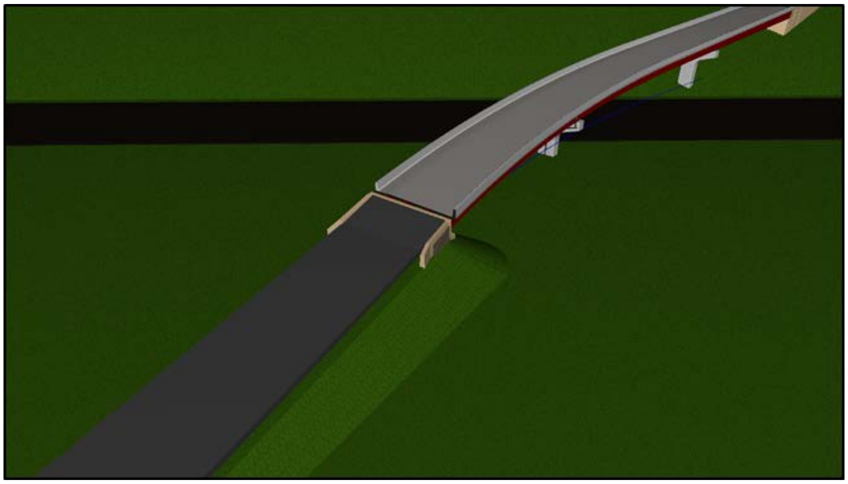
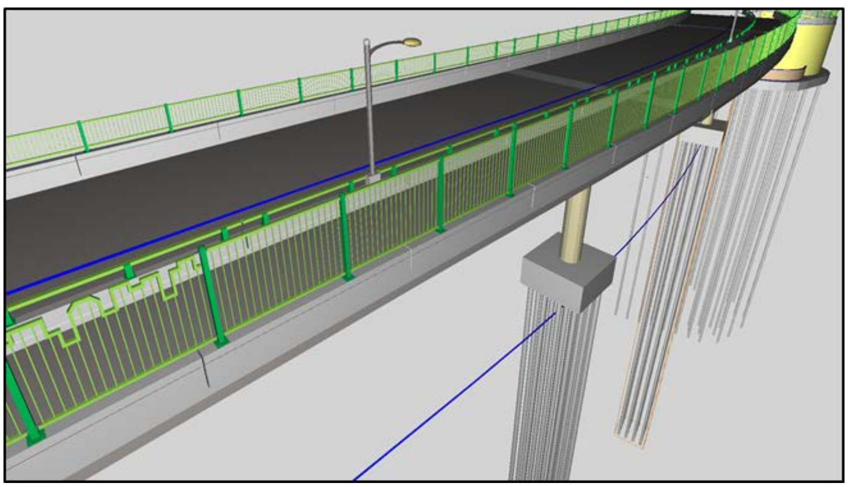
Bridge Information Model Standardization
VOLUME III: COMPONENT MODELING

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

This document is disseminated under the sponsorship of the U.S. Department of Transportation in the interest of information exchange. The U.S. Government assumes no liability for the use of the information contained in this document.

The U.S. Government does not endorse products or manufacturers. Trademarks or manufacturers' names appear in this Report only because they are considered essential to the objective of the document.

Quality Assurance Statement

The Federal Highway Administration (FHWA) provides high-quality information to serve Government, industry, and the public in a manner that promotes public understanding. Standards and policies are used to ensure and maximize the quality, objectivity, utility, and integrity of its information. FHWA periodically reviews quality issues and adjusts its programs and processes to ensure continuous quality improvement.

Technical Report Documentation Page

1. Report No. **FHWA-HIF-16-011**	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Bridge Information Modeling Standardization Report Volume III – Component Modeling		5. Report Date 2016-04 6. Performing Organization Code
7. Author(s) Primary - Tim Chipman Contributing – Donghoon Yang Editor – Roger Grant		8. Performing Organization Report No.
9. Performing Organization Name and Address National Institute of Building Sciences 1090 Vermont Avenue, NW, 7th Floor Washington, DC 20005		10. Work Unit No. 11. Contract or Grant No. DTFH6114C00047
12. Sponsoring Agency Name and Address Office of Bridges and Structures Federal Highway Administration 6300 Georgetown Pike McLean, VA 22101-2296		13. Type of Report and Period Covered 14. Sponsoring Agency Code

15. Supplementary Notes
This document is the fourth of a multi-volume set of documents on Bridge Information Modeling Standardization. The volumes can be read individually or sequentially as part of the set. Reading the Introduction first is recommended to provide context and a summary of the work and its findings.

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 an comprehensive exchange specification to assist software developers to implement the recommended alternative to the benefit of bridge owners. This Volume, Component Modeling, the fourth of four volumes, describes the modeling of specific components of bridges to the level of detail as conveyed on design contract plans, using two real-world case studies and three candidate schemas, to identify methodologies that would be valuable in a national bridge information modeling standard and to evaluate the viability of one of those schemas, buildingSMART Industry Foundation Class (IFC), as a candidate for a bridge modeling standard.

17. Key Words bridge, design, construction, modeling, models, open, standards,	18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22161.

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages - 101	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")
		TEMPERATURE (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
		TEMPERATURE (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

1. Component Modeling 1
1.1 Alignment 5
1.1.1 Bentley OpenBridge 8
1.1.2 Industry Foundation Classes 10
1.1.3 OpenBrIM 10
1.2 Abutments 11
1.2.1 Bentley OpenBridge 11
1.2.2 Industry Foundation Classes 12
1.2.3 OpenBrIM 21
1.3 Piers 22
1.3.1 Bentley OpenBridge 22
1.3.2 Industry Foundation Classes 22
1.3.3 OpenBrim 28
1.4 Bearings 29
1.4.1 Bentley OpenBridge 29
1.4.2 Industry Foundation Classes 29
1.4.3 OpenBrIM 32
1.5 Girders 33
1.5.1 Bentley OpenBridge 33
1.5.2 Industry Foundation Classes 33
1.5.3 OpenBrIM 39
1.6 Framing 40
1.6.1 Bentley OpenBridge 40
1.6.2 Industry Foundation Classes 43
1.6.3 OpenBrIM 44
1.7 Deck 45
1.7.1 Bentley OpenBridge 46
1.7.2 Industry Foundation Classes 48
1.7.3 OpenBrIM 57
1.8 Barriers 58
1.8.1 Bentley OpenBridge 58
1.8.2 Industry Foundation Classes 59
1.8.3 OpenBrIM 66
1.9 Lighting 67
1.9.1 Bentley OpenBridge 67
1.9.2 Industry Foundation Classes 67
1.9.3 OpenBrIM 68

1.10 Sitework ... 69
1.10.1 Bentley OpenBridge ... 69
1.10.2 Industry Foundation Classes ... 70
1.10.3 OpenBrIM ... 71
1.11 Soil Borings ... 72
1.11.1 Bentley OpenBridge ... 72
1.11.2 Industry Foundation Classes ... 72
1.11.3 OpenBrIM ... 73
1.12 Quantities ... 74
1.12.1 Bentley OpenBridge ... 74
1.12.2 Industry Foundation Classes ... 74
1.12.3 OpenBrIM ... 74
1.13 Construction Staging ... 75
1.13.1 Bentley OpenBridge ... 75
1.13.2 Industry Foundation Classes ... 76
1.13.3 OpenBrIM ... 76
1.14 Structural Analysis ... 77
1.14.1 Bentley OpenBridge ... 78
1.14.2 Industry Foundation Classes ... 80
1.14.3 OpenBrIM ... 81
2 Implementation ... 82
2.1 Data Formats ... 82
2.2 Correlation of Plan Model to Integrated Data Model ... 83
2.2.1 Penn Turnpike Bridge ... 83
2.2.2 Van White Bridge ... 84
3 Findings ... 87
3.1 Data Modeling ... 87
3.2 Vendor Feedback ... 88
3.3 Verification ... 88
3.4 Information Standards ... 89

Appendix C:1 IFC-XML Format Sample ... 90
Appendix C:2 IFC-SPF Format Sample ... 94

List of Figures

Figure 1 - Penn Turnpike bridge rendering 2
Figure 2 – Penn Turnpike bridge plan and elevation 2
Figure 3 –Van White bridge rendering 3
Figure 4 - Van White bridge plan and elevation 3
Figure 5 - Horizontal Alignment Plan 5
Figure 6 - Alignment Rendering 6
Figure 7 – Penn Turnpike Horizontal Alignment Parameters 6
Figure 8 - Van White Horizontal Alignment Parameters 7
Figure 9 – Penn Turnpike Vertical Alignment Parameters 7
Figure 10 - Van White Vertical Alignment Parameters 8
Figure 11- Horizontal Alignment Parameters for Bentley 9
Figure 12 - Vertical Alignment Parameters for Bentley 9
Figure 13 - Creating Abutment for Bentley 12
Figure 14 – Penn Turnpike Abutment 1 Components 13
Figure 15 – Penn Turnpike Abutment 1 Modeling 14
Figure 16 – Penn Turnpike Wall Parameters of Abutment 1 14
Figure 17 – Penn Turnpike Abutment 1 Foundation Plan 15
Figure 18 – Van White Abutment 1 Reinforcement Plan 16
Figure 19 – Van White Abutment 1 Elevation 16
Figure 20 – Penn Turnpike Abutment 2 17
Figure 21 - Van White south abutment model 18
Figure 22 - Van White abutment rebar definition 19
Figure 23 - Van White abutment rebar mapping 19
Figure 24 - Van White north abutment model 20
Figure 25 - Pier definition with Bentley 22
Figure 26 - Pier components 23
Figure 27 - Penn Turnpike Pier 1 model 24
Figure 28 - Penn Turnpike Pier 2 model 25
Figure 29 - Van White pier model 26
Figure 30 - Penn Turnpike bearing plan 29
Figure 31 - Penn Turnpike bearing definition model 30

Figure 32 - Penn Turnpike bearing schedules ... 30
Figure 33 – Penn Turnpike Bearing occurrences model ... 31
Figure 34 - Girder for Bentley ... 33
Figure 35 – Penn Turnpike girder model ... 34
Figure 36 - Penn Turnpike Girder 1 framing plan ... 35
Figure 37 - Girder parameters ... 35
Figure 38 - Table column mapping ... 36
Figure 39 - Van White girder definition model ... 37
Figure 40 - Van White girder transformation model ... 38
Figure 41 - Cross frame with Bentley ... 40
Figure 42 - Member section with Bentley ... 41
Figure 43 - Framing working points with Bentley ... 42
Figure 44 - Penn Turnpike framing model ... 44
Figure 45 - Deck plan example ... 45
Figure 46 - Deck for Bentley ... 46
Figure 47 - Deck slabs for Bentley ... 47
Figure 48 - Penn Turnpike deck profile ... 48
Figure 49 - Deck dimension table ... 49
Figure 49 - Penn Turnpike deck model ... 50
Figure 50 - Van White deck profile ... 51
Figure 51 - Van White deck model ... 52
Figure 52 - Van White pier cap geometry values ... 53
Figure 53 - Van White pier cap model ... 54
Figure 54 - Van White rebar definition ... 56
Figure 55 - Van White diaphragm model ... 57
Figure 56 - Barriers with Bentley ... 58
Figure 57 - Penn Turnpike barrier model ... 59
Figure 58 - Penn Turnpike barrier profile ... 60
Figure 59 - Van White railings for walkway ... 61
Figure 60 - Van White railings for divider ... 62
Figure 61 - Van White railing post for divider ... 63
Figure 62 - Van White railing post for walkway ... 63
Figure 63 - Van White railing rails straight ... 64
Figure 64 - Van White railing rails decorative ... 64

Figure 65 - Van White railing spindles 65
Figure 66 –Van White railing chain transform 65
Figure 67 - Van White railing warp transform 66
Figure 68 - Van White lighting 68
Figure 69 - Sitework with Bentley 69
Figure 70 - Penn Turnpike terrain 70
Figure 71 - Soil layers 72
Figure 72 - Quantity schedule 74
Figure 73 - Sequencing with Bentley 75
Figure 74 - Sequencing tasks 76
Figure 75 - Structural analysis girder moments 77
Figure 76 - Structural support condition with Bentley 78
Figure 77 - Structural loads with Bentley 79
Figure 78 - Structural load cases with Bentley 80
Figure 79 - Structural boundary condition 80
Figure 80 - Structural load 81
Figure 81 - Van White bridge loaded into Tekla Structures 89

List of Tables

Table 1 - Van White deck geometry 51
Table 2 - Van White rebar patterns 55
Table 3 - Files for test cases 82
Table 4 - Penn Turnpike plan contents 83
Table 5 - Van White plan contents 84

1. Component Modeling

This Volume of the Report describes the modeling of specific components of bridges to the level of detail as conveyed on design contract plans, using two real-world case studies. Information contained within accompanying construction specifications (special provisions) is not captured as part of this scope; as such information is primarily textual, it may be incorporated by reference. Information contained in shop drawings is considered to be part of a different exchange and is not covered here. Three candidate schemas are applied in this effort to identify methodologies that would be valuable in a national bridge information modeling standard.

To ensure that details are captured that reflect common usage and reflect the level of information presented in design plans, two specific case study bridges were chosen for illustration deemed to be representative of highway bridges in the U.S., yet containing certain complexities that stretch the limits of the proposed information model.

The first bridge evaluated is Pennsylvania Turnpike Bridge MF-145 - Ramp 1195N over SR 51. This bridge follows a horizontal alignment consisting of circular and straight sections at a constant vertical slope, with varying super-elevation and varying cross-section. It is a 3-span continuous curved steel bridge with spans of 130’-180’-130’. It consists of steel I-girder framing, with reinforced concrete abutments, piers, and decking.

The second bridge evaluated is the Van White Memorial Overpass in Minneapolis, MN. This bridge follows a horizontal alignment consisting of circular and straight sections with a parabolic vertical curve, with varying super-elevation and constant cross section. It consists of a reinforced concrete box girder, abutments, and piers. As this bridge is situated in an urban area, it consists of decorative railings, walkways, and lighting, and makes use of geometry consisting of curved surfaces that cannot be described by polygons alone but requires B-Spline surfaces and Constructive Solid Geometry (CSG).

While it would certainly be preferable to model additional bridges, for this exercise it was deemed critical to first go through the process of modeling a limited set of specific bridges in the same detail as described in design plans before attempting to accommodate additional bridges at a lower level of detail. As with any such modeling effort, the Pareto principle applies, where the last 20% takes 80% of the time: the initial layout of the bridge deck, girders, and piers ended up being rather trivial (i.e. several days effort) compared to capturing the more detailed aspects found in the plans such as rebar, architectural railings, electrical (i.e. multiple weeks effort).

There are various other scenarios that may be encountered on other bridges which are not captured by this specific bridge, such as diverging roadways and interchanges.

Details are provided indicating how each component was modeled in the schemas evaluated. For such detail, familiarity with the schemas as described in Volume II may help in understanding.

Figure 1 illustrates a 3D rendering of the sample Penn Turnpike steel bridge. Figure 2 shows the first page of the plans, representing the overall alignment and positioning of major components.

