheet, including all LandXML and IFC parameter for verification

The eveBIM was then used to import and re-interpret the IFC alignment data files.

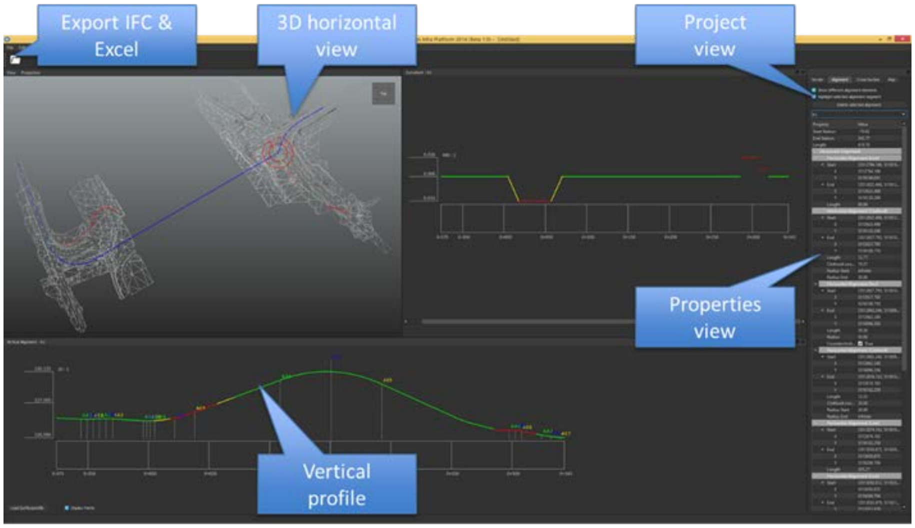


Figure 15: Screen shot of the OIP software showing a real case alignment

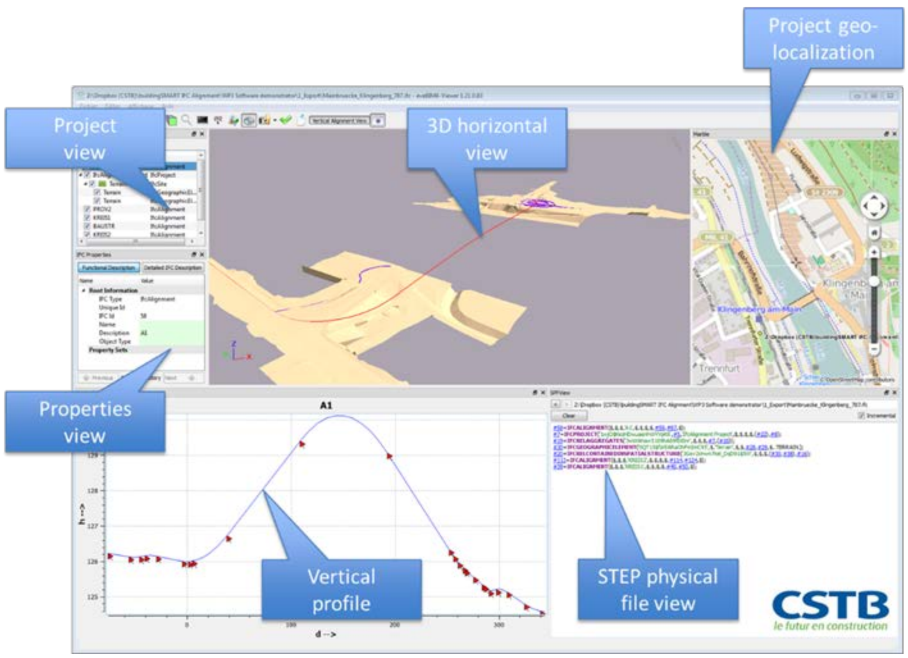


Figure 16: Screen shot of eveBIM with the import of the real case alignment

Towards the end of the alignment project, other software developers demonstrated the ability to import and use the IFC alignment data files. A few examples are shown below:

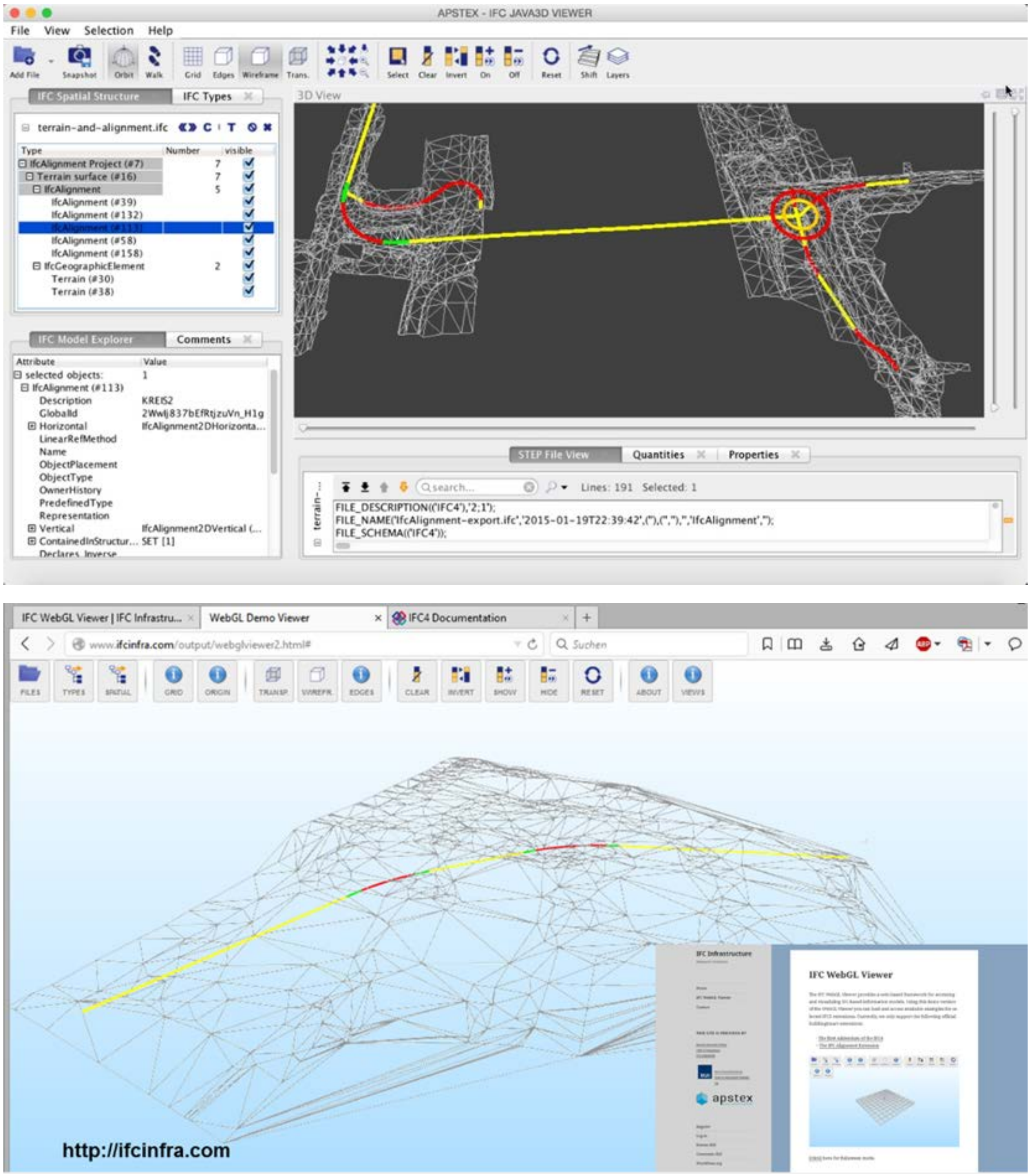


Figure 17: Import in desktop viewer and webGML viewer, developed by Apstek and RUB

2.1.1.7 Review process

The review process by an international group of 30 experts representing owners, engineers, and software developers, has been an essential and integral part of the IFC alignment project. The project accepted participation of 22 domain experts and 11 software experts with a total of 13 nations being represented. During 6 review meetings (4 by telecommunication, 2 in person), the progress had been reviewed, open questions were answered and conflicts were resolved.

The establishment of such an international group of experts demonstrated the ability to work globally and to establish an international body of knowledge. In return, it is now expected that the IFC Alignment data standard is accepted and deployed internationally as well.

2.1.1.8 Next steps

The detailed description of the IFC Alignment project has been provided to also demonstrate the ideal organization and sequence of an international data standard development project.

The important reasons for its success had been a well-defined scope of work, focusing on a particular demand and thereby allowing for a lean process to standardize, a small project team with the technical knowledge to execute the work, and a steering and expert group guiding the work and representing an international source of knowledge.

Three future lines of action are identified now to continue this work:

- A deployment project bringing software developers together with their key customers to implement the standard in commercial software solutions,
- A small extension, IFC Alignment 1.1 to add some of the identified additional scope of work, and
- Support for other ongoing projects on how to use the alignment information for upcoming IFC standards in roads, bridges, tunnels and waterways.

As of May 2015, some early prototype work demonstrates the ability to reuse the buildingSMART alignment standard for bridge design. Work within the U.S. on the feasibility study for the FHWA, executed by the National Institute of Building Sciences (NIBS), and within Germany on bridge design and maintenance, done at the Technical University in Munich has proven the successful implementation of the IFC alignment for Bridge Information Modeling, BrIM.

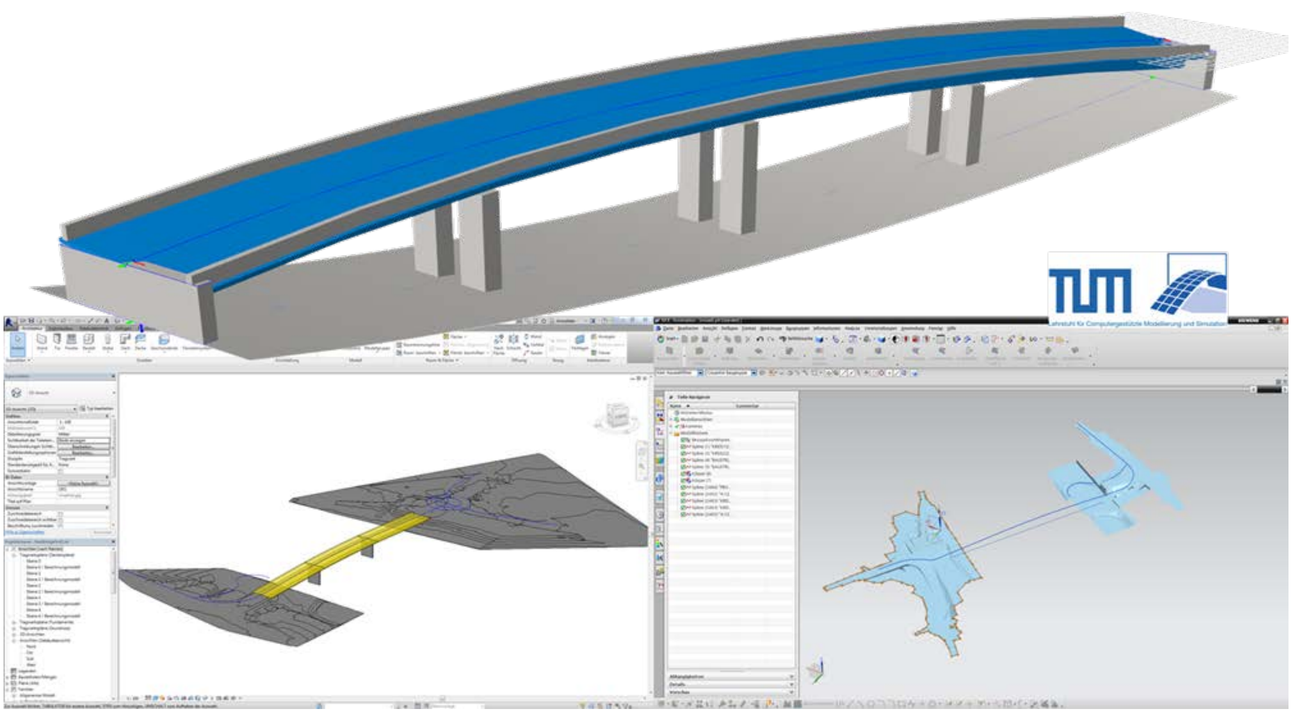


Figure 18: Implementation of IFC Alignment using Autodesk Revit and Siemens NX

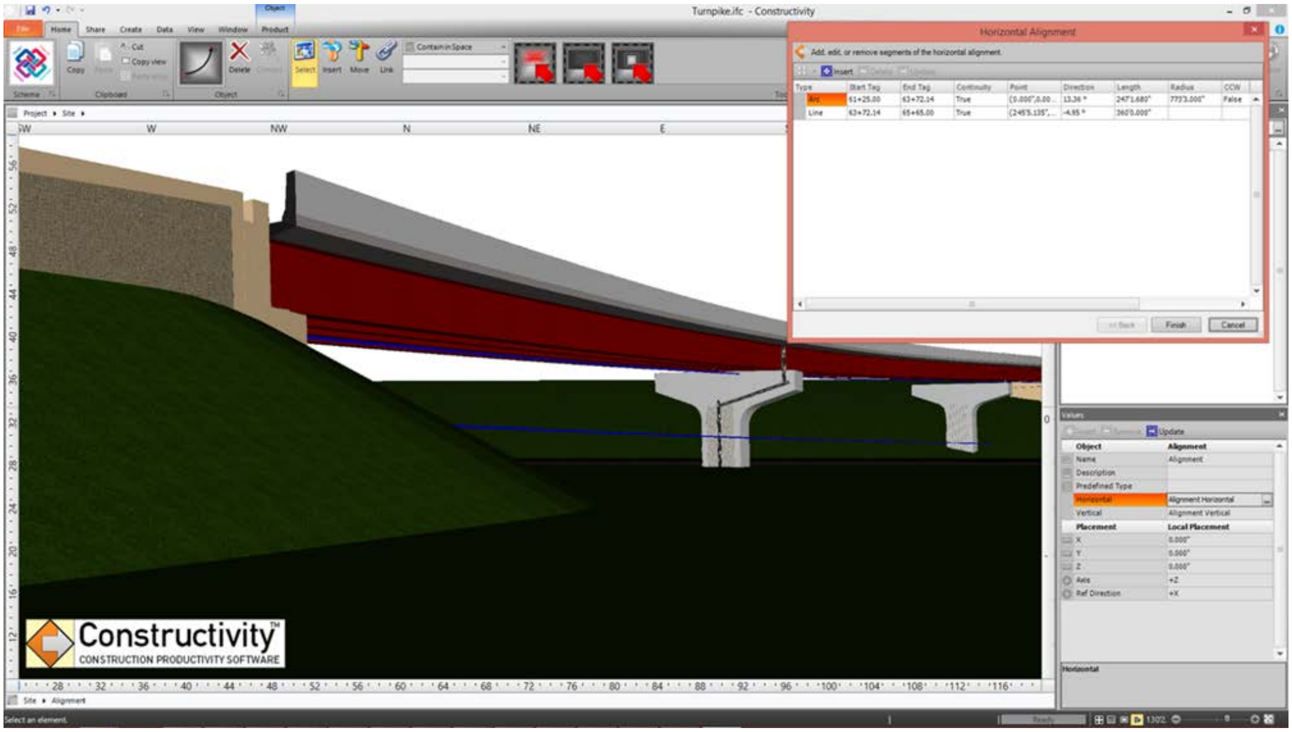


Figure 19: Use of the IFC Alignment in BrIM feasibility study (Constructivity)

2.1.2 IFC-Road

The next major undertaking for increasing the coverage of infrastructure work within the buildingSMART IFC standard is the international IFC-Road project. It is currently in a project development phase.

The overall scope is defined as "*Development of International IFC model extensions and data exchange standards for planning, design, cost estimation, scheduling and construction of roads and associated structures and earthworks*". Under a steering committee, initiated by the buildingSMART Infrastructure Room, stakeholders are invited to participate in this critical endeavor.

As a first step, existing international work is reviewed. The most prominent project is the Korean IFC-Roads project developed by KICT. The KICT project team has indicated its interest and willingness to participate. More details on the Korean IFC-Roads project is provided in section 1.2.1.

2.1.3 IFC-Bridge

Certainly, among all infrastructure works, bridges are among the most complex and have many structural elements with considerable commonalities to building elements. Therefore the extension of the exiting IFC data schema, primarily focusing on buildings to the inclusion of bridges would be a natural evolution.

An initial attempt had been made in the past by the initiative of the French Chapter of buildingSMART with support of SETRA, CSTB and several French construction companies. This early IFC-Bridge extension had been based on the IFC2x3 platform and dealt with the bridge specific structures and elements. One important, but difficult task had been the positioning and derivation of shape for the elements along the reference line of the bridge. This part is now revisited using the IFC Alignment solution.

The European research project V-Con has now taken over the work on the French IFC-Bridge project, together with an additional French working group, MINnD. The outcome will be an input into the international IFC-Bridge project, more information is provided in section 2.2.2.

2.2 Infrastructure standard developments connected to buildingSMART International

Besides the ongoing infrastructure data standard development projects already accepted by buildingSMART International, there are several initiatives either by regional chapters of buildingSMART or by organizations and consortia, which link to the IFC (ISO 16739) development and often suggest extensions to IFC to better incorporate the needs of various infrastructure works.

The buildingSMART Infrastructure Room then acts as the meeting point of those projects and could take proposed specifications onboard for consideration as international standards after thorough international participation and review. It can also liaise with other organizations for joint developments, in particular with the Open Geospatial Consortium (OGC).

Currently the following international initiatives are active:

- IFC-Roads by the Korean Institute of Construction Technology, KICT, Korea
- IFC-Bridge, previously supported by SETRA, now through the MIND'S project, France
- Various tunnel projects to define an IFC-Tunnel extension (early developments)
- Development on a Rail standard based on IFC4 for the Chinese Railway

The following sections introduce each of these project in further detail.

2.2.1 Korean IFC-Roads project

The Korean government is heavily supporting the development, deployment and use of BIM within the construction industry. Within a large 5 year project, for a total of about $3M, the Korean Institute of Construction Technology (KICT) has been tasked to develop a standard for BIM in road design and construction (Choi, 2016). The project lasts from 2012-2016.

After analyzing existing work, KICT decided to base its developments on top of IFC4 (ISO 16739). The following scope has been addressed:

- Roads
- Supporting structures
    - Bridges for roads
    - Tunnels for roads
- Drainage
- Earthwork
- Subsidiary facilities (signals, lights, etc.)

The goal is to define a Korean standard that is backed by an international standard. The international standard is identified as the next version of IFC issued by buildingSMART International with the intension to submit it to ISO as ISO 16739:ed. 2. The team of KICT is actively engaged with buildingSMART International to bring the intermediate results forward to internationalization.

2.2.1.1 Scope definition

While the project looks at the overall life-cycle of road facilities, the main focus is the handover of detailed design to construction and to the governing agencies. The terrain model, the civil elements organized in the various structures, and subsidiary objects shall all be created as 3D objects and enriched by predefined property data. The KICT IFC-Road schema defines the structure for these object definitions and it includes reference data in terms of property set definitions.

Figure 20: Scope of the IFC-Road project (source: KICT, presentation by H. Moon)

The development of the KICT-Road schema was executed in three rounds of continuous refinements. At the start, it included a deep survey of existing breakdown structures, an object structure based on the IFC object hierarchy and the more traditional work breakdown structure. In addition, other specifications such as LandXML, the Finnish InfraModel, and the Japanese JHDM work had been analyzed.

A refinement of scope was part of the third round of development that has the target to internationalize the work. The refined scope definition is:

“The main scope of the IFC Road project is to extend product data model of road facilities with earthwork enabling open data access based on IFC4 (ISO16739) schema in order to secure interoperability in delivering the as-built design model to government. In this project, the critical facility scope includes spatial structure related to road, roadway with cross-section, earthwork with cut & fill, drainage facilities.”

The main life-cycle phase considered, and therefore also the predominant level of detail for the shape representations and the property set definitions is the delivery of the detailed road model for the detailed design phase before preparing shop drawings for construction process.

Previous and parallel work on IFC-Alignment, IFC-Bridge and IFC-Tunnel would be integrated, once released.

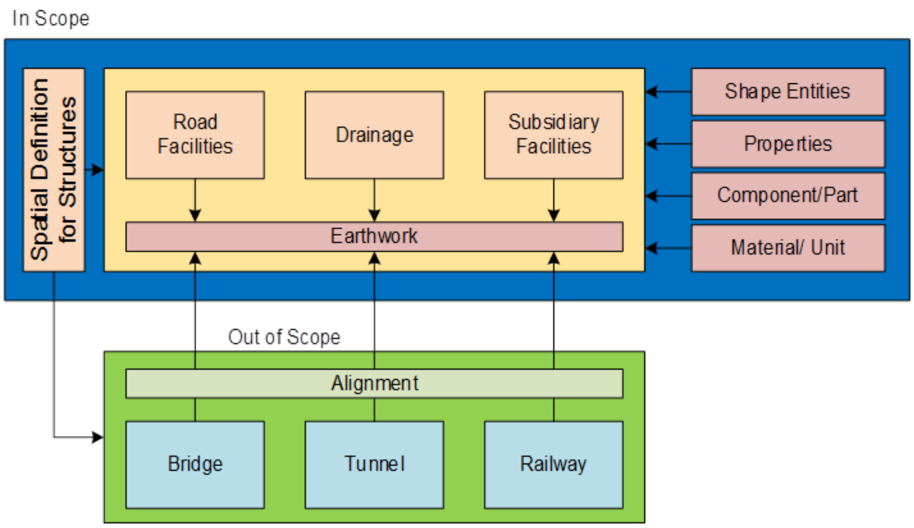


Figure 21: New scope definition of Korean IFC-Road (source KICT)

The IFC Road definition developed by KICT provides a common definition layer for road extension schemas. The data schema definition of road includes:

- Spatial structure: spatial structure breakdown suitable for road constructions
    - Local engineering and geospatial coordinate reference system
    - Alignment for linear placement and positioning
- Physical structures: road facilities, earthwork, civil common, culvert, retaining wall, drainage, and subsidiary facilities
    - Assemblies to group elements into the next higher facility level
    - Shapes to describe complex road elements
- Properties: common properties of road project suitable for the level of information "design to operation phase", suggesting unique properties for each element
- Earthwork model: an original terrain model for the construction site, cut & fill entities as an volumetric shape object, other terrain models

2.2.1.2 Process map

Following the general IDM approach, developed by buildingSMART to capture exchange requirements for construction projects (ISO 29481-1), the KICT project defined a process map for the overall life-cycle phase of road projects, and later a detailed process map for the detailed design and the construction phase. Both life-cycle phases are in focus of this project.

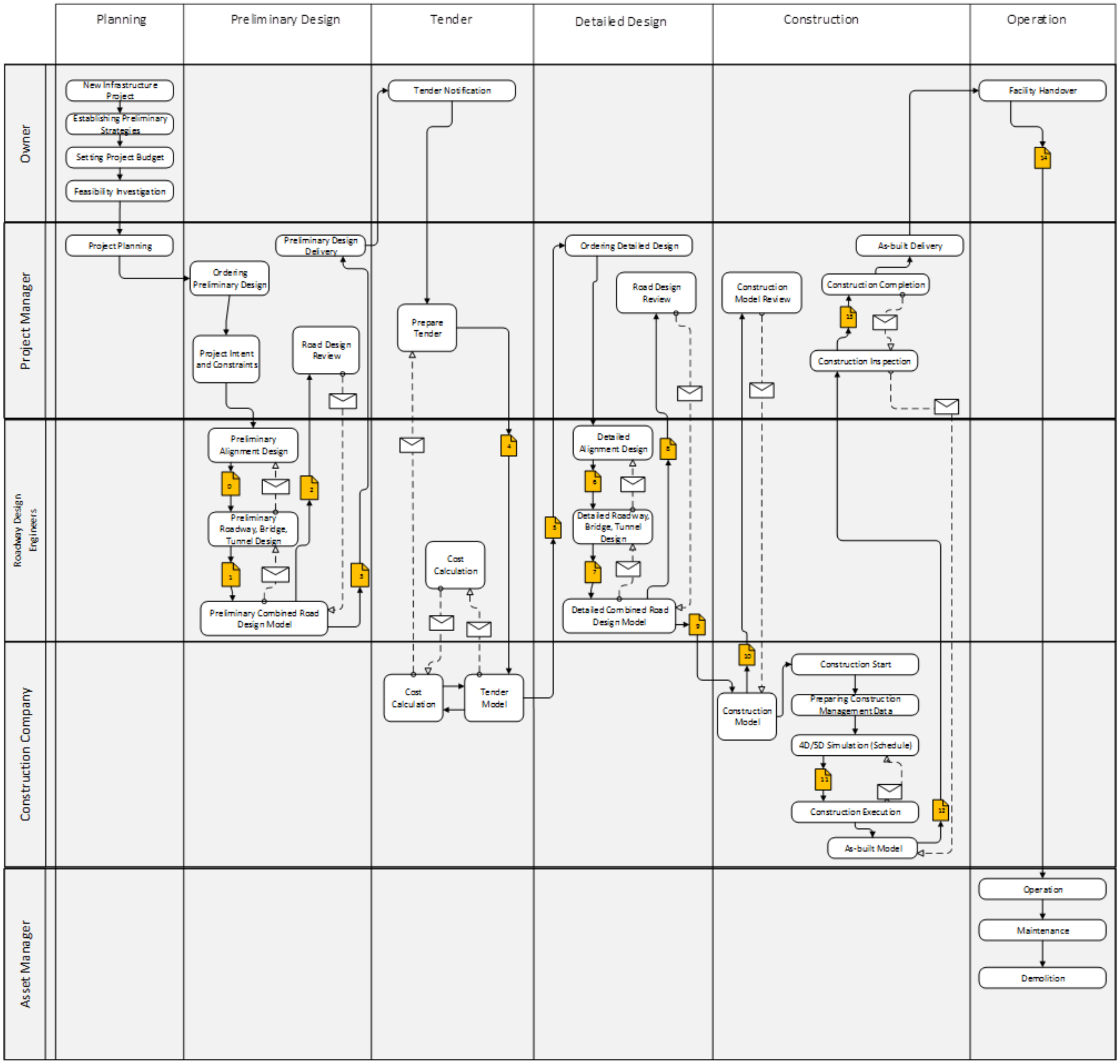


Figure 22: The overall process map developed by KICT (Moon, 2016)

2.2.1.3 Exchange definitions

Following the same approach as the IFC Alignment project, the next step has been to identify the information exchanges (the document symbols within the diagram) to determine the main content. Those are analyzed in regard to the sender, receiver, the purpose and the required details of the road information.

No.	Sender	Receiver	Purpose	Content	Level of Definition
0	Project Manager	Roadway Design Engineers	Establishment of Plan, Optimum Alignment Study	Project Planning Report	
1	Roadway Design Engineers	Roadway Design Engineers	Establishment of Facility Plan	Alignment Planning Model	Alignment Planning Mod
2	Roadway Design Engineers	Roadway Design Engineers	Beginning the Surveying and Inspection	Facilities Planning Models	Facilities Planning Models
3	Roadway Design Engineers	Roadway Design Engineers	Beginning the Preliminary Design	Planning Models (Alignment, Facilities, Structures), Surveying and Inspection Reports	Planning Models
4	Roadway Design Engineers	Roadway Design Engineers	Preliminary Structures Design	Preliminary Design Road Facilities Models	Preliminary Design Road
5	Roadway Design Engineers	Roadway Design Engineers	Integration of Preliminary Design Road Model	Preliminary Design Road Structures Models	Preliminary Design Road
6	Roadway Design Engineers	Roadway Design Engineers	Preliminary Cost Calculation	Integrated Preliminary Design Road Model	Integrated Preliminary De
7	Roadway Design Engineers	Project Manager	Preliminary Road Design Model Review	Integrated Preliminary Design Road Model	Integrated Preliminary De
8	Project Manager	Project Manager	Preparaing Tender	Integrated Preliminary Design Road Model (Reviewed)	Integrated Preliminary De
9	Project Manager	Roadway Design Engineers	Beginning the Detailed Design	Detailed Design Planning Report	
10	Roadway Design Engineers	Roadway Design Engineers	Structural Analysis of Bridge and Tunnel	Detailed Design Bridge and Tunnel Models	Detailed Design Bridge an
11	Roadway Design Engineers	Roadway Design Engineers	Integration of Detailed Design Road Model	Detailed Design Road, Facilities, Stuctures Models	Detailed Design Road, Fac
12	Roadway Design Engineers	Roadway Design Engineers	Detailed Cost Calculation	Integrated Detailed Design Road Model	Integrated Detailed Design
13	Roadway Design Engineers	Project Manager	Detailed Road Design Model Review	Integrated Detailed Design Road Model	Integrated Detailed Design
14	Project Manager	Roadway Design Engineers	IFC-Road Model Preparation	Integrated Detailed Design Road Model (Reviewed)	Integrated Detailed Design
15	Roadway Design Engineers	Roadway Design Engineers	Completed Road Design Model Delivery	IFC-Road Model	IFC-Road Model
16	Roadway Design Engineers	Roadway Design Engineers	Preparation and Submission of Permission Documents	As-built Road Model	As-built Model
17	Roadway Design Engineers	Roadway Design Engineers	Preparation of Design Documents Handover	Permission Documents with Final Road Models to be delivered	As-built Model
18	Roadway Design Engineers	Project Manager	Delivery Inspection	Overall Design Documents	As-built Model

Figure 23: The table with main information exchanges developed by KICT (Moon, 2016)

2.2.1.4 Concept for IFC extension development

In order to plan for the extension development work, the KICT team had considered the following methodology:

- Existing work breakdown structures for civil facilities
- Structures of road reference models (such as LandXML, InfraModel, JHDM and OKSTRA)
- The existing schema hierarchy of IFC4 (ISO 16739)

The main dependency diagram between spaces, facilities, physical elements, (sub-) components and construction resources represents the outcome of this work.

Figure 24: Underlying structure for the IFC Road extension schema (Choi, 2016)

Physical elements are governed by type definitions that hold common shape, material and property information. This adds another main part to extend the IFC schema.

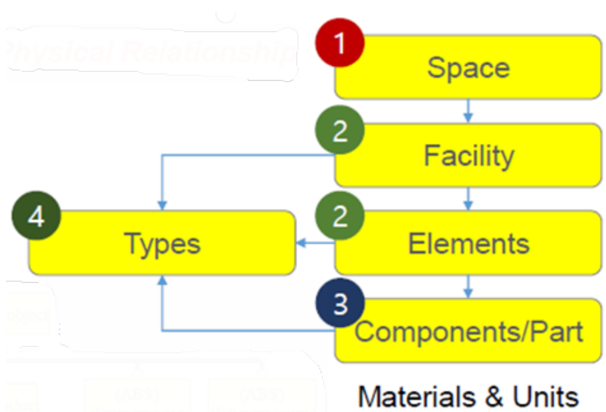


Figure 25: Adding types to the underlying structure (Choi, 2016)

These IFC classes needs to be added for the scope, comprising all aspects of road construction. Bridges and tunnels, being part of the overall road construction are handled internally for the Korean standard.

However, if an international IFC-Bridge and IFC-Tunnel Standard would become available, it should be replaced by the international definitions (see also Figure 20).

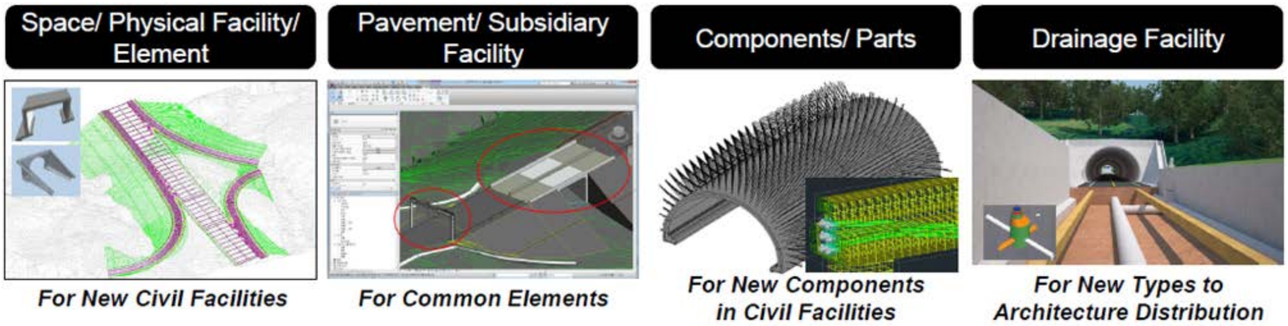


Figure 26: Example of spatial, structural and physical elements in scope (Choi, 2016)

The necessary property information for the new road infrastructure extension is defined as property sets using the existing IFC definitions (Property Set Definition PSD Language). Since many individual properties reflect local usage, such as classifying properties, commissioning properties, etc. a division is made between property definitions suggested for international usage, and those for local usage only.

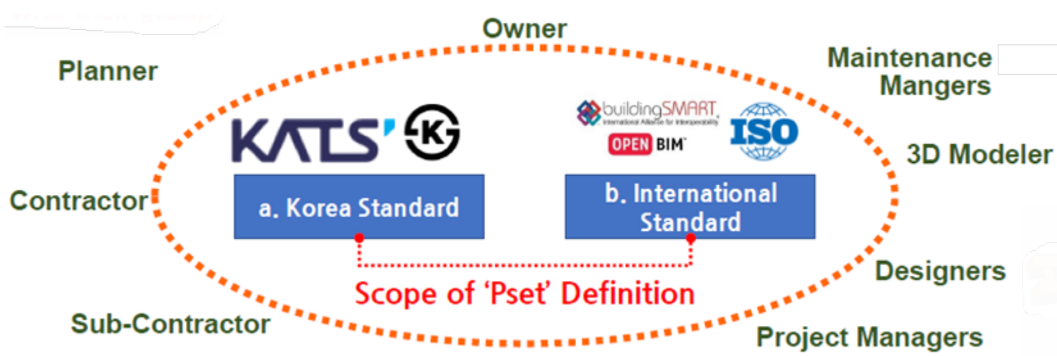


Figure 27: Division of property definitions internationally and locally (Choi, 2016)

2.2.1.5 IFC Extension

An IFC Extension schema is currently defined in its third revision. This takes into account the review week, where the second revision was analyzed by the chair of the buildingSMART Model Support Group, the team responsible for the development and maintenance of the IFC schema.

The work had been presented at the recent meetings of the buildingSMART Infrastructure Room meetings and is accepted as one source for the upcoming international IFC-Road project.

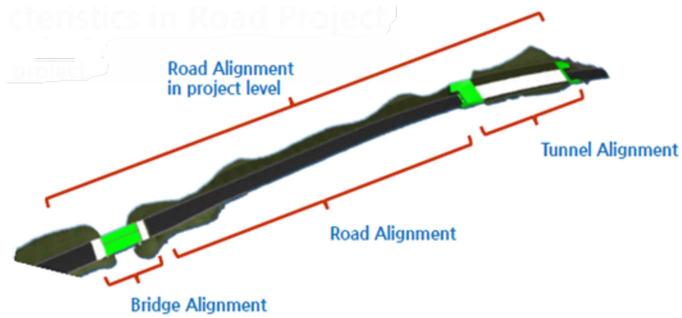


Figure 28: Curvilinear characteristics in road projects

One difficult task, as in any standardization project leading to a data schema, is the determination of the right level of class breakdown structure, not too shallow (so that only "proxy" elements are available"), but also not too deep to create too many classes and be too rigid for easy extensions for future needs.

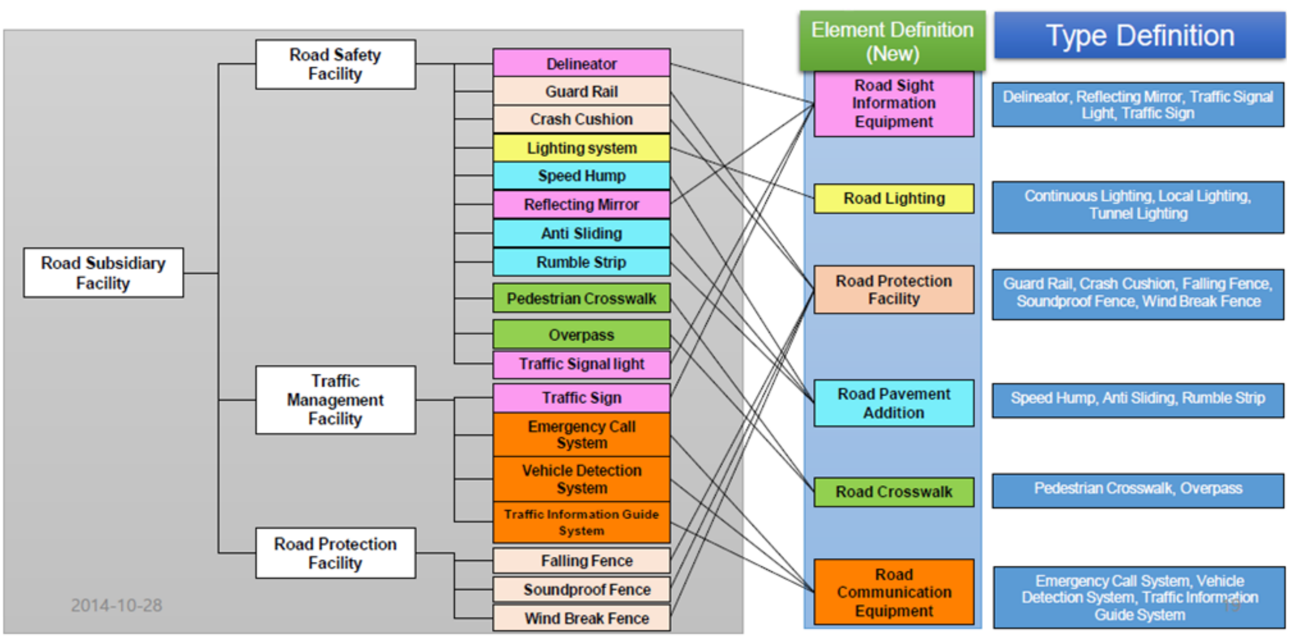


Figure 29: Determine the class breakdown structure (Choi, 2016)

2.2.1.6 Software prototypes

The development of IFC-Road converter and viewer to enable quality control of the IFC-Road extension schema was a parallel project that ran for 6 months in 2014 (and for a total of $75K).

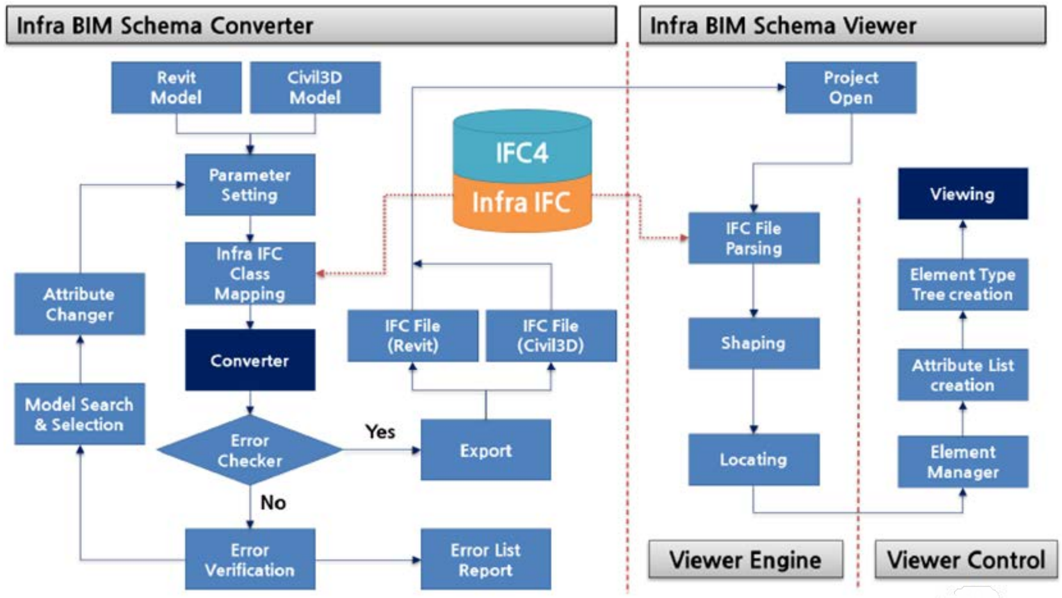


Figure 30: Workflow for the validation of converter and viewer (Choi, 2016)

Two add-ins have been developed, one for Autodesk Revit (for structures) and one for Civil3D (for road and earthwork). A separate view had been implemented to visualize the results of the export through those add-ins.

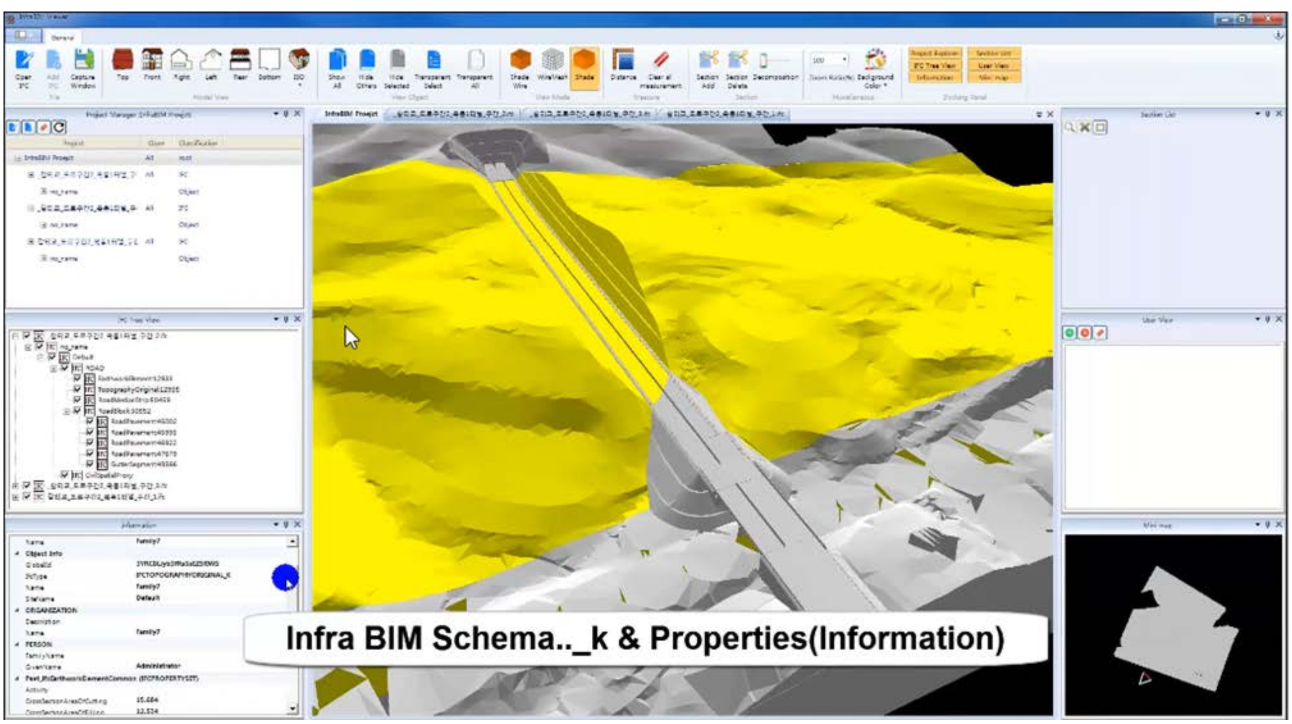


Figure 31: Korean IFC-Road Viewer (Choi, 2016)

2.2.2 French IFC-Bridge project

The IFC-Bridge project begun on the basis of the IFC2x(3) platform as a French initiative (Ferries, 2001). It was based on an earlier national standard, called OA-Express. The goal of the IFC-Bridge project is the description of 3D object models for bridges over their life-cycle phases.

In 2014, the previous work had been revitalized by two new projects, the European research project V-Con and the French project MINnD (Claude Dumoulin, 2016). The latter also included actual test of use cases with selected software tools.

The IFC-Bridge extension is currently upgraded to IFC4, and the previous part dealing with the bridge reference line is replaced by the new IFC Alignment standard. The most difficult task now is to describe the shape of bridge elements along the alignment, using a sub section of the alignment as a directrix for sweeping operation.

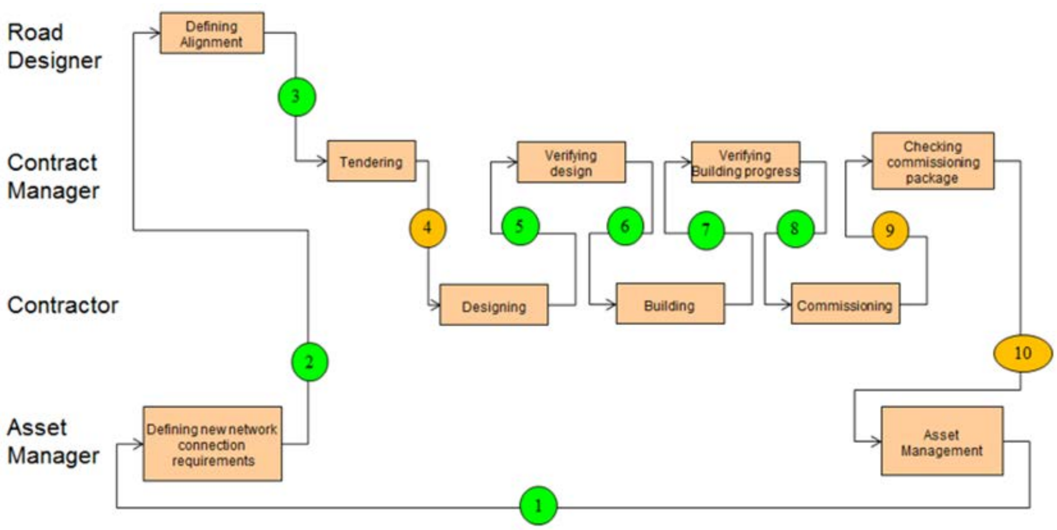


Figure 32: Overall process definition and information exchanges (source V-Con)

The main breakdown structure of the bridge model includes (very similar to the IFC-Road breakdown) spatial structure, physical element structure, (sub-) component structure and necessary extensions to the geometry to cover curvilinear element shape.

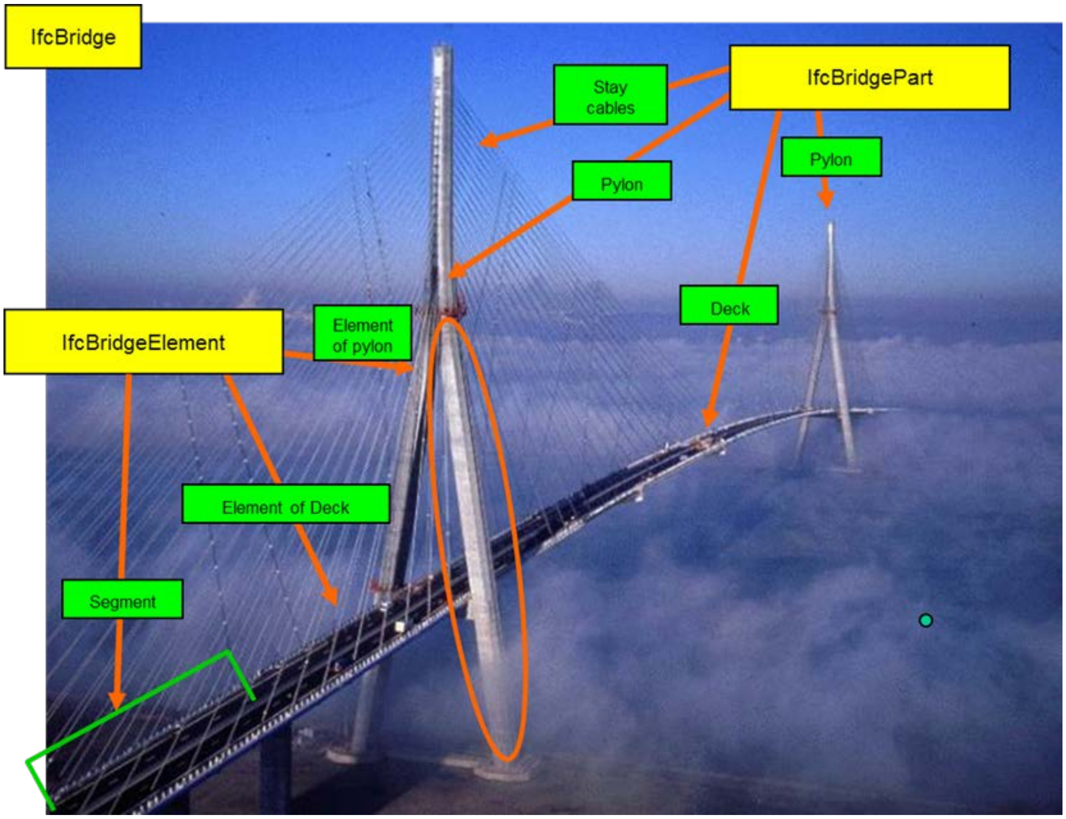


Figure 33: Illustration of the IFC-Bridge project (source CSTB)

Bridge decks, and segments of the bridge deck represent complex geometric forms, swept along a directrix defined by an alignment, but with tapering along the sweep. It is a current challenge to correctly define the necessary IFC geometry extension.

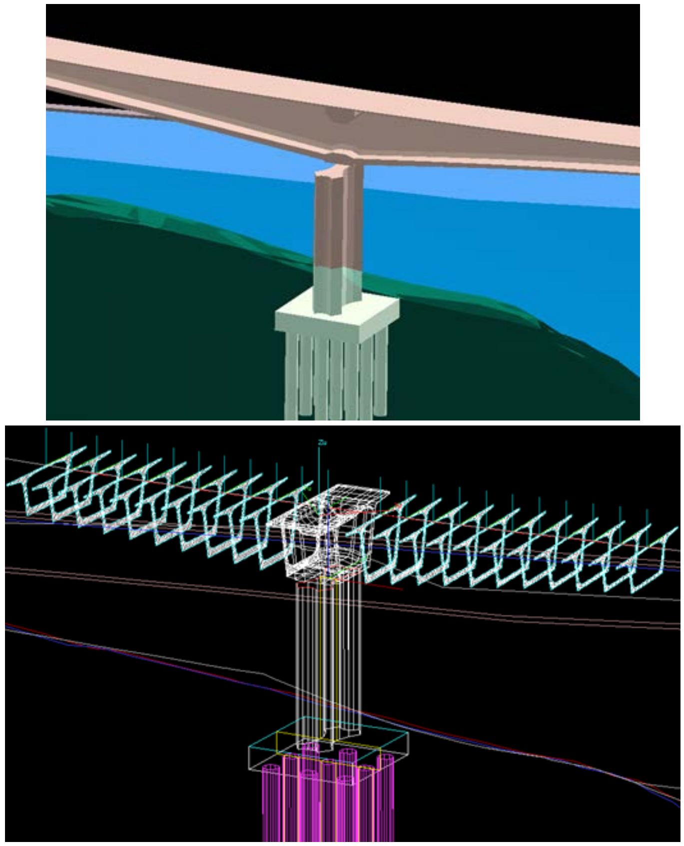


Figure 34: Example of a complex form of a bridge deck (source CSTB)

An IFC-Bridge viewer had been developed, currently with limited capabilities, to visualize the test examples for bridge structures.

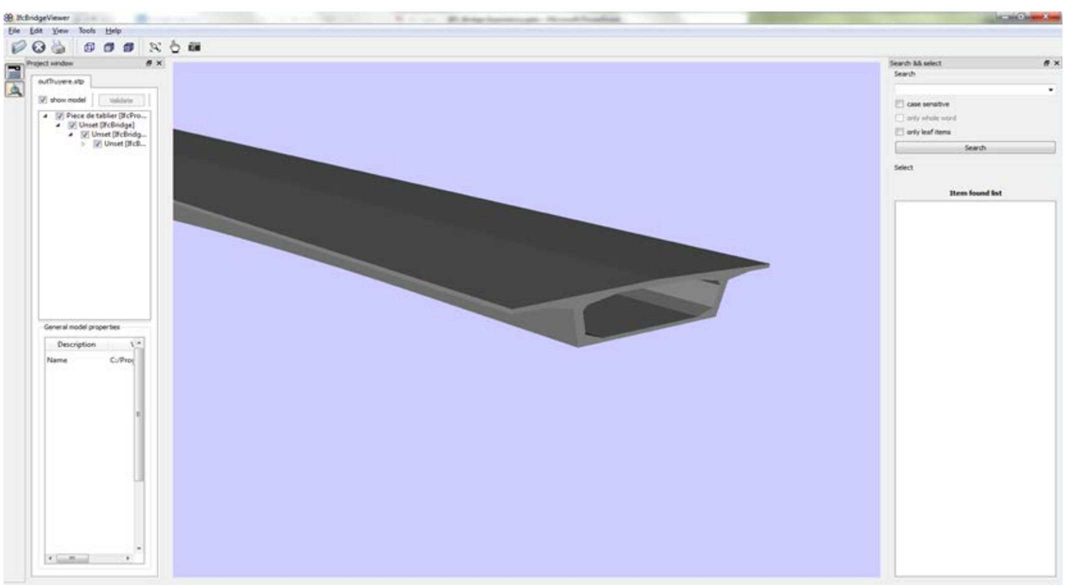


Figure 35: IFC-Bridge prototype viewer

The new French MINnD project has identified four bridge use cases:

- A standard bridge as underpass
- A standard bridge as overpass
- A steel bridge with approaching viaduct (more complex alignment)
- A suspension bridge (as a more outstanding case)

The first three use cases were modelled in existing 3D BIM software and exported as IFC files (using the current IFC2x3 schema) to test what can already be achieved today and where are the shortcomings. Software used included Autodesk Revit and Tekla Structures.

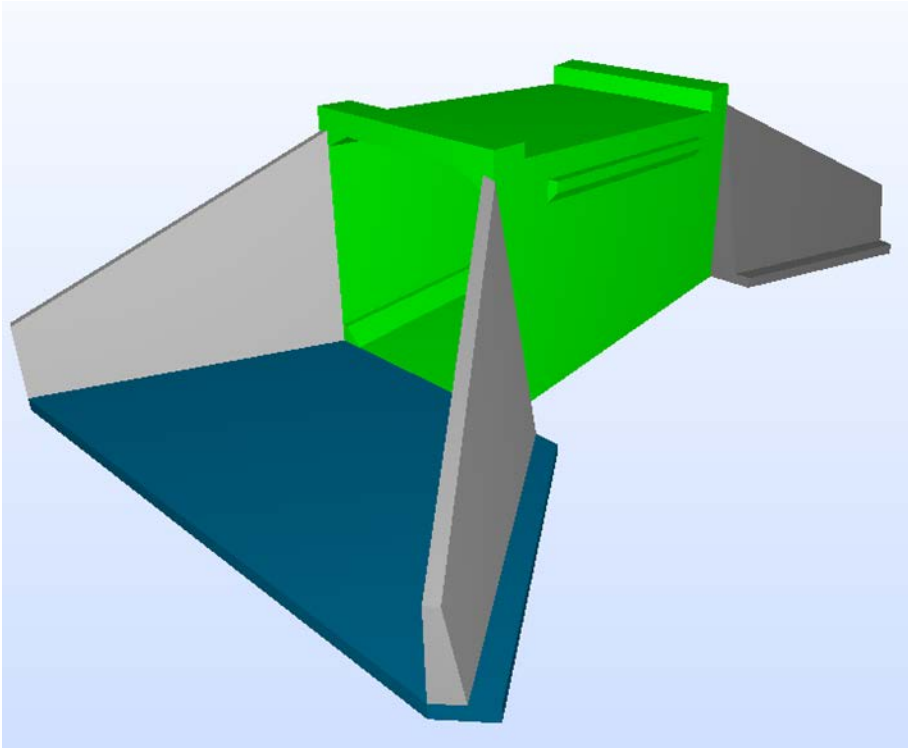


Figure 36: Underpass use case (source MINnD, Mobyus, Bouygues)

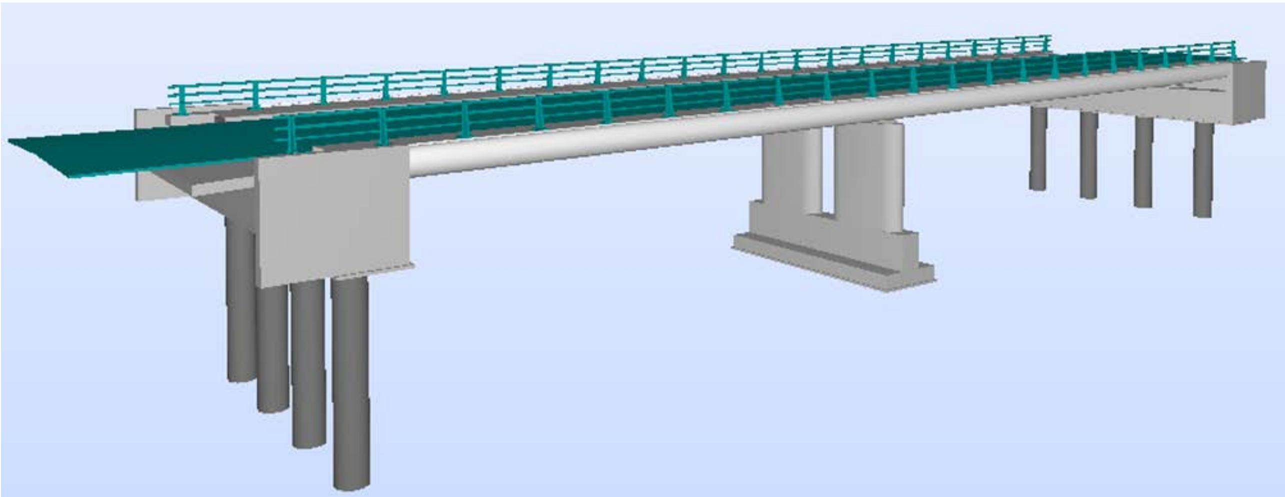


Figure 37: Overpass use case (source MINnD, Mobyus, Bouygues)

The results of the as-is analysis had been obvious. The current IFC2x3 export from the modeling tools are based on the previous IFC breakdown structure mainly focusing on buildings. Whereas the geometry as boundary representations, object names and properties are maintained, the more semantically rich bridge information (alignment, bridge breakdown structure, sweep and CSG geometry, linear placement and bridge element classes) is not present. This gap needs to be closed with the IFC-Bridge extension.

The third use case had been an existing steel bridge in Oslo Norway that was designed using a steel construction BIM software, here Tekla.

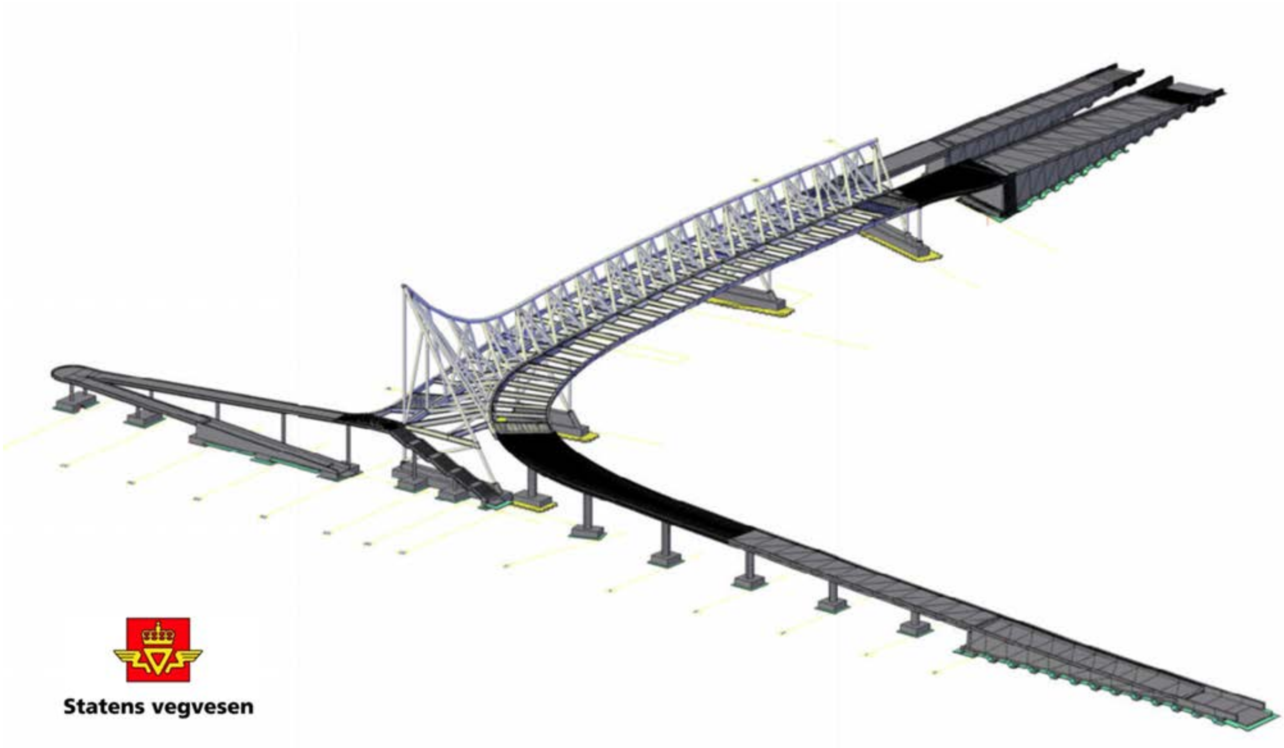


Figure 38: Use case steel bridge (source Statens vegvesen, MINnD)

Here, more complex geometric forms (sweeps, CSG geometry) had been used. In the final conclusion, the project stated that IFC is ready as a baseline schema to bridges, and:

- IFC can be used for bridges and most of the needed semantics are already there,
- IFC bridge specific entities need to be added
- IFC use (workflows on how to use IFC models) needs to be defined (using IDM methodology) and proper Level of Detail / Definition have to be described
- IFC export/import of bridge structures need to be tested and finally certified (same rigor as for buildings)
- IFC version management needs to be improved

2.2.3 Various IFC-Tunnel projects

Several initial studies have been done to explore the possibility to define an IFC extension for tunneling. Initial work had been performed in Japan (Yabuki, 2015), and recently a research project in Germany allowed for the continuation of the work (Amann, et al., 2013). In contrary to the road and bridge work, there is not yet an ongoing project preparation at the international level of buildingSMART.

Two work programs at Germany universities supported by a research grant had recently intensified the work on IFC-Tunnel. It includes the IFC definitions for the tunnel and the shied tunnel machinery. It also adds Definitions of Detail for tunnels.