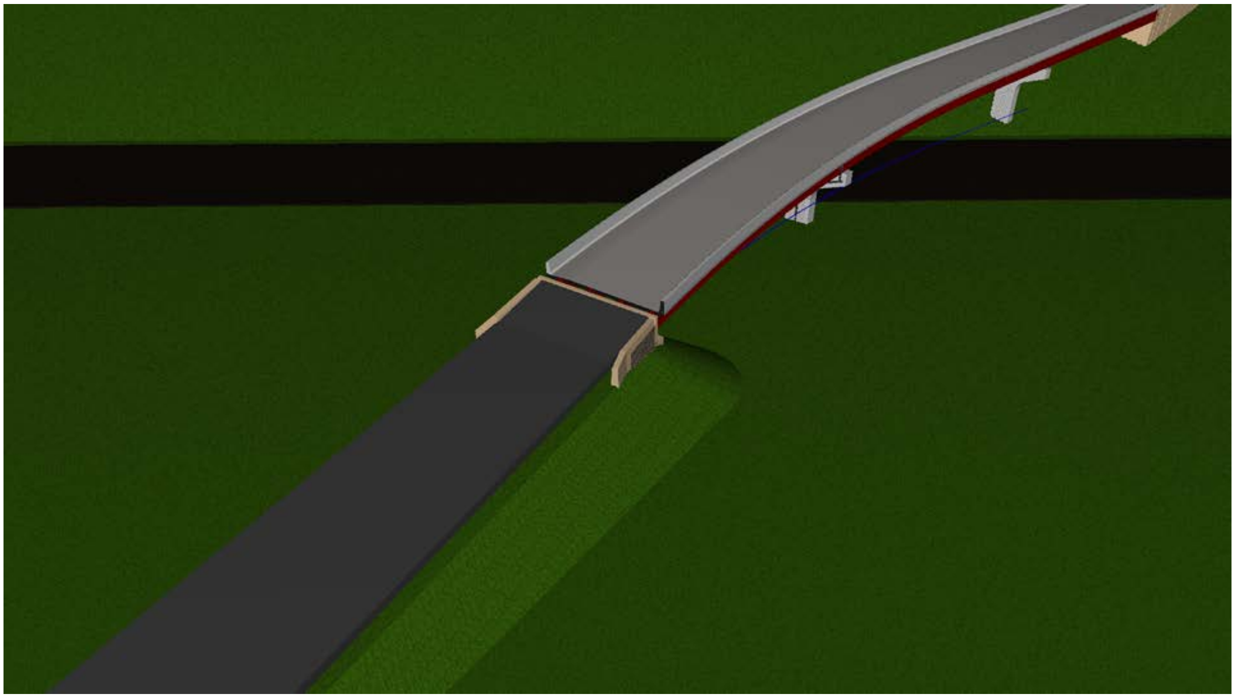


Figure 1 - Penn Turnpike bridge rendering

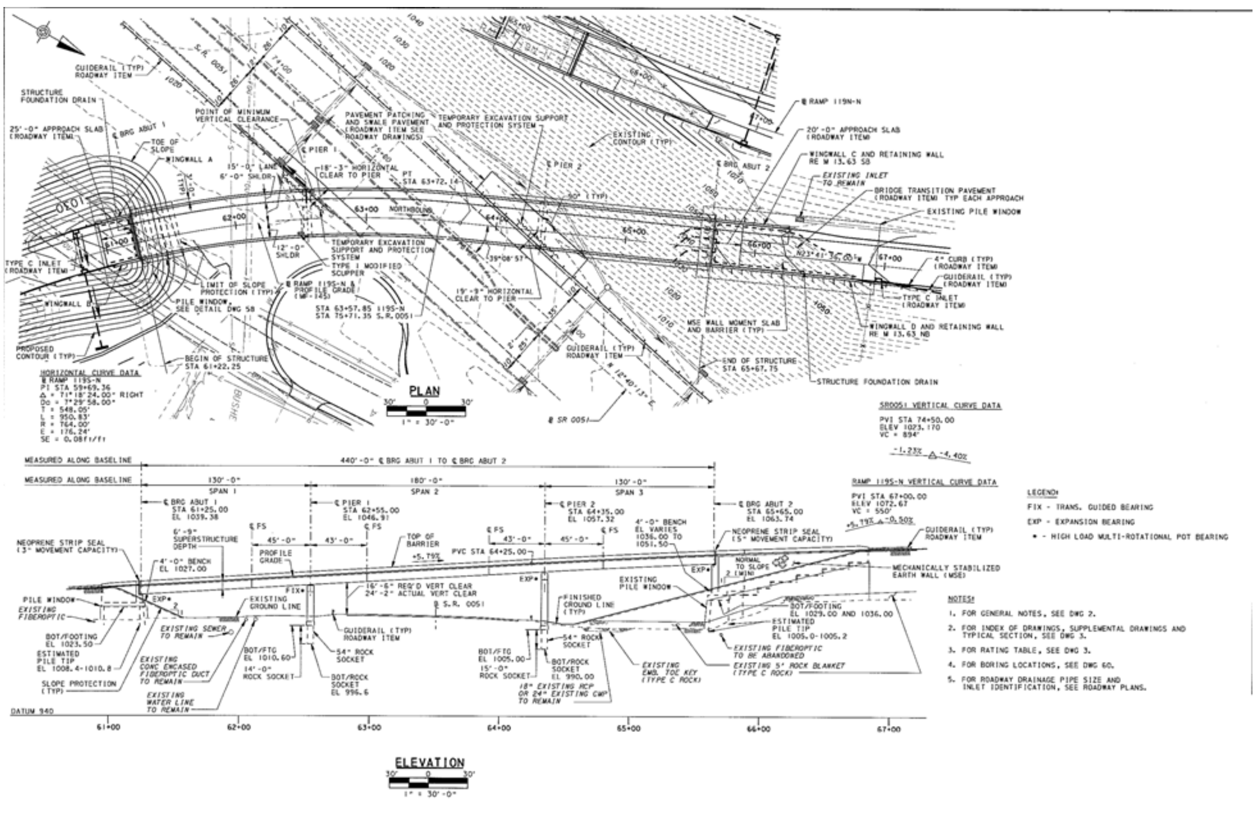


Figure 2 – Penn Turnpike bridge plan and elevation

Figure 3 illustrates a 3D rendering of the Van White sample bridge. Figure 4 shows the first page of the plans indicating the overall alignment of this bridge.

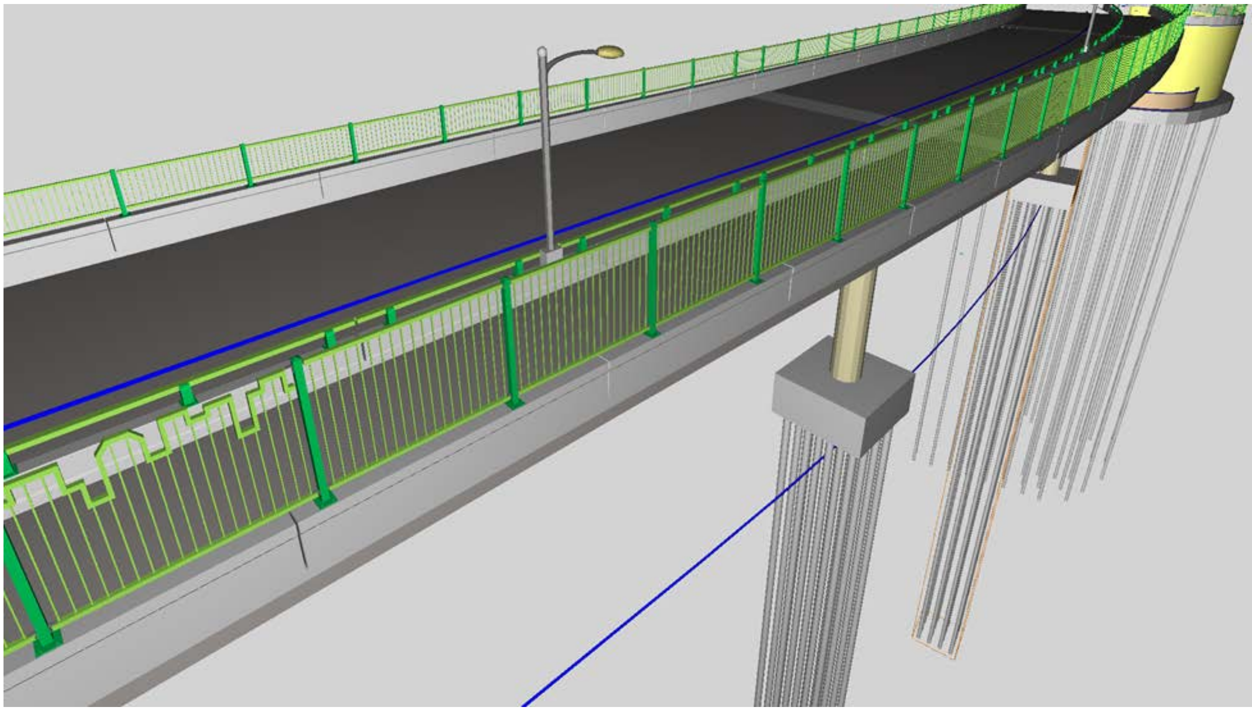


Figure 3 –Van White bridge rendering

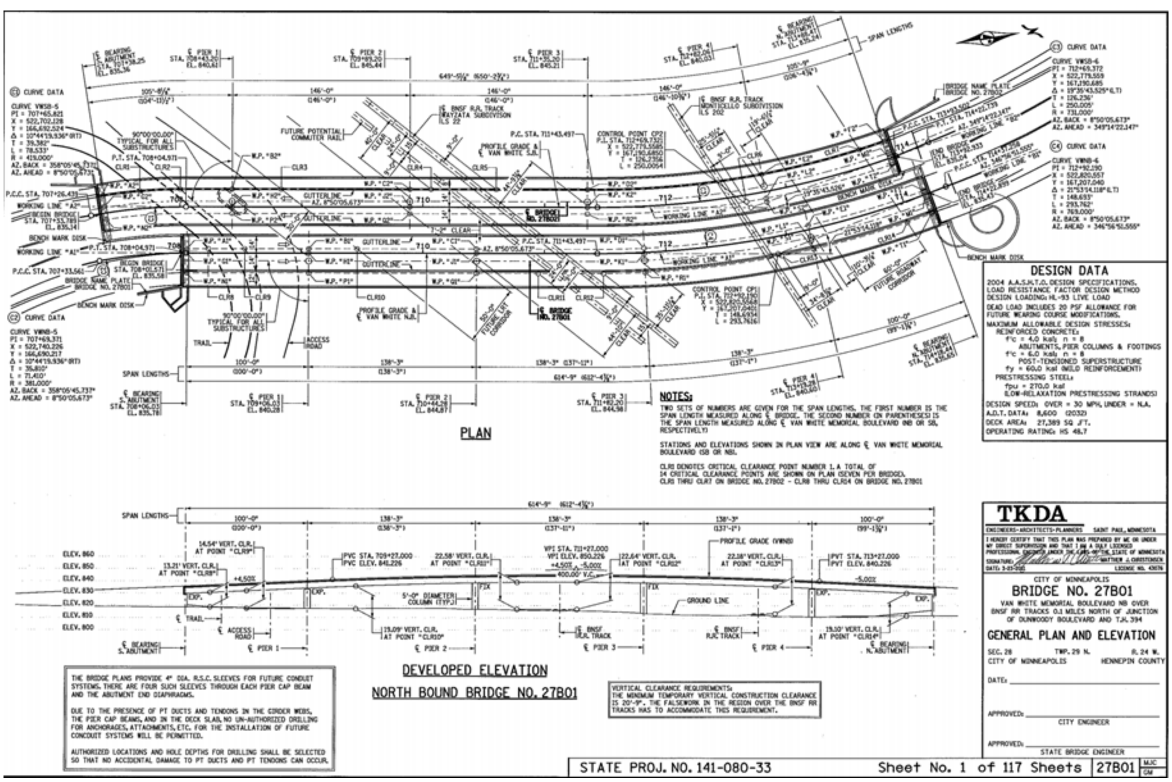


Figure 4 - Van White bridge plan and elevation

For each of the components modeled, approaches are detailed for each of the schemas evaluated. For Bentley OpenBridge/iModel, as the specification had not been published at the time of this review (Bentley indicates it will be available in 2016), Bentley software applications are presented instead, most notably Bentley LEAP Bridge Steel V8i release that was digitally signed February 23, 2015. This application captures design parameters for constructing an initial model that may be later detailed in downstream products such as Bentley RC-Pier and Bentley Microstation.

While applications should not be conflated with data formats, such detail is provided based on the assumption that the resulting data format would likely carry the data presented in this application, and that including this information may be informative to readers having experience with Bentley software. Many caveats should be noted – just because a software application does not support a particular feature does not necessarily mean that the underlying data format does not support carrying the relevant data, and vice-versa; thus, where Bentley software is shown, it is not possible to make any definitive statements on what would or would not be supported by any future interoperability specification. The intention is to illustrate to readers familiar with Bentley software, how such software could foreseeably work in the context of generating output data formats described.

The analysis presented in this Volume is underpinned by and in part derived from a fully detailed Model View Definition which contains a detailed analysis of data structures, schema extensions, process workflows, and testing of sample data for use by software developers. It is provided within separate documentation referred to as the ***IFC Bridge Design to Construction Information Exchange (U.S.)*** available from the Federal Highway Administration and National Institute of Building Sciences (NIBS)

1.1 Alignment

Horizontal and vertical alignment curves provide the underlying placement for all components in a bridge plan. For terminology, some software applications (e.g. Bentley LEAP Bridge) refer to Horizontal Alignment as simply “Alignment”, and Vertical Alignment as “Profile”, whereas this document and referenced standards use the term “Profile” to mean any arbitrary cross-section that may be applied in any direction. The deck surface cross-section is considered to be a “Profile” which is not part of the alignment itself but may be used in conjunction with the alignment to derive the overall 3D shape of the bridge deck. For all terms used within this document, refer to the Terms section of Volume I of this Report for the specific meanings.

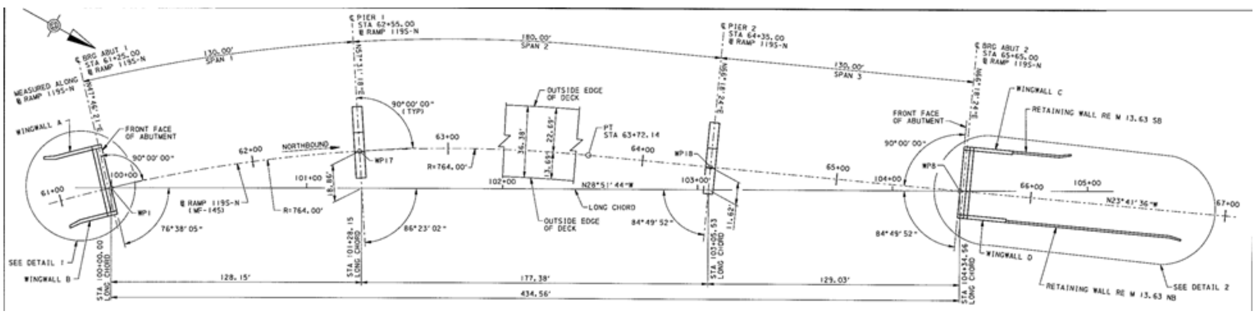


Figure 5 - Horizontal Alignment Plan

In this example in Figure 5, the origin of the horizontal alignment curve is considered to be the first station, having notation “61+25.00” as commonly used in the U.S. for indicating 6125.0 feet as a reference offset location. The particular placement referencing used results in both abutments lining up at the Y=0 location on Cartesian axis.