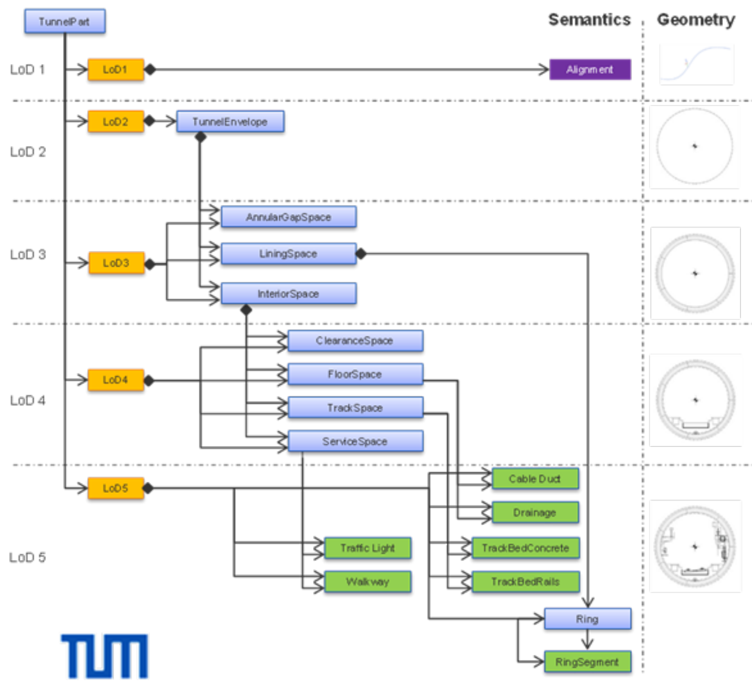


Figure 39: Definition of Level of Details for IFC-Tunnel (source Tech. Univ. of Munich)

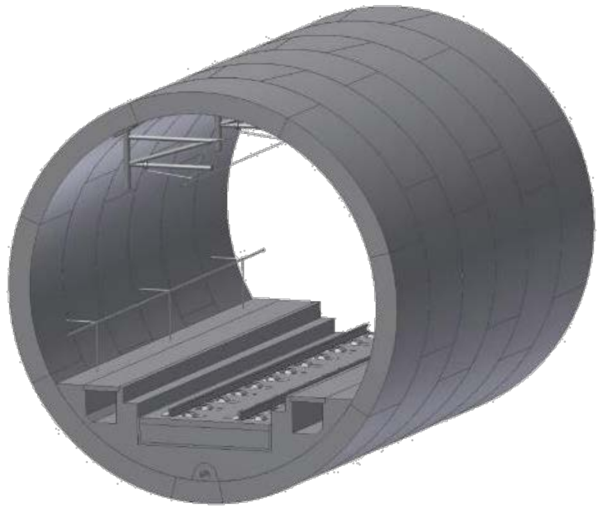


Figure 40: Tunnel segment as use case (Amann, et al., 2013)

2.2.4 Chinese IFC-Rails project

The China Railway BIM (CRBIM) Alliance presented their latest developments in standardizing information exchanges for railways at the buildingSmart International conference in Singapore in October 2015. (Ge, 2015). It consists of three parts: (1) Railway BIM Classification Standard, completed in December 2014; (2) Railway BIM Data Standard, draft version completed in June 2015; and (3) Railway BIM Delivery Standard, expected to be available in 2016.

Figure 41: Presentation from China Railway BIM Alliance

The Data Standard contains data definitions for track, subgrade, bridge, and tunneling components, built on top of IFC 4 specification. Figure 42 illustrates a diagram of definitions for rail track.

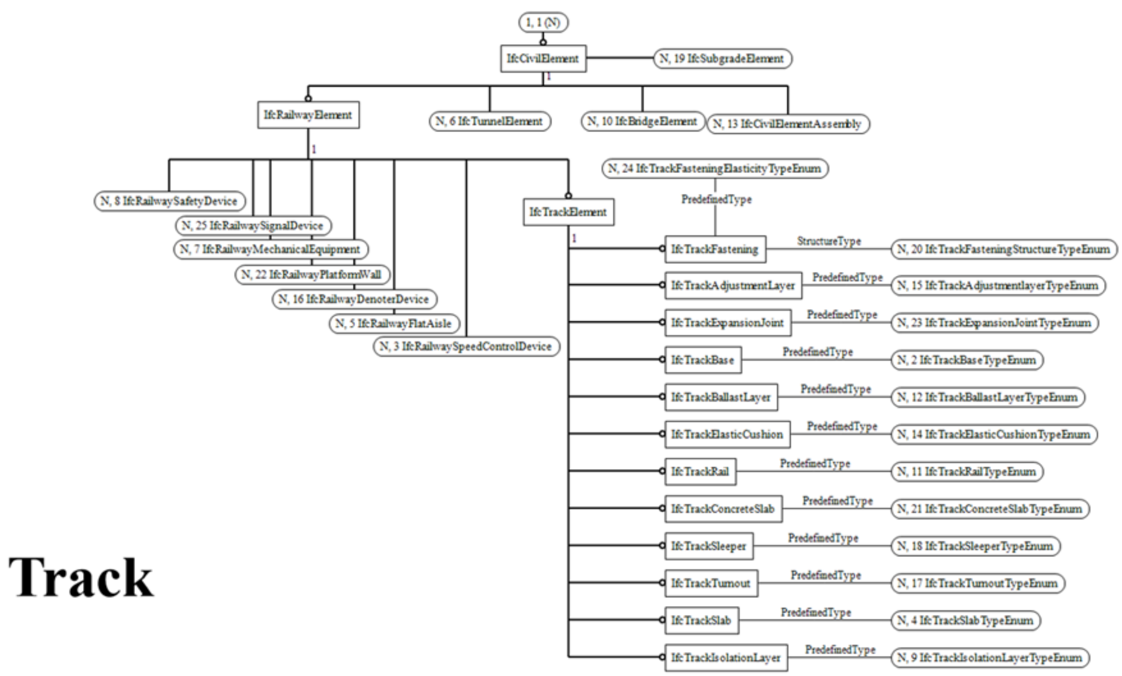


Figure 42: China Railways proposed definitions for rail track

2.3 Worldwide Integration Strategy

In February 2016, project technical leaders from Korea Institute of Construction Technology (KICT), China Railway (CRBIM), Japan Construction Information Center Foundation (JACIC), this project (U.S. Federal Highway Administration), and buildingSmart International met to review recent work and chart a path for integration of efforts.

KICT hosted meetings in Seoul for discussing integration of IFC standards for roadway design and construction. In addition to detailing an expanded IFC schema, KICT also developed plug-ins for Autodesk Revit and Civil 3D for exporting IFC files conforming to the expanded schema, along with a custom IFC viewer program containing many features useful for infrastructure.

While KICT has referred to their project as "IfcRoad Standard", the scope also includes bridges, where they have developed an expanded classification scheme for physical elements found in roads and bridges. KICT has leveraged the new alignment definitions found in IFC4.1, and has leveraged geometry definitions already available and widely supported in software. KICT has generated various sample IFC files for bridges and roadways.

During the meetings at KICT, the FHWA project team was able to load files developed by KICT simply by adding several new element classifications as shown in Figure 45, demonstrating true interoperability for bridge models across independent software applications.

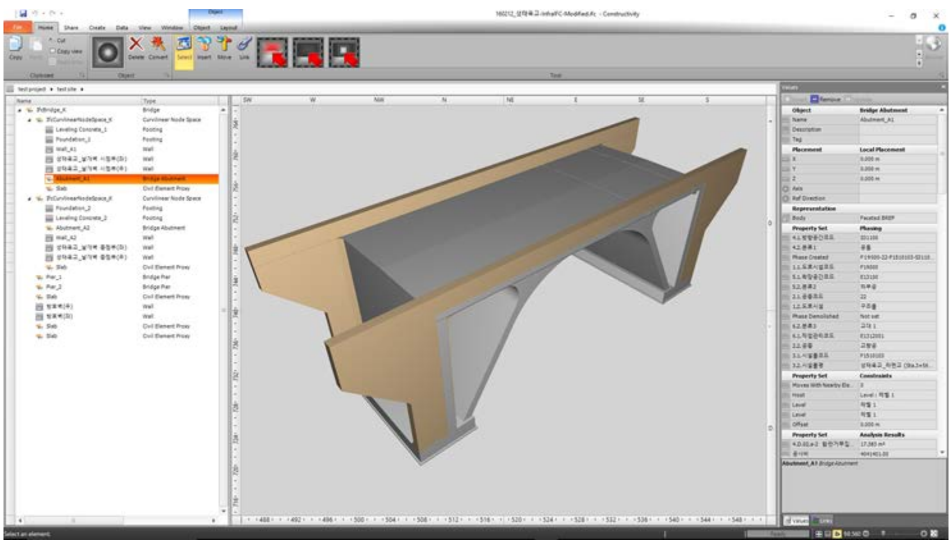


Figure 43: KICT IFC model loaded into FHWA-based software

China Railway hosted meetings in Beijing in February 2016 for discussing integration of IFC standards for railway design and construction. Leveraging the massive scale of their market, China Railways has lined up key software vendors in supporting IFC railway standardization efforts.

While the focus of China Railway is specific to their railway and subway system, like KICT their scope also includes bridges, where they have developed an expanded classification scheme.

China Railway has also leveraged the new alignment definitions found in IFC4.1, and has leveraged geometry definitions already available and widely supported in software.

For alignment positioning, China Railway Corp has identified additional transitional curve types (besides clothoid) that are used in their railway system, and has proposed new data definitions to be incorporated into the IFC 4.2 extension for alignment positioning now undergoing development. This FHWA effort has proposed a new data definition for positioning and transforming geometry that may be leveraged by other efforts and also incorporated into the IFC 4.2 alignment extension.

Both China Railways and KICT have proposed new data definitions for earthwork, including cut and fill volumes, which are to be integrated and leveraged.

As each of these efforts have occurred independently and have been driven by schedules requiring results sooner than could be achieved through international standardization, integration of these efforts has not happened at a detailed level until now. While each organization has been aware of concurrent efforts, and has coordinated overall scope to avoid general overlap, inevitably some of the details have overlapping definitions which need to be harmonized, as all of these efforts detail with bridge construction in some form. The consensus from these meetings was that the overall modeling approaches were very similar and primarily differed in naming conventions and classification depths.

As of March 2016, key stakeholders from IFC-extension projects in China, France, Korea, and U.S. have started the process of integrating bridge information modeling efforts and have put forth a joint proposal to be presented at the buildingSmart International meeting in Rotterdam, Netherlands in April 2016.

The overall impact of this integration to the FHWA specification is anticipated to include several new data definitions such as IfcPier, IfcAbutment, etc. to be used rather than the fallback IfcCivilElement. It is also expected that the proposed IfcRelPositions data type may undergo some additional refinement. However, it is expected that the overall specification will remain intact, and the sample IFC bridge files developed will require minimal modification.

2.4 Summary and conclusions

Examples from various countries and from international associations and standardization groups show, that the existing BIM data standard IFC4 (ISO 16739) is seen as a foundation for extensions to also cover infrastructure works.

Over the recent years, the initial limitations of IFC in regard to the schema, but also to software implementation and quality of the exchanges, have been overcome, and a rigorous certification program is now in place. The main organization to support IFC development, buildingSMART International, has been a stable organization for several decades, and in recent years has experienced a surge in growth, now with full-time staff, and positioned as a strong international leader for BIM standards.

The IFC standard is accepted by the International Standardization Organization (ISO), and there are plans to accept it for all European Member States through CEN in 2015/16.

The first infrastructure extension of IFC4, the IFC-Alignment, has been completed in time after a one year development project with an international project team, steering group and expert panel. Integration efforts for bridges, railways, roadways, and tunnels are now also underway, with software providers actively involved with development.

3 Analysis of Existing Schemas

This section provides overviews of existing schemas that may be leveraged for bridge modeling. Summaries of the existing schemas are provided herein; details on specific data structures are provided in the Model View Definition documentation targeted towards software developers, published separately and available on request from FHWA or NIBS.

The primary audience of this section is software developers, however information is presented in tables and graphs (as opposed to programming languages) in the hope that domain practitioners may more easily interpret the information.

A data schema refers to the composition of data structures, and is not to be confused with data formats which describe how data structures are presented in files or messages transmitted over the Internet. Examples of data schemas are IFC, OpenBrim, and LandXML. Examples of data formats are XML, SPF, HDF5, JSON, and ASN.1. Any data schema may be stored in any data format, though it is common for certain data schemas to be associated with specific formats. A data standard encompasses a data schema in one or more defined formats and may be evaluated according to many other criteria in addition to the technical composition of underlying schemas.

This section compares and contrasts schemas by focusing on function (not form), independent of the originating specification format (UML, XSD, Express, etc.). To encourage the widest participation, schemas are presented in a consistent form readable to domain experts (cross-referenced tables with diagrams), whereas the resulting specification will also include formats familiar to programmers of various backgrounds (XML/XSD, IFC/EXPRESS, C#, Java, etc.).

The detailed schema analysis and the resulting technical specification for software developers (the IFC Bridge Design-to-Construction Information Exchange (U.S.)) is published at the U.S. National Institute of Building Sciences (U.S. Federal Highway Administration, 2016).

To achieve comprehensive review of the subject matter, it is critical to base such analysis on real projects, including the same level of detail published on such projects. The analysis within this volume was based on the efforts of modeling two bridge models in detail found on design plans, as described in *Volume III*.

3.1 Open Bridge Information Model (OpenBrIM 2.0)

The Open Bridge Information Model (OpenBrIM) was developed in cooperation between the Federal Highway Administration and Red Equation Corporation.

The OpenBrIM 2.0 schema (Chen, 2013) models components of bridges using several generic data structures. Since this schema was initially defined in the referenced report, there have been multiple iterations, where documentation has been evolving, and the classes, attributes, syntax, and usage has substantially changed each time presented (Red Equation Corporation, 2016). However, the general architecture and functionality appear to be remain similar as of August 2015.

3.1.1 Data Types

The fundamental data type of OpenBrIM is called "Obj" and contains the following attributes:

Table 1: OpenBrIM Obj data type

Attribute	Type	Description
Name	String	User-readable name for identifying the object
ID	String	Machine-readable persistent identifier for an object
Type	String	Defines the type of object according to well-known identifier
RelObj	String	Refers to a template defining inherited values
Alignment	String	Refers to the alignment curve object for which the object is placed
X	String	Position X as a floating-point length or formula
Y	String	Position Y as a floating-point length or formula
Z	String	Position Z as a floating-point length or formula
RX	String	Rotation X as a floating-point ratio or formula
RY	String	Rotation Y as a floating-point ratio or formula
RZ	String	Rotation Z as a floating-point ratio or formula
AboutX	String	Axis X as a floating-point ratio or formula
AboutY	String	Axis Y as a floating-point ratio or formula
AboutZ	String	Axis Z as a floating-point ratio or formula
Units	Obj []	Optional units indicating measure of values at object and within
Parameter	Parameter []	Optional parameters indicating semantics according to the Type
Obj	Obj []	Optional decomposition of objects
Repeat	Repeat	Describes repetitive patterns of placement of the object
Surface	Surface	Representation of 3D geometric surface
Circle	Circle	Primitive for round surfaces
Line	Line	Primitive for straight segments
Volume	Volume	Representation of 3D closed volume

Within the OpenBrIM schema, this generic structure captures all rooted data. Specific meaning of objects depends on understanding particular object types of reserved identifiers, which are not defined within the schema – the data schema itself (XSD file) contributes a minimal role in validation of such file, while specific custom validation would be required to enforce correct usage of attributes and string encodings. For example, the flange width of a beam would be captured as a Parameter, and the 3D volume would consist of data structures that reference this parameter. The naming of such parameters also needs to be standardized for software applications to extract such information consistently, for which a standard set of objects is defined within the OpenBrIM specification.

Parameters are defined using the "Parameter" data type which contains the following attributes:

Table 2: OpenBrIM Parameter data type

Attribute	Type	Description
Name	String	Machine-readable name for identifying the attribute
Label	String	Human-readable label for displaying the attribute
Type	Parameter_Type	Type of the parameter, indicating data type and measure
Desc	String	Human-readable description for the attribute
Value	String	Literal value or formula expression

Repetition intervals are defined using the "Repeat" data type which contains the following attributes:

Table 3: OpenBrIM Repeat data type

Attribute	Type	Description
Param	String	Machine-readable identifier of the parameter to be repeated
StartValue	String	Literal value at starting occurrence
EndValue	String	Literal value at ending occurrence
Increment	String	Literal value to increment
Values	String	Sequence of values to be used at intermediate intervals
Obj	Obj	An inner object to be repeated at each interval
Repeat	Repeat	Nested repetition structure such as for multiple dimensions

Geometric volumes are defined using the "Volume" data types (subtype of "Obj"), which is specialized to contain "Surface" elements. Surface elements contain an additional "Point" attribute consisting of a list of Cartesian points indicating the boundaries of the surface.

Higher-level geometric constructs (e.g. extruded profiles) are composed by defining parameters, repetition intervals, and formulas referencing such parameters.

Note: Since the initial analysis, OpenBrIM 3.0 has been under development. As the technical specifications were not available at the time this report was prepared (i.e. no XSD file or supporting documentation was available), the schema analysis and preparation of data for the *sample bridge was specific to OpenBrIM 2.0. However, notes have been added to indicate differences that have been observed based on presentations for OpenBrIM 3.0.*

3.2 Land Topography (LandXML)

LandXML (LandXML.org Industry Consortium, 2015) is a data model that describes terrain, road alignments, pipe networks, and other information of interest to land surveying and development. LandXML is widely adopted across civil design software platforms; according to landxml.org, Version 1.1 (2006) is supported by 13 registered applications (many with multiple versions). Initially sponsored by Autodesk, this data model was driven to support the needs of various U.S. Departments of Transportation. As of 2014, its principal sponsor is Carlson Software (carlsonsw.com). A working draft for LandXML 2.0 has been published as of 2014. According to landxml.org, Nathan Crews has been the primary contributor since its formation and has indicated intent to build an official organization behind the standard to further support the initiative. Meanwhile, other industry organizations including OGC and BuildingSmart International have taken initiatives to document LandXML.

Documentation posted at landxml.org includes an XML Schema Definition (XSD) file capturing the schema, and a Software Development Kit (SDK) based on C++ for Windows. The XSD file provides descriptions for many entities, but does not indicate the meaning of specific attributes – in many cases such meaning and usage can be assumed according to attribute names and data types, while other cases leave room for interpretation or may require inspection of example files to understand how they are used in practice.

3.2.1 Data Types

All LandXML files contain a single "LandXml" instance which defines default units, spatial location, and collections of elements representing various types of information: Alignments, CgPoints, Amendment, GradeModel, Monuments, Parcels, PlanFeatures, PipeNetworks, Roadways, Surfaces, and Survey. All semantic objects have a name and description. Objects may also have extensible name/value pairs organized into "features" which may be defined by derivative specifications, software vendors, or end-users (if made possible by the authoring application).

- Alignments capture horizontal alignment records, which may be referenced by other data structures.
- Grade Models indicate land terrain geometry.
- Amendments indicate title changes that affect naming of elements.
- Parcels indicate regions of land defined by legal boundaries, and capture any restrictions such as easements.
- Plan Features indicate miscellaneous physical structures for which detailed information is out of scope, but referencing the item may be of use to provide additional context, such as building footprints, guard rails, light posts, and signage.
- Pipe Networks indicate open and closed channels for supply (e.g. water) or drainage (e.g. wastewater), that are positioned according to alignments.
- Roadways indicate roads, bridges, and other structures for carrying traffic. The detail captures design information with regards to lanes, speeds, road classifications, traffic

volume, etc. With regards to bridges, the only parameter captured is "width"; no information is captured for the physical design of a bridge, structural parameters, or maintenance information – all are out of scope.

3.3 National Bridge Inventory (NBI)

The National Bridge Inventory (NBI) is both an inventory and current condition record of bridges in the USA. It is organized as a flat set of records for a particular bridge describing bridge owner authority, location, functional classification of routes, bridge history, design criteria, the bridge structure in terms of approach spans, main spans, and a set of predefined bridge components used in assessment. The components include: Deck, Superstructure, Substructure, Channel and channel protection, Culverts.

The NBI also defines a binary format for encoding such records within a file.

These data collection classifications identify future integration requirements with this project.

3.4 Industry Foundation Classes (IFC)

Industry Foundation Classes is a schema that describes details of buildings throughout their lifecycle of design, construction, and maintenance. Initially developed by Autodesk in 1994, an independent organization was established to promote and further this standard, initially called International Alliance for Interoperability, then later renamed to BuildingSmart. IFC is the most widely implemented standard for exchanging building information between leading CAD/BIM software platforms, supported by approximately 150 registered software applications (BuildingSmart International, 2016). IFC is also registered as an international standard - ISO 16739 (ISO, 2013).

IFC has evolved from several iterations since its inception. Designed to leverage the existing software base of manufactured product models (which also find their way into building models), IFC was initially based on STEP (STandard for Exchange of Product information). Like many standards initiatives, there was significant change between the first several versions until it became widely implemented and issues were discovered and resolved. Later versions (since IFC2x in 2002) have maintained substantial compatibility with features added incrementally.

At a high level, IFC organizes all information about buildings into six abstract object types: actors, controls, groups, products, processes, and resources. For purposes of modeling physical building elements, products are used. An IFC data exchange always includes a single project instance which defines default settings for information contained such as units and coordinate systems. This project instance contains a spatial hierarchy of other objects, such as sites containing buildings, elaborated into multiple levels containing rooms. Each of these logical spatial structures may contain physical elements of several hundred classified types (e.g. slab, wall, beam, pipe segment, electrical outlet). Such physical elements may also be decomposed into parts (e.g. reinforcing bars).

All semantic objects in IFC have a 128-bit globally unique ID, object ownership and merge state, name, and description. Objects may also have extensible name/value pairs organized into "property sets" which may be defined in the IFC specification itself or by derivative specifications (called "model view definitions"), software vendors, or end-users (if made possible by the authoring application).

The shape representations in IFC are designed to match the method for creating shapes in design tools, allowing most shapes to be represented consistent with how they were modeled. IFC supports various geometric models derived from the STandard for Exchange of Product (STEP) information pervasive in manufactured product models. Geometry structures include Constructive Solid Geometry (CSG) models, swept solids, boundary representations (B-Rep), surface models, and tessellated models. Four basic means of representing geometry can be further described as follows.

1) A Swept Solid is sweep with a wide range of profiles and sweep paths (e.g. directrix). Profile types include a set of parametric profiles or explicit profiles whose profile

boundary may be a polyline connecting points or a composite curve made up of line segments on a plane or the boundary of a derived 2D shape.

2) Sweep path can be a composite curve, straight line, or more complex curve, including non-uniform B-spline. In IFC, the profile is usually normal along the sweep, but explicit control of profile rotation, skew, and profile transformation along the sweep is also supported.
3) Constructive Solid Geometry (CSG) involves primitives (box, cylinder, cone, pyramid, or sphere) and other solid geometry combined within operations of union, intersection, or subtraction to define functions that generate new solids. The result of the Boolean operations is a boundary representation (or B-Rep), which is also an operand of CSG.
4) Boundary Representation (BRep) is a set of connected faces that satisfy the set of well-formed rules that define a solid (e.g. Euler equation, Mobius Law, consistent facial orientation).

While IFC supports 3D geometry of various forms, it also captures information for engineering and construction. Any physical element may have materials identified with properties such as structural behavior (e.g. elastic modulus), thermal behavior (e.g. heat transmittance), and others. Any physical element may also have relationships with other physical elements such as embedding (e.g. rebar), voiding (e.g. openings), filling (e.g. doors), covering (e.g. fire retardant), connecting (e.g. bolts), assemblies, and others. Physical building models may also have derivative models assigned to indicate existing or future conditions, or theoretical idealized models such as for structural analysis.

3.4.1 Encodings

IFC data may be encoded in multiple formats, each capable of carrying the same information:

Table 4: IFC encoding formats

Format	Standard	Description
SPF (.ifc)	ISO 10303-21	Text format with flat list of objects, sequential attributes
HDF (.ifchdf)	ISO 10303-26	Binary indexed format with hierarchy of data tables (HDF5)
XML (.ifcxml)	ISO 10303-28	Text hierarchical markup format with named attributes
ZIP (.ifczip)	(IFC-specific)	Zip compressed file containing one of the above formats

In practice, the SPF format (STEP Physical File) is the most widely used of the above formats. SPF provides a balance of readability and compact encoding, typically 10-30% the size of XML. Readability is important for efficiently resolving issues between contracting parties who may be using different software, in the absence of software tools for performing such comparison. Size is important as a matter of time and cost. For buildings, SPF files are commonly in the 200MB range, whereas equivalent XML files would be in the 1 GB range. Similar to other data models used for large quantities and/or high frequency of engineering information (e.g. OPC, BACnet, SNMP), XML primarily serves for documentation and debugging; economics dictate more efficient encodings in practice due to time, bandwidth, or cost considerations particularly for use at remote construction sites with limited Internet access. With ZIP compression, both formats are reduced to similar size (10:1 for XML, 5:1 for SPF); the tradeoff of such compression is additional processing time and memory required of software applications.

HDF, while documented as an ISO standard, has not yet been promoted as an IFC format; it may be of interest for modeling large infrastructure networks, as the indexed structure facilitates partial loading, supporting essentially unlimited amounts of data in a scalable manner. Ultimately, IFC-HDF is a standardized object-relational database format, though may be accessed sequentially, such as by specifying byte ranges over HTTP. Other encodings may also accomplish the same, such as SQLite.

3.4.2 Data Types

The fundamental data type for physical objects within the IFC schema is called “IfcElement” and contains the following attributes:

Table 5: IFC element data type

Attribute	Type	Description
GlobalId	IfcGloballyUniqueId (String)	Machine-readable persistent identifier for an object
OwnerHistory	IfcOwnerHistory	Access control and merge behavior
Name	IfcLabel (String)	Human-readable name for identifying an object
Description	IfcText (String)	Human-readable description
HasAssociations	IfcRelAssociates[]	External or parametric information (documents, classifications, libraries, constraints, approvals, material and shape parameters)
HasAssignments	IfcRelAssigns[]	Links to derivative data for which this object serves as input (e.g. tasks, structural models)
IsNestedBy	IfcRelNests[]	Externally configurable components (e.g. ports)
IsAggregatedBy	IfcRelAggregates[]	Internally embedded components (e.g. rebar)
ObjectType	IfcIdentifier (String)	Type identifier for non-standard objects
IsTypedBy	IfcRelDefinesByType[]	Type definition of this object
IsDefinedBy	IfcRelDefinesByProperties[]	Extensible parameters for object
ObjectPlacement	IfcObjectPlacement	3D position and orientation
Representation	IfcProductRepresentation	Geometry of various parametric levels (e.g. Axis, FootPrint, Profile, Box, Surface, Body, Lighting,)
ConnectedTo	IfcRelConnects[]	Links to other objects anchored to this object
IsInterferedBy	IfcRelInterferes[]	Links to other objects intersecting this object
HasProjections	IfcRelProjectsElement[]	Projections placed along edges of this object
HasOpenings	IfcRelVoidsElement[]	Openings placed within volumes of this object
HasCoverings	IfcRelCoversBldgElements[]	Coverings placed on surfaces of this object

The IfcElement data structure described above is an abstract entity, for which many subtypes are defined for particular product classifications, (e.g. IfcBeam, IfcSlab, IfcWall). This data structure also inherits from other abstract entities (IfcProduct, IfcObject, IfcObjectDefinition, IfcRoot).

Of note, when stored in files, such attributes may be represented differently: for example, the SPF format serializes object references in a specific direction (direct attributes and not inverse attributes), while the XML format serializes attributes as hierarchies and linked references according to configured mappings.

The IFC model may be derived, extended, and constrained in downstream specifications, which are referred to as “Model Views” by the publishing organization (BuildingSmart International). Model views are designed for a specific domain and define one or more data exchanges. Model Views may be described using a computer-interpretable format called “mvdXML”. Such format may be used to automatically generate derivative schema formats or programming language encodings (e.g. XSD, EXP, Java, C#, C++) with rule checking intact.

3.4.3 Model View Definitions

3.4.3.1 Precast Concrete National BIM Standard

The Precast Concrete National BIM Standard (PCNBS) is an IFC Model View initiated in 2008 to cover the significant exchanges centered on the precast fabricator, that deal with design, engineering, production and erection. It was the first effort to address all the major exchanges needed to facilitate data workflows for a building domain. The project was sponsored by the Charles Pankow Foundation with initially funding from the Precast/Prestressed Concrete Institute. At this time, the PCI has taken over support. The process followed was that described in the National BIM Standard, Version One (2008). A working committee of PCI (BIM Committee) oversaw and advised the initiative.

The functional scope of the (PCNBS) was buildings, addressing both structural and architectural precast. It is a sub-schema of IFC, Release 2.3. It includes the standard building elements, such as beams, columns, slabs, spandrels, stairs, ramps, footings, piles, and roof, as well as generic accessory elements. Precast has many components that are subjoint to the precast piece they are part of: discrete accessories, components, parts, and reinforcing elements. Reinforcing elements include regular mesh, rebar with ACI bending patterns, and prestressed tendons. Building Elements can be represented as cambered, battered and twisted. Finishes can cover the whole element or a partial surface with the geometrical bounding shape representation. Shape representation for building element modeling is based on the IFC geometric model provisions.

Temporary supports, such as for temporary structural support for precast, are not included in the scope. The schema does not explicitly represent 4D construction simulation, though such capability is supported by the IFC base schema. It incorporates minimal site information, with limited utilities or details for external systems.

The initial development goals were broad and included 51 exchanges. Many were similar in that they called for slight differences in model content. Based on a similarity analysis the range of exchanges was reduced to eleven exchanges. These still cover a broad range of the precast lifecycle.

The PCNBS, by nature of mvdXML, makes use of the BuildingSmart International “IfcDoc” validation and documentation tool. A version of IfcDoc is pre-loaded with a set of modularly loaded rules. When an IFC instance file is loaded into IfcDoc, it tests the instance file and identifies code segments that violate the loaded rules. The validation tools include test models and error handling guidelines.

The PCNBS is documented and is beginning to be implemented and validated. It has not yet been field tested.

3.4.3.2 Reinforced Concrete BIM Model

The Reinforced Concrete BIM Model was initiated and funded by the Charles Pankow Foundation with Chuck Eastman of GA Tech as a technical advisor. Its focus is reinforced concrete detailing and production. The Reinforced Concrete BIM Standard Committee (ACI-

131) chose to focus on RC production and developed an IFC Model View, working with the following Omniclass phases: Design Development, Construction Documentation, Concrete Resource and Placement Planning, Concrete Execution, and Erection Phase and Turnover

The potential actors considered in the exchange process include the Architect, Structural Engineer, Civil Engineer, Mechanical Engineer, Concrete Contractor, Batch Plant Reinforcing Fabricator, Testing Agency, Reinforcing Detailer, Reinforcing Contractor, Formwork Contractor, Finish Contractor, Concrete Contractor, General Contractor, Site Contractor, Rebar & Tendon Distributor, and Owner Client. These roles were assigned to 38 distinct tasks, with 24 exchanges between them.

Reinforced concrete is complex with many production operations, dealing with different elements: reinforcing, formwork and scaffolding, finishes, and the dynamic volumes that address overlapping structural elements. These elements are merged and broken up using pour breaks, to which reinforcing, structural connections, embeds and pass-thrus are provided as required to reflect the pours (as planned and as realized).

Reinforcing objects are often dense with thousands of reinforcing pieces, and may consume significant graphic processing resources. Conventional practice with CAD drawings uses patterns of layout that are defined, and applied repeatedly according to schedules. If they are laid out as repeating patterns, then the patterns can be defined and called in the same manner, lightening the model. There are multiple levels of reinforcing modeling:

1. Individual reinforcing piece – defined with lightweight geometry with a centerline composite curve (connected lines and arcs), and a diameter plus its various attributes. This is easily mapped to the automated bending schedules. Reinforcing attributes include: steel grade, nominal diameter, section area, surface type, and coatings.
2. Reinforcing aggregations – defined as a master shape and a list of placements of the master shape; one reinforcing assembly can be the master for a larger one, resulting in arrays of arrays. This level uses multiple hierarchical examples of Reinforcing Type to optionally represent a logical tree of reinforcing assemblies.
3. Reinforcing arrays – This level uses a master rebar object, material and geometry, a vector offset and the number of rebar copies to be laid out according to the vector.
4. Top Reinforcing Assembly – This entity takes the top level aggregation with possibly multiple levels of sets of reinforcing, with the overall groups of rebar, with bending schedule, materials, coating, etc. quantities and counts. It assigns this structure to the concrete element that carries this assembly. If the assembly is part of more than one pour, it is treated like a single reinforcing element.
5. In some cases, assembles and types are not sufficient for very large concrete structures and further compacting is required. The option exists to define a master shape and array of like precast pieces, defined as a single object, but with an attribute indicating the count. For example, seven rebar geometries might be represented in a single shape, with a count of seven.

The originally defined 24 exchanges were considered too extensive to implement initially and the exchanges were filtered to initially implement three. The three exchanges are EM6: Structural design model; EM15: Reinforcement placement sequence; and EM20: Construction reference schedule.

3.4.3.2.1 Structural Design Model

EM6: Structural design model contains a report of detail structural design that can be used by receivers to determine steel reinforcing sections, lap standard details, and special connections. It can optionally provide information for early mill order for reinforcing and early shoring needs.

The detail structural design carries information on cross section layouts requirement, spacing and lap requirements, standard details, lap lengths, special connections, major embeds and cutouts, concrete strength, steel reinforcing specs, tendon specs, expansion joints, PT tendon placement joints, geophysical data, and foundation spatial requirements.

EM6 uses parametric definition of structurally derived longitudinal and transverse structural requirements. These requirements are passed to Reinforcing Detailer for 3-D layout and spatial detailing and resolution; includes lap joints and coatings; also used in later structural reviews to depict design intent. For detailing, EM6 defines post-tension tendon layout patterns, and tendon placement joints, and specifies foundation spatial requirements and geophysical data for shoring needs.

3.4.3.2.2 Reinforcement Placement Sequence

EM15: Reinforcement Placement Sequence is to coordinate reinforcement and tendon placement with placement sequence and schedule. Its data type includes complete physical detail of all reinforcement and tendon items, embeds, and formwork, including includes formwork for special finishes, block outs, and insulation. Formwork can be associated with schedule and placement. The reinforcement placement sequence exchange also includes a concrete detailed model, with the definition of placement schedule related to pour sequence.

EM15 uses **B-Rep** geometric representation for all reinforced concrete elements including control joints and other features, 3D physical layout of reinforcement bar, mesh, PT tendons, anchors, and other embeds for construction coordination and scheduling. Sequencing of reinforcement assemblies is represented in association with each pour. Formwork designation, as needed for scheduling of embed placement sequencing, is also included.

3.4.3.2.3 Construction Reference Schedule

EM20: Construction reference schedule coordinates the layout of all systems for clashes and coordinates the schedule of installation, especially with formwork and finishing tasks. Optionally a 4D configurator can also be also used to verify coordination with mechanical systems and architectural intent.

EM20 represents all major systems: structure, MEP, architectural detailing interfacing clash detection and coordination as well as concrete placement & discrepancy report. EM20 integrates all of the CIP concrete elements in a project with their associated pours. Component sequences of placements are identified in a previous application and noted for the elements making up each pour, and are integrated with the overall project schedule.

3.4.3.3 BIMsteel

BIMsteel of the American Institute of Steel Construction (AISC) aims to develop new processes and improved methods of data transfer and sharing, which include AISC Interoperability Strategy, AISC Steel Information Delivery Manual (IDM), Automating Steel Fabrication, Shop Model Review, Design to Detailing, steelXML, and Joist and Deck Specification. The AISC Steel IDM serves as basis for two IFC Model Views: Automating Steel Fabrication and Design to Detailing.

AISC effort was originally motivated to replace and broaden the CIMSteel Integration Standard (CIS), developed through a European Community project Eureka project EU130, part of the ESPIRIT program. CIS engaged European, Japanese and US steel industry interests. CIS was very thorough and largely complete for steel design. It was, like IFC, a derivative of the ISO-STEP product model technology, using the starting base of ISO-10303 Integrated Resources and the EXPRESS language. CIS ended in 1998, with input and revisions from the involvement of the US steel industry. After review, a new version of CIS, called CIS/2 was adopted by the AISC in 2001 and promoted as a data exchange format for steel building fabrication. It was adopted widely within the steel industry and used broadly.

But with use, it was found that CIS/2 was seriously limited by its lack of modeling of non-steel elements and lack of integration with data models for these other non-steel systems. In considering integration of the two standards, CIS/2's internal structure and schema logic was very different from IFC. The steel design, engineering, fabrication and erection had no need for extension and did not contain abstractions to facilitate extension into other domains. CIS/2 continues to be used. In 2010, AISC undertook to explore the transition to IFC, utilizing the lessons learned from CIS/2. AISC Steel Information Delivery Manual (IDM), Automating Steel Fabrication, and Design to Detailing are three main initiatives toward IFC transition.

AISC Steel Information Delivery Manual (IDM) includes a standard process model and exchange requirements for structural steel and later used for developing Automating Steel Fabrication exchange and Design to Detail exchange.

Steel shapes are largely rolled from extrusions or welded together from individual plates, which can be defined using extrusion geometry with profiles. Automating Steel Fabrication exchange focuses on CNC fabrication in order to enhance productivity of cellular robotic fabrication and assembly, especially the representations that support robotic cutting and welding of structural steel assemblies.

The geometry representation in the Automating Steel Fabrication exchange relies on parameterized profiles according to the AISC shape database and other shape databases. Features associated with the pieces and assemblies also use parametric values that can be used in downstream software. Connection details including welding and bolt assembly also use parameterized values. Geometric representation is limited to 3D curve representation for welding path or bolt direction without explicit 3D representation.

This approach helps to reduce data size while enhancing accuracy of data exchange. The model addresses surface treatment with or without geometric representation of that treatment, especially for partial surface treatment.

FIATECH undertook the effort to define Design to Detailing exchange. This exchange is a detailed structural steel model to help the steel detailer to design final steel structure layout, to help the steel manufacturer provide the detailed material take-off, and also to help the contractor develop the bid documents. While this model does not provide the same level of detail as provided in the Automating Steel Fabrication exchange, it does address the structural analysis model including structural performance requirements, structural connections, structural members, loads, and reactions.

3.5 Bentley OpenBridge iModel

Bentley® Systems, Inc. has announced their intention of publishing a standard for bridge information modeling in 2016 as part of their "i-model" initiative (Bentley Systems, 2015).

As this schema has not yet been published, there is nothing that can be incorporated within this schema review, however, the Bentley® LEAP Bridge Steel software application is referenced in *Volume III* of this report, as it is assumed that the data modeled by this application or its successors would correspond to the "iModel" schema.

4 Gap Analysis

This section describes concepts used and capabilities provided in candidate information models, and compares approaches across various schemas where applicable. The goal of such a comparison is to understand how other applications may construct or consume data, and to consider alternatives and document the rationale for using a particular approach.

This section discusses concepts at various levels of specificity – concepts are listed in order from those that are generic and would apply in many scenarios (e.g. object identification) to those that are more specific that address targeted scenarios (e.g. test borings).

4.1 Identification

Object identification refers to the capability of identifying an object persistently, regardless of any future changes made to the object. This implies that once such an identifier is assigned to particular information, that identifier must never change; otherwise any references to such information would no longer refer to the same target information. For information models comprised of content that is authored separately and later combined, global uniqueness may be achieved by generating such identifier that is unique in time and space (e.g. associated with the MAC address of a network interface which is issued by a universal registry), or statistically unique if the identifier is of high-enough resolution (e.g. 128-bits).

4.1.1 OpenBrIM

Identification is indicated according to a generic string type, which could map to any arbitrary-sized identifier. For interoperability with software, the usage must be made specific to indicate the size and format of the identifier.

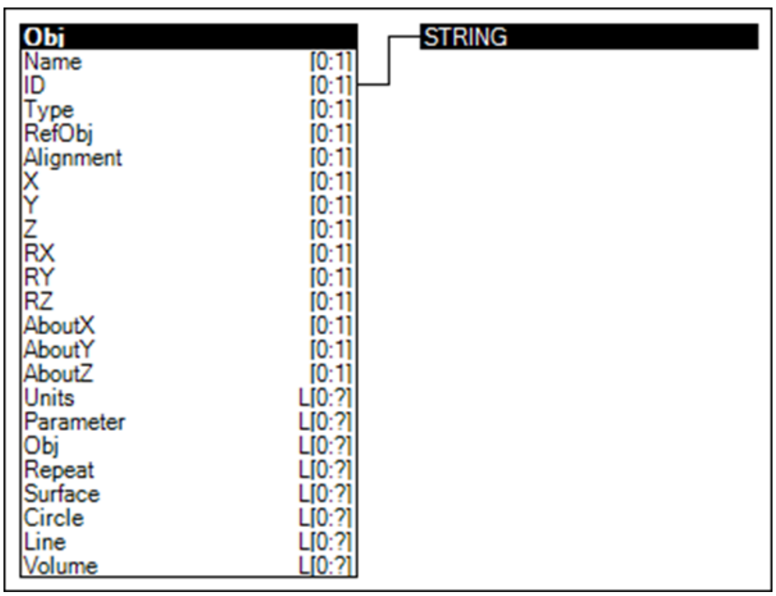


Figure 44: Identification in OpenBrIM

4.1.2 IFC

IFC uses a 128-bit GUID for unique identification, which uses a custom string encoding. Various file formats may use additional identifications of instances for serialization purposes; however there is no requirement or guarantee for such identifications to remain the same between revisions or across applications. For example, the IFC-SPF file format lists each instance with a 64-bit integer that is unique within the particular file. The IFC-XML format requires identifiers only for objects that are referenced within the same file, where such identifiers may use any format.

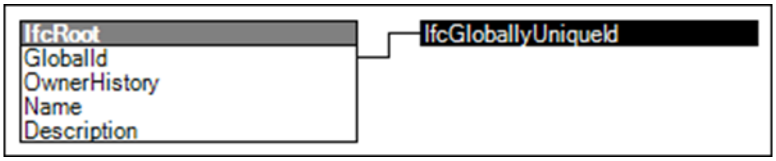


Figure 45: Identification in IFC

4.1.3 LandXML

While LandXML does not make use of inherited definitions where a single attribute may be defined for object identity, most objects contain an attribute called "oID" which serves as the identifier. The specific format and resolution of this identifier is undefined.

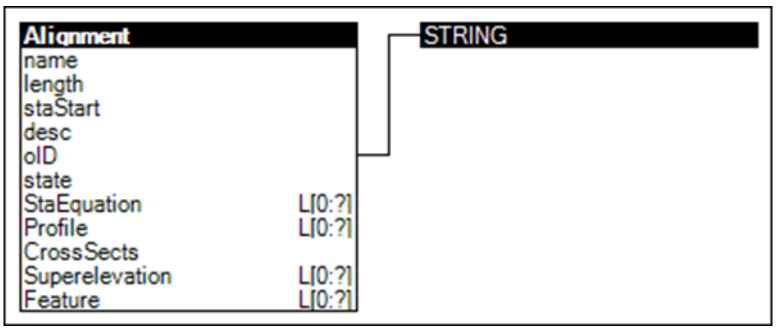


Figure 46: Identification in LandXML

4.2 Naming

Object naming refers to the capability of humans to name objects that may be later retrieved by understanding the meaning. If such identifiers are renamed, then any external usages of these identifiers must also be changed.

4.2.1 OpenBrIM

The *Name* attribute is a string of Unicode characters with unbounded length.
The *Desc* attribute provides further context in identifying or locating the object.

Note: OpenBrIM 3.0 abbreviates "N" for name and "D" for description.

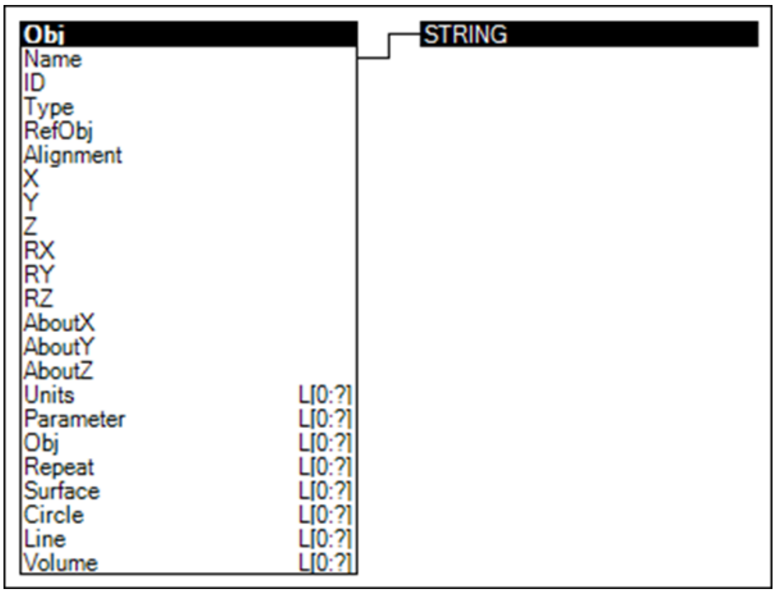


Figure 47: Naming in OpenBrIM

4.2.2 IFC

The *Name* attribute is a string of Unicode characters with maximum length of 255.
The *Description* attribute provides further context in identifying or locating the object.Specific subtypes introduce additional attributes for naming:

- Spatial objects may be further identified via the *LongName* attribute. This value should generally correspond to building signage describing floor levels or rooms. While the *Name* attribute generally provides a coded or abbreviated identifier, the *LongName* provides a functional name for the location such as "Reception Area".
- Physical elements may be further identified via the *Tag* attribute. This is a human readable identifier such as an element or item number. While there is no restriction on usage of such tags, it is recommended that the *Tag* be unique within its containing scope.

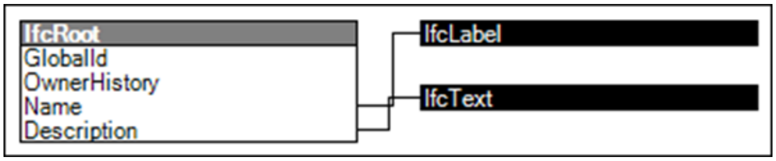


Figure 48: Naming in IFC

4.2.3 LandXML

The **name** attribute is a string of Unicode characters with unbounded length.
The **desc** attribute provides further context in identifying or locating the object.

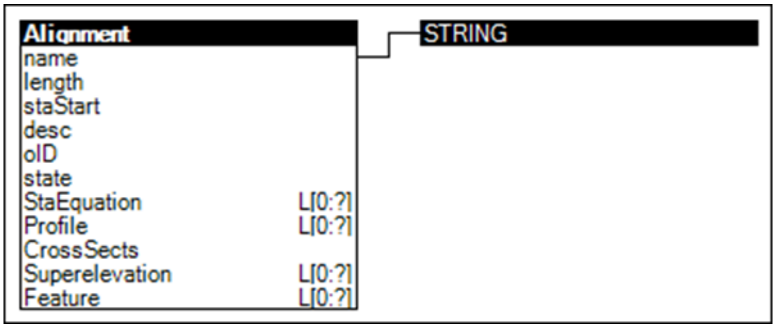


Figure 49: Naming in LandXML

4.3 Object Metadata

For data that is used in a contractual context and multiple parties are responsible for defining the data, a mechanism for indicating the person and organization responsible for each object may be required. For cases where a file has one or more electronic “engineering stamps” in the form of digital signatures, there needs to be a way to relate the signing party with object(s) they are certifying, and distinguishing from other data out of their scope.

For scenarios of plan revisions or change orders, there may also be a need to indicate creation, modification, or deletion of objects. While such information may be deduced by comparing two files containing the same scope, there may be other scenarios where only subsets of data are provided, which require distinction between an object that has been deleted, and an object that isn’t included in a particular exchange.

4.3.1 OpenBrIM

No such concept has been defined.

4.3.2 IFC

Objects may be marked according to the individual and company responsible. If a file is digitally signed indicating engineers stamp of approval, the identification of the person and organization may correspond to that on the referenced certificate.

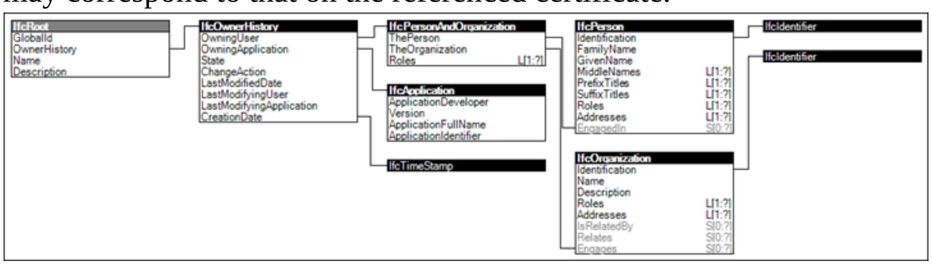


Figure 50: Object metadata in IFC

4.3.3 LandXML

No such concept has been defined.

4.4 Object Types

For objects that occur in the same configuration or in parameterized configurations multiple times, it is necessary to model an object type separately from the object occurrence.

The most obvious use of object typing is for manufactured product models. However for bridge design in particular, there are often bridge types requiring standard compositions of components subject to parameters that may vary at the occurrence, such as spans and slopes.

4.4.1 OpenBrIM

Types are indicated by well-known strings, where the meaning of such strings is determined by an external dictionary. A particular type name may imply a required set of parameters to be included on the object.

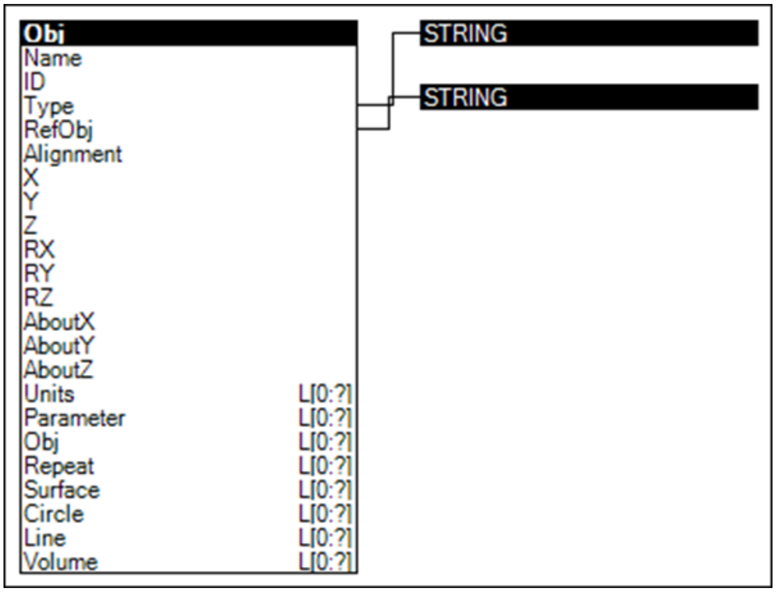


Figure 51: Object typing in OpenBrIM

4.4.2 IFC

Object occurrences can be defined by a particular object type. A pair of entities is defined for most semantic objects - an object occurrence entity and a corresponding object type entity. For example, the *IfcBeam* is the object occurrence entity that has a corresponding *IfcBeamType* as the object type entity.

On instance level, an object occurrence instance may have:

- similar state as its object type instance by applying all characteristics defined at the type;
- overridden state for particular characteristics;
- no defined object type instance.

Characteristics defined at the object type level may include:

- common naming and predefined type;
- common properties within a type driven property set;

- common geometry representations, applied as mapped representation to each occurrences;
- common material assignments (with exception of material set usages);
- common definition of a decomposition structure.

Many object occurrence and object type entities have an attribute named *PredefinedType* consisting of a specific enumeration. Such predefined type essentially provides another level of inheritance to further differentiate objects without the need for additional entities. Predefined types are not just informational; various rules apply such as applicable property sets, part composition, and distribution ports.

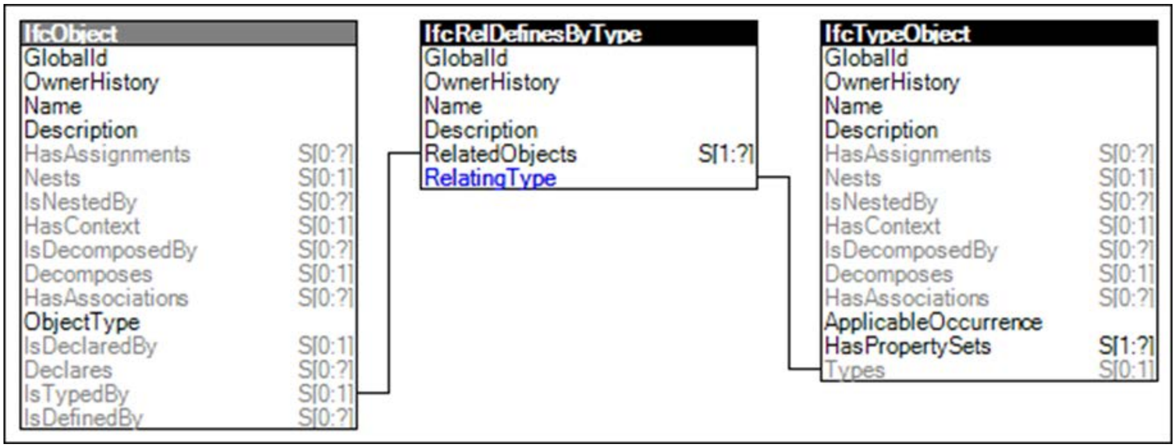


Figure 52: Object typing in IFC

4.4.3 LandXML

No such concept has been defined.

4.5 Composition

Objects may be composed into parts to indicate levels of detail, such as a building having multiple levels, a framed wall having studs, or a task having subtasks. Composition may form a hierarchy of multiple levels, where an object must have a single parent, or if a top-level object is declared within the single project or a project library.

4.5.1 OpenBrIM

Objects may be organized into parts recursively using the "Objs" element. Placement and other contexts such as units are inherited from outer objects.

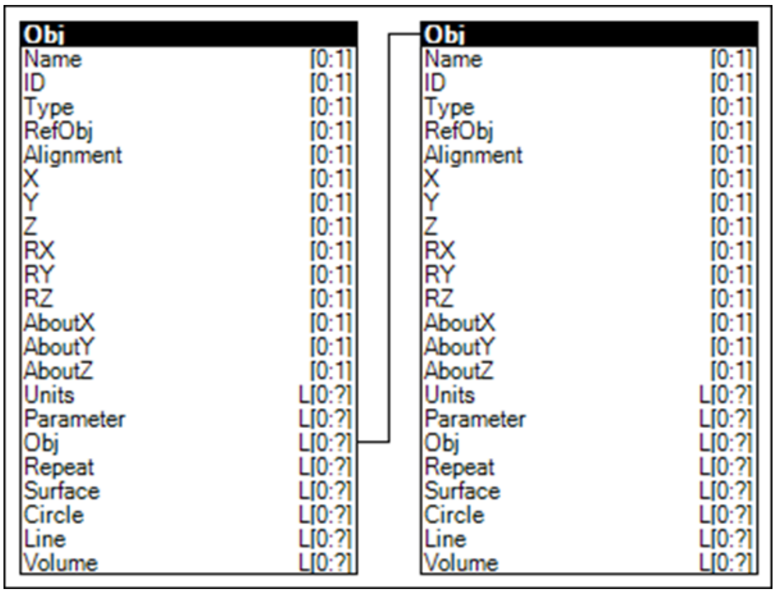


Figure 53: Composition in OpenBrIM

4.5.2 IFC

An aggregation indicates an internal unordered part composition relationship between the whole structure, referred to as the "composite", and the subordinate components, referred to as the "parts". The concept of aggregation is used in various ways. Examples are:

- Aggregation is used on building elements to indicate parts such as studs within a wall;
- Aggregation is used on spatial elements to indicate a spatial structure such as a story within a building;
- Aggregation is used on systems to indicate subsystems such as branch circuits.

Aggregation is a bi-directional relationship, the relationship from the composite to its parts is called Decomposition, and the relationship from the part to its composite is called Composition. A nesting indicates an external ordered part composition relationship between the hosting structure, referred to as the "host", and the attached components, referred to as the "hosted elements". The concept of nesting is used in various ways. Examples are:

- Nesting is used on product elements to indicate external connectable parts such as faucets mounted on a sink, or switches within a junction box.

- Nesting is used on control objects to indicate specification hierarchies.
- Nesting is used on process objects to indicate subordinate processes which may occur in parallel or in series.
- Nesting is used on resource objects to indicate subordinate resource allocations which may occur in parallel or in series.

Nesting is also a bi-directional relationship. The relationship from the hosting structure to its attached components is called Nesting, and the relationship from the components to their containing structure is called Hosting.

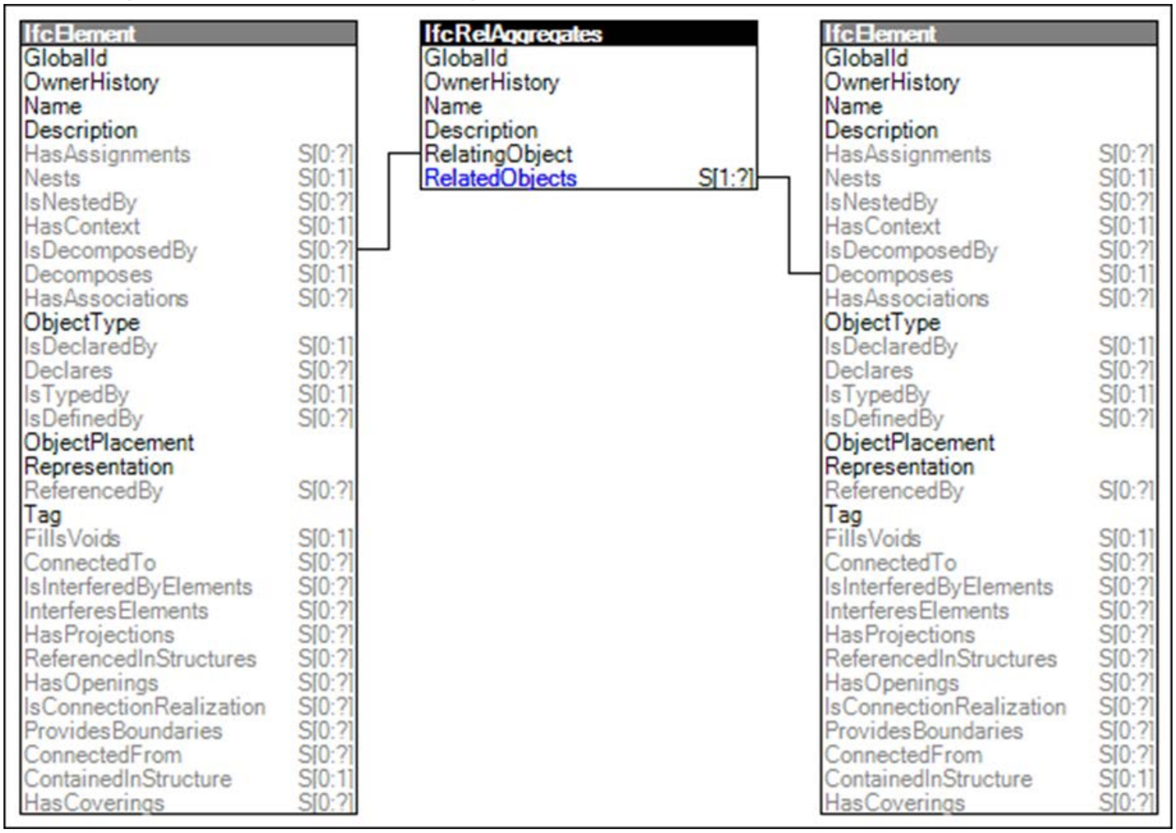


Figure 54: Composition in IFC

4.5.3 LandXML

No such concept has been defined.

4.6 Placement

Placement refers to locating physical elements in space, ultimately at specific latitude, longitude, and elevation. As physical measurement tools and existing conventions rely on specific coordinate systems, information models must also follow suit to achieve design interoperability. While a specific reference latitude, longitude, and elevation may be defined, individual elements are typically positioned at relative offsets. Such offsets may rely on a Cartesian coordinate system, and may make use of multiple matrix transformations. For roads and bridges, placement is typically defined relative to an alignment curve which consists of a vertical curve relative to a curve in the horizontal plane (at constant elevation from earth). In addition to the final placement, schemas may also define intermediate structures such as connectivity relationships that constrain how objects are placed.

4.6.1 OpenBrIM

Placement of objects is defined as position and orientation relative to another object, where the underlying values may be literal constants or formulas. If an alignment is referenced, then such position and orientation is relative to the alignment object.

Repetition intervals may be defined for any element using the Repeat structure.

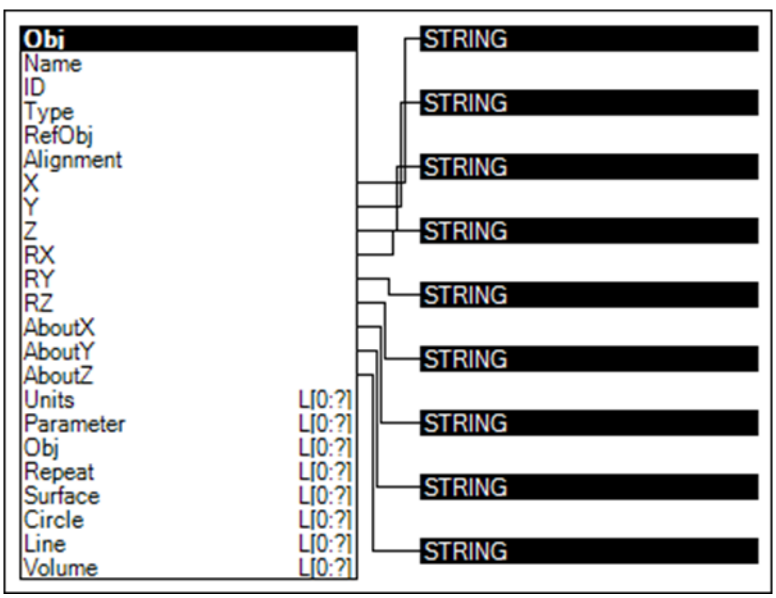


Figure 55: Placement in OpenBrIM

4.6.2 IFC

Product occurrences can be placed in 3D space relative to where they are contained.
There are two variations of placement:

- Local Placement describes a relative position and orientation relative to a parent element
- Grid Placement describes a relative offset and rotation relative to coordinates within a grid, where such grid may be rectangular, circular, or triangular

Placement is defined by a relative position (X, Y, Z coordinates), a horizontal reference direction, and a vertical axis direction. At the outermost level, relative directions are defined according to representation context; for example, +X may point east, +Y may point north, and +Z may point up.

Placement follows aggregation and containment relationships as follows:

- at the outermost level, a site is globally positioned according to latitude, longitude, and elevation;
- for spatial structures, positioning is relative to aggregation. For example, a site may aggregate multiple sub-sites,
- for aggregated parts, positioning is relative to aggregation. For example, a pier may aggregate piles, footings, a column, and a member;
- for feature elements, positioning is relative to the affected building element. For example, an opening element for a drain is positioned relative to the slab it voids,
- for fillings, positioning is relative to the filled opening. For example, a drainage structure is positioned relative to an opening which in turn is positioned relative to a slab;
- for distribution ports, positioning is relative to the containing distribution element. For example, a waste terminal may have a port connection for a pipe segment;

If a containing spatial structure contains a grid, then placement may also be based relative to grid coordinates.

In certain use cases, an absolute placement may be used by omitting the IfcObjectPlacement. In this case, the shape representation is defined within the world coordinate system.

As of IFC 4.1, there is no structure defined for placing an object relative to an alignment curve. A proposed solution for this is to introduce a connection relationship for this purpose, which enables forward and backward compatibility with existing IFC-compatible software, and allows a wider variety of applications (e.g. steel fabrication) to make use of the data outside of the bridge industry. Such solution is described in a later section. When using such relationship, IfcLocalPlacement is defined – while redundant, this is also consistent with design plans commonly seen for bridges which include both alignment-based parameters and resulting Cartesian coordinates.