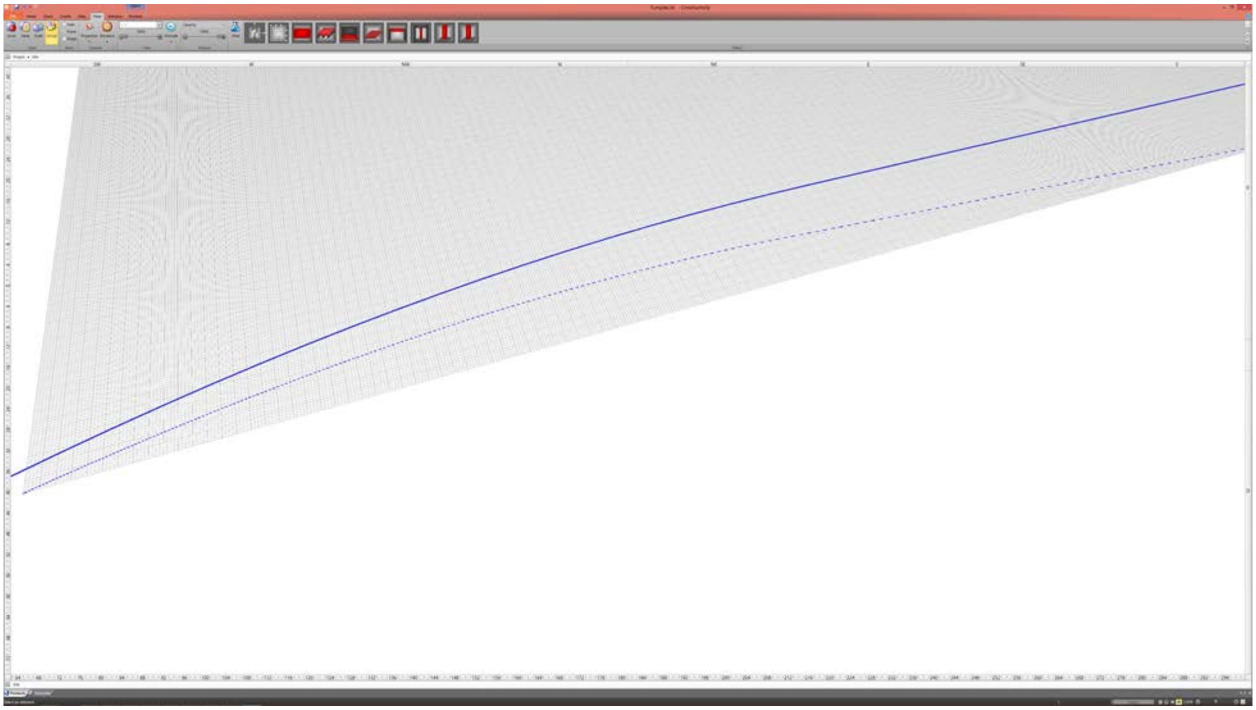
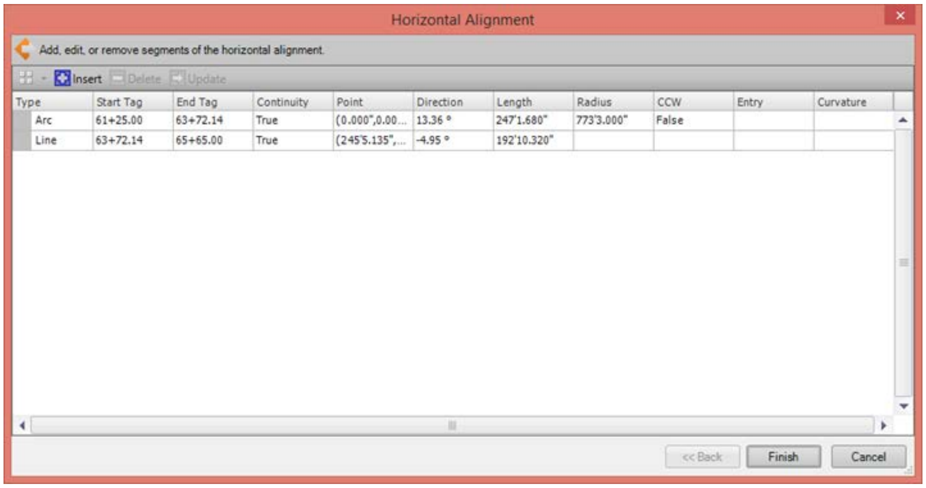


Figure 6 - Alignment Rendering

Figure 6 shows the horizontal alignment curve (dotted line relative to the grid), and the vertical alignment curve (projected above the horizontal curve at constant slope) for the Penn Turnpike bridge. All components are positioned relative to either the horizontal alignment (e.g. piers and abutments) or the vertical alignment (e.g. girders, deck, and terrain for approaches).

Figure 7 shows the horizontal alignment curve for the Penn Turnpike bridge, consisting of a circular arc segment and a linear segment.



Type	Start Tag	End Tag	Continuity	Point	Direction	Length	Radius	CCW	Entry	Curvature
Arc	61+25.00	63+72.14	True	(0.000",0.00...	13.36 °	247'1.680"	773'3.000"	False		
Line	63+72.14	65+65.00	True	(245'5.135",...	-4.95 °	192'10.320"				

Figure 7 – Penn Turnpike Horizontal Alignment Parameters

Figure 8 shows this horizontal alignment curve for the Van White bridge, consisting of a linear segment and a circular arc segment.

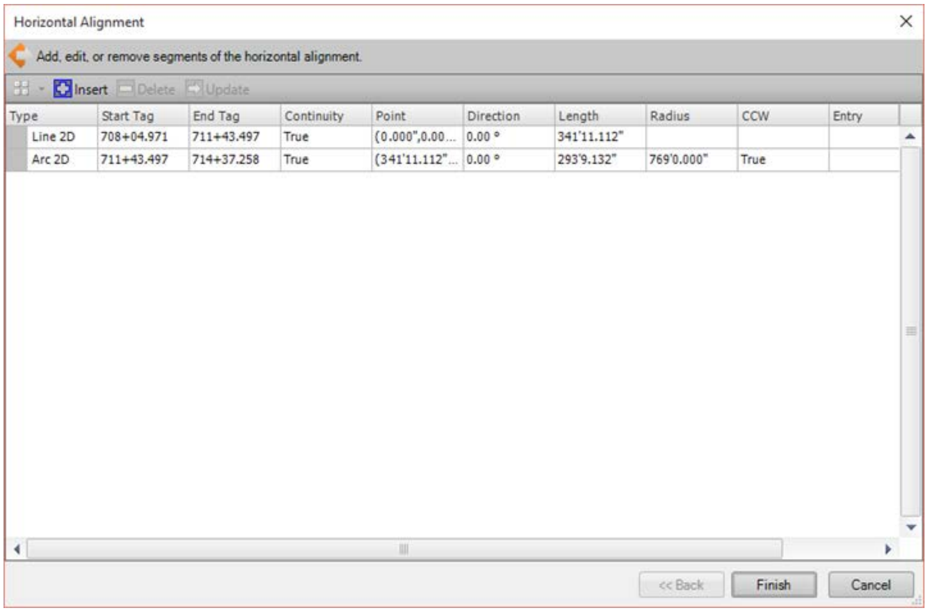


Figure 8 - Van White Horizontal Alignment Parameters

A vertical curve is also defined with a starting elevation and constant slope.

Figure 9 shows the vertical curve for the Penn Turnpike bridge which corresponds to the outer lane marking on the roadway surface.

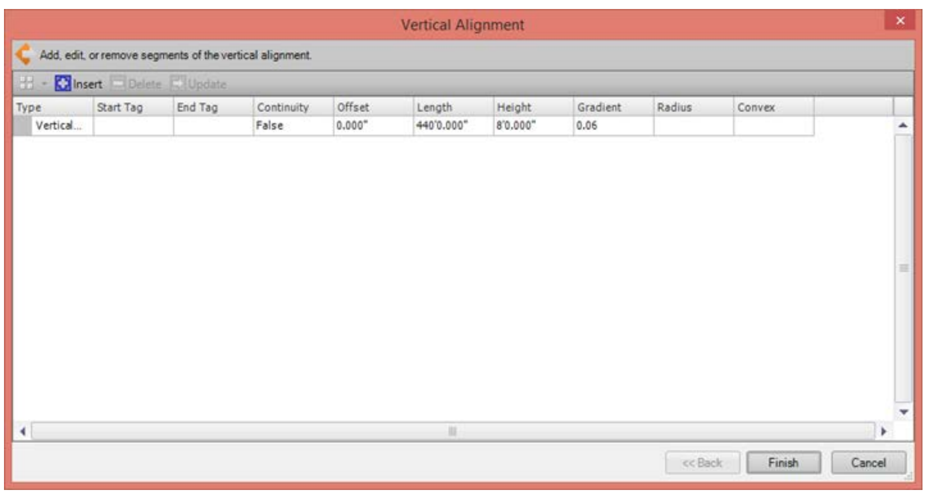


Figure 9 – Penn Turnpike Vertical Alignment Parameters

Figure 10 shows the vertical curve for the Van White bridge which corresponds to the physical center of the surface of the bridge deck (though not corresponding to lane markings).

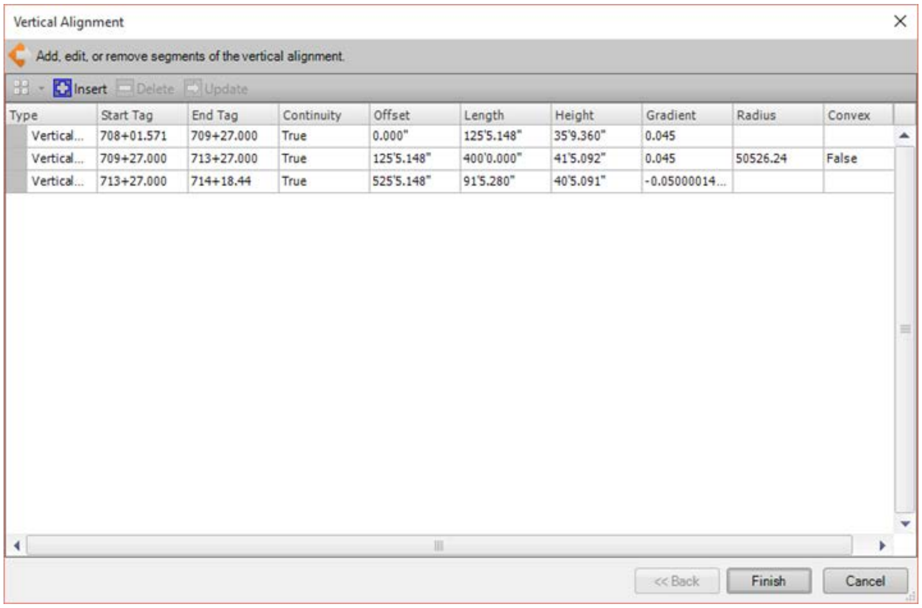


Figure 10 - Van White Vertical Alignment Parameters

1.1.1 Bentley OpenBridge

Parameters for horizontal alignment and vertical alignment (referred to as Profiles in Bentley software) follow the tables shown in the Figure 11 and Figure 12 respectively. The software also supports importing such information from other formats including LandXML.