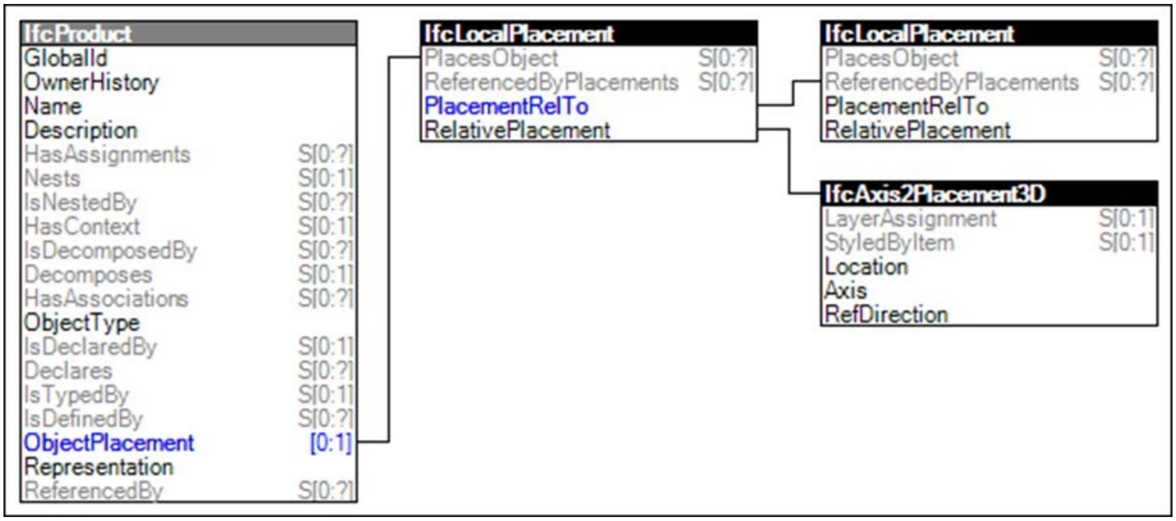


Figure 56: Placement in IFC

4.6.3 LandXML

Placement of bridge elements is done according to starting and ending station along a referenced alignment curve, along with the width of the bridge. Positioning of elements of bridges is not captured.

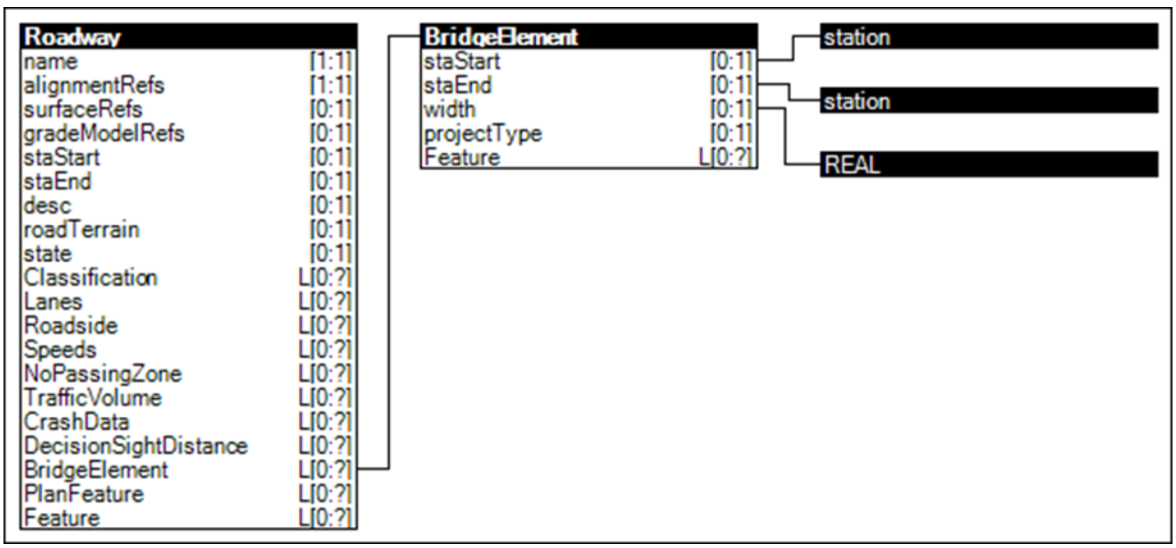


Figure 57: Placement in LandXML

4.7 Alignment

Placement of bridges and roads may be defined relative to offsets along an alignment curve. In the U.S., positions along an alignment curve are customarily referred to as "stations" at every 100 feet, where the notation of a station offset takes the form "190+34.25" (indicating 19034.25 feet)

4.7.1 OpenBrIM

An alignment is represented generically using the "Obj" structure consisting of inner "line" data structures.

Note: the OpenBrIM 2.0 XSD defines "RoadwayGeometry", and "HorizontalAlignmentSegment" subtypes, however these are not referenced anywhere from other data types, so it is not possible to encode these according to the formal XSD definition. This is presumed to be a work in progress, where such data types are to be further defined.

Note: the OpenBrIM 3.0 examples currently illustrate objects having reserved names to indicate alignment segments of particular types.

4.7.2 IFC

An alignment is an identifiable object consisting of horizontal and vertical curve segments. Usage of the alignment, including placement of objects, is independent of the alignment curve, though references this curve.

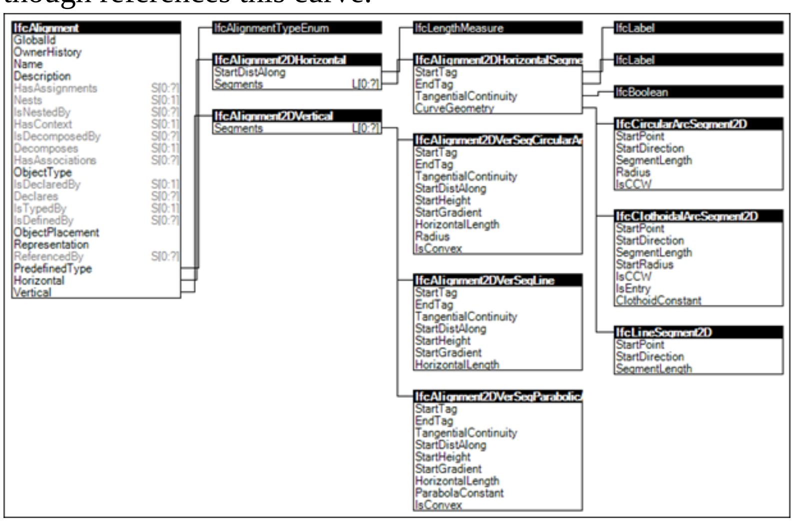


Figure 58: Alignment in IFC

The horizontal curve consists of one or more continuous segments, which may be linear, circular, or spiral. Each horizontal curve segment is defined by a starting point, starting direction, and length.

- The StartingPoint parameter indicates X and Y Cartesian coordinates, which are relative to the project coordinate system that defines placement within the geospatial coordinate system. After the first segment, such points may be derived from proceeding segments.
- The StartingDirection parameter indicates a unit vector within the horizontal plane. After the first segment, such directions may be derived from proceeding segments.
- The SegmentLength parameter indicates the length of the segment, according to the particular curve.
- The StartingRadius parameter (applicable to circular arcs and clothoidal arcs) indicates the radius at the beginning of the curve, where the IsCWW parameter indicates whether the curvature is clockwise or counterclockwise.
- The ClothoidConstant parameter (applicable to clothoidal arcs) indicates the rate that the radius of curvature changes along the segment, where the IsEntry parameter indicates whether the curvature increases or decreases along the segment.

The vertical curve is defined relative to the 2D horizontal curve and adds a vertical dimension. It consists of one or more continuous segments which may be linear, circular, or parabolic.

- The StartDistAlong parameter indicates the relative distance along the path of the horizontal curve. After the first segment, this parameter may be derived from proceeding segments.
- The StartHeight parameter indicates the vertical height at the beginning of the segment. After the first segment, this parameter may be derived from proceeding segments.
- The StartGradient parameter indicates the percent grade at the beginning of the segment, where positive values indicate upwards and negative values indicate downwards.
- The HorizontalLength parameter indicates the length of the curve as projected onto the horizontal curve.
- The TangentialContinuity parameter indicates whether a segment starts at the same tangent as at the end of the proceeding segment, which for bridges and roadways in general is most likely True.
- The ParabolaConstant parameter (applicable to parabolic curves) indicates the radius of the parabola at its vertical axis, where the IsConvex parameter indicates whether curvature turns upwards (through a valley) or downwards (over a hill)
- The Radius parameter (applicable to circular curves) indicates the radius of curvature. This is not applicable to roadways or bridges, but is provided for other domains.

The Alignment data structure may be referenced using the IfcRelPositions relationship described later within this document, which applies positioning, superelevation, and geometry transformation to physical elements of a bridge.

4.7.3 LandXML

The “Alignment” data types captures the horizontal alignment, along with specific usage information including profile, cross-sections, and super-elevations.

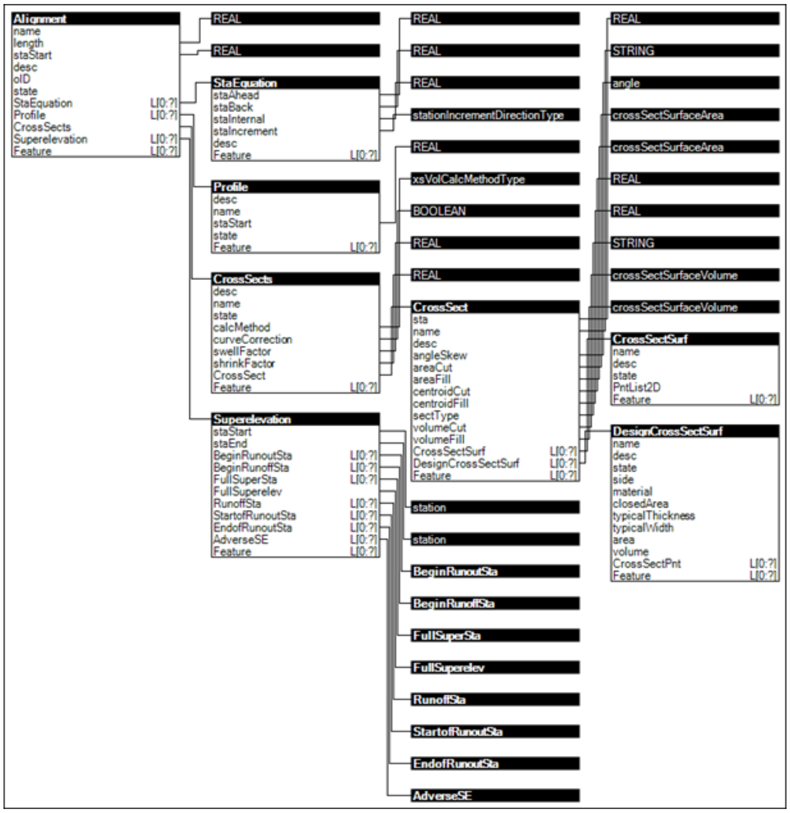


Figure 59: Alignment in LandXML

4.8 Cross-Sections

Cross-sections may define relative positioning of elements at constant or variable points along an alignment curve. While cross-sections are typically incorporated in plans, they may also be definitional such that 3D geometry may be derived by sweeping, rotating, and transforming cross-section definitions along an alignment curve.

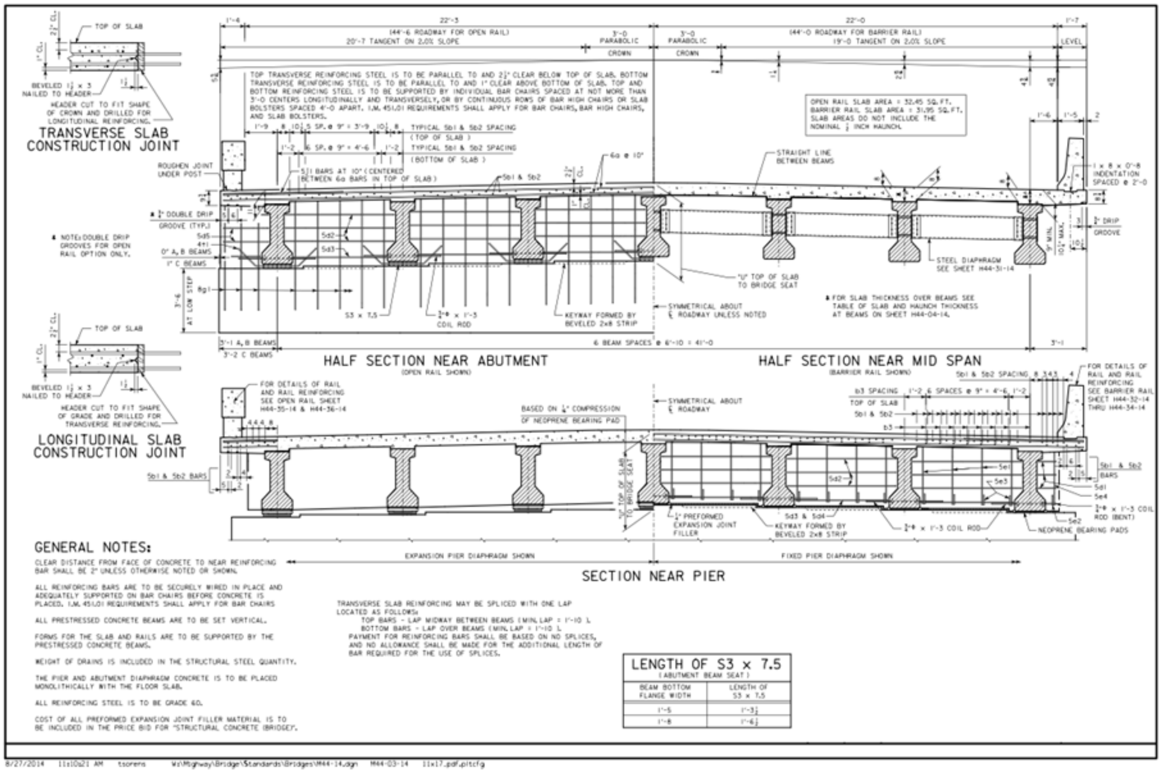


Figure 60: Cross section usage

4.8.1 OpenBrIM

No such concept has been explicitly defined in the schema, however it may be implied within the generic "Obj" element where placement offsets may be defined for components relative to alignment curve.

Profiles may be defined by objects containing lists of points, forming a polyline. Arcs or other curves are not directly supported but may be approximated by defining points at intervals along such curves.

4.8.2 IFC

Cross-sections are referred to as "profiles" in IFC. Profiles of materials used for individual elements are captured using IfcMaterialProfileSet. For extended placement, IfcMaterialProfileSetUsage provides indication of cardinal point positioning and reference extent.

For bridge decks along curved alignments, such profiles may be rotated or transformed (described by points that vary) along the alignment curve. Simple rotation may be achieved by using IfcDerivedProfileDef, where an angle may be applied to a fixed profile. For cross-sections where the shape varies and cannot be described by rotation alone, constraints (IfcRelAssociatesConstraint) may be used to attach formulas to points of an arbitrary profile which constrain the lateral and vertical positions to those of particular alignment curves. Such general formulas may then be used to form discrete geometry.

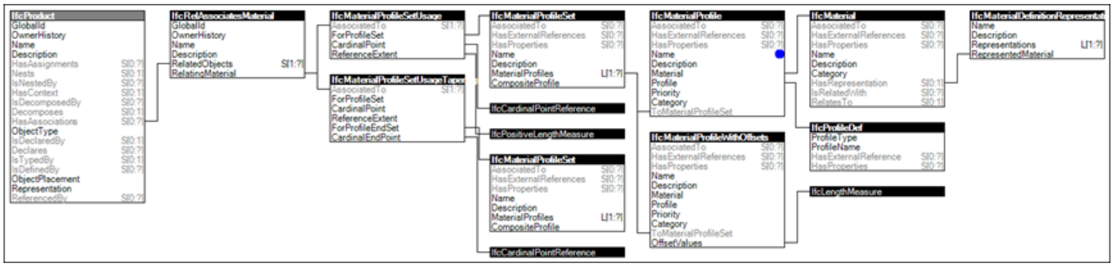
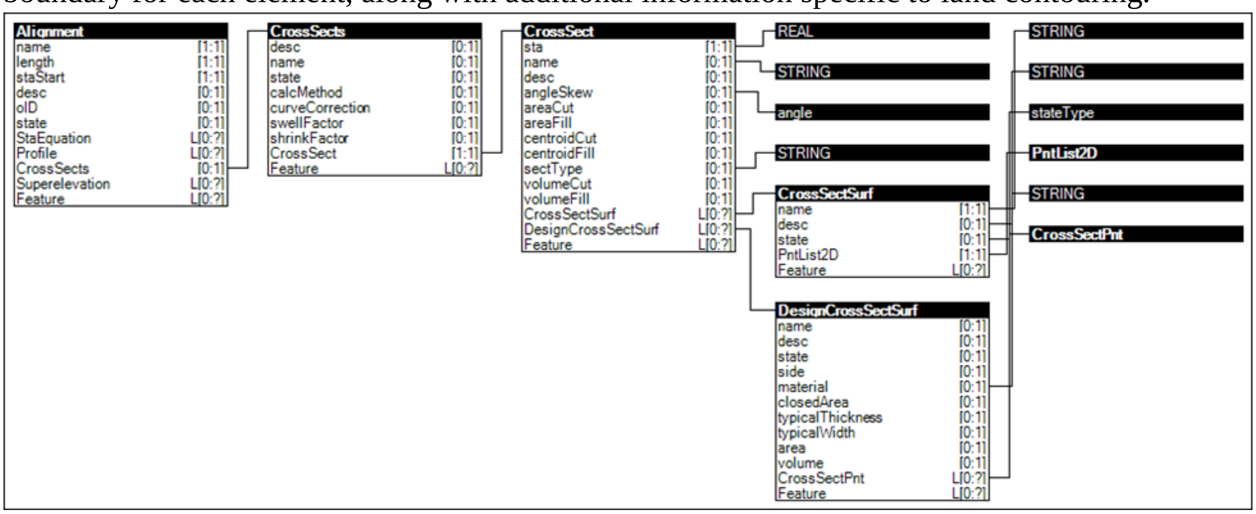


Figure 61: Cross-sections in IFC

4.8.3 LandXML

The “CrossSect” data type captures section information generically reduced to closed polyline boundary for each element, along with additional information specific to land contouring.



4.9 Geometry

Geometry refers to the 3D representation of elements which may be used to build the represented structure. While geometry may be used for rendering purposes for visualization, specific data structures may be required that capture parameters needed for construction or fabrication. For example, CNC machines understand identification of stock material by well-known identifiers and dimensions (e.g. “W” to identify wide-flange W-shape beams according to AISC dimensions), lengths for cutting, hole locations and sizes, surface markings, copings (CSG Boolean subtractions), etc. While some of this information could also be described using boundary representations (the outer faces) or tessellations (triangles of visible faces), such representation would not be sufficient for current software systems within this domain such as described by the DSTV format (Tekla, 2014). Visually, any of these representations may look exactly the same to a user of a CAD system, as they all result in the same tessellated representation (triangles) produced by computer graphics processors.

Geometry constructs may range from explicit to parametric, where the most useful form for a particular usage may vary somewhere in between. At the extreme of explicit geometry is tessellated geometry consisting of triangles and normal vectors at vertices, where the subdivision of triangles may be performed at fine intervals necessary achieve desired precision. At the other extreme are explicit formulas defining the positions and orientations of primitives. The former is ideal for visualization software, while the latter is more ideal for design software. In the middle of these extremes are intermediate constructs such as:

- Boundary representation (B-Rep): set of surfaces described by closed curves;
- swept solids: closed curves for cross-sections that are swept along open curves;
- primitives: 3D solids with specific parameters such as spheres, blocks, cylinders, cones, pyramids;
- Constructive Solid Modeling (CSG): unions, intersections, and subtractions of solids;
- Non-Uniform Rational B-Spline (NURBS): precise mathematical description of any solid;

Different modeling platforms may use different constructs to come up with the same result. In general, any physical element that is to be constructed or fabricated should be defined using the simplest form of parametric geometry that exactly describes the resulting shape. Tessellation should only be used to capture predefined manufactured products (such as light fixtures on a bridge), for which dimensional detail is not required.

The sample bridge models evaluated make extensive use of extruded solids with and without tapering. There were limited incidents of voiding (e.g. conduit within guardrail, drainage within slabs). The sample bridge models contained circular curves for some geometry (e.g. piers, terrain). All geometry could be described using swept geometry with voiding and composition relationships.

4.9.1 OpenBrIM

A single geometric representation is defined describing the 3D Surface of the object.

Several data types are defined:

- Line: collection of points describing path; curves interpolated
- Surface: collection of points describing boundary; curves interpolated
- Volume: collection of surfaces

Such capability enables boundary representation of flat surfaces with linear segments, and no voids.

Higher-level geometry for sweeps makes use of parameters and formulas. For example, an extruded beam may have a profile defined as a sequence of points, and multiple surfaces defined that reference these points along repetition intervals. The structure of such higher-level geometry is not captured in the schema itself, but relies on reserved names for object types. Swept geometry does not support tapering (differing start and end profiles).

All geometry follows point and line segments, such that any arcs (circular, parabolic, or other) must be interpolated with line segments.

Note: OpenBrIM 3.0 supports tapered extrusions (differing profiles at each end) and voids within cross-sections

4.9.2 IFC

Multiple geometric representations are defined:

- Surface: 3D surface tessellation after subtracting all dependencies
- Body: 3D parametric shape before applying any voids
- Axis: 3D path for sweeping material profiles
- FootPrint: 2D closed curve for material layers

The following extended representations have been proposed:

- Repeat: 3D path(s) of repetition patterns

The Body representation supports the following types:

- Boundary representation: IfcFacetedBrep, IfcFacetedBrepWithVoids, IfcAdvancedBrep
- Swept solids: IfcExtrudedAreaSolid, IfcExtrudedAreaSolidTapered, IfcRevolvedAreaSolid, IfcRevolvedAreaSolidTapered, IfcSweptDiskSolid, IfcSweptDiskSolidPolygonal, IfcSurfaceCurveSweptAreaSolid, IfcFixedReferenceSweptAreaSolid, IfcSectionedSpine;
- Primitives: IfcBlock, IfcSphere, IfcRightCircularCone, IfcRightCircularCylinder, IfcRectangularPyramid;
- CSG: IfcCsgSolid, IfcHalfSpaceSolid, IfcBooleanResult
- NURBS: IfcBSplineSurface, IfcBSplineCurve (and subtypes)

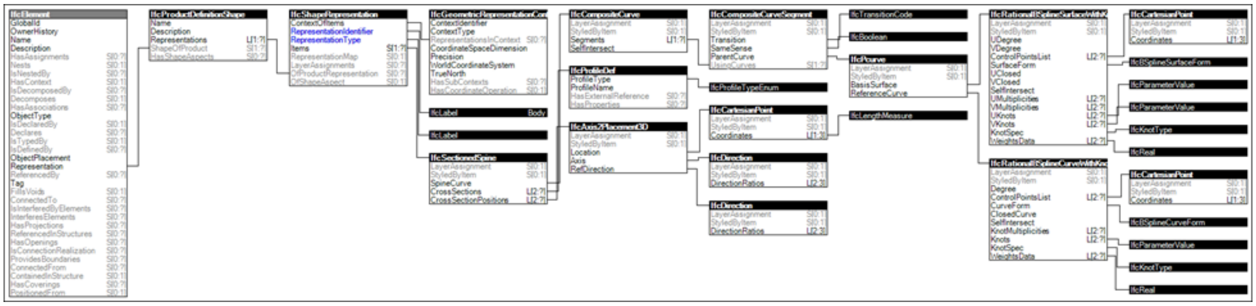


Figure 62: Geometry in IFC

Geometry for bridge decks fits four general cases, where 'Alignment curves' indicates whether the horizontal and/or vertical alignment has any curvature (rather than straight line), 'Cross section rotates' indicates whether there is super-elevation such that the same profile may be rotated along the alignment, and 'Cross section varies' indicates that working points of the profile vary independently along the alignment, such as for keeping surfaces in contact with girders in the horizontal plane while the cross-section has an incline overall.

Table 6: IFC swept solid geometry

Entity	**Alignment curves**	**Cross section rotates**	**Cross section varies**
IfcExtrudedAreaSolid	No	No	No
IfcFixedReferenceSweptAreaSolid	Yes	No	No
IfcSurfaceCurveSweptAreaSolid	Yes	Yes	No
IfcSectionedSpine	Yes	Yes	Yes

4.9.3 LandXML

Cross-section geometry may be used to approximate elements along alignments, and tessellated geometry (triangles referencing shared points) may be used for all other elements.

4.10 Presentation

To describe architectural details as found in plans, colors and textures may be applied to geometry. Such information may also be used for visualization.

For bridges, this is of particular use for describing architectural treatments, and also provides quick visual indication to help distinguish between materials such as steel and concrete girders. For bridges and roadways in general, lane striping may be conveyed according to line colors and dash patterns.

For deriving plans from digital models, presentation information also provides a place to encode drafting conventions.

The sample bridge model evaluated made use of textures to describe architectural detailing as described in the plans. Coloring was also used to support visual identification of materials.

4.10.1 OpenBrIM

No such concept has been explicitly defined in V2.0 schema. Of note, example files include extensions to indicate colors on objects.

4.10.2 IFC

Style information may be applied to any element, element type, material, or presentation layer.

Surface styles are used for 3D representations to indicate color, transparency, light reflectance, and textures.

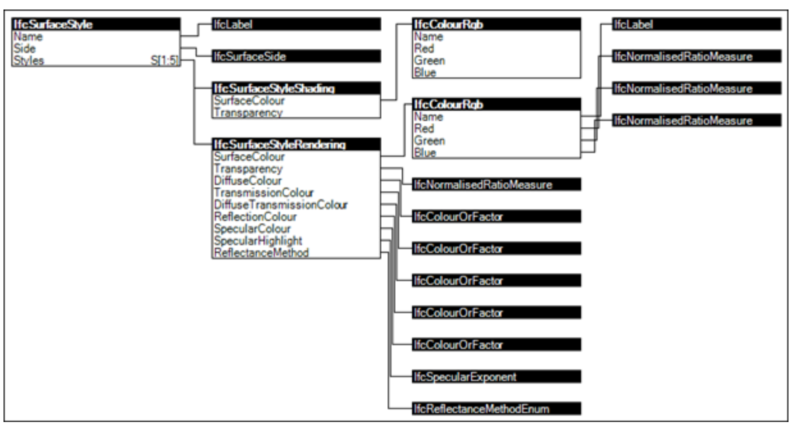


Figure 63: IFC presentation instance diagram

Textures may be stored in external files or embedded within the IFC or IFCXML file, supporting PNG, JPG, GIF, and BMP formats.

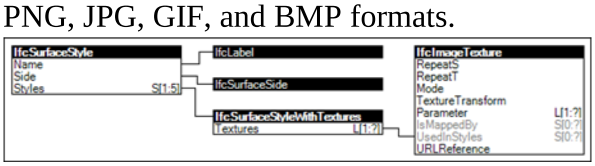


Figure 64: IFC surface style instance diagram

Fill styles are used for 2D plan representations to indicate fill color, cross-hatching, or tiling.

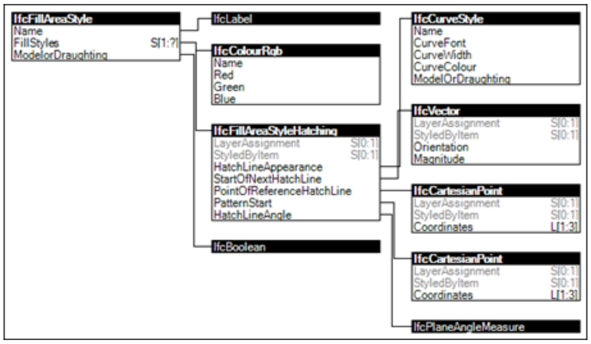


Figure 65: IFC fill style instance diagram

Curve styles are used for 2D plan representation to indicate line color, thickness, and dash style.

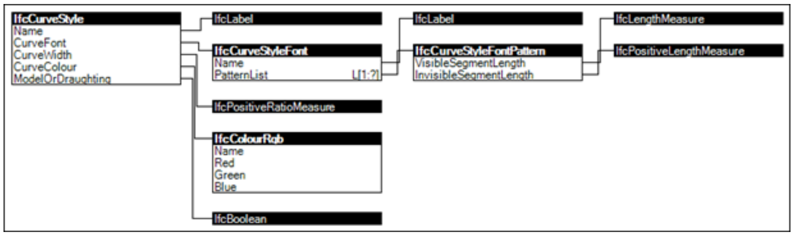


Figure 66: IFC curve style instance diagram

4.10.3 LandXML

No such concept has been defined.

4.11 Attributes

For schema definitions, attributes may be defined within the schema itself, or defined dynamically by future standard extensions, software applications, or end-users.

The balance between statically and dynamically defined attributes impacts usability for different scenarios: in general, software applications that consume data can be more efficiently developed with smaller schemas and dynamic attributes, while software applications that generate data can be more efficiently developed where schemas are defined statically, as validation is built-in to the schema itself.

Such balance also impacts performance and memory limitations of software that reads or writes such data. For representing physical structures such as buildings and bridges, this has historically been a limiting factor and still is as of 2015, such that the level of detail of files is often constrained by practical limits of computer hardware.

Typically, data structures are defined in native programming languages (e.g. C#, C++, Java) that correspond to the published data exchange schema, often automatically by programming tools. A statically defined attribute (where the type is defined by the schema) may be compiled to consume the minimum space required in memory rounded up to the size of the processor architecture. For example, a floating-point number such as for representing a distance would consume 8 bytes on a 64-bit system (most common as of 2015). If such floating-point number is stored on a dynamically defined attribute, it consumes at least 32 bytes on a 64-bit system: 8 bytes to identify the field, 8 bytes to reference the value of the field which points to a dynamically instantiated object (where value types are referred to as "boxed" data types in the context of Java or .NET environments), and 16 bytes for a boxed floating point instance (8 bytes to identify the type as an Int64, and 8 bytes to hold the actual value).[2] Thus, dynamically defined attributes require a minimum of 4 times more memory than statically defined attributes (32 vs. 8), plus additional memory to hold data structures for dynamic lists on each object instance (minimum of 32 bytes for an array-based collection).

The capability of supporting attributes that may be extended beyond the fixed schema is beneficial, but it comes at the cost of performance. It is recommended that any schema for data exchange should not rely on dynamically defined attributes for critical data that is known in advance, but only for more rarely used data or extensions not anticipated at the time of the schema development.

4.11.1 OpenBrIM

Attributes are defined on objects according to well-known identifier. Values may be literals or expressions calculated according to other parameters.

---

[2] https://www.simple-talk.com/dotnet/.net-framework/object-overhead-the-hidden-.net-memory--allocation-cost/

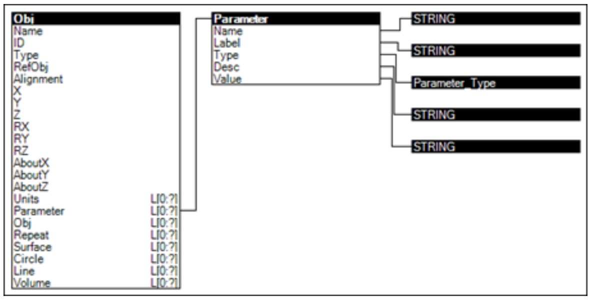


Figure 67: Attributes in OpenBrIM

4.11.2 IFC

Attributes are defined on objects according to well-known identifier of property set and well-known identifier of property.

Property templates may be provided to define data types and usage. Properties may be single values, bounded values (upper and lower limits), enumeration values (selecting from set of discrete values), lists, tables, time series, or nested properties.

Values are always literals; any expressions that control values are defined as constraints independent of the value.