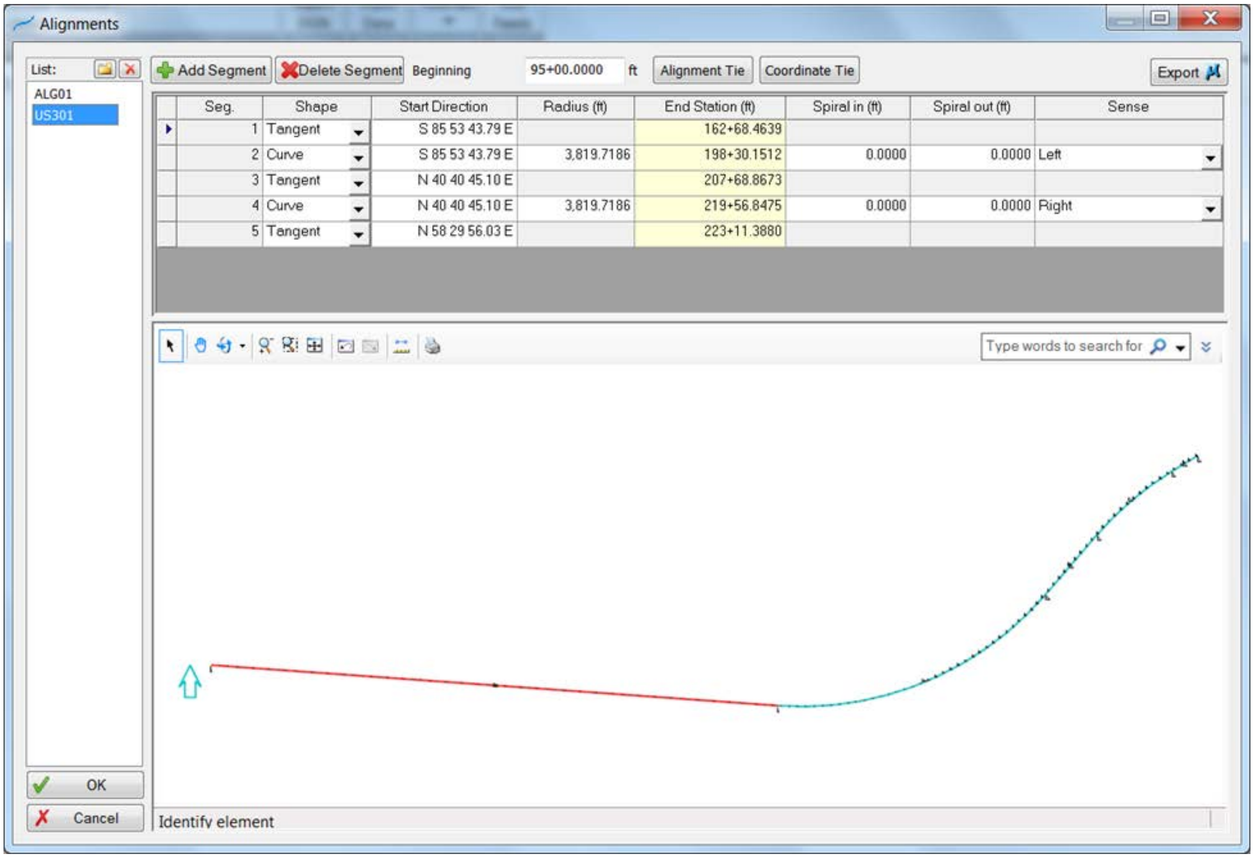


Figure 11- Horizontal Alignment Parameters for Bentley

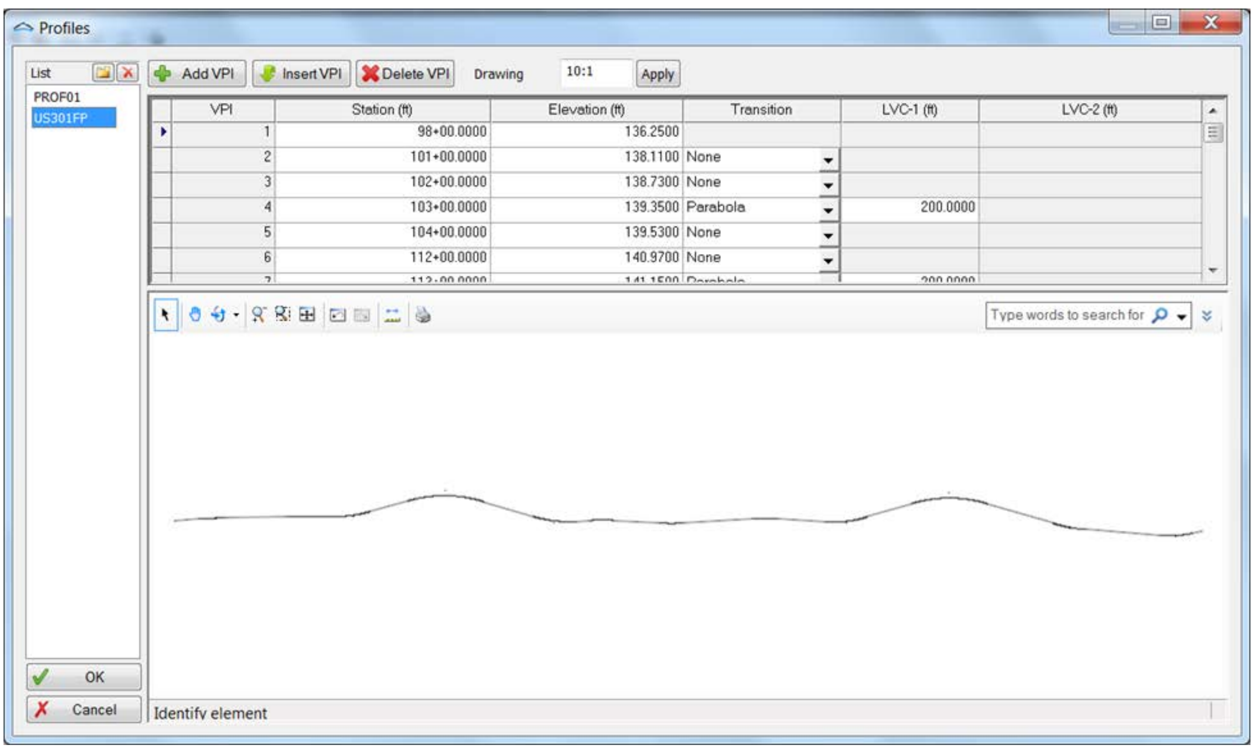


Figure 12 - Vertical Alignment Parameters for Bentley

1.1.2 Industry Foundation Classes

Using Industry Foundation Classes 4.1 published at www.buildingsmart-tech.org, all components are placed relative to an IfcAlignment instance, however, data definitions for relating an IfcAlignment to elements across multiple infrastructure domains are scheduled to be defined in a follow-on project ("IFC Alignment 1.1") which foreseeably would be integrated into IFC 4.2.

To advance such integration for purposes of bridges, a new entity, IfcRelPositions has been proposed in this Report (Volume II), indicating a horizontal offset along the curve (which also implies rotation within the horizontal plane), a lateral offset, a vertical elevation, and an optional transformation of components and geometry. For compatibility with existing software supporting IFC, and also reflecting plan documents, coordinates of every element are also provided according to Cartesian placement using IfcLocalPlacement.

Note: The IfcRelPositions structure is proposed as part of a future IFC 4.2 release containing work produced by the "IFC Alignment 1.1" extension project at BuildingSmart International and is not yet officially approved.

1.1.3 OpenBrIM

Using the OpenBrIM 2.0 schema and inferring usage based on the online viewer at www.openbrim.org, all components are placed relative to a RoadwayGeometry instance within the Obj object hierarchy, where the HorizontalAlignment entity contains a Curve where Type="Circular", Direction="Right", Length="2965.68", and Radius="9279", followed by a Curve where Type="Line" and Length="2314.32".

Note: the "Steel I Grider" example illustrates this instance underneath an "Obj" instance, however the schema does not support such usage – an assumption was made that the examples contained within OpenBrim 2.0 viewer use a later version of the schema than what is posted.

1.2 Abutments

As design parameters for abutments are highly dependent on the terrain and soil conditions, there is limited use of discrete parameters that can be applied to standardized formulas; rather any automated approach must take advantage of specialized algorithms. For this reason, and for particular bridges modeled, the abutments were modeled as discrete components with explicit geometry.

In addition to structural components such as piles, footings, wing walls, and cheek wall, components also include drainage piping, conduit, and architectural paneling.

Each abutment is placed relative to the horizontal alignment curve, with components placed according to Cartesian placement within. This reflects the positioning indicated on the plans, where all dimension lines are based on Cartesian positioning relative to the position and orientation of the station along the horizontal alignment curve.

1.2.1 Bentley OpenBridge

The Bentley LEAP Bridge Steel application contains several templates for creating abutments. For the particular bridges modelled, these templates may be used as a starting point, but do not account for the stepped footings, tapered and spiral wall geometry, irregular patterns of piles, reinforcing, drainage, or conduit. However, downstream applications may accommodate these details. As such, any resulting data format relying solely on the parameters presented would be insufficient for detailed design. That said this is not be construed as conflating an application with a data format, but before such data format specification is published, inferences are made based on software to be used for generating data in this format. The lack of such detail in a particular format is not necessarily a deficiency, but rather such format may not be intended to be used for a particular purpose, which is detailed design for the scenario evaluated.

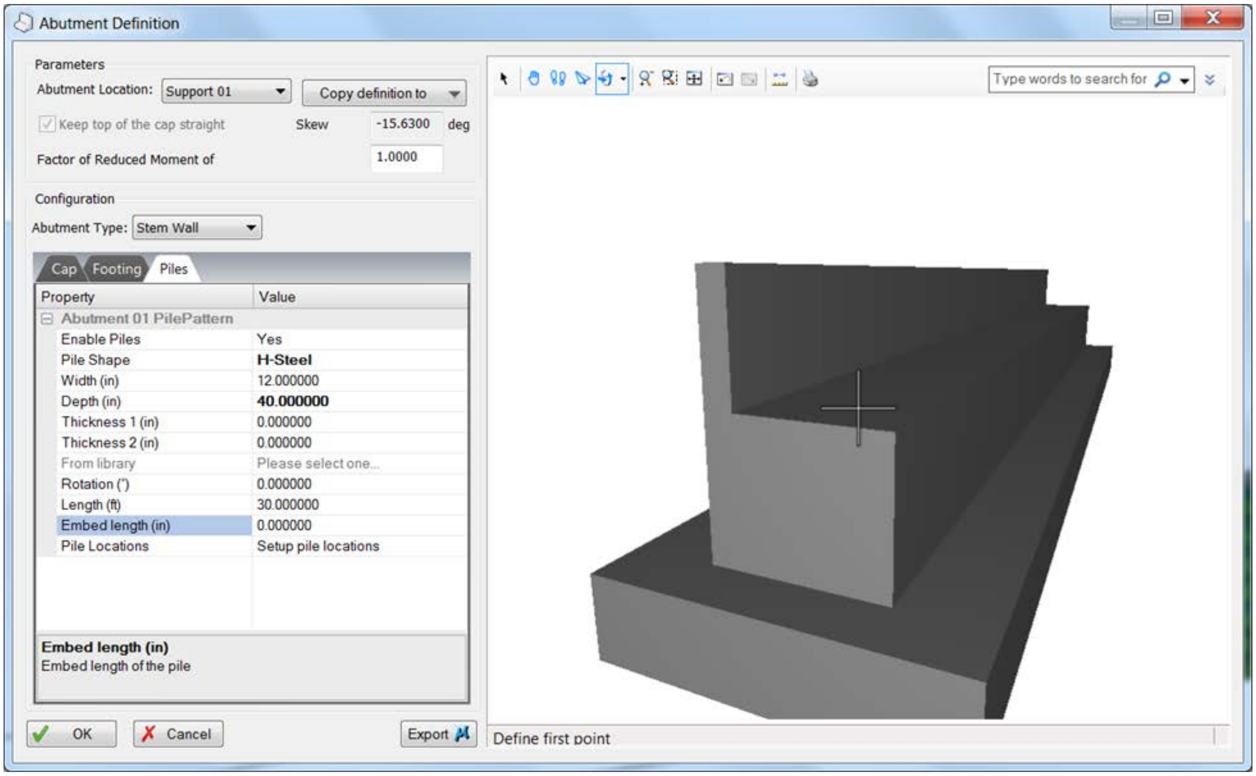


Figure 13 - Creating Abutment for Bentley

1.2.2 Industry Foundation Classes

Abutments are modelled as an aggregation of components including IfcWall, IfcFooting, IfcPile, IfcPlate, IfcPipeSegment, IfcCableCarrierSegment, IfcReinforcingBar, and other accessories.

Rebar is modelled using IfcReinforcingBar with a Body representation one or more IfcMappedItem's, each of which refer to the same IfcSweptDiskSolidPolygonal, but position each bar occurrence separately. Repetition intervals may optionally be defined with a Pattern representation consisting of a parameterized curve, where the length of the curve indicates the extent of repetition, and the parameter of the curve indicates the repetition interval. Each IfcReinforcingBar is typically bound to a type definition, where IfcReinforcingBarType may be used to capture bending shape definitions as found in a plans in a rebar schedule, where IfcReinforcingBarType.Name refers to the identifier as found in plans, IfcReinforcingBarType.BendingShapeCode identifies the bending code according to ACI 318, and IfcReinforcingBarType.BendingParameters indicates 1-4 parameters (depending on the BendingCode) which adjust the shapes. Optionally, IfcRelAssociatesConstraint may be associated with the IfcReinforcingBarType to explicitly indicate the formulas applied to the rebar geometry (IfcSweptDiskSolidPolygonal.Directrix\IfcPolyline.Points[n]) according to the bending parameters.

Piles supporting abutments are modelled using IfcPile, where multiple instances may be indicated using IfcMappedItem. Piles are connected to the footing using the IfcRelConnectsElements relationship, indicating the intersecting volume of each instance.

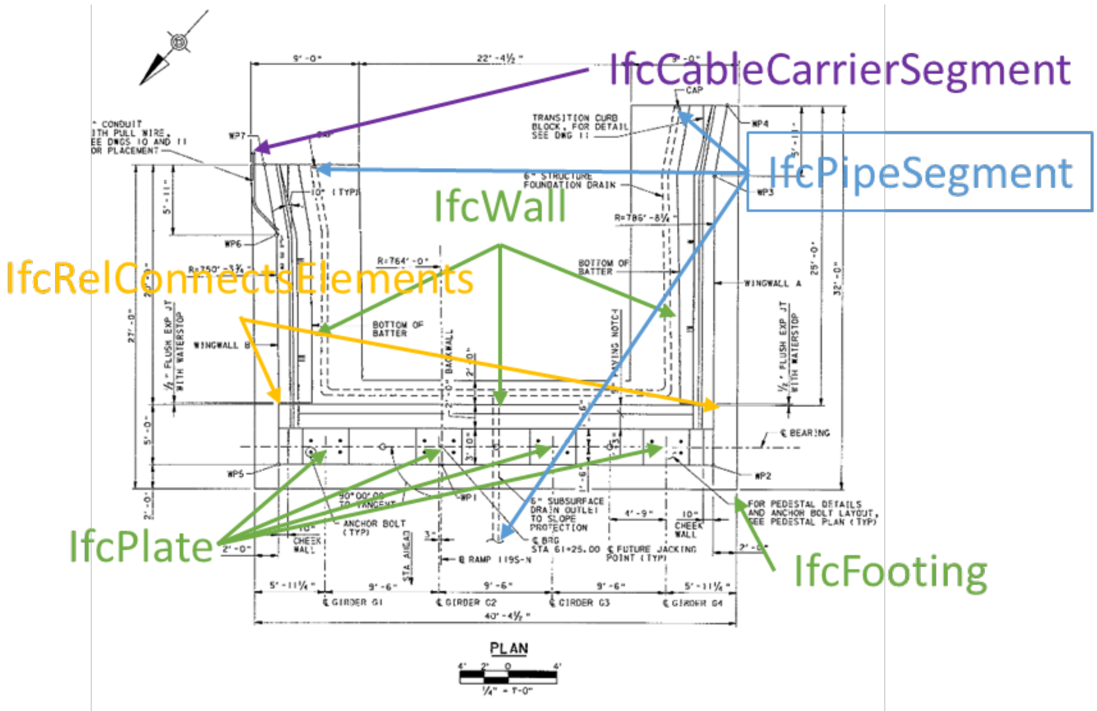
Architectural detailings applied to surfaces of the abutment are defined using IfcCovering and related to surfaces using the IfcRelCoversBldgElements relationship. The pattern to be used is applied as a styled representation (IfcStyledRepresentation) consisting of IfcSurfaceStyle holding

IfcSurfaceStyleWithTextures, with IfcImageTexture indicating the surface detailing, and qualifying as a bump pattern to be embedded or a treatment to be applied to the surface.

Wing walls and cheek walls are modeled using IfcWall (an arbitrarily shaped wall), as described further below, with construction joints interconnected using IfcRelConnectsElements. Drainage piping is modeled using IfcPipeSegment assigned to IfcDistributionSystem of type DRAINAGE with geometry using IfcSweptDiskSolid of IfcCircleHollowProfileDef. Conduit for wire is modeled using IfcCableCarrierSegment with geometry using IfcSweptDiskSolid of IfcCircleHollowProfileDef.

1.2.2.1 Penn Turnpike Bridge

Figure 14 illustrates the plan details and associated entities used to model the first abutment of the Penn Turnpike sample bridge. Figure 15 provides a graphical view of this abutment as modeled.



TOP OF PEDESTAL ELEVATIONS	
GIRDER NO.	ELEVATION
G1	1031.09
G2	1031.85
G3	1032.61
G4	1033.05

Figure 14 – Penn Turnpike Abutment 1 Components

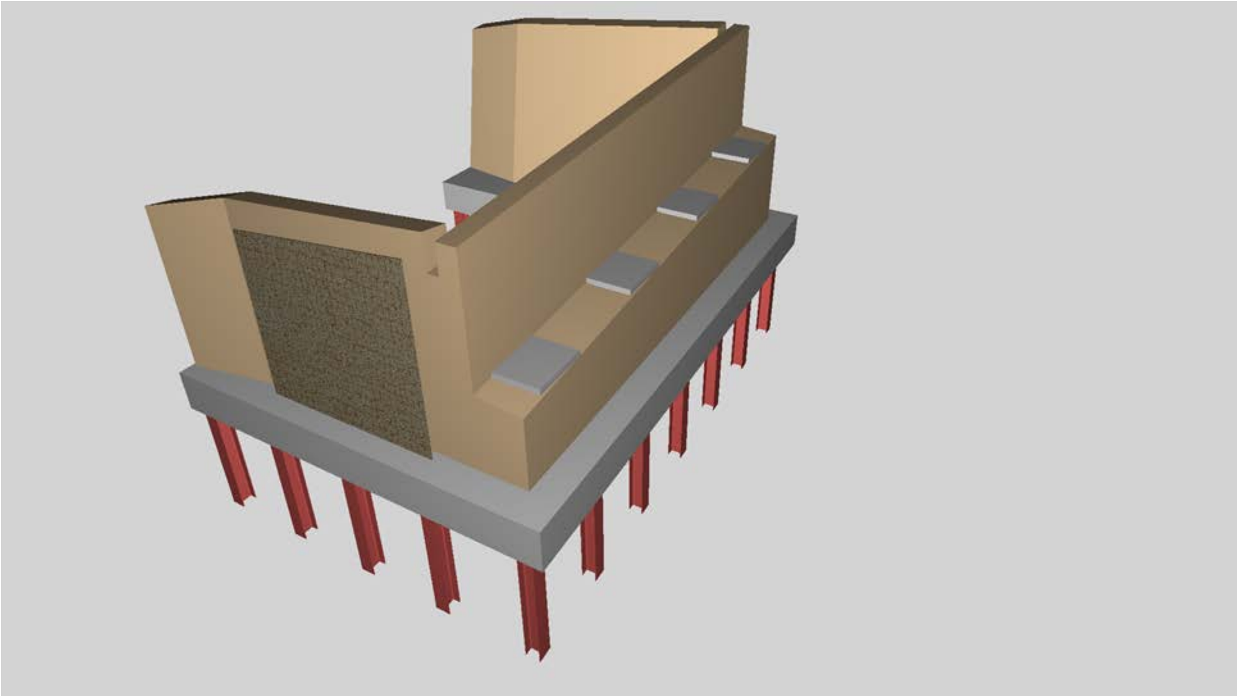


Figure 15 – Penn Turnpike Abutment 1 Modeling

For the Penn Turnpike, the wall supporting the girders is defined by IfcWall and has geometry represented by IfcExtrudedAreaSolidTapered as shown in Figure 16, where the left side (as facing the upstation/positive direction of the alignment curve) is higher than the right side. The wing walls on each end are also defined by IfcWall and have geometry represented by IfcExtrudedAreaSolidTapered, where the top is at an incline inwards, the wing walls lower in elevation away from the bridge.