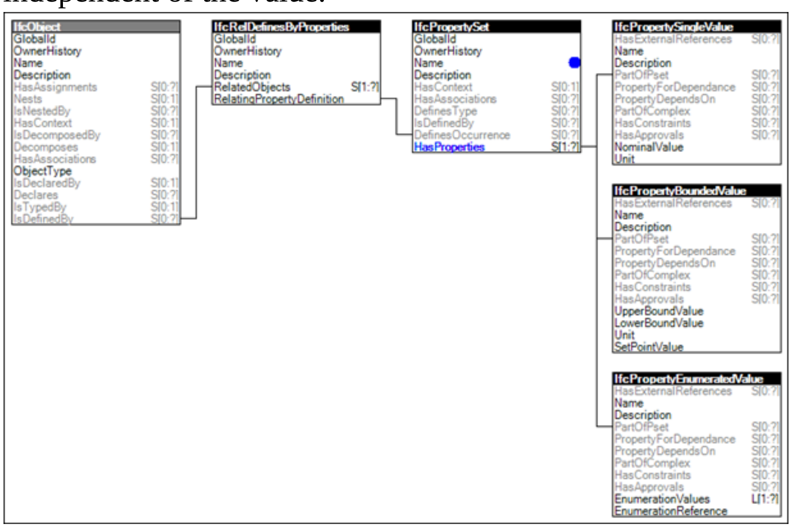


Figure 68: Attributes in IFC

4.11.3 LandXML

Attributes are defined on objects according to well-known identifier of feature (“code” and “source”) and well-known identifier of property (“label”). Values are always literals. Properties may be nested.

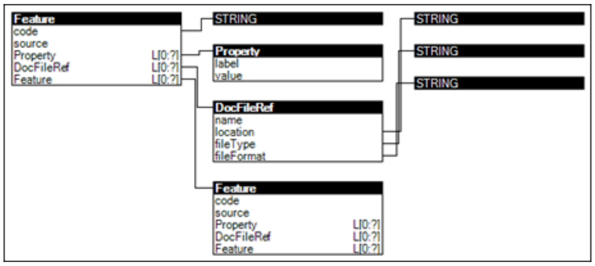


Figure 69: Attributes in LandXML

4.12 Parameters

For data that is extensively derived from parameters, it may be useful to capture formulas used for producing the resulting data, such as positioning and dimensioning of physical elements. While most design software supports such capability, software significantly differs in the structuring of such formulas, and some software may use higher-level constructs that conceal the underlying formulas. An example of such construct is anchoring, where elements are positioned at offsets relative to specified sides of another element (e.g. top/bottom, left/right, near/far).

To support the widest interoperability between software platforms, standardization of parametric relationships must balance flexibility with uniformity. At ultimate extremes, the highest flexibility is essentially encoding a programming language, while the highest uniformity would mean defining specific constructs for every envisioned scenario.

In addition to using explicit formulas, functions, and programming constructs, parametric behavior may be defined declaratively such as by indicating how elements are connected, and which elements and dimensions may be resized.

4.12.1 OpenBrIM

Any STRING-based value may be a literal or an expression calculated according to other parameters. The syntax of the expression includes arithmetic operators (+, -, *, /), parentheses to indicate operator precedence, parameter names (see Parameter object), and a list of built-in functions which include Sin, Cos, Tan, Asin, Acos, Atan.

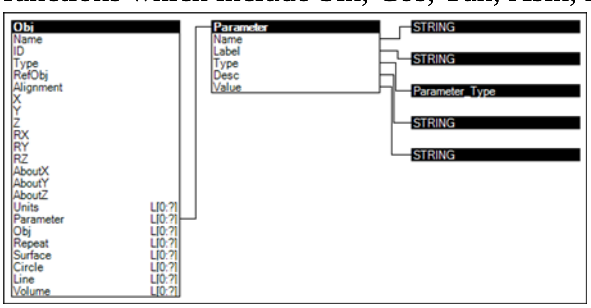


Figure 70: Parameters in OpenBrIM

Note: Some examples for OpenBrIM 3.0 illustrate extended syntax such as values that vary at particular intervals; definition of this syntax is not yet available.

4.12.2 IFC

Any value of any type may have a formula or lookup table applied using the IfcRelAssociatesConstraint relationship. The resulting value of such formula is encoded independently, which enables downstream applications to directly access resulting data without calculating formulas, while enabling authoring applications to encode formulas in a rigid structure that can be validated by the schema itself.

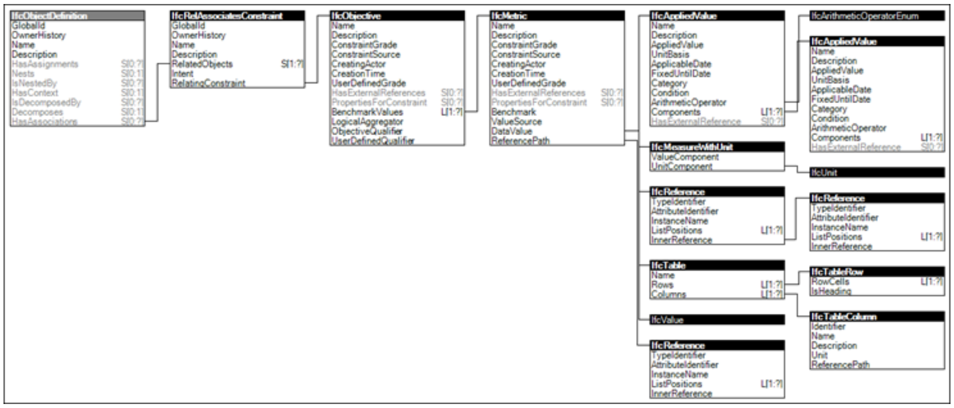


Figure 71: Parameters in IFC

The IfcRelAssociatesConstraint relationship references an IfcObjective which qualifies the constraint and lists formulas applied to values on the referenced objects. IfcMetric is used for each attribute, where IfcMetric.ReferencePath identifies the attribute (directly or along a graph of object references), and IfcMetric.DataValue indicates the formula. The data value may be any one of the following:

- IfcAppliedValue: an arithmetic operation on other values or parameters
- IfcTable: a table containing available combinations of values
- IfcReference: a reference to a parameter on objects
- IfcMeasureWithUnit: a value with explicit unit
- IfcValue: a value assuming default project units

An advantage of using explicit data structures to encode formulas is to ensure that data is encoded uniformly across implementations, which also eliminates the possibility of vendors introducing proprietary extensions. A disadvantage of using data types instead of strings is that it may require additional space in file formats, and may be less readable.

In addition to explicit constraints, IFC also provides several intermediate structures to enforce consistency of common parametric scenarios. For path-based extrusions such as beams or columns, cross-sections may be defined that are to be swept along arbitrary curves using IfcMaterialProfileSet. For boundary-based extrusions such as walls or slabs, layer thicknesses may be defined that are stretched to fill a boundary using IfcMaterialLayerSet. IFC also defines various subtypes of objects having “StandardCase” suffix, which indicates that such elements support parametric resizing behavior.

For assemblies consisting of multiple components, several connectivity relationships are defined, which in addition to describing physical and logical connectivity, may also be used to enforce parametric resizing behavior. The relationship IfcRelConnectsElements may be used to indicate that one element is at a fixed position on another element. The relationship IfcRelConnectsPathElements may be used to indicate that one path-based element (such as a beam) connects to the head, tail, or along the path of another path-based element. The relationship IfcRelConnectsWithRealizingElements may be used to indicate the connection mechanism specifically, such as a pot bearing, where additional mechanical behavior may be described. For all such relationships, the connection may be constrained to a point, along a line, within an area, or within a volume.

4.12.3 LandXML

No such concept has been defined.

4.13 Arrays

Another aspect of parametric design is defining elements which may repeat in multiple directions forming arrays. Wall studs commonly repeat in one direction along the axis of a wall, while floor tiles commonly repeat along two directions of a floor region. Such repetition is not necessarily linear; for example, rebar may repeat following a path constrained by enclosing stirrups, which could be polygonal or circular

For repetition in one direction, consider the scenario of stud framing in a building – while positioning of studs may be described by spacing at particular intervals, such a positioning algorithm quickly becomes more complicated when dealing with openings, blocking, lintels, pipe cavities, etc. where attribute-driven equations cannot adequately describe the layout.

For repetition in two directions, consider the scenario of tile layout on a floor – in the most simple scenario, tiles may have a square layout, starting at offsets in either direction, and repeat until filling boundaries. In more complex scenarios, there may be multiple colors and shapes of tiles, some place at recurring patterns potentially rotated or offset at each course, and some placed irregularly.

For bridge design, in reviewing standardized plans at DOT websites, it has been observed that there are many scenarios that can benefit from parametric relationships where positioning may be described by attribute-driven formulas. However, such parameters cannot be a substitute for the resulting positioning and dimensioning data, as the sample bridges evaluated suggest that there will always be scenarios requiring explicit positioning or users and software that produce layouts requiring programmatic algorithms.

It is envisioned that placement of longitudinal rebar and stirrups would most benefit from parametric extensions, where such usage would define uniform placement offsets. But here, such offsets are not simply linear; longitudinal bar positioning is based on the shape of the stirrups, and stirrup positioning is not necessarily constant, but may have variable spacing becoming denser where the maximum shear occurs near supports.

The bridge test cases evaluated contained some elements that could be described according to repetitive patterns (e.g. rebar meshes), though most repetitive elements had exceptions to placement that precluded use of repeatable patterns (e.g. piles, foundation walls, girder segments, bridge deck segments, rebar in piers and abutments). The bridge test cases revealed the following scenarios of repetition:

- Repeating at constant spacing in one direction (e.g. rebar within bridge decks)
- Repeating at variable spacing in one direction (e.g. stirrups within guard rails)
- Repeating at stepped offsets (e.g. stirrups within pier caps)
- Repeating at variable angles (e.g. stirrups within pier caps)
- Repeating about an axis (e.g. rebar within circular columns)

- Repeating with linear scaling in one direction (e.g. stirrups within pier cap)

4.13.1 OpenBRIM

Every element has a Repeat structure that indicates how multiple occurrences may be constructed at various offsets.
Note: OpenBrim 3.0 drops this structure in favor of a generic Obj having well-known type identifier of "Repeat".

4.13.2 IFC

An element may have multiple placements using multiple instances of IfcMappedItem. However there is no place to record any parameters describing repetition patterns.

An extension is proposed in this document to support repetition patterns defined by a custom representation called "Pattern".

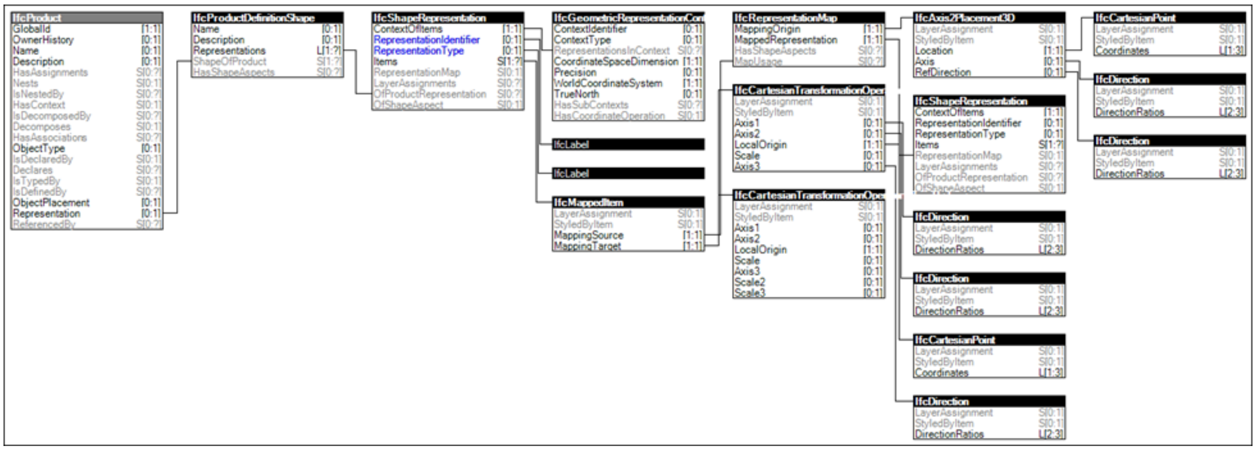


Figure 72: Arrays in IFC

4.13.3 LandXML

No such concept has been defined.

4.14 Soil Conditions

For bridge design, of utmost importance are the ground conditions supporting the foundations of bridges. Such conditions include physical properties of different layers of soils and expected water table elevations. In capturing such information, samples are taken at various points along a site.

In addition to soil topography of the resulting construction, capturing existing conditions is also required for estimating the amount of cut/fill for different soil types, and the equipment required.

The sample bridge model evaluated includes soil layers and properties for each boring.

4.14.1 OpenBrIM

No such concept has been defined.

4.14.2 IFC

IFC4 captures the top ground elevation as a set of points and break lines at IfcSite and IfcGeographicElement.
Specific layers are captured using IfcMaterialLayerSet at each sample point, where IfcMaterial and IfcMaterialProperties capture properties specific to each layer.
The water table is captured as IfcExternalSpatialElement with PredefinedType of Water, and surface geometry indicating the elevation. Note that due to drainage and other conditions, the water table is not necessarily a constant or static elevation; it may change over time and vary at different locations.

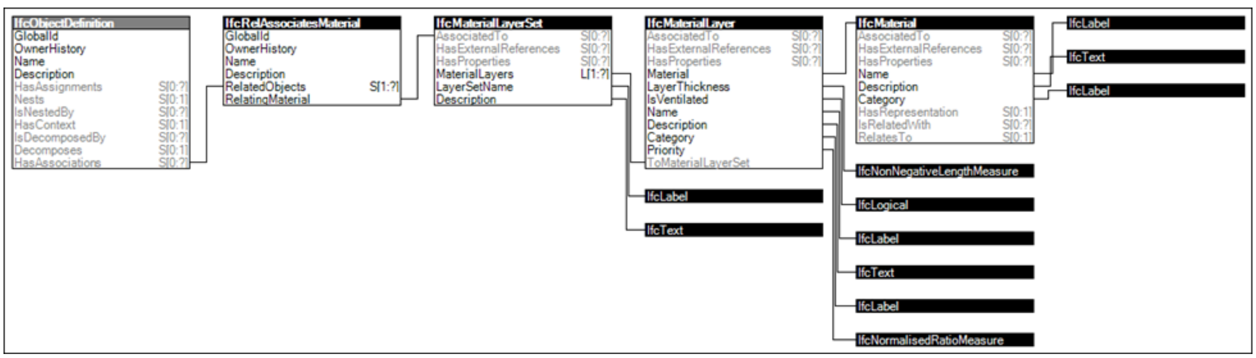


Figure 73: Soil conditions in IFC

4.14.3 LandXML

LandXML captures the top ground elevation along with specific soil layers and their properties.

4.15 Quantities

Quantity information involves calculating and summarizing counts, lengths, areas, volumes, and masses of physical elements. Such information may be used for estimating or screening projects for suitability according to a contractor's capacity and capabilities. While such information is inherently derived, and may be calculated by users or software, it may be desirable to include such information for downstream processes, and as verification for the contractor preparing a bid. Such information also provides the basis for transportation agencies to prepare budgets on projects in advance of receiving bids.

The sample bridge model evaluated includes quantities for all components.

4.15.1 OpenBrIM

No such concept has been defined.

4.15.2 IFC

Standard quantity definitions are defined as extensible values using *IfcElementQuantity* for every physical element (e.g. footing, pile, beam, slab, rebar, etc.) which may include gross and net variations (accounting for openings or voids). Standardized quantity sets are defined for all types of products, processes, and resources, though may be extended to capture additional quantities.

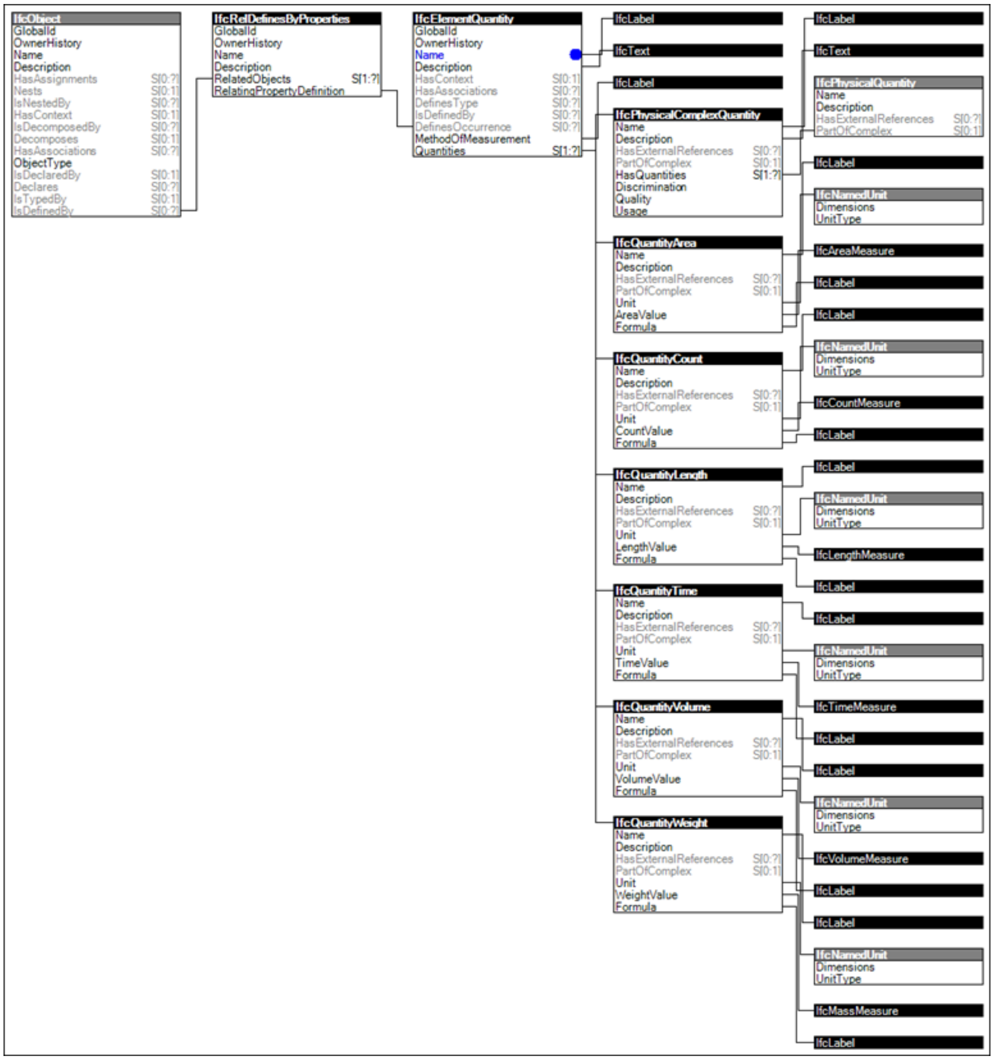


Figure 74: Quantities in IFC

Quantities may be assigned to schedules (IfcCostSchedule) with or without costs applied, and such schedules may be incorporated into contracts (IfcWorkOrder) or bid requests (IfcActionRequest).

4.15.3 LandXML

Quantities are captured on definitions as specific attributes. For example, CrossSect includes areaCut, areaFill, volumeCut, volumeFill.

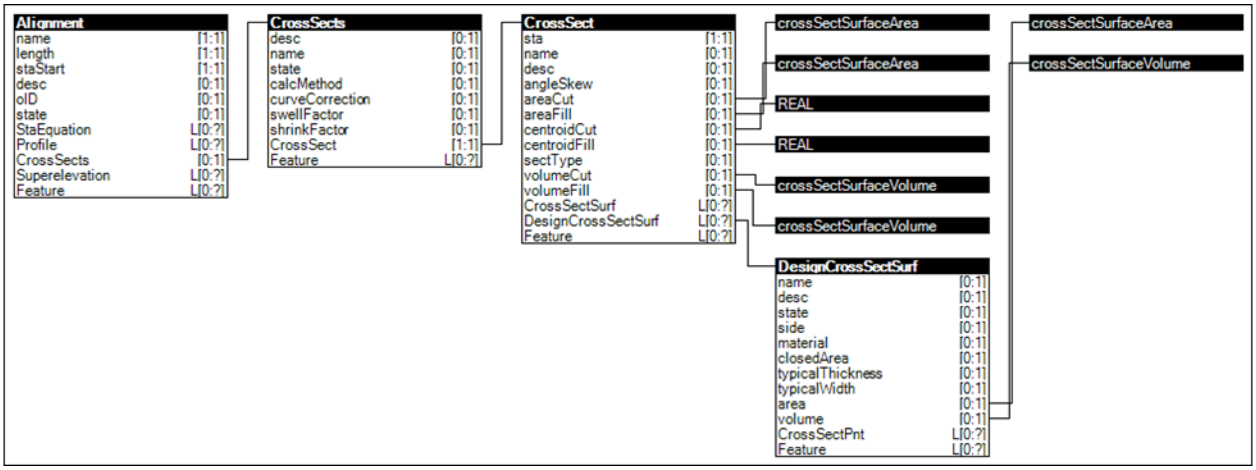


Figure 75: Quantities in LandXML

4.16 Structural Analysis

Structural analysis of bridges involves mapping physical elements to analytical representations for purposes of determining physical behavior of elements in response to loads.

Fundamentally, structural analysis models are comprised of members, connections, and loads. A member may be a linear segment having point connections at each end, a surface having linear or point connections, or a volume having surface, linear, or point connections. A connection may be a point, line segment, or surface, whose translation or rotation may be constrained along any axis. Loads may take the form of points, lines, or areas, and may be applied to members accordingly. Loads may be organized into load cases and combinations, where analysis may be done for each combination.

For performing analysis, analytical surfaces and volumes may be converted to meshes of linear members and connections, where the resolution of such a conversion may be application-dependent.

For construction purposes, it may be required to determine the expected stresses in various members. Contractors may use such information in determining allowable loads for temporary equipment or material storage during construction. Engineers may use such information to verify the anticipated loads and that the design meets the intended usage.

The sample bridge model evaluated includes design loads and reactions for structural members.

4.16.1 OpenBrIM

There are no specific data structures defined for structural information, however there are reserved object types with specific parameters are defined. The OpenBrIM 2.0 documentation covers structural point loads, and appears to have made provisions for other load types, load cases, and load combinations, however the documentation does not elaborate on such data types.

4.16.2 IFC

There may be zero, one, or more analytical models, each of which may be assigned to the overall bridge structure or as multiple separate models that may be analyzed in sequence or in parallel – for example, a bridge deck, framing, and supports may be analyzed as one unified analysis model or as separate analysis models where the support reactions of one system correspond to loads within another system. Analytical models may be derived from other models – for example, a model containing surface members may be reduced to a model containing meshes of linear members.

Structural members and connections are represented separately from the physical building components, and may have assignment relationships to the corresponding building components.

Structural loads may be captured as point loads, linear loads, or area loads with constant or variable force and moment, displacement and rotation, or temperature change.

Structural results may be captured at supports and at elements as fixed-end-reactions, such that stresses and deflections at any point along a member may be arithmetically derived (e.g. shear, moment, deflection diagrams).

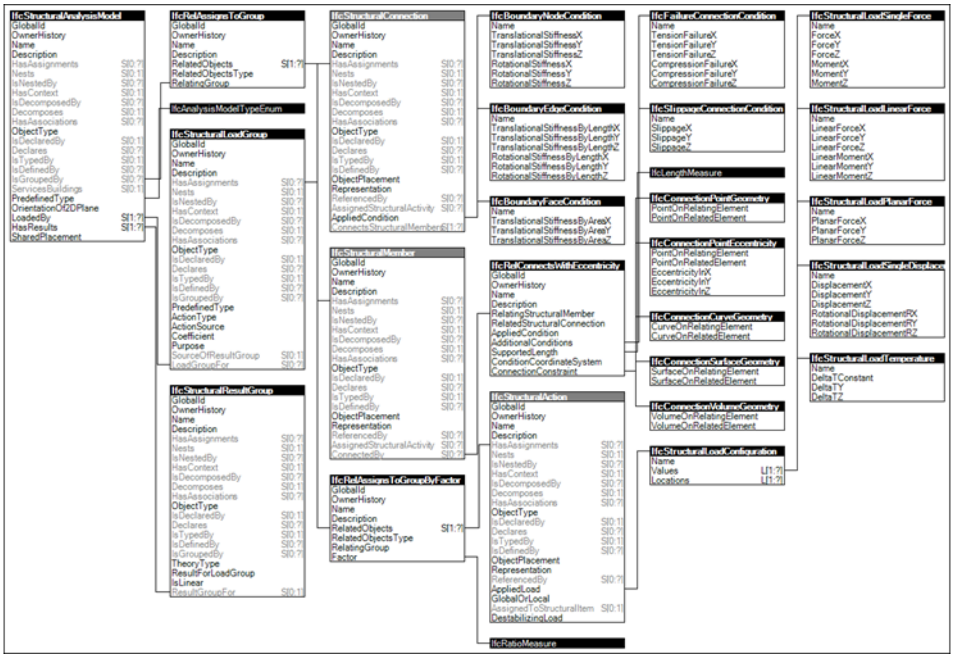


Figure 76: Structural analysis in IFC

The following capabilities are currently lacking in the IFC4 model, which are recommended to be added to support structural analysis of bridges:

- Volumetric members: For resisting shear in multiple directions and to capture more complex volumes often used for bridge supports, volumetric modeling may be necessary. For analysis, such volumes may be reduced into meshes of linear members and point connections.
- Physical-to-analytical relations: Also required is the capability to link analytical members to failure plans of soil. The current IFC model requires a 1:1 linkage between a structural member and the physical element it idealizes, whereas for soil, one physical object may correspond to many structural members.
- Dynamic loads: For modeling loads of traffic that vary at positions along a bridge span, such analysis may be converted into static models by integrating loads along paths. Supporting dynamic loads would not necessarily require a schema change, but rather documenting the use of time-based constraints to simulate forces of varying range, varying direction, and varying position over a time interval. However, as this is a more common scenario for traffic, defining explicit structures may better enforce consistent usage by implementers.
- Staged construction: For modeling conditions that may vary during construction, such as girders that may be temporarily cantilevered or supported by temporary structures, there is a need to relate a structural analysis model to a particular phase of construction (IfcTask). This can be accomplished using existing assignment relationships, however such usage would need to be documented.

4.16.3 LandXML

No such concept has been defined.

4.17 Connectivity

Relationships between objects are required to extract physical connectivity information to understand construction dependencies. Relationships are also necessary to define structural information included in construction plans such as design loads, reactions within elements (e.g. maximum stresses and deflections), and connectivity between elements such that analytical models may be derived. Such structural models may be used for construction to determine anticipated equipment loadings, and for maintenance in the future to understand design criteria for a bridge.

4.17.1 OpenBrIM

No specific data structures are defined for connectivity. It may be possible to describe such relations using parameters, though there is no referential integrity or indexing capability that can be derived from the schema.

4.17.2 IFC

Connectivity relationships may be defined using IfcRelConnects subtypes:

- Host connectivity: General connections are defined using IfcRelConnectsElements where RelatingElement refers to the anchoring element which must be installed before the RelatedElement.
- Realizing connectivity: Connections with specific elements used for attachment (e.g. plates with bolts) are defined using IfcRelConnectsWithRealizingElements. Such connectivity relationship may be used to derive structural connection parameters for use within a structural analysis model.
- Axis connectivity: Connections between elements following parametric paths may use IfcRelConnectsAxisElements, where intersecting behavior may be defined (i.e. order of precedence for components).

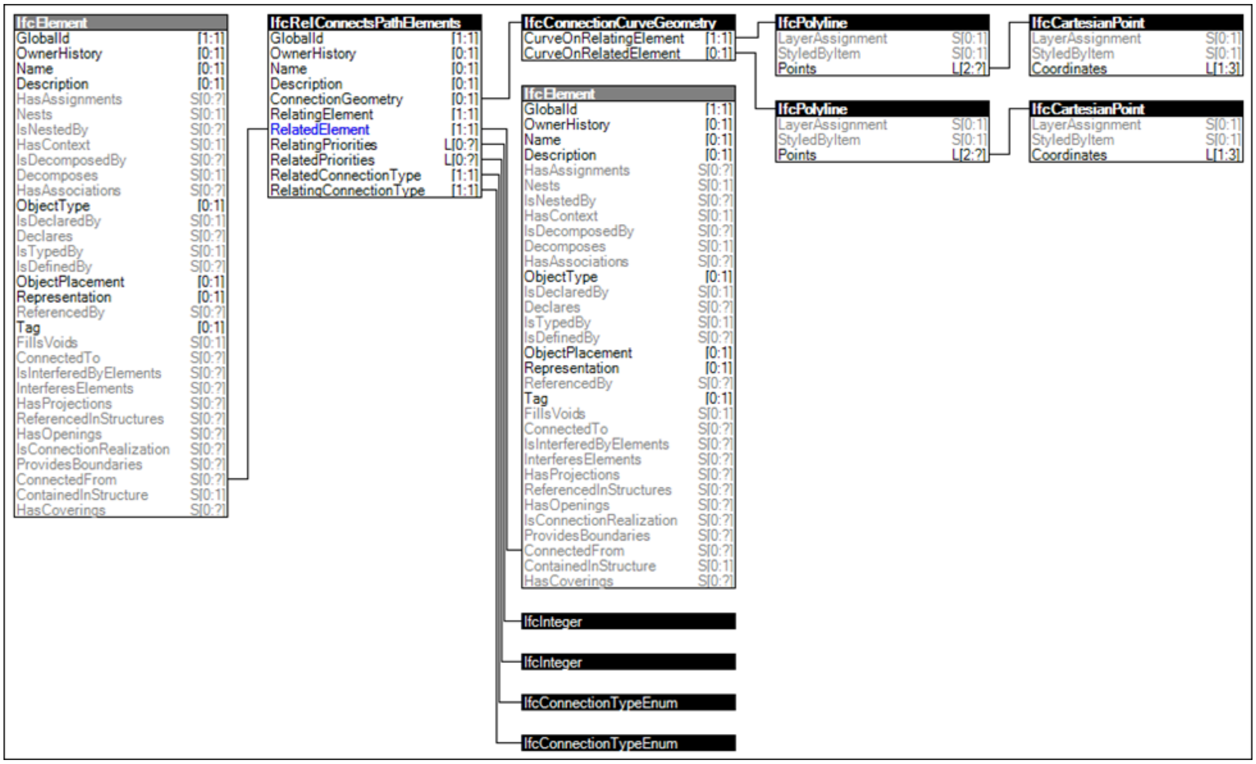


Figure 77: Element connectivity in IFC

4.17.3 LandXML

No specific data structures are defined for connectivity.

5 Proposed IFC Definitions for Immediate Use

In consideration of proposing additions or modifications to the IFC schema, heavy emphasis must be placed on retaining compatibility with current software and limiting such expansion to only that which is absolutely required to fit use cases.

In accelerating usage of IFC for bridges, several compatible extensions are proposed. One schema addition is proposed to define placement of bridge elements relative to alignment curves, while other additions do not require schema changes, but specific usages of existing structures.

It is anticipated that such additions or adaptations thereof, will become part of the “Alignment 1.1” initiative, which has been funded as of October 2015, and is scheduled to be completed in 2016, at which time IFC 4.2 will be published to contain such definitions.

5.1 Alignment Placement

The proposed entity *IfcRelPositions* provides a mechanism to indicate placement and transformation requirements of a physical element relative to a positioning element such as an alignment curve or grid. This entity derives from *IfcRelConnects* and works similarly to other such connectivity relationships deriving from *IfcRelConnects*. For example, walls in buildings are placed using *IfcRelConnectsPathElements* to ensure that if repositioned, walls move or stretch according to such relationships. Similarly, *IfcRelConnectsElements* is also used more generally for attaching fixtures to walls.

Using a separate connectivity relationship allows for local placement (*IfcLocalPlacement*) to be indicated in parallel at *IfcProduct.ObjectPlacement*, which ensures compatibility with existing IFC software as well as future IFC software that may not support alignment curves but still supports targeted uses of such models such as for fabrication. The proposed relationship may be used to not only describe positioning, but also transformation of geometry along alignment curves.

Using an *IfcRoot*-based relationship also provides referential integrity between alignments and elements, such that schema definitions alone are sufficient for software to automatically maintain data integrity, such that if software updates an *IfcAlignment* definition, the links to dependent elements are explicitly defined. This also enables generic data integrity – for example, deleting an *IfcAlignment* instance results in cascading deletion of *IfcRelPositions* (according to the 1:1 *RelatedElement* requirement). Such data integrity enforcement would not be possible without *IfcRoot*-based relationships, as explained in IFC documentation and further below.