While it is possible to represent this abutment in other ways within IFC (e.g. a single unified object with BREP or tessellated geometry), the particular composition used was chosen based on the functional definitions of each component and geometry driven by parameters that correspond to the dimensions as shown on the plans.

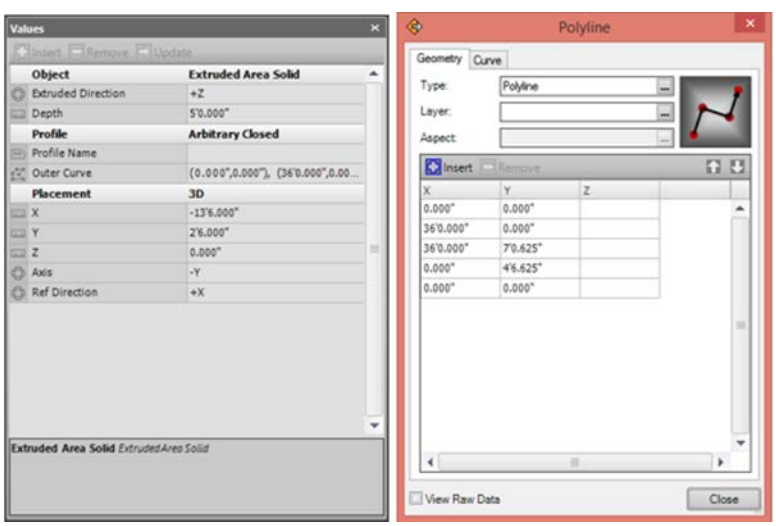


Figure 16 – Penn Turnpike Wall Parameters of Abutment 1

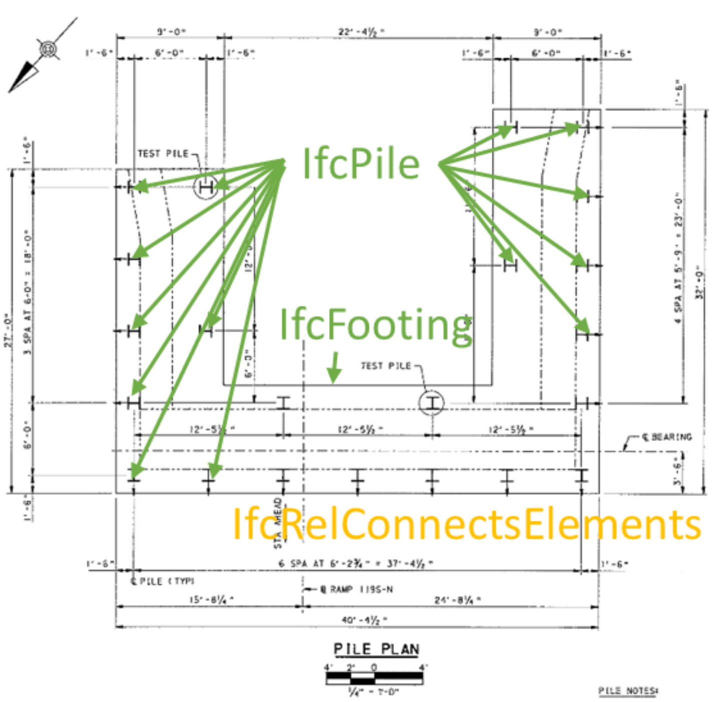


Figure 17 – Penn Turnpike Abutment 1 Foundation Plan

As illustrated in Figure 17, the foundation of the abutment consists of a single IfcFooting with shape defined by IfcExtrudedAreaSolid of an IfcArbitraryProfileDef extruded upwards. Each pile is modeled using a separate IfcPile with shape defined using IfcExtrudedAreaSolid of an IfcIShapeProfileDef extruded upwards, and is connected to the footing using IfcRelConnectsElements where RelatingElement refers to the pile. The footing connects to each wall using IfcRelConnectsElements where RelatingElement refers to the footing.

As illustrated in Figure 18, Rebar is modeled using IfcReinforcingBar and is embedded in the footing using the IfcRelAggregates relationship.

Subbase layers are represented by IfcGeographicElement having an IfcMaterialLayerSet describing the fill materials, depths, and required mechanical properties, with 2D ‘FootPrint’ representation of an IfcIndexedPolyCurve indicating the boundaries, and 3D ‘Body’ representation of an IfcExtrudedAreaSolid indicating the 3D extent.

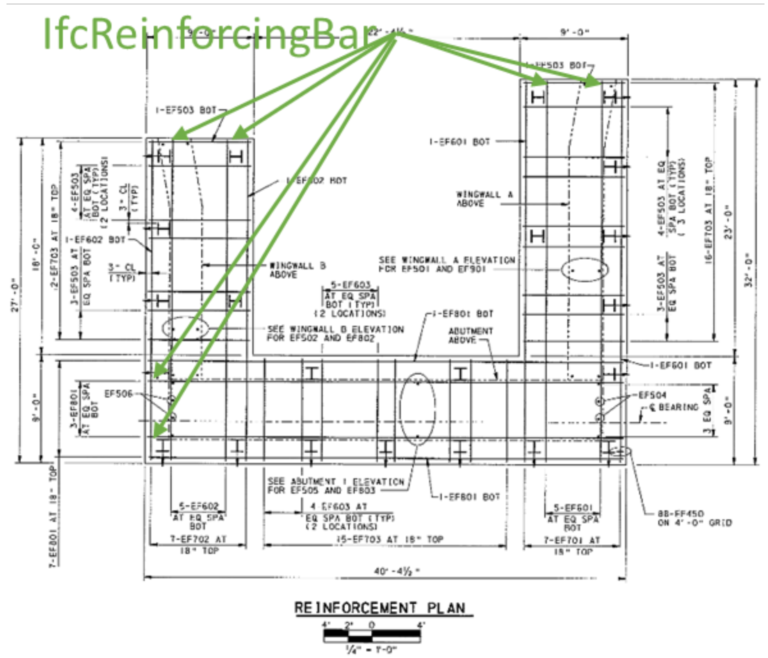


Figure 18 – Van White Abutment 1 Reinforcement Plan

Figure 19 shows an elevation of the abutment example.

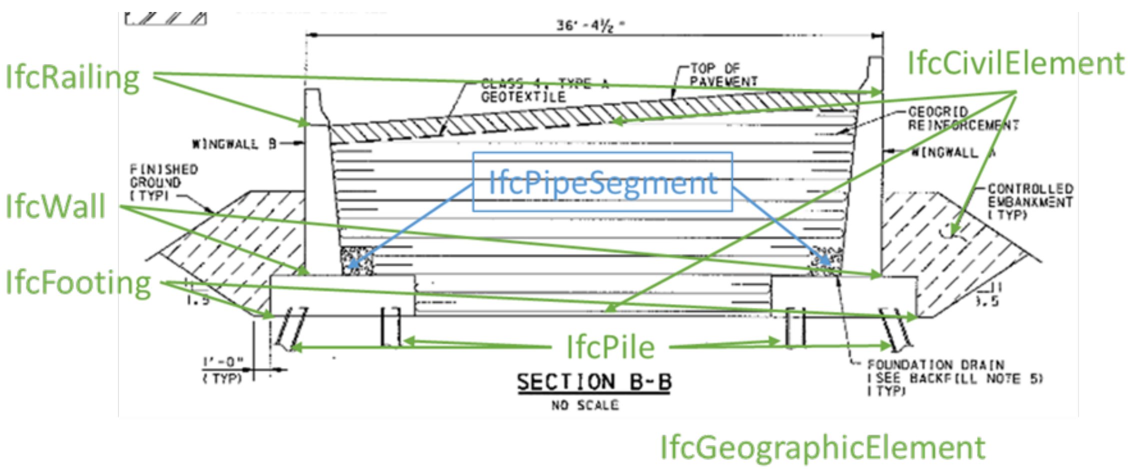


Figure 19 – Van White Abutment 1 Elevation

The pavement is defined using IfcCivilElement with the pavement material and thickness indicating using IfcMaterialLayerSet. The Profile representation is used to indicate the cross-section, using a closed IfcIndexedPolyCurve. The geogrid reinforcement beneath the road is defined using IfcCivilElement, where the geometry of the solid volume is defined using IfcAdvancedBrep.

As shown in Figure 20, the abutment at the other end has similar geometry, but uses stepped footings and a retaining wall of varying depth.

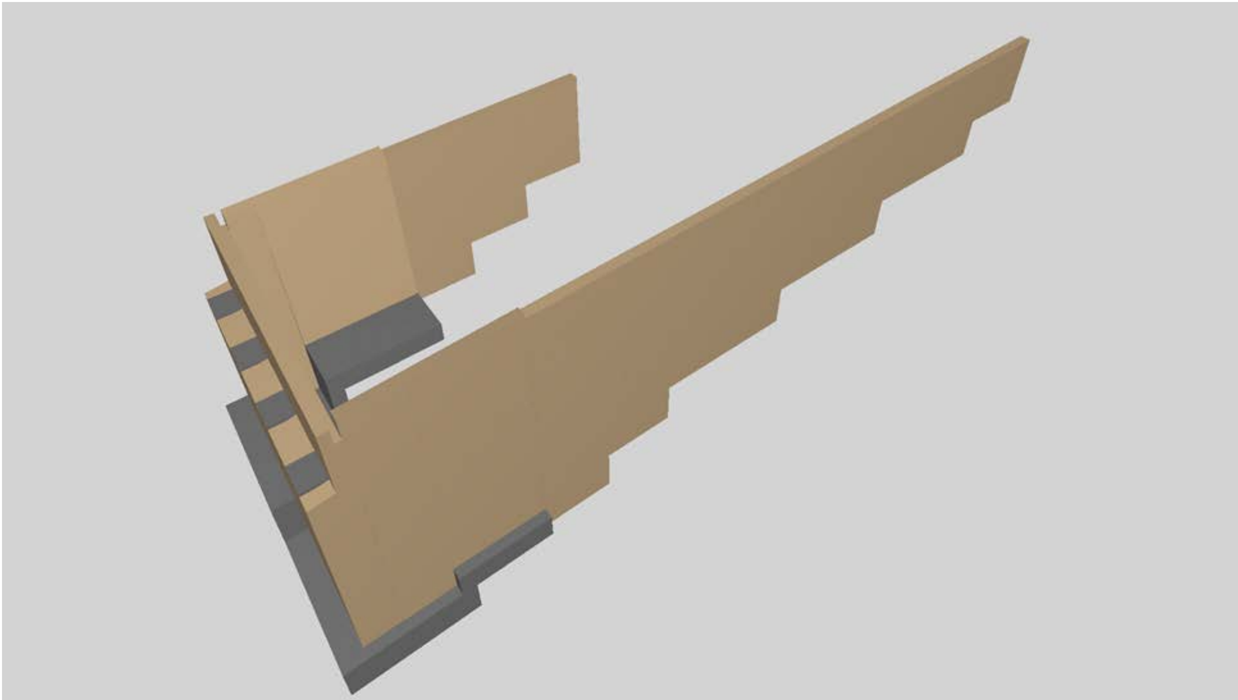


Figure 20 – Penn Turnpike Abutment 2

1.2.2.2 Van White Bridge

Figure 21 illustrates the south abutment as modeled for the Van White concrete box girder bridge.

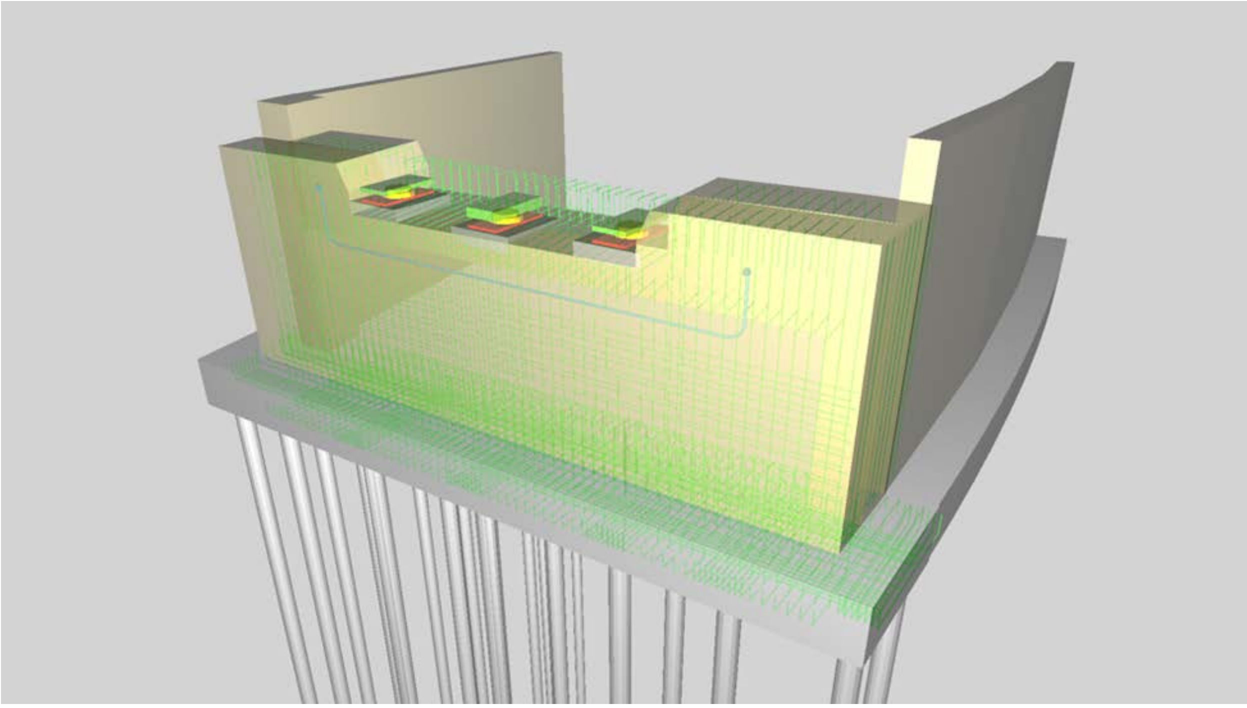


Figure 21 - Van White south abutment model

The cheek wall consists of a construction joint, so is modeled by two separate IfcWall instances, where the construction joint is related using the IfcRelConnectsElements relationship, and each instance consists of IfcExtrudedAreaSolidTapered geometry. The southeast wing wall is modeled as a single IfcWall instance based on IfcExtrudedAreaSolidTapered geometry. The southwest retaining wall is modeled as a single IfcWall instance based on IfcFixedReferenceSweptAreaSolid geometry to capture curvature.

Footings are modeled using IfcFooting based on IfcExtrudedAreaSolid geometry with IfcArbitraryClosedProfileDef footprint.

Piles are modeled using IfcPile based on IfcMappedItem indicating repetitive placement of IfcExtrudedAreaSolid geometry with IfcCircleProfileDef cross-section.

Reinforcing within this abutment follows several forms of repetition: linear, linear with angle transition, and staggered (L-shaped instances alternate directions of hooked ends). Reinforcing near piles uses larger bars spaced closer together, compared to smaller bars spaced further together between piles.

All rebar shapes are defined within a schedule for this set of plans, where parameters are applied to a set of standard bending shapes, where such information is captured at IfcReinforcingBarType, where the geometry is described using IfcSweptDiskSolidPolygonal. Figure 22 illustrates such rebar type definition.

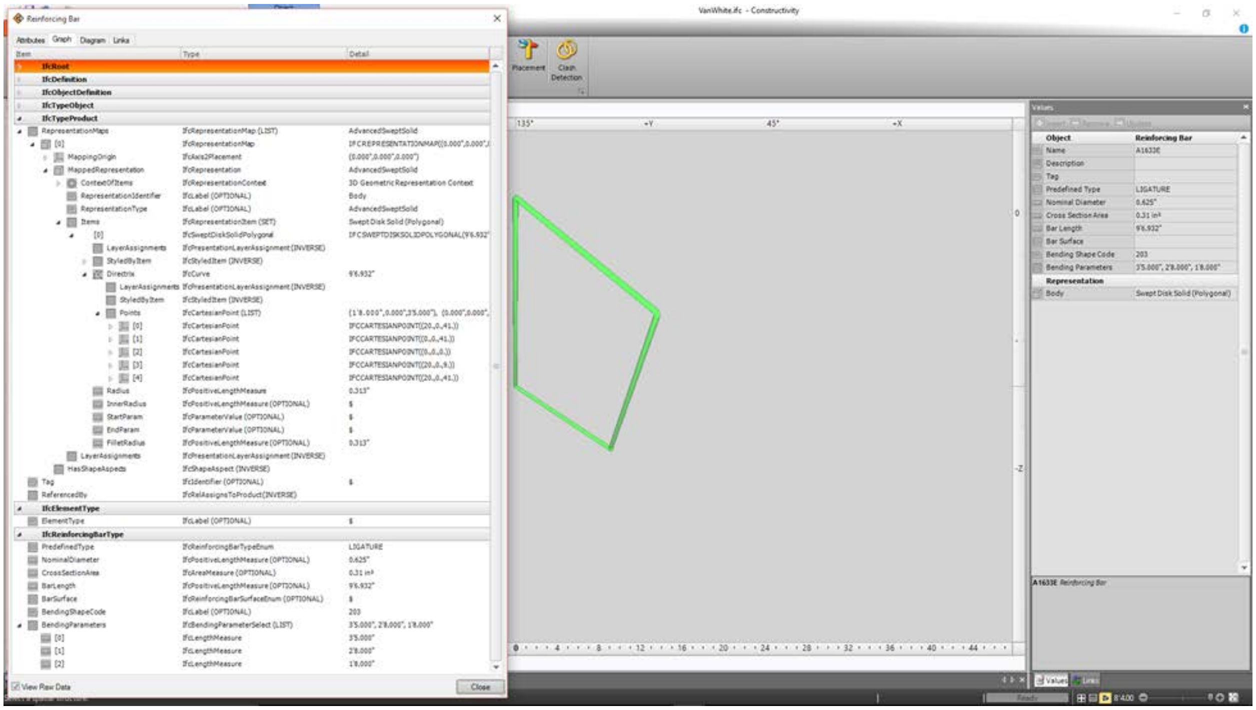


Figure 22 - Van White abutment rebar definition

Each usage of a group of bars of the same schedule entry is captured at IfcReinforcingBar, with each bar instance is positioned according to IfcMappedItem. Figure 23 illustrates such rebar occurrences.

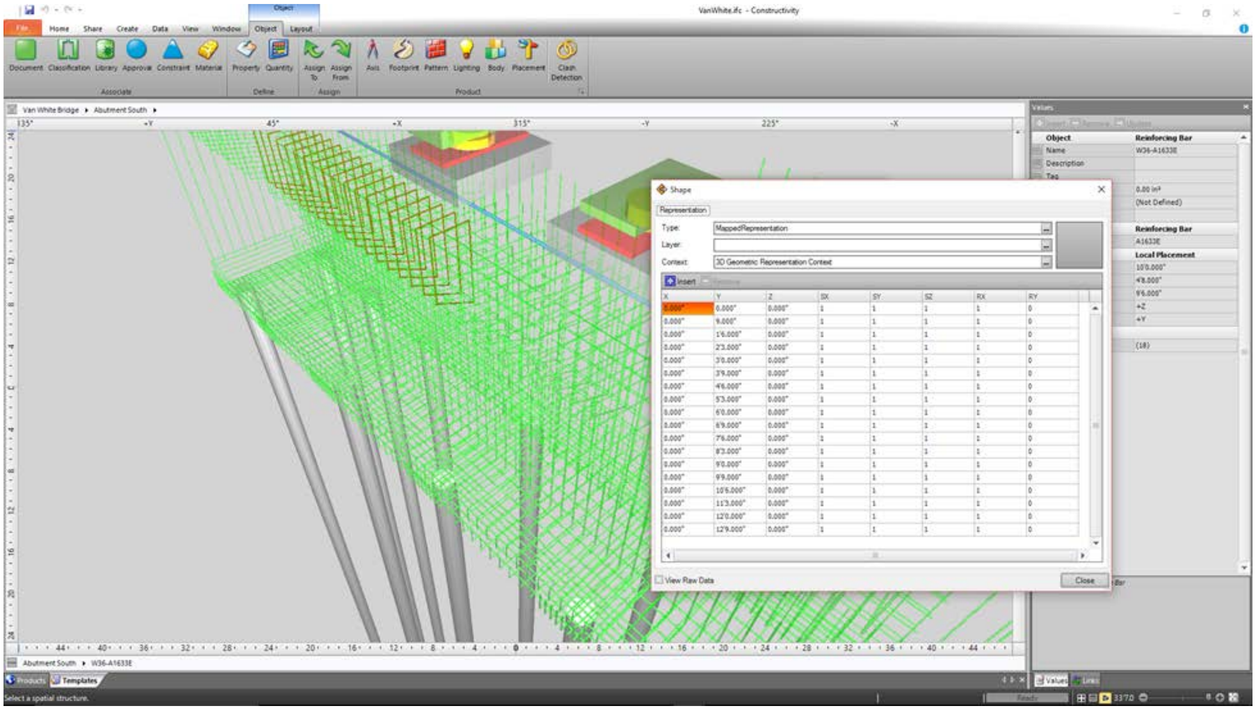


Figure 23 - Van White abutment rebar mapping

The embedded electrical conduit is captured using IfcCableCarrierSegment having PredefinedType=CONDUIT. Two junction boxes are captured using IfcJunctionBox, each of which have ports (IfcDistributionPort) connected to the IfcCableCarrierSegment using IfcRelConnectsPorts.

To explicitly indicate the embedding of conduit and junction boxes, the relationship IfcRelInterferesElements is used to link each IfcWall (separated by construction joint) to the elements. Unlike rebar, aggregation cannot be used for conduit, as it spans multiple elements and the single conduit instance must be provisioned for both in parallel.

Figure 24 illustrates the north abutment as modeled for the Van White concrete box girder bridge.

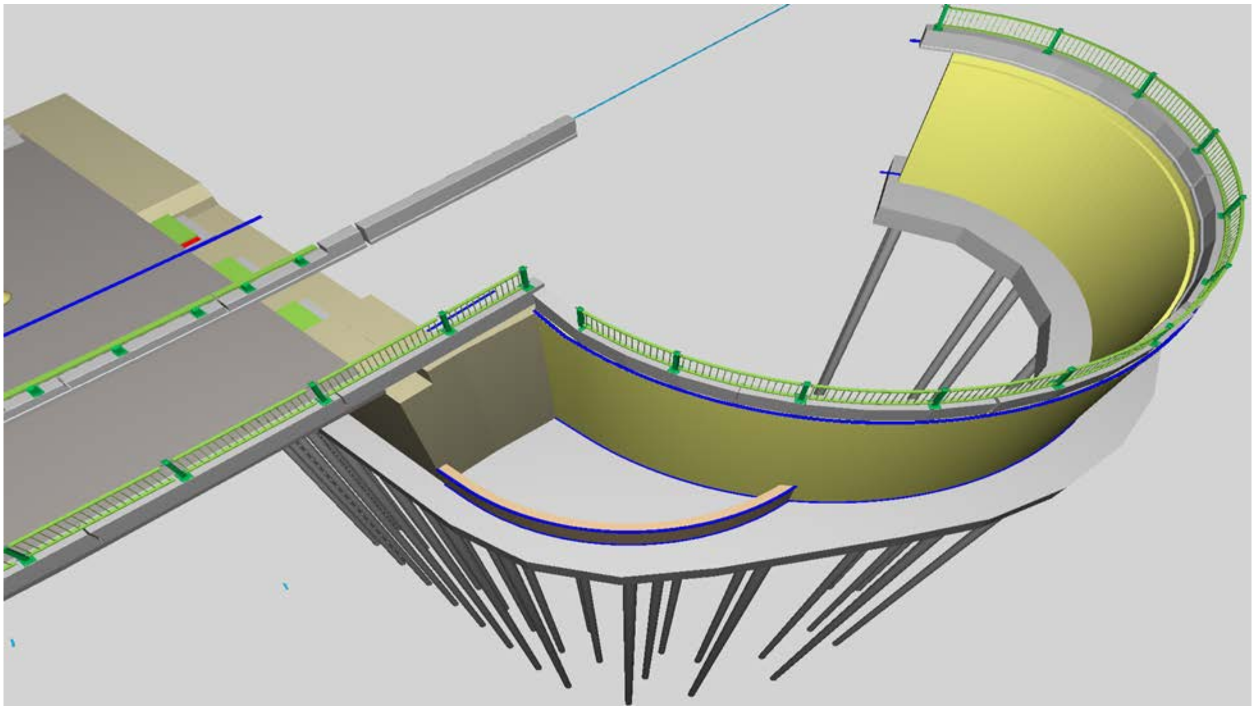


Figure 24 - Van White north abutment model

The north abutment is modeled similarly to the south abutment, with the exception of the retaining wall surrounding a plaza which follows a spiral curve. The same alignment-based positioning of the bridge deck and girders is also leveraged for the retaining wall, where the horizontal alignment curve is used to position the base of the wall, and the vertical alignment curve is used to determine the height of the wall which varies parabolically. The curved portion of the retaining wall consists of three segments divided by construction joints, using IfcMaterialProfileSetUsage to define the general cross-section, which is swept along the IfcAlignment that is referenced by IfcRelPositions.

The railing segments above are positioned according to the vertical alignment curve and use TransformType=WARP to indicate that the geometry must be transformed (not just visually but for fabrication), differing from railing segments along the bridge which are simply repositioned but not transformed, as the curvature is minimal so as to not require additional fabrication cost. Railing posts use TransformType=CHAIN, as there is the posts themselves are positioned but not transformed in either case.

1.2.3 OpenBrIM

Components are modeled in OpenBrim using hierarchy of Obj instances with geometry represented by Volume instances containing sets of Surface instances defined by polylines where any curves are discretized. This representation is equivalent to a polygonal Boundary Representation or “B-Rep” as commonly called in graphics applications. It is possible to convert from parameterized geometry to B-Rep, but not always the reverse.

Modeling of embedded components such as reinforcing bars, drainage structures and conduit would need to be elaborated. Connectivity information for purposes of deriving structural analysis information and construction dependencies also needs to be provisioned.

1.3 Piers

Design of piers were evaluated with the sample bridge models, where the Penn Turnpike bridge used hammerhead piers with super-elevation, while the Van White bridge used circular piers.

For these particular bridge models, geometry for concrete may be described as extruded area solids based on composite curves containing lines and circular arcs. Several construction joints are indicated with specific notched geometry.

1.3.1 Bentley OpenBridge

As shown in Figure 25, the Bentley LEAP Bridge Steel application contains several templates for defining piers. For the particular bridges evaluated, these templates can approximate the geometry, but do not account for the circular arcs below the cap, projections to support bearing plates at constant elevation, reinforcing, architectural detailing, or accessory components such as drainage piping. This is not necessarily a deficiency of the application or the resulting data; rather the intended usage is not for the particular purpose of detailed design. For example, Bentley RC-Pier may be used for detailed design of piers.

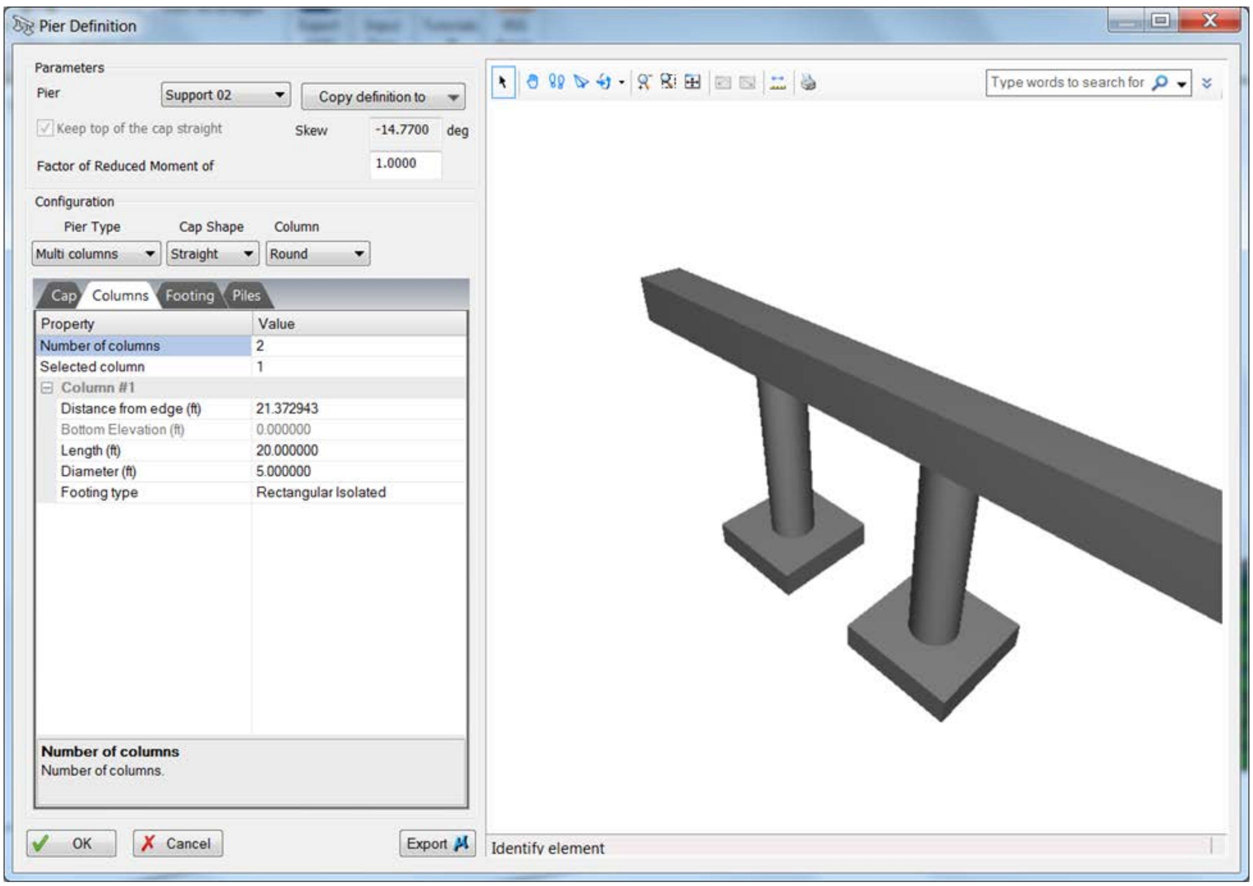


Figure 25 - Pier definition with Bentley

1.3.2 Industry Foundation Classes

Piers are modeled as a combination of interconnected components, typically comprising IfcPile, IfcFooting, IfcColumn, and IfcMember. Each of these components are connected to the next using

IfcRelConnectsElements or a subtype, where RelatingElement refers to the anchoring component and RelatedElement refers to component attached, which also implies the order of construction. The IfcRelConnectsElements relationship with ConnectionGeometry implies construction joints (no specific fasteners or adhesion other than rebar); otherwise specific connectivity is indicated using the IfcRelConnectsWithRealizingElements subtype instead.