The definition for *IfcRelPositions* is defined in Table 7, deriving from *IfcRelConnects*.

Table 7: IfcRelPositions attributes

Attribute	Type	Description
RelatingElement	IfcElement	The element to be positioned.
RelatedPositioning Element	IfcPositioningElement	The positioning element defining one or more axes for relative positioning.
GridOrdinates	IfcInteger[0:3]	Optional indices of the axes defined by the related positioning element, where 1 refers to the first item in a list. For IfcAlignment, the first index identifies an IfcAlignmentStation according to the Stations list at IfcAlignment, and the second index identifies an IfcAlignmentAxis according to the Axes list at IfcAlignment.
EndGridOrdinates	IfcInteger[0:3]	For elements that span between two points, indicates optional ending indices of the axes defined by the related positioning element. For longitudinal elements (e.g. girders, bridge decks, guard rails), the first index varies and the second index (lateral) is constant. For lateral elements (e.g. cross-bracing), the first index is constant and the second index varies. For vertical elements such as piers, this value is not provided.
RelativePlacement	IfcAxis2Placement3D	Optional relative placement after applying positioning from GridOrdinates, DistanceAlong, OffsetLateral, and OffsetVertical to support special cases of positioning at particular Cartesian offsets and orientations, and in spaces that cannot be addressed from alignment curves having discontinuous transitions. The RefDirection is defined relative to the horizontal alignment curve at the referenced position. If PlacementType is VERTICAL, then the Axis is defined relative to the vertical slope.
DistanceAlong	IfcLengthMeasure	Distance along the alignment curve, where positive is in the direction of the curve. If the distance is less than zero or greater than the extent of the alignment curve, then the alignment curve is assumed to be

Attribute	Type	Description
		linear following the orientation at the corresponding boundary. If GridOrdinates are provided, then this value is relative.
OffsetLateral	IfcLengthMeasure	Lateral offset from the alignment curve where positive values indicate to the left and negative values indicate to the right, as facing in the direction of the alignment. If the alignment curve has a super-elevation surface defined, then the vertical position is calculated accordingly. If GridOrdinates are provided, then this value is relative.
OffsetVertical	IfcLengthMeasure	Vertical offset where positive values indicate globally upwards and negative values indicate globally downwards. Note: to indicate vertical offsets that are normal to the alignment (as opposed to gravitational), RelativePlacement may be used for that purpose.
PlacementType	IfcAlignmentPlacementEnum	Indicates whether positioning is calculated according to only the horizontal alignment curve, or the vertical alignment curve relative to the horizontal alignment curve.
TransformType	IfcAlignmentTransformEnum	Indicates whether element is either positioned as a whole, has components and mapped geometry positioned separately, or has geometry warped to follow the alignment curve.
ProfileOrdinates	IfcInteger[1:?]	Optional list of indices that relates cross-section points within a polyline from an IfcArbitraryClosedProfileDef to a corresponding IfcAlignmentAxis, such that the vertical position of profile points along a sweep is made relative to the AxisElevations provided at IfcAlignmentStation.

The IfcAlignmentTransformEnum enumeration provides indication of how the positioned element is transformed to fit the alignment curve. For example, railings along the main span of a bridge (having very subtle curvature) may use CHAIN to indicate that each individual railing segment retains its shape (cheaper to fabricate) but is rotated, while railings along bump-out plaza areas (having pronounced curvature) may use WARP to indicate that the each railing segment is specifically fabricated as curved. Rebar would typically use CHAIN (typically not long enough to require specific bending), while tendons would typically use WARP (to reflect final representation, even though tendons are flexible).

Table 8: IfcAlignmentTransformEnum constants

Constant	Description
PLACE	Element is placed such that the local position corresponds to the point along the referenced curve, such that any components retain their relative Cartesian positions, and no geometry is transformed. Horizontal rotation is relative to the horizontal alignment curve. Vertical rotation is relative to the vertical axis (independent of slope).
CHAIN	Element is placed at the global origin, and geometry at the element and components is transformed by mapped item corresponding to the point along the referenced curve.
WARP	Same behavior as CHAIN, except geometry is transformed, where X follows the alignment curve, Y is lateral to the alignment curve, and Z is vertical (upwards) regardless of slope.

Software implementations that support editing functionality of alignment transformation may transform shapes defined as object types by converting all longitudinally swept geometry (IfcExtrudedAreaSolid and IfcExtrudedAreaSolidTapered having extrusion in +X direction) into IfcSectionedSpine, and all other geometry into transformed boundary representations (IfcFacetedBrep and IfcAdvancedBrep).

The IfcRelPositions definition is designed to support positioning structures generically, including both IfcGrid and IfcAlignment. In addition to the connectivity relationships, inverse attributes are defined at IfcElement and IfcPositioningElement, as shown in the diagram.

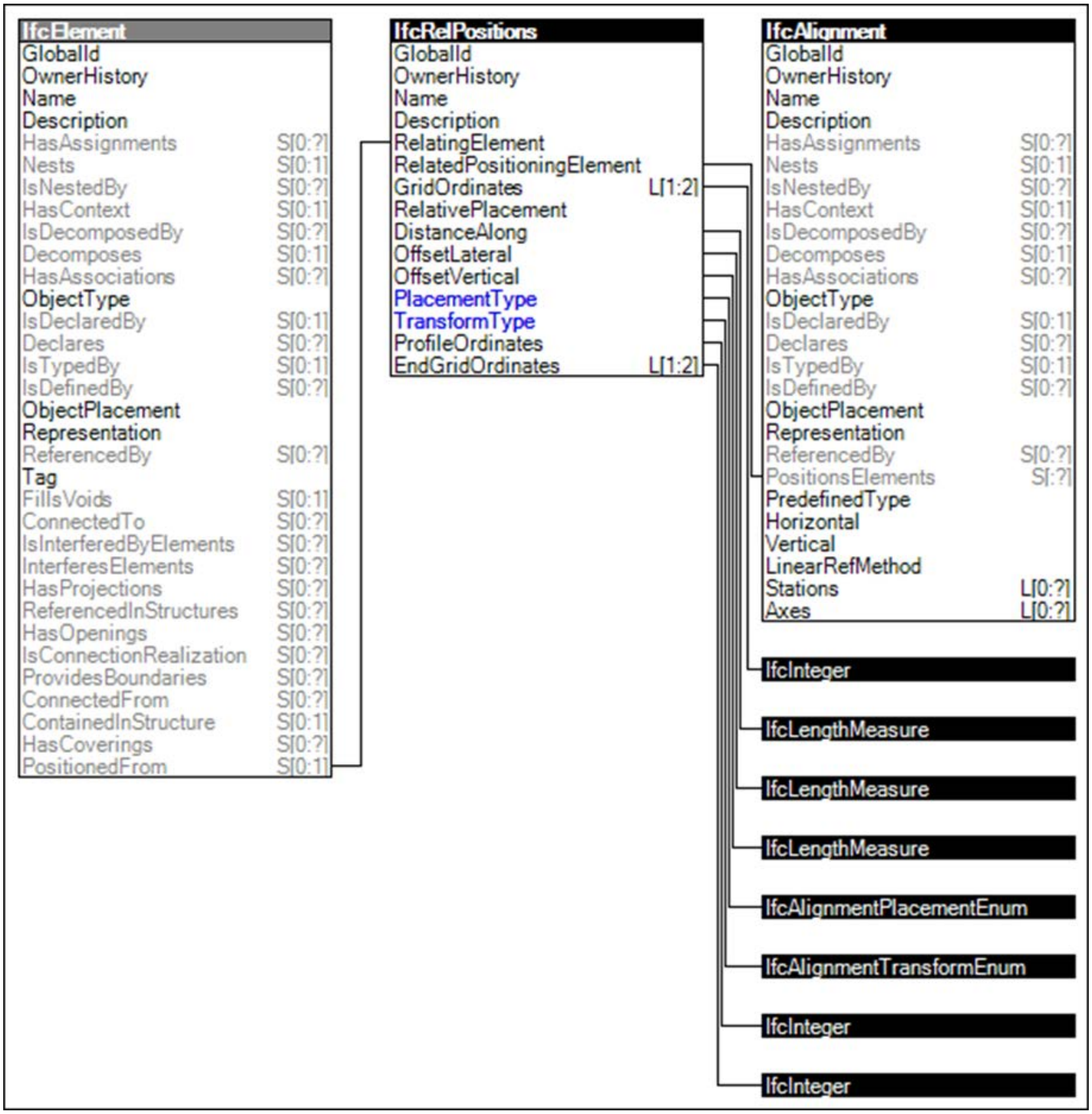


Figure 78: Element positioning instance graph

For grids, this also replaces usage of IfcGridPlacement (which has not been widely supported by software). This also makes usage of grids more easily adopted, as virtually all IFC software understands IfcLocalPlacement, while a subset supports IfcGridPlacement. This also solves a technical issue with IfcGridPlacement, which identifies grid axes by reference to IfcGridAxis – as only IfcRoot-based objects support object referencing, such relationship by itself cannot be relied upon to uniquely determine an axis; rather the Name of an axis must be used. As a general rule in IFC to support efficient data retrieval and storage, pointers to non-IfcRoot instances do not imply any semantic relationship, and such instances may be shared for data interning purposes – for storage in IFC-SPF format, such instances are optimally interned to minimize file size, while for storage in IFC-XML format, such instances are optimally duplicated to maximize readability. For database implementations, such instances are optimally duplicated by encoding such instance graphs within fields of a single row of data for an IfcRoot-based object such that retrieval time is minimized by avoiding unnecessary joins of tables for which the data would otherwise not need to be indexed.

As an alternative to the above structure, separate relationships could be defined that are more specific to IfcAlignment and IfcGrid.

Usage of this relationship is illustrated in the two example bridge files and accompanying documentation included in the full Model View Definition specification along with this report.

In addition to the new positioning relationship, a grid may be defined in relation to the alignment that defines longitudinal working points, lateral working points, and reference elevations as commonly defined for steel girder bridges as shown in Figure 61.

DECK ELEVATIONS OVER ℄ OF GIRDER AND DECK BREAK POINTS

LOCATION	PGL STA	PGL	G4	G3	G2	G1	A	B	C	D
℄ BRG ABUT 1	61+25.00	1039.38	1040.58	1040.12	1039.36	1038.60	1040.54	1040.62	1040.58	1038.42
	61+38.00	1040.13	1041.34	1040.87	1040.11	1039.35	1041.29	1041.37	1041.33	1039.17
	61+51.00	1040.88	1042.09	1041.62	1040.86	1040.10	1042.04	1042.12	1042.08	1039.92
	61+64.00	1041.64	1042.84	1042.38	1041.62	1040.86	1042.80	1042.88	1042.84	1040.68
	61+77.00	1042.39	1043.60	1043.13	1042.37	1041.61	1043.55	1043.63	1043.59	1041.43
	61+90.00	1043.14	1044.35	1043.88	1043.12	1042.36	1044.30	1044.38	1044.34	1042.18
	62+03.00	1043.90	1045.10	1044.64	1043.88	1043.12	1045.06	1045.14	1045.10	1042.94
	62+16.00	1044.65	1045.85	1045.39	1044.63	1043.87	1045.81	1045.89	1045.85	1043.69
	62+29.00	1045.40	1046.61	1046.14	1045.38	1044.62	1046.56	1046.64	1046.60	1044.44
	62+42.00	1046.15	1047.36	1046.89	1046.13	1045.37	1047.31	1047.39	1047.35	1045.19
℄ BRG PIER 1	62+55.00	1046.91	1048.11	1047.65	1046.89	1046.13	1048.07	1048.15	1048.11	1045.95
	62+73.00	1047.95	1049.15	1048.69	1047.93	1047.17	1049.11	1049.19	1049.15	1046.99
	62+91.00	1048.99	1050.20	1049.73	1048.97	1048.21	1050.15	1050.23	1050.19	1048.03
	63+09.00	1050.03	1051.24	1050.77	1050.01	1049.25	1051.19	1051.27	1051.23	1049.07
	63+27.00	1051.08	1052.28	1051.82	1051.06	1050.30	1052.24	1052.32	1052.28	1050.12
	63+45.00	1052.12	1053.32	1052.86	1052.10	1051.34	1053.28	1053.36	1053.32	1051.16
	63+63.00	1053.16	1054.36	1053.90	1053.14	1052.38	1054.32	1054.40	1054.36	1052.20
	63+81.00	1054.20	1055.37	1054.92	1054.18	1053.45	1055.33	1055.41	1055.37	1053.27
	63+99.00	1055.24	1056.34	1055.92	1055.23	1054.54	1056.29	1056.37	1056.33	1054.37
	64+17.00	1056.29	1057.30	1056.91	1056.27	1055.63	1057.26	1057.34	1057.30	1055.48
℄ BRG PIER 2	64+35.00	1057.32	1058.27	1057.90	1057.31	1056.71	1058.22	1058.30	1058.26	1056.57
	64+48.00	1058.05	1058.94	1058.60	1058.04	1057.48	1058.89	1058.97	1058.94	1057.34
	64+61.00	1058.76	1059.58	1059.27	1058.75	1058.22	1059.53	1059.61	1059.59	1058.10
	64+74.00	1059.45	1060.20	1059.93	1059.44	1058.94	1060.15	1060.23	1060.23	1058.83
	64+87.00	1060.12	1060.80	1060.57	1060.11	1059.65	1060.75	1060.83	1060.84	1059.54
	65+00.00	1060.77	1061.38	1061.18	1060.76	1060.34	1061.34	1061.42	1061.44	1060.24
	65+13.00	1061.40	1061.95	1061.78	1061.39	1061.00	1061.90	1061.98	1062.02	1060.91
	65+26.00	1062.01	1062.49	1062.36	1062.00	1061.62	1062.44	1062.53	1062.57	1061.53
	65+39.00	1062.61	1063.02	1062.92	1062.60	1062.22	1062.96	1063.06	1063.11	1062.13
	65+52.00	1063.18	1063.52	1063.46	1063.17	1062.79	1063.45	1063.57	1063.63	1062.70
℄ BRG ABUT 2	65+65.00	1063.73	1064.01	1063.98	1063.72	1063.34	1063.93	1064.07	1064.13	1063.25

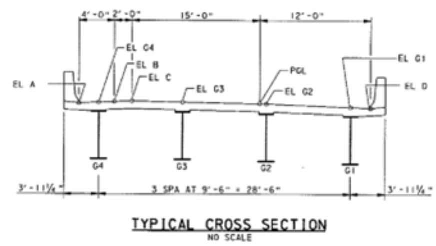


Figure 79: Alignment working points schedule and drawing

The IfcAlignment data structure is extended to capture two additional fields as shown in Table 9.

Table 9: IfcAlignment extended attributes

Attribute	Type	Description
Stations	IfcAlignmentStation[0:?]	List of positions along alignment that may be used to derive longitudinal and vertical offsets for physical elements.
Axes	IfcAlignmentAxis[0:?]	List of positions lateral to an alignment that may be used to derive lateral offsets for physical elements.

IfcAlignmentStation indicates a position along an alignment curve that may be referenced to place physical elements.

Table 10: IfcAlignmentStation attributes

Attribute	Type	Description
Tag	IfcLabel	Tag to annotate the alignment station.
DistanceAlong	IfcLengthMeasure	Distance along the horizontal alignment, measured along the IfcAlignment2DHorizontal given in the length unit of the global IfcUnitAssignment.
ReferenceAngle	IfcPlaneAngleMeasure	Angle indicating a reference super-elevation, where zero indicates the surface normal faces upwards and positive values indicate counter-clockwise rotation as facing in the positive direction of the alignment.
AxisElevations	IfcLengthMeasure	List of elevations corresponding to the Axes list at IfcAlignment, indicating the vertical positions of working points.

IfcAlignmentAxis indicates a lateral position that is offset from an alignment curve, for which "Working Points" may be derived in combination with IfcAlignmentStation.

Table 11: IfcAlignmentAxis attributes

Attribute	Type	Description
AxisTag	IfcLabel	Tag to annotate the alignment axis
OffsetLateral	IfcLengthMeasure	Lateral offset from the alignment curve where positive values indicate to the left and negative values indicate to the right, as facing in the direction of the alignment.

An instance diagram of the extended attributes on IfcAlignment is shown in Figure 74.

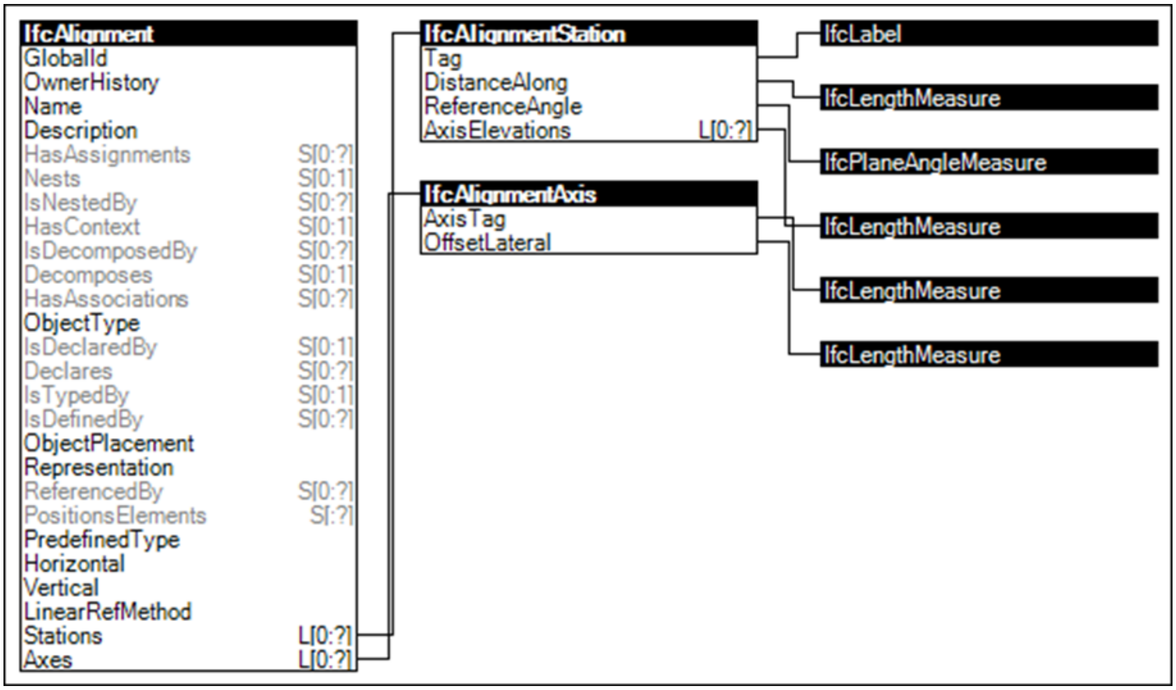


Figure 80: Alignment working points instance diagram

The buildingSmart "P6 Extension" project, also referred to as "IfcAlignment 1.1" could capture such proposed scope, along with any additional functionality required as found in other regions and domains.

5.2 Pattern Placement

Another aspect to be captured that is common in bridge structures (though also in buildings) is describing repetitive placement of objects according to patterns. Such patterns may consist of repeating in one direction at a fixed length interval (e.g. rebar within a slab, spindles of a bridge railing), rotating around a point (e.g. rebar within a circular column), or repeating in multiple directions (e.g. pavers along walkways). As these patterns of repetition take many forms, and different software supports different parameterized approaches, a custom shape representation is proposed to indicate such repetition. Such shape representation is similar to existing parametric representations such as 'Axis' and 'FootPrint' as documented, where software implementations only need to be interested in such intermediate representations if they support editing of the particular component based on such parameters.

The initial documentation of such representation uses the representation identifier 'Pattern' and documents usage of a single IfcVector instance, where the Orientation and Magnitude indicate how geometry is to be duplicated. Such duplication continues until reaching the extent of the item, which is defined by the bounding parametric representation such as 'Axis', 'Footprint', or 'Box'.

As such representation is optional, and is unknown to current IFC software, it does not remove the need to indicate each geometric placement specifically as is currently done for repetitive elements using IfcMappedItem. However, this representation provides the information necessary to regenerate the quantity and transformation of such mapped items if editing such definitions.

This representation may be used in combination with other parametric representations: 'Axis' for repeating along a curve; 'FootPrint' for repeating within a horizontal boundary; 'Profile' for repeating within a vertical boundary; or 'Box' for repeating within a volume.

More complex repetitive patterns may need to be defined to accommodate variations encountered, and are discussed in a following section for future definitions to be considered.

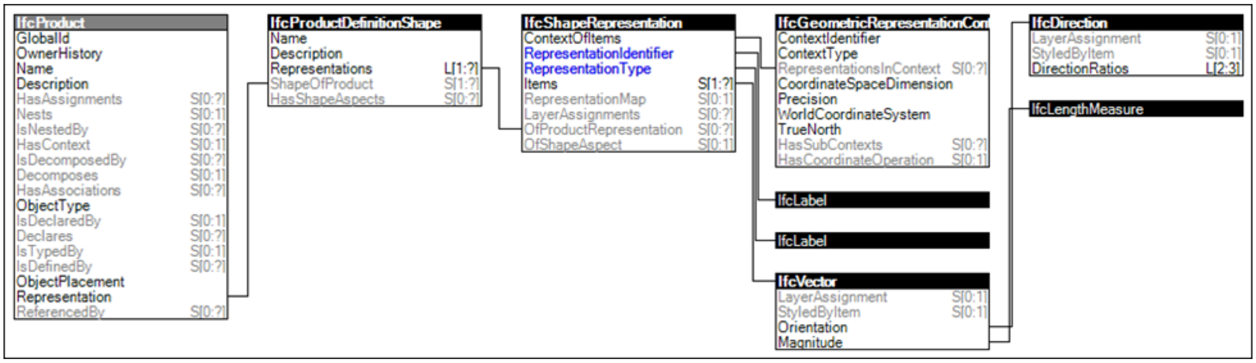


Figure 81: Pattern placement instance graph

5.3 Camber Ordinates

Camber ordinates are another item commonly required for bridge girders, where a custom representation could be used to indicate such information. However, IFC already provides fully documented data structures for doing so, and such structures not only indicate resulting camber, but also qualify such camber according to the load combination used (e.g. total dead load, self-weight of member only).

Camber ordinates may be derived from structural load results related to total dead load. For geometry that resides within spatial structures, it is assumed that all dimensions reflect the conditions as constructed in place (where such camber would be balanced out by resulting loads), therefore any camber must be captured separately. To relate camber to specific load results and load cases, the IfcBeam may link to an idealized structural model using the assignment relationship IfcRelAssignsToProduct, where *RelatedObjects* refers to the IfcBeam and *RelatedObjects* contains one or more idealized IfcStructuralCurveMember instances, where load results may be traversed following the *AssignedStructuralActivity* inverse attribute where IfcRelConnectsStructuralActivity.*AppliedLoad* refers to an IfcStructuralCurveReaction instance within a result set (*HasAssignments* related to IfcStructuralResultGroup via IfcRelAssignsToGroup) corresponding to the load combination (IfcStructuralLoadGroup) and analytical member (IfcStructuralCurveMember).

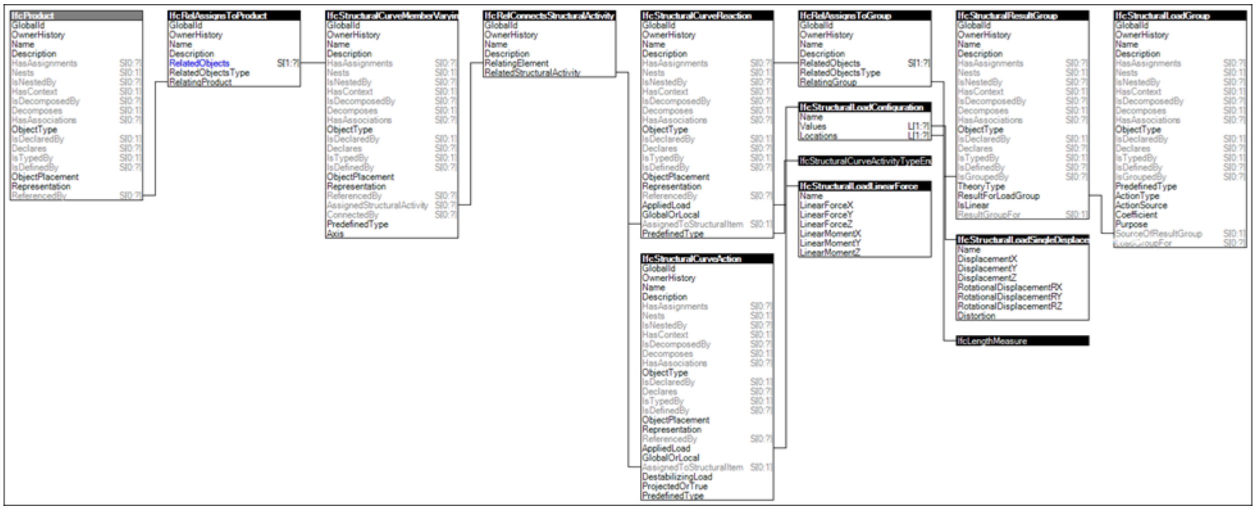


Figure 82: Camber ordinates instance graph

6 Definitions for Future Consideration

This section describes possible extensions to the IFC schema (with IFC4 as the baseline) that are not required, but may better accommodate bridge modeling. The proposed definitions will be compared with other definitions proposed within extension projects in other domains and countries. It is anticipated that the next major IFC release will contain such additions or adaptations thereof.

As extending a schema adds complexity, particularly for very large schemas such as IFC, a modeling bias is to re-use existing definitions where applicable (but without overloading the meaning), and only propose extensions where there is no integrated alternative.

Thus, none of the definitions included in this section are required to capture information for bridges, and none were relied upon in modeling sample bridges in IFC as part of this effort. The definitions described are only proposed as future convenience, and for future deliberation as alternatives to extensions, and would be expected to go through modification and balancing with ongoing initiatives in other countries in extending the IFC schema.

Of note, there is an entity called “IfcBuilding” within the IFC schema, which would suggest that there should also be an “IfcBridge” defined for parity. However, the actual usage of IfcBuilding (and related spatial elements such as IfcBuildingStorey and IfcSpace) is for relating spatial information within a building which is required for many uses cases in design and facilities management (but not construction other than being able to reference the location of elements). For bridges, the corollary of such spatial elements would involve roadways and lanes. While there certainly may be use for capturing such spatial information, such use was not required for the purposes of exchanging design-to-construction information, and therefore was not considered. There may be other exchanges where such information would be useful, for which such definitions may be described that meet requirements specific to such exchanges.

Various physical elements in bridge domains use particular terms not currently used within the IFC schema. For example, while there is no “IfcGirder”, there is an IfcElementAssembly entity with “GIRDER” as a predefined type enumeration, defining an assembly of beam segments using IfcBeam. IfcBeam is described as “horizontal, or nearly horizontal, structural member that is capable of withstanding load primarily by resisting bending”. Such definition encapsulates beams within structural framing as commonly understood, as well as floor joists, lintels above doors or windows, pre-stressed hollow-core slab components, as well as segments of girders in buildings (defined as either a “large” beam, or one that supports other beams)[3] For any such definition to be standardized, there needs to be clear differentiation between classifications, and particular uses requiring such distinction. Here, the distinction between a “beam” and a “girder” based on size is ambiguous (unless some specific measurement were to be defined), and the distinction between supporting other beams or not can already be deduced from connectivity relationships. IFC has defined a girder to mean a series of one or more connected beams over one or more spans.

[3] http://dictionary.reference.com/browse/girder

Likewise, there is no particular data structure defined for abutments or piers, other than IfcCivilElement which serves as a general placeholder. For purposes of exchanging design-to-construction information, such classification usage was not required, and therefore not considered. There may be other exchanges that would require such distinction, for example if there are local codes that make such distinction according to measurable criteria. Pragmatically, to accelerate industry adoption by supporting existing software in a compatible way, there is very good reason to not introduce new data structures if not absolutely needed. Such approach favoring compatibility is recommended to achieve more rapid adoption by software vendors in the near term. As of February 2016, China Railway Corp and Korea Institute of Construction Technology (KICT) have introduced additional elements that may be leveraged, where such distinctions are required for their particular uses, and may also be leveraged in the United States.

6.1 Geometric Model Resources

For representing bridge girders, particularly longitudinal beams along vertically curved sloped alignments, the IfcFixedReferenceSweptAreaSolid data structure is the most suitable. This provides for any arbitrary vertical curve, including parabolic curves, circular curves, or straight lines as defined by the vertical alignment curve. While this structure allows the profile to be oriented relative to the directrix (variable roll angle), the profile is always orthogonal to the directrix, which is incompatible with beams having vertical slope. An attribute needs to be added to describe this vertical orientation.

The proposed change is to add a "FixedAxis" attribute of type "IfcDirection", which indicates the direction of the Y axis of the profile relative to the directrix. Or, for compatibility, a new subtype may be introduced called IfcFixedAxisSweptAreaSolid which inherits from IfcFixedReferenceSweptAreaSolid.

For representing bridge decks, particularly along curved sloped alignments, the IfcFixedReferenceSweptAreaSolid data structure is also the most suitable, however it also requires specification of a fixed axis. Additionally, bridge deck segments are defined by separate profiles at each end – i.e. same shape but different vertical dimensions - which requires the addition of an attribute to capture the opposite end of the shape similar to IfcExtrudedAreaSolidTapered.

The proposed change for tapering is to add a new entity IfcFixedAxisSweptAreaSolidTapered, which includes the additional attribute "EndSweptArea" of type "IfcProfileDef".

The Table below indicates proposed attribute additions.

Table 12: IfcFixedAxisSweptAreaSolidTapered attributes

Attribute	Type	Description
SweptArea	IfcProfileDef	The surface defining the area to be swept. It is given as a profile definition within the xy plane of the position coordinate system.
Position	IfcAxis2Placement 3D	Position coordinate system for the resulting swept solid of the sweeping operation
Directrix	IfcCurve	The curve used to define the sweeping operation
StartParam	IfcParameterValue	The parameter value on the Directrix at which the sweeping operation commences
EndParam	IfcParameterValue	The parameter value on the Directrix at which the sweeping operation ends.
FixedReference	IfcDirection	The direction providing the fixed axis1 (x-axis) direction for orienting the swept area during the sweeping operation along the Directrix.
FixedAxis	**IfcDirection**	**The direction providing the fixed axis2 (y-axis) direction for orienting the swept area during the**

Attribute	Type	Description
		sweeping operation along the Directrix.
EndSweptArea	**IfcProfileDef**	**The surface defining the end of the swept area.**

In addition to the new attributes and data structures, several other related changes are proposed to support usage of this structure.

There is a need for a single beam segment to span multiple horizontal alignment curve segments. This could be accomplished by making IfcAlignment2DHorizontal derive from IfcBoundedCurve, or else using IfcCompositeCurve where IfcCompositeCurveSegment may reference IfcCurveSegment2D subtypes. Without such capability, then the beam must be split into multiple geometric shapes.

There is a need for a single beam segment to be placed relative to a vertical alignment curve. This could be accomplished by making IfcAlignment2DVertical derive from IfcBoundedCurve and contain an attribute referencing the underlying IfcAlignment2DHorizontal for which the offsets are based, such that the definition is complete by itself.

An alternative to the above addition would be to use IfcSectionedSpine, but also adapt the underlying vertical and horizontal curves to be used by IfcSectionedSpine.SpineCurve directly rather than converting into IfcCompositeCurve.

6.2 Profile Resources

To support shapes commonly used for bridge deck cross-sections, several standardized profile types are introduced. As indicated in *Volume I*, the term "profile" refers to an arbitrary cross-section within this document, and not necessarily the vertical alignment of a bridge for which that term is also used.

IfcDeckSpanProfileDef is introduced to capture common bridge deck profiles. Rather than using IfcArbitraryProfileDef with dozens of vertices, a parameterized form may be more efficiently described. The downside of such extension though is the additional work required of implementers, most of which may not be supporting bridges as a main focus. Alternatively, constraint-based models may be used, where formulas may be defined for coordinates of every point along an IfcArbitraryClosedProfileDef. A compromise may be to introduce this as a property set rather than a static entity, for which formulas may also be encapsulated.

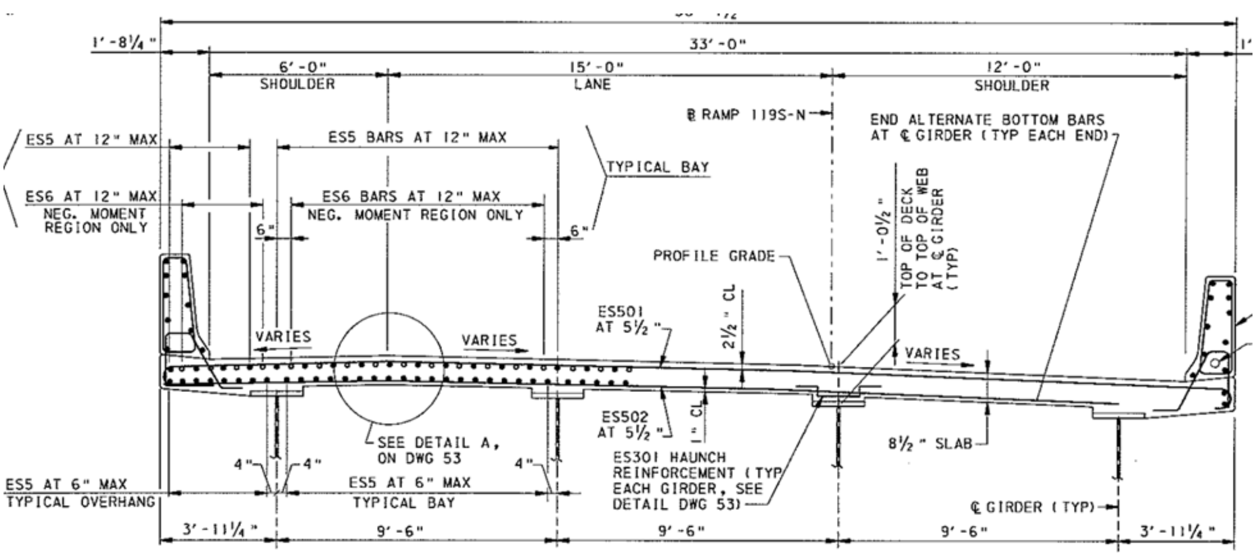


Figure 83: Deck cross-section

Table 13: IfcDeckSpanProfileDef attributes

Attribute	Type	Description
SpacingCount	IfcInteger	Number of spans between girders (girder count - 1)
GirderSpacing	IfcPositiveLengthMeasure	Center-to-center distance between each span
EffectiveFlangeWidth	IfcPositiveLengthMeasure	Width above each flange
OverhangWidth	IfcPositiveLengthMeasure	Width on each edge from the center of outer girder
BarrierWidth	IfcPositiveLengthMeasure	Width of barriers on each edge
RoadwayCrown	IfcLengthMeasure	Distance between center and where slope changes

Attribute	Type	Description
ThicknessMin	IfcPositiveLengthMeasure	Thickness of slab between girders
ThicknessMax	IfcPositiveLengthMeasure	Thickness of slab over girders
SlopeLeft	IfcPlaneAngleMeasure	Slope on the left side of the High Position
SlopeRight	IfcPlaneAngleMeasure	Slope on the right side of the High Position
OverhangSlope	IfcPlaneAngleMeasure	Slope underneath edges
BarrierSlope	IfcPlaneAngleMeasure	Slope underneath barriers

IfcBarrierFShapeProfileDef is introduced to capture common concrete barriers. Similar assumptions apply as in the previous profile, where it may be preferable to introduce this definition as a property set.

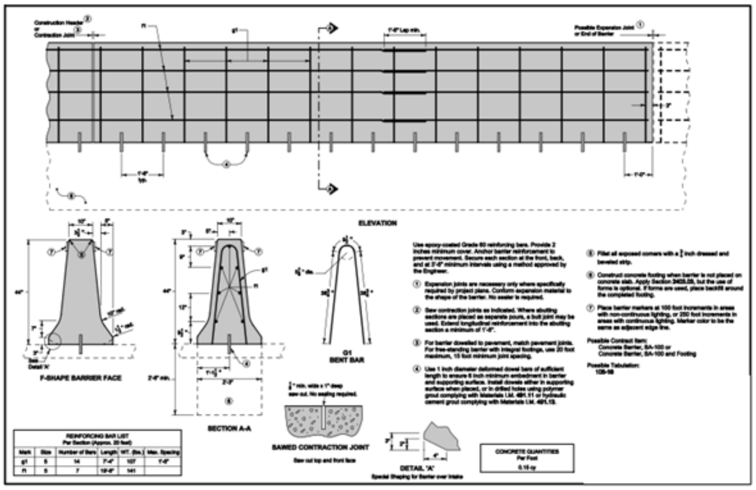


Figure 84: Guardrail cross-section

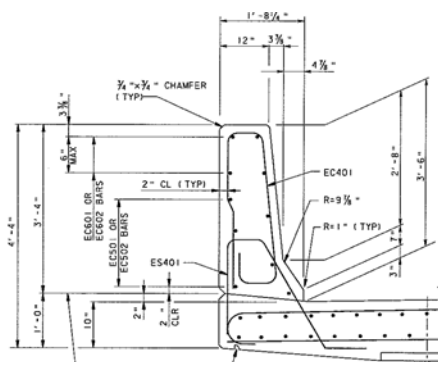


Figure 85: Guardrail dimensions

Table 14: IfcBarrierFShapeProfileDef attributes

Attribute	Type	Description
BaseWidth	IfcPositiveLengthMeasure	Overall width at largest dimension
OffsetTopLeft	IfcNonNegativeLengthMeasure	
OffsetBaseLeft	IfcNonNegativeLengthMeasure	
OffsetTopRight	IfcNonNegativeLengthMeasure	
OffsetBaseRight	IfcNonNegativeLengthMeasure	
OverallHeight	IfcPositiveLengthMeasure	Overall height at largest dimension
TopHeight	IfcPositiveLengthMeasure	Height of top
BaseHeight	IfcPositiveLengthMeasure	Height of base
BaseSlope	IfcPlaneAngleMeasure	Slope at base

6.3 Geometric Constraint Resources

IFC 4 supports placement according to position and orientation relative to an enclosing element, or relative to axes of a grid.

With the proposed extension of placing physical elements relative to alignment curves, also of interest is basing geometry on such alignment curves directly, rather than converting to existing swept solids.

To enable referencing of alignment curves from existing geometry such as IfcExtrudedAreaSolid, IfcFixedReferenceSweptAreaSolid, IfcSurfaceCurveSweptAreaSolid, or IfcSectionedSpine, either the alignment curve must derive from IfcBoundedCurve or such geometry definitions must be more substantially amended or modified. In absence of such support, alignment curves may be converted to IfcBSplineCurve or subtypes to obtain exact shape based on Cartesian coordinates, or IfcPolyline to obtain discretized shape reflecting intervals sufficient for construction.

To support this generic curve placement, in addition to supporting geometry taking advantage of curve functionality, the following entities proposed as part of the BuildingSmart "P6 Alignment" project are recommended to be adapted within a future extension:

IfcAlignment2DHorizontal: Make it a subtype of IfcBoundedCurve and move to Geometry resource schema. Or more drastically, to achieve uniformity, this entity and IfcAlignment2DHorizontalSegment could be deleted, and be replaced with IfcCompositeCurve. A new IfcCompositeCurveSegment subtype could be introduced that captures tag information.

IfcAlignment2DVertical: Make it a subtype of IfcBoundedCurve, move to Geometry resource schema, and add an attribute to explicitly reference the horizontal curve. This would make the definition encapsulated such that it could be used as a curve to define profile sweeping geometry for bridge girders or bridge slabs.

Table 15: IfcAlignment2DVertical attributes

Attribute	Type	Description
Segments	IfcAlignment2DVerticalSegment[]	Vertical segments
ReferenceCurve	**IfcBoundedCurve**	**The relative horizontal curve**

To handle specific patterns in a more discrete way (rather than relying on particular interpretations of curves or vectors at a custom representation), several additional definitions are proposed.

The initial documentation of such representation uses the corresponding representation identifier of the pattern source, and representation type 'Pattern'. The proposed entity IfcPatternItem is a subtype of IfcRepresentationItem. The IfcPatternItem is the inserted instance of a source definition. The instance is inserted by applying transformation defined in its PatternDefinition attribute. The PatternDefinition attributes is defined by the proposed entity IfcPattern. The

IfcPattern is a supertype of IfcLinearPattern and IfcRotationalPattern. The IfcPattern has Occurrence attribute which indicates the number of occurrences of instances within the pattern (at least 1). The attributes of IfcLinearPattern are Direction and Offset attributes. The IfcRotationalPattern contains RotationAxis and Angle. The origin of the pattern source is defined at the PatternOrigin attribute of IfcPatternItem. Patterns with multiple dimensions can be instantiated by referencing another IfcPatternItem instance.

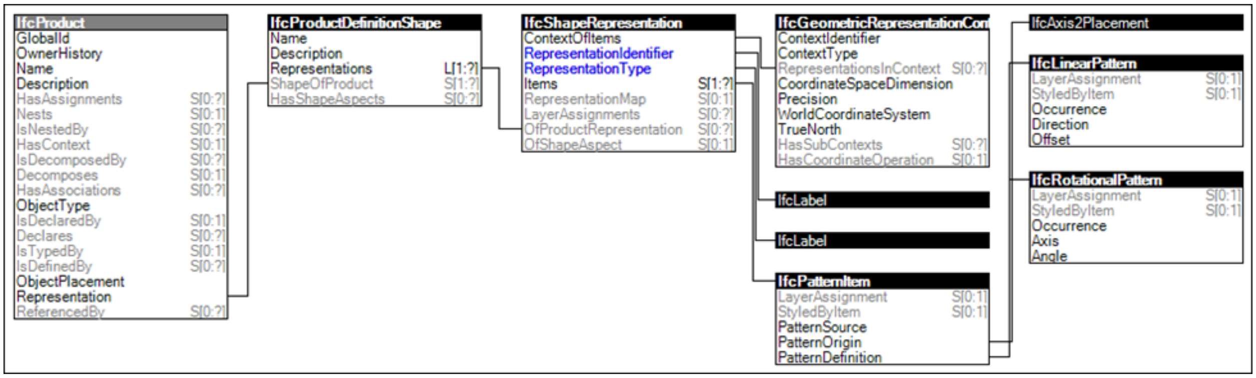


Figure 86: IfcPatternItem instance graph

6.4 Product Extensions

This section describes several high level "Product Extensions" proposed to be added to IFC for classification purposes. In product lifecycle information standards such as IFC, the term "product" refers to "physical or conceptual object that occurs in space", which includes both physical elements to be constructed, volumes of spaces, or any other arbitrary reference within space. An "element" is a subtype of product that refers to "tangible physical product that can be described by its shape representation, material representations, and other properties". A product is specialization of an "object", defined as "anything perceivable or conceivable that has a distinct existence, albeit not material". All such data definitions are described by "entities". These terms are further described within *Volume I* of this report.

Several chapters of buildingSmart International have proposed similar additions, while the proposed extensions within this document are kept intentionally minimal, as such classification does not impact the particular exchanges evaluated.

IfcBridge serves as a spatial structure in parallel to IfcBuilding and IfcSite. The bridge itself is described by parameters but not 3D geometry; geometry is defined on physical elements (IfcElement subtypes) related using IfcRelContainedInSpatialStructure.

Bridges may be classified by form or function. Omniclass (Construction Specifications Institute, 2016) provides related classification notations as shown in Table 28.

Table 16: Omniclass table for bridges

Notation	Description
11-00 00 00	Construction Entities by Function
11-51 65 00	Bridge
11-51 65 11	Vehicular Bridge
11-51 65 15	Rail Bridge
11-51 65 19	Pedestrian Bridge
12-00 00 00	Construction Entities by Form
12-14 14 00	Bridge
12-14 14 11	Trabeated Bridge
12-14 14 14	Arch Bridge
12-14 14 17	Truss Bridge
12-14 14 21	Cable-Stayed Bridge
12-14 14 24	Suspension Bridge
12-14 14 35	Pedestrian Bridge

Oddly, "Construction Entities by Form" includes "Pedestrian Bridge", which describes a function that may overlap with any of the indicated forms, and duplicates an entry of the same name under "Construction Entities by Function". Furthermore, it is very common for one bridge to carry multiple modes of transportation. We suspect that Omniclass needs further review and development to provide more meaningful distinctions for bridge structures. In the absence of specific usage requirements, Wikipedia provides specific types based on form, which are proposed to be used for general purposes.

The proposed IfcBridgeTypeEnum defines constants as shown in Table 29, as described on Wikipedia (Wikipedia, 2016):

Table 17: IfcBridgeTypeEnum constants

Constant	Description
BEAM	Supported by horizontal beams along span
TRUSS	Supported by trusses along span
CANTILEVER	Supported by beams cantilevered at each end
ARCH	Supported by abutments at each end
TIEDARCH	Suspended by arch superstructure
SUSPENSION	Suspended by cables along span
CABLESTAYED	Suspended by cables connected to towers at each end

Before establishing any fixed classification criteria, it is recommended that uses cases are applied that demonstrate specific need for such classification, such as for bridge inventory purposes or code requirements.

A bridge may carry roads, railroads, or other spatial structures (even buildings), where the relationship IfcRelReferencedInSpatialStructure links a bridge (RelatingStructure) to the spatial elements carried (RelatedStructures). A bridge itself has no concept of route connectivity; that is reserved for the roads or other routes it carries. A bridge is not necessarily limited to one route or one mode of transportation; it may carry roads shared by vehicles and bicycles, railways, and/or walkways in a single direction or opposite directions. A bridge may also carry intersections of roads, such as for the case of a Single Point Urban Interchange (SPUI).

For identification of major bridge components, IfcBridgeElement is introduced, derived from the existing IfcCivilElement. IfcBridgeElement aggregates other physical elements (IfcElement subtypes) into functions specific to bridges. For example, a bridge pier may consist of piles, footings, columns, and beams. IfcBridgeElement is intended to be a physical structure rather than a spatial structure, however it would typically not define its own geometry except for high-level parameters that drive the dimensions of components.

IfcBridgeElementTypeEnum defines constants as shown in Table 30.

Table 18: IfcBridgeElementTypeEnum constants

Constant	Description
ABUTMENT	Supporting structure at end of bridge consisting of walls, footings, piles
PIER	Supports at intermediate locations consisting of columns, footings, piles
CHANNEL	Structure to control water flow below the bridge
CULVERT	Structure to restrict water flow under bridge

While such classification elements do not impact detailing of bridges themselves, they provide a mechanism to organize information in a way that is compatible with typical bridge plans, such that information may be rolled up into a particular component when extracting plans, and searched later based on categorized information. Such differentiation also provides a common way for software to show/hide or lock/unlock elements, as commonly done within workflows that are specific to subsets – for example, while constructing the foundations of bridges, contractors may find it useful to turn off all superstructure elements for ease of navigation or to reduce the time/cost downloading very detailed information (e.g. bridges used for test cases used 50MB in entirety).

6.5 Shared Physical Elements

Certain components of bridges can be modelled using existing IFC definitions, where the relationship IfcRelAggregates is used:

- IfcWall captures vertical planar segments
- IfcSlab captures horizontal planar segments.
- IfcColumn captures vertical linear segments with axial loading.
- IfcBeam captures horizontal linear segments with shear loading.
- IfcPile captures vertical linear segments distributing loads into the ground.
- IfcFooting captures horizontal linear segments distributing loads into the ground.
- IfcRailing captures guard rails and ornamental railings.
- IfcMember captures webbing between beams, and components of cross-framing.
- IfcPlate captures gusset plates used to connect components of cross-framing.

Assemblies of components can be modelled using the existing IfcElementAssembly entity, where predefined types are used as follows:

- BRACED_FRAME captures cross-framing (IfcMember, IfcPlate) between girders.
- GIRDER captures girders consisting of IfcBeam segments.
- SLAB_FIELD captures bridge decks consisting of IfcSlab segments.

Assembled ground structures such as piers and abutments are modelled using IfcCivilElement.

No new elements are proposed for physical elements. However, several assumptions about usage are proposed to be changed.

To support shape modifications such as shear studs, as well as integrated reinforcing, it is proposed that geometry may be supported at parent elements AND at aggregated elements where any overlap is implied to be a subtraction from the parent element.

To support bridge decks consisting of slabs defined by profiles, material profile sets are more suitable than material layer sets. Similarly, for retaining walls that interconnect with guard rails or otherwise have cross-section details, material profile sets are more suitable than material layer sets. Rather than IfcSlab or IfcWall documenting usage of IfcMaterialProfileSet, it is proposed that any element in IFC support any method of applying material parameters, with an exception for “standard case” subtypes such as IfcWallStandardCase which are specifically documented.

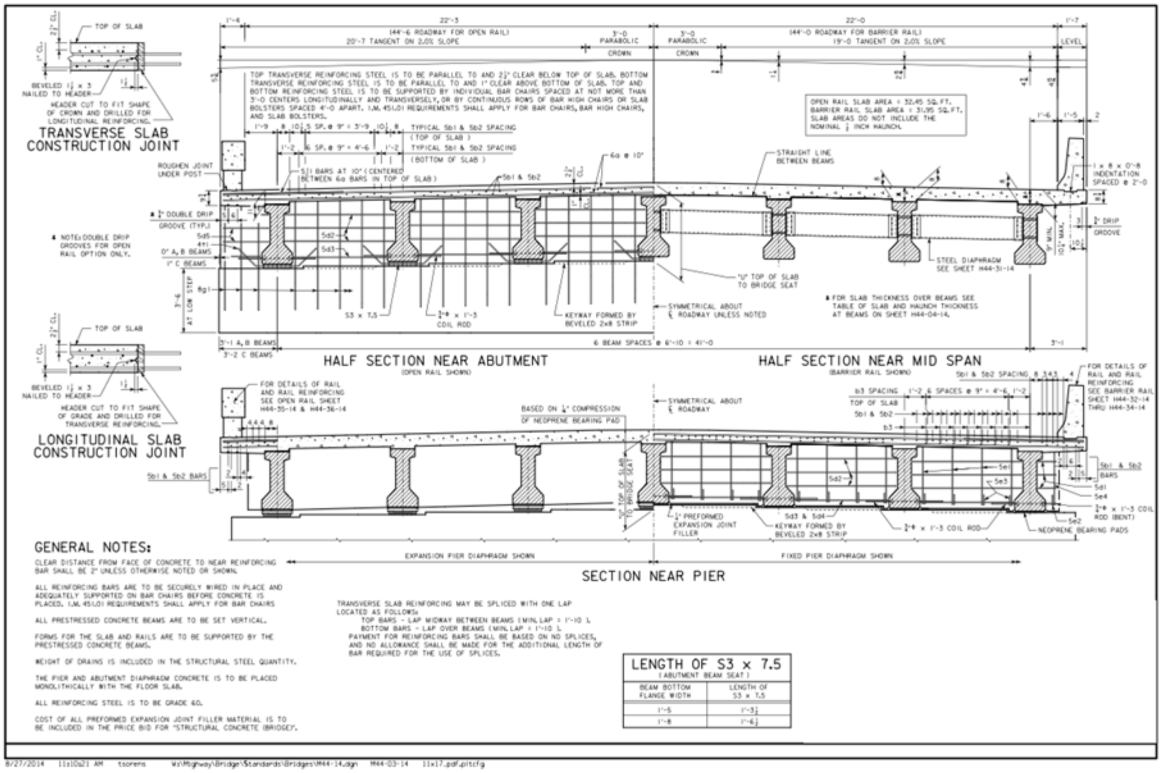


Figure 87: Bridge cross section

In the above illustration, entities are used as follows:

Beams (IfcBeam) have their cross-section defined with IfcMaterialProfileSet and are placed with Axis point up, and Placement offset in the X and Z directions. This is consistent with beam usage in buildings. Connected series of beams (forming a girder) are defined with IfcElementAssembly having PredefinedType=GIRDER.

Slabs (IfcSlab) have their cross-section defined with IfcMaterialProfileSet and are placed with Axis point up, and Placement set at 0 relative to the alignment curve. This diverges from slab usage in buildings, which are typically of uniform thickness, flat (no slope), and fill a closed boundary. Connected series of slabs (forming a deck) are defined with IfcElementAssembly having PredefinedType=SLAB_FIELD.

6.6 Plumbing Domain

Drainage systems for bridges can be modelled using existing IFC definitions, where the relationship IfcRelContainedInSpatialStructure is used:

- IfcWasteTerminal captures drainage inlets typically embedded at outer lower points along bridge decks above piers.
- IfcPipeSegment captures pipe segments
- IfcPipeFitting captures pipe transitions
- IfcValve captures cleanout plugs

6.7 Electrical Domain

Electrical systems for bridges can be modelled using existing IFC definitions, where the relationship IfcRelContainedInSpatialStructure is used:

- IfcCableCarrierSegment captures conduit
- IfcCableSegment captures wiring within conduit
- IfcLightFixtures captures lighting
- IfcLamp captures bulbs
- IfcJunctionBox captures boxes where wiring may be accessed

6.8 Structural Elements Domain

Shear studs can be modelled as a predefined type of IfcProjectionElement, attached to IfcBeam using IfcRelProjectsElement. Such projection relationship is used to indicate that the projected aspects are treated as voids within the connecting element(s) such as the bridge deck.

The newly introduced 'Pattern' representation is used for IfcReinforcingBar to support repetitive positioning of longitudinal bars and stirrups.

The positioning of longitudinal bars along stirrups may be achieved based on the IfcRelConnects relationship with additional semantics implied where RelatingElement provides the boundary for the RelatedElement.

Specific examples of rebar layouts are detailed in *Volume III*.

6.9 Shared Transportation Elements

While the purpose of this project is to capture bridge information, bridges are often identified according to the routes they carry and the routes they intersect (e.g. National Bridge Inventory), and the design is driven by parameters relating to such routes – for example, design speed and minimum clearance may influence the super-elevation of bridge ramps at an interchange. However, such parameters are specific to intended uses, and not of bridges or alignments themselves.

The information in this section should be considered as very secondary – not critical to construction of bridges, and not essential to be adopted. However it is documented here to provide a conceptual framework for how the IFC model could be cohesively extended to capture higher level traffic flow and route connectivity information.

Segments of roads, rails, or other elements designed to carry traffic are modelled using the abstract types IfcTransportationSegment and IfcTransportationJunction, each of which may have ports to indicate directional connectivity (IfcTransportationPort), and may participate in systems using IfcTransportationSystem. These entities are subtyped into IfcRoadSegment/IfcRoadJunction and IfcRailSegment/IfcRailJunction. Such modeling is in parallel to IfcFlowSegment/IfcFlowFitting for which subtypes are defined for IfcPipeSegment, IfcDuctSegment, IfcCableSegment, and corresponding fitting types.

To link indicate road linkages, for purposes of identifying route connectivity, the IFC port concept is extended. In IFC4, the abstract IfcPort object has one subtype, IfcDistributionPort which is used to model the flow of utility services through pipes, ducts, cables, or conduit. The concept for modeling traffic flow is similar, and warrants the introduction of IfcTransportationPort. A tranportation port indicates connectivity between roads or junctions. Similar to distribution ports, transportation ports may also have properties indicating flow, such as traffic counts and traffic speeds. Such information may also be captured as time series at scheduled intervals (IfcPerformanceHistory), similarly as is done for distribution systems.

IfcTransportationPort contains attributes as shown in Table 31.

Table 19: IfcTransportationPort attributes

Attribute	Type	Description
FlowDirection	IfcFlowDirectionEnum	Direction of traffic
PredefinedType	IfcTransportationPortTypeEnum	Type of port connection
SystemType	IfcTransportationSystemTypeEnum	Transportation system

IfcTranportationPortTypeEnum contains constants as shown in Table 32.

Table 20: IfcTransportationPortTypeEnum constants

Constant	Description
Roadway	Connection to a roadway
Railway	Connection to a railway
Pathway	Connection to a pathway

The entity IfcTransportationSystem is introduced, deriving from IfcSystem. It has a corresponding type enumeration IfcTransportationSystemType defining constants as shown in Table 33.

Table 21: IfcTransportationSystemTypeEnum constants

Constant	Description
Pedestrian	Carries vehicular traffic
Bicycle	Carries bicycles
Automobile	Carries vehicles
Rail	Carries trains
Water	Carries boats

Note that the transportation system type implies modes of transportation, where multiple modes may share the same components – for example, road segments may be designed to support both vehicles and bicycles. This is similar to how pipe segments (IfcPipeSegment) may be designed to carry different types of fluids or gases (IfcDistributionSystemType.DOMESTICCOLDWATER), however only one for a particular network, whereas multiple networks may share the same elements (i.e. mixed use).

Merge behavior at road junctions may be linked to IfcTrafficSignal using IfcRelTrafficControl (in parallel to IfcRelFlowControl), where IfcTrafficSignalTypeEnum contains constants as shown in Table 34.

Table 22: IfcTrafficSignalTypeEnum constants

Constant	Description
Open	Free-flowing
Merge	Traffic must merge with other traffic
Yield	Traffic must yield to other traffic
Stop	Traffic must stop before proceeding
Signal	Traffic must continue or stop according to signal

6.10 Bridge Systems

In parallel with buildings having systems of components (IfcBuildingSystem, IfcDistributionSystem), bridges also have systems defined, which organize related elements for purposes of group analysis.

The entity IfcBridgeSystem is introduced, deriving from IfcSystem. It has a corresponding type enumeration IfcBridgeSystemTypeEnum defining constants as shown in Table 35.

Table 23: IfcBridgeSystemTypeEnum constants

Constant	Description
Deck	Supported slabs
Superstructure	Girders and bearings
Substructure	Columns and abutments

Such differentiation is driven by FHWA MAP-21 reporting requirements, where this breakdown corresponds to reporting of bridge conditions. In composite element scenarios, it is possible for one element to belong to multiple systems; thus a system membership is used instead of element composition.

Property sets may be defined specific to each bridge system overall, as well as for specific subtypes.

As with any other IfcSystem type, elements of a bridge are assigned to the IfcBridgeSystem using the IfcRelAssignsToGroup relationship.

For modeling transportation networks, which comprise roads, railroads, water channels, or similar, an object is required to relate such information.

6.11 Structural Analysis Extensions

To support dynamic loads, a subtype of IfcStructuralLoadConfiguration is introduced to simulate loads evaluated over a range of positions. For bridges, AASHTO defines such load cases. For analysis purposes, this is the equivalent of evaluating the load at every position between the starting and ending point; or pragmatically by iteratively selecting positions to find local points resulting in maximum stresses.

The IfcStructuralLoadDynamic entity derives from IfcStructuralLoad and is defined as shown in Table 36.

Table 24: IfcStructuralLoadDynamic attributes

Attribute	Type	Description
Load	IfcStructuralLoad	The load to evaluate across the range
Start	IfcCartesianPoint	2D point on surface for which to start
End	IfcCartesianPoint	2D point on surface for which to end

To support volumetric members such as bridge abutments, the new IfcStructuralVolumeMember is proposed, deriving from IfcStructuralMember. This entity defines a volumetric mass, where analysis is performed by converting the mass into a 3D mesh of linear members.

The IfcStructuralVolumeMember entity derives from IfcStructuralMember and is defined as shown in Table 37:

Table 25: IfcStructuralVolumeMember attributes

Attribute	Type	Description
PredefinedType	IfcStructuralVolumeMemberTypeEnum	Indicates how derivative mesh should be built

The IfcStructuralVolumeMemberTypeEnum is defined as shown in Table 38.

Table 26: IfcStructuralVolumeMemberTypeEnum constants

Constant	Description
Membrane	Mass should be analyzed as grid interconnections (no diagonal members)
Shell	Mass should be analyzed as diagonal interconnections

Bibliography

Amann, J., Borrmann, A., Hegemann, F., Jubierre, J. R., Flurl, M., Kock, C., & Konig, M. (2013). *A refined product model for shield tunnels based on a generalized approach for alignment representation.* Retrieved from Technical University of Munich: http://www.cms.bgu.tum.de/publications/Amann_2013_ICCBEI.pdf

Bentley Systems. (2015). *iModel*. Retrieved from Bentley Systems: http://www.bentley.com/en-US/Products/imodel/

BuildingSmart International. (2015, 7 29). *IFC Alignment.* Retrieved from BuildingSmart: http://www.buildingsmart-tech.org/infrastructure/projects/alignment

BuildingSmart International. (2016, 1 10). *All Applications by Category*. Retrieved from BuildingSmart Tech: http://www.buildingsmart-tech.org/implementation/implementations

buildingSmart International. (2016, 1 10). *Technical Vision*. Retrieved from BuildingSmart: http://www.buildingsmart.org/standards/technical-vision/

Chen, S. S. (2013). *Model View Definitions for Highway Bridge Interoperable Data Protocols.* Buffalo: State University of New York at Buffalo.

Choi, W. (2016, 1 10). *Infra BIM Project in Korea*. Retrieved from BuildingSmart: http://iug.buildingsmart.org/resources/itm-and-iug-meetings-2013-munich/infra-room/infra-bim-project-in-korea/view

Claude Dumoulin, P. B. (2016, 1 10). *MINnD Use Case 3 - IFC-Bridge.* Retrieved from BuildingSmart: http://iug.buildingsmart.org/resources/itm-and-iug-meetings-2014-stockholm/infra-room-meeting/minnd-use-case-nb3-ifcbridge

CSTB. (2016, 1 10). *eveBIM.* Retrieved from CSTB: http://editions.cstb.fr/Products/eveBIM-Viewer

Ferries, B. (2001, 12 18). *Contribution to an IFC Extension in the Bridge Domain.* Retrieved from Laurenti: http://www.laurenti.com/publis/Bridge2001.pdf

International Standards Organization. (2016, 1 10). *Benefits of International Standards*. Retrieved from iso.org: http://www.iso.org/iso/home/standards/benefitsofstandards.htm

ISO. (2013, 4 1). *ISO 16739:2013.* Retrieved from ISO: http://www.iso.org/iso/catalogue_detail.htm?csnumber=51622

LandXML.org Industry Consortium. (2015, 7 1). *LandXML*. Retrieved from LandXML: http://landxml.org/

Moon, D. H. (2016, 1 10). *buildingSmart Singapore Standards Summit.* Retrieved from Dropbox: https://www.dropbox.com/home/bSI%20Singapore%20Standards%20Summit/Room%20 documents%20(Infrastructure)/13%20Oct?preview=IR2+Moon+-+IfcRoad+Extension+with+IDM.pdf

Red Equation Corporation. (2016, 1 10). *OpenBRIM*. Retrieved from OpenBRIM: https://openbrim.org/www/brim/

Technical University of Munchen. (2016, 1 10). *TUM Open Infra Platform.* Retrieved from Technical University of Munchen: https://www.cms.bgu.tum.de/de/forschung/projekte/31-forschung/projekte/397-tum-open-infra-platform

Tekla. (2014, 3 4). *DSTV file description*. Retrieved from Tekla: http://teklastructures.support.tekla.com/200/en/int_nc_dstv_file_description

U.S. Federal Highway Administration. (2016). *IFC Bridge Design-to-Construction Information Exchange (U.S.)*. Retrieved from National Institute of Building Sciences: http://www.nibs.org/?page=bsa_bridge

Wikipedia. (2016, 1 10). *Bridge*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Bridge

Yabuki, N. (2015, 10 14). *Integration Issues: Tunnels.* Retrieved from DropBox: https://www.dropbox.com/home/bSI%20Singapore%20Standards%20Summit/Room%20documents%20(Infrastructure)/14%20Oct?preview=IR6+Yabuki+-+Integration+Tunnels.pdf