All components are positioned locally within an IfcCivilElementType describing the pier, and an instance of the pier is instantiated using IfcCivilElement and is placed relative to an IfcAlignment based on a horizontal curve at the geometric center of the column. The components of the pier type are reflected at the occurrence, which enables specific connectivity between major components, such as the drain positioned relative to the pier but embedded within one of the slabs of the bridge deck.

Rebar is embedded using the IfcRelAggregates relationship, where both the outer concrete element and inner steel elements have geometry defined.

1.3.2.1 Penn Turnpike Bridge

Figure 26 indicates use of object types and relationships for each pier within the Penn Turnpike bridge example.

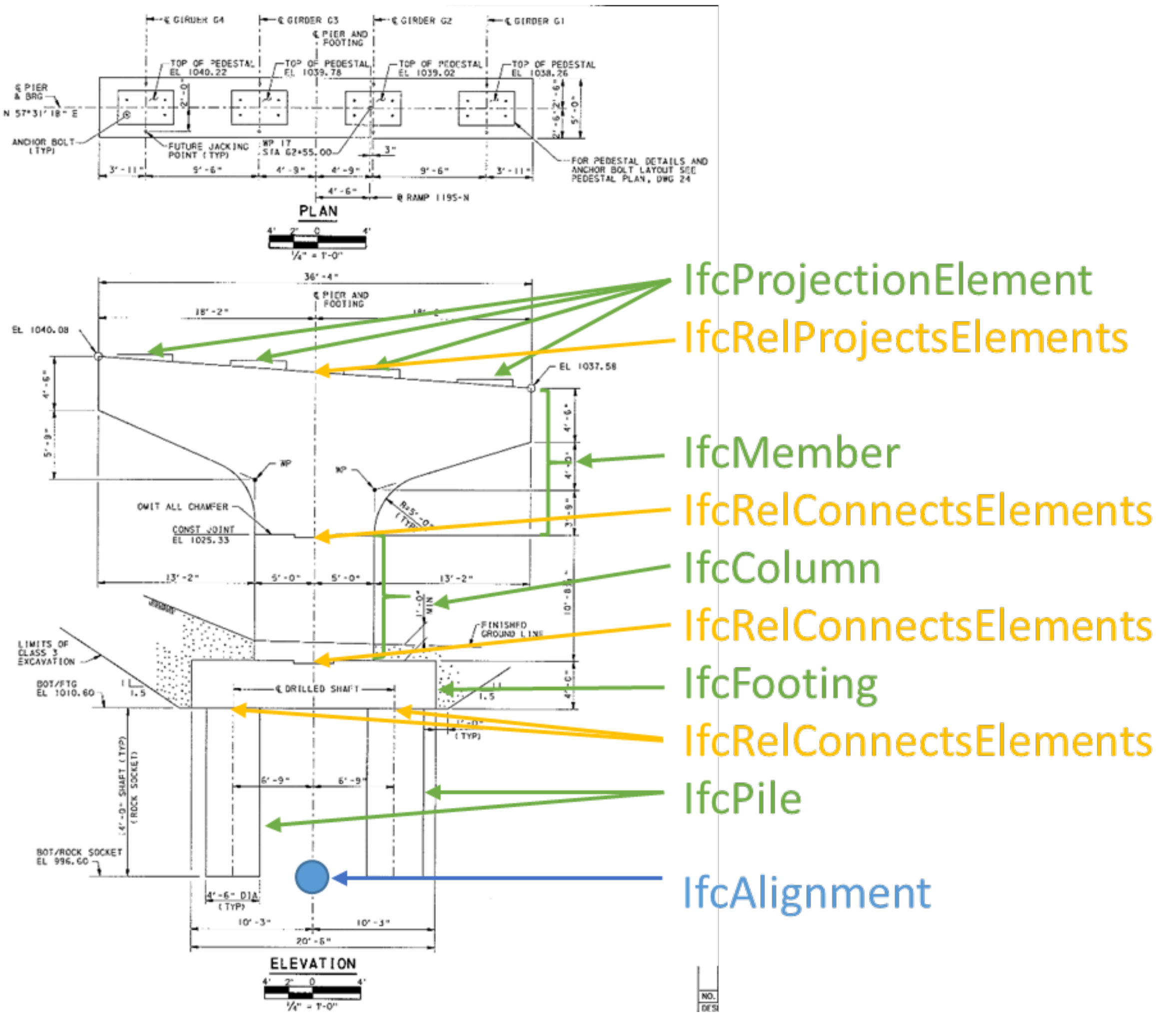


Figure 26 - Pier components

Figure 27 illustrates the resulting model of the first pier of this sample bridge.

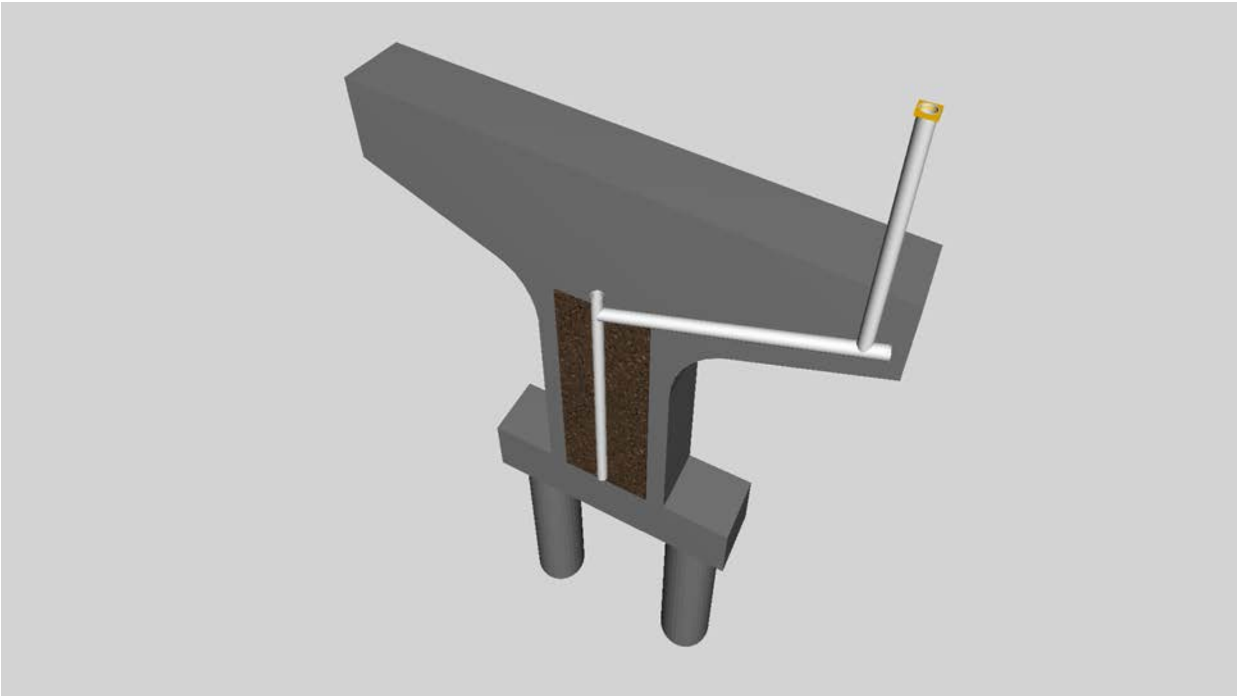


Figure 27 - Penn Turnpike Pier 1 model

For this particular example, the each pile is modeled using IfcPile. The concrete geometry is represented by IfcExtrudedAreaSolid extruded upwards (in the +Z direction) with the cross-section defined by IfcCircleProfileDef.

The footing is modeled using IfcFooting. The concrete geometry is represented by IfcExtrudedAreaSolid extruded towards the alignment curve with the cross-section defined by IfcArbitraryClosedProfileDef to accommodate the notch used for the construction joint.

The column (the section above the footing between construction joints) is modeled using IfcColumn (as an axial compression member). The concrete geometry is represented by IfcExtrudedAreaSolid extruded towards the alignment curve with the cross-section defined by IfcArbitraryClosedProfileDef to accommodate the notches for construction joint, consisting of IfcIndexedPolyCurve.

The top section is modeled using IfcMember (as an arbitrary load-carrying member) and with its base at a vertical offset (+Z) and zero horizontal offset. The concrete geometry is represented by IfcExtrudedAreaSolid extruded towards the alignment curve with the cross-section defined by IfcArbitraryClosedProfileDef to accommodate the irregular shape, consisting of IfcIndexedPolyCurve.

The architectural treatment is modeled using IfcCovering, with IfcExtrudedAreaSolid of IfcRectangleProfileDef indicating the shape and the rock texture indicated using IfcStyledItem, IfcSurfaceStyle, IfcSurfaceStyleWithTextures, and IfcImageTexture.

The drainage pipes are modeled using IfcPipeSegment, with IfcExtrudedAreaSolid of IfcCircleHollowProfileDef indicating the shape. The drain above is modeled using IfcWasteTerminal. All piping components belong to an IfcDistributionSystem of predefined type STORMWATER, and have ports at each end indicating connectivity using IfcDistributionPort. Ports are connected according to flow using IfcRelConnectsPorts.

This separate modeling associated with the construction joints facilitates connectivity of the supported elements in fashion similar to that used for structural analysis and construction sequencing.

Figure 28 illustrates the resulting model for the second pier of this sample bridge.

Figure 28 - Penn Turnpike Pier 2 model

1.3.2.2 Van White Bridge

For the Van White Memorial bridge, there are four piers using the same overall geometry but with variable height. Such similarity allows for a template to be used that describes the pier once, and then instantiated four times, where the only parameters that vary are the position and height. However, the plans for this bridge did not use such templated approach, and described each pier separately.

Figure 29 illustrates the resulting model for the first pier of this sample bridge.

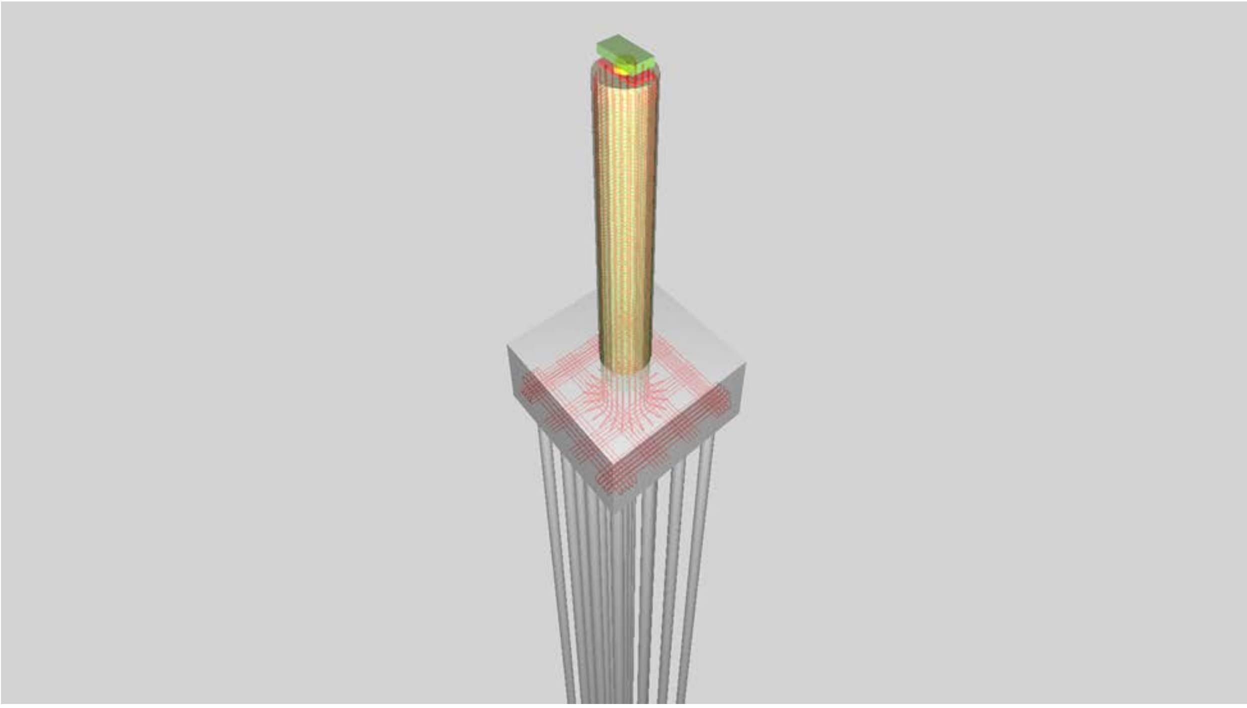


Figure 29 - Van White pier model

The template of piers is modeled as IfcCivilElementType. The IfcCivilElementType contains an 'Axis' representation of an IfcPolyline having a single segment of an arbitrary length, which is used to establish parametric resizing behavior.

The IfcCivilElementType is aggregated into four components using the IfcRelAggregates relationship to contain IfcPile, IfcFooting, IfcColumnStandardCase, and IfcMechanicalFastener (for the bearing). These components are connected sequentially using IfcRelConnectsPathElements, indicating that each start and end according to connectivity along the 'Axis' path. While such relationship indicates physical connectivity that may be used for construction and analytical purposes, this relationship may also be used for instantiation of such template to indicate how components resize.

The IfcPile corresponds to 16 shape occurrences forming a 4x4 grid. The pile has a corresponding template, IfcPileType, which defines the geometry using IfcExtrudedAreaSolid based on IfcCircleProfileDef. To indicate the repeat placement, the 'Body' representation of IfcPile consists of 16 instances of IfcMappedItem, each indicating the position relative to the 'Axis' of the enclosing IfcCivilElementType.

The IfcFooting defines the shape of the concrete using IfcExtrudedAreaSolid based on IfcRectangleProfileDef. This footing is aggregated into components for rebar using the IfcRelAggregates relationship to contain IfcReinforcingBar and IfcReinforcingMesh, where one IfcReinforcingBar instance describes lateral rebar forming a complete loop (on the outside in each direction), another IfcReinforcingBar instance describes lateral rebar forming a straight line (on the inside in each direction), and an IfcReinforcingMesh instance describes longitudinal rebar and stirrups forming a cage within the column. While such mesh is primarily part of the adjoining IfcColumnStandardCase, it must be embedded within the IfcFooting before the concrete is poured for the footing; therefore such rebar is aggregated within the IfcFooting rather than the IfcColumnStandardCase. Each rebar assembly is detailed at IfcReinforcingBarType and IfcReinforcingMeshType respectively. Geometry for all rebar types makes use of IfcSweptDiskSolidPolygonal, which allows parametric definition of the bending radius applied at each transition of an IfcPolyline. Geometry for all rebar occurrences makes use of IfcMappedItem to indicate the position and orientation within the IfcFooting relative to the 'Axis' representation.

The IfcColumnStandardCase captures the variable-height stem of the pier. The "StandardCase" subtype provides for parametric definition here in the same way as done in the general IFC schema, differentiating from the more generic IfcColumn. This IfcColumnStandardCase consists of an 'Axis' representation and a relationship to the material cross-section definition using IfcRelAssociatesMaterial linked to IfcMaterialProfileSetUsage. As this column consists of a single circular cross-section, the referenced IfcMaterialProfileSet contains a single IfcMaterialProfile having a reference to IfcCircleProfileDef and IfcMaterial. Concrete strength and other requirements are indicated at IfcMaterialProperties using "Pset_MaterialMechanical" and others linked to the IfcMaterial instance.

The IfcMechanicalFastener captures the pot bearing. Functionally, a pot bearing fits the same definition as already documented at IfcMechanicalFastener, though is of a much larger form factor compared to more common existing uses such as for bolts or bearings used with building equipment. The mechanical behavior of the IfcMechanicalFastener may be described by an assigned IfcStructuralPointConnection, as is done for any physical element having an analytical representation. This IfcStructuralPointConnection defines mechanical degrees of freedom using IfcBoundaryNodeCondition, indicating which axes are fixed and which are free within the range of movement.

Each pier instance is modelled using a unique IfcCivilElement instance referencing the IfcCivilElementType. Each IfcCivilElement has a unique ObjectPlacement describing its location, an IfcRelPositions relationship indicating the placement relative to the horizontal curve of the IfcAlignment, and an 'Axis' representation indicating the path, which essentially stretches the template to fit according to the specific height.

Each IfcCivilElement is connected to the corresponding pier cap beam (IfcBeam) of the box girder using IfcRelConnectsPathElements, where RelatingElement refers to the pier using 'ATEND' positioning, and the RelatedElement refers to the box girder using 'ATPATH' positioning. Each CivilElement is connected to the ground (IfcGeographicElement) using IfcRelConnectsElements, where volume geometry defines the required ground penetration. Such relationships provide information for applications to automatically resize piers according to the ground and the vertical alignment of the bridge deck.

1.3.3 OpenBrim

Piers are modelled as a hierarchy of Obj instances using Volume geometry based on extruded polylines. Circular arcs are discretized into line segments. Connectivity information such as construction joints, sequencing, rebar embedding, and drainage flow are not captured; it is foreseeable that extensions could be made for capturing such relationship information either built into the static schema or as dynamic extensions based on well-known identifiers.

1.4 Bearings

This Penn Turnpike sample bridge uses pot bearings for supporting girders, where longitudinal-guided bearings are used at abutments, lateral-guided bearings are used at the first pier (supporting circular alignment), and non-guided bearings are used at the second pier (supporting linear alignment).

The Van White sample bridge uses longitudinal-guided bearings at abutments, lateral-guided bearings at outer piers, and fixed connections at inner piers.

Figure 30 illustrates bearing details for the Penn Turnpike bridge.

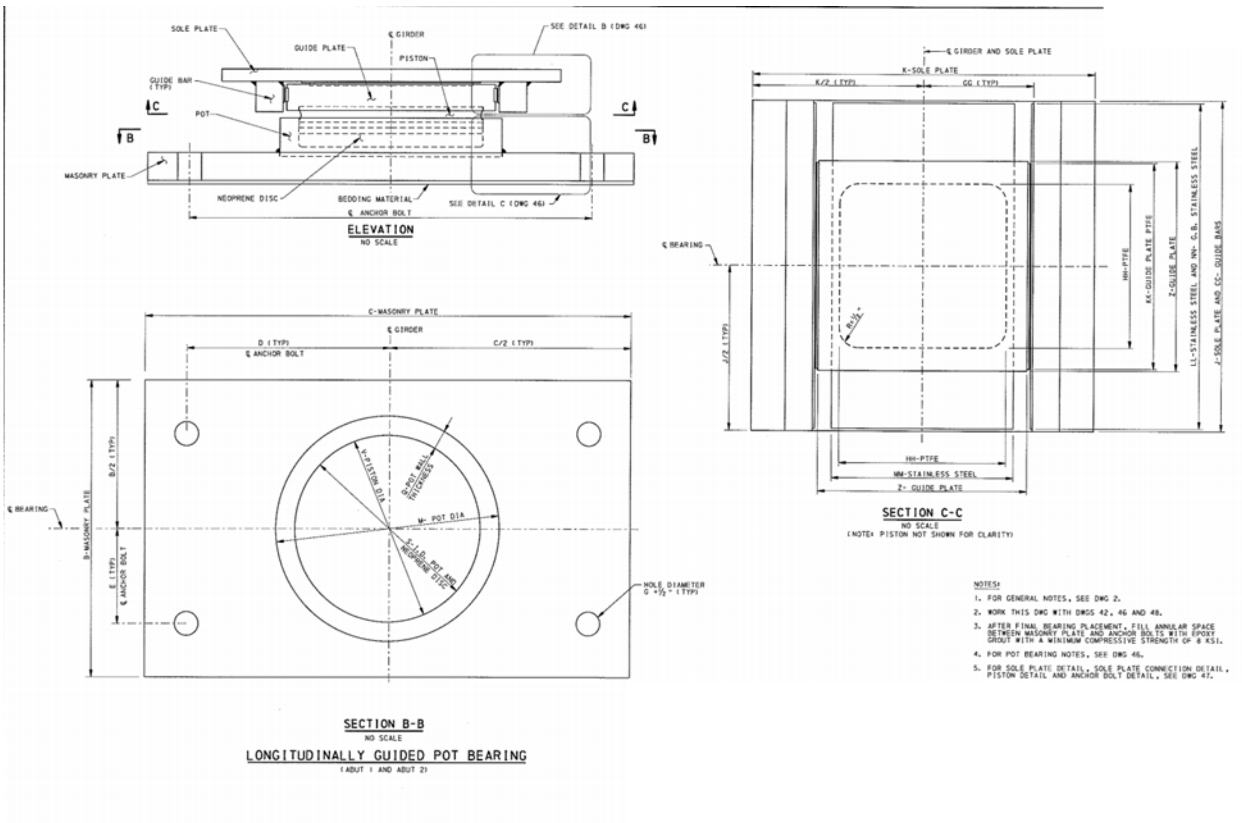


Figure 30 - Penn Turnpike bearing plan

1.4.1 Bentley OpenBridge

Bentley LEAP Bridge Steel does not capture information for bearings.

1.4.2 Industry Foundation Classes

Each pot bearing type is defined using IfcMechanicalFastenerType with PredefinedType set to USERDEFINED and ObjectType set to “PotBearing”.

The mechanical behavior of the bearing may be described using IfcStructuralPointConnection linked to the IfcMechanicalFastenerType using the IfcRelAssignsToProduct relationship. The IfcStructuralPointConnection may defines the support conditions using IfcBoundaryNodeCondition. This usage is consistent with existing IFC documentation for structural analysis, however does not require a full analytical model – only relationships as indicated.

1.4.2.1 Penn Turnpike Bridge

As shown in Figure 31, the geometry for the bearings is represented by swept solids for each shape. Masonry plates use IfcExtrudedAreaSolid with a profile cross-section based on IfcArbitraryClosedProfileWithVoids consisting of IfcCircle for each of the anchor bolt holes. Pots use IfcRevolvedAreaSolid with a profile cross-section based on IfcArbitraryClosedProfileDef and IfcIndexedPolyCurve, which is necessary to capture the fillet radius (which would not be possible with IfcExtrudedAreaSolid). Sole plates use IfcExtrudedAreaSolid with a profile-cross-section based on IfcRectangleProfileDef.

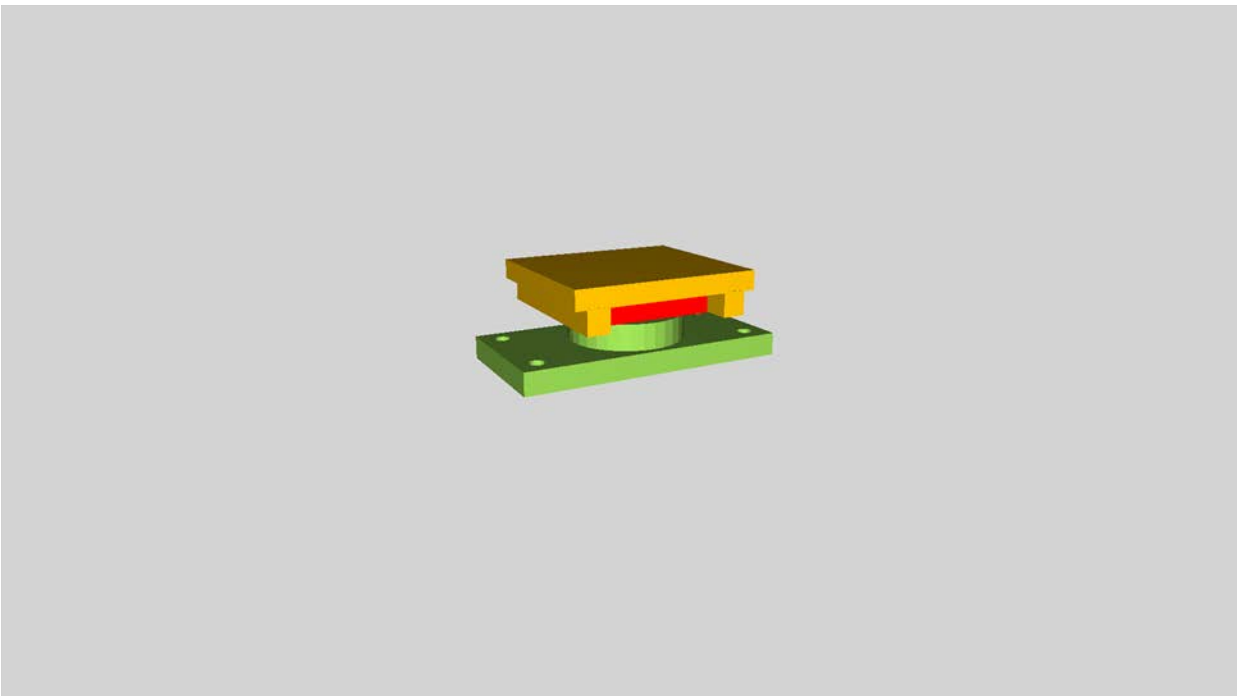


Figure 31 - Penn Turnpike bearing definition model

POT BEARING DIMENSION TABLE (INCHES)

LOCATION	GIRDER MARK	BEARING TYPE	ROTATION (RADS.)	MASONRY PLATE A	P	B	C	D	E	ANCHOR BOLT QTY	G	QQ	RR
ABUT 1	G1	LONGIT. GUIDED	0.03	2.625	0.25	16.75	34.5	14.625	5.750	4	1.50	5.00	3.50
	G2	LONGIT. GUIDED	0.03	2.625	0.25	16.75	34.5	14.625	5.750	4	1.50	5.00	3.50
	G3	LONGIT. GUIDED	0.03	2.625	0.25	16.75	34.5	14.625	5.750	4	1.50	5.00	3.50
	G4	LONGIT. GUIDED	0.03	2.625	0.25	16.75	34.5	14.625	5.750	4	1.50	5.00	3.50
PIER 1	G1	TRANS. GUIDED	0.03	3.500	0.25	21.75	45.0	19.125	7.500	4	2.00	6.50	5.00
	G2	TRANS. GUIDED	0.03	3.500	0.25	21.75	45.0	19.125	7.500	4	2.00	6.50	5.00
	G3	TRANS. GUIDED	0.03	3.500	0.25	21.75	45.0	19.125	7.500	4	2.00	6.50	5.00
	G4	TRANS. GUIDED	0.03	3.500	0.25	21.75	45.0	19.125	7.500	4	2.00	6.50	5.00
PIER 2	G1	NON-GUIDED	0.03	2.750	0.25	21.75	32.0	13.750	8.625	4	1.25	5.00	3.50
	G2	NON-GUIDED	0.03	2.750	0.25	21.75	32.0	13.750	8.625	4	1.25	5.00	3.50
	G3	NON-GUIDED	0.03	2.750	0.25	21.75	32.0	13.750	8.625	4	1.25	5.00	3.50
	G4	NON-GUIDED	0.03	2.750	0.25	21.75	32.0	13.750	8.625	4	1.25	5.00	3.50
ABUT 2	G1	LONGIT. GUIDED	0.03	2.625	0.25	16.75	34.5	14.625	5.750	4	1.50	5.00	3.50
	G2	LONGIT. GUIDED	0.03	2.625	0.25	16.75	34.5	14.625	5.750	4	1.50	5.00	3.50
	G3	LONGIT. GUIDED	0.03	2.625	0.25	16.75	34.5	14.625	5.750	4	1.50	5.00	3.50
	G4	LONGIT. GUIDED	0.03	2.625	0.25	16.75	34.5	14.625	5.750	4	1.50	5.00	3.50

LOCATION	GIRDER MARK	SOLE PLATE H	J	K	POT L	M	N	O	Q	NEOPRENE DISC R	S	T	PISTON U	V	W	X	GUIDE PLATE Y	Z
ABUT 1	G1	2.25	21.5	24.0	2.5625	14.875	2.0000	0.5625	1.625	1.250	11.625	0.375	1.5000	11.605	0.500	0.2500	2.625	14.875
	G2	2.25	21.5	24.0	2.5625	14.875	2.0000	0.5625	1.625	1.250	11.625	0.375	1.5000	11.605	0.500	0.2500	2.625	14.875
	G3	2.25	21.5	24.0	2.5625	14.875	2.0000	0.5625	1.625	1.250	11.625	0.375	1.5000	11.605	0.500	0.2500	2.625	14.875
	G4	2.25	21.5	24.0	2.5625	14.875	2.0000	0.5625	1.625	1.250	11.625	0.375	1.5000	11.605	0.500	0.2500	2.625	14.875
PIER 1	G1	2.25	30.0	27.5	3.4375	20.250	2.6250	0.8125	2.250	1.625	15.750	0.375	1.8750	15.730	0.625	0.3125	3.375	20.250
	G2	2.25	30.0	27.5	3.4375	20.250	2.6250	0.8125	2.250	1.625	15.750	0.375	1.8750	15.730	0.625	0.3125	3.375	20.250
	G3	2.25	30.0	27.5	3.4375	20.250	2.6250	0.8125	2.250	1.625	15.750	0.375	1.8750	15.730	0.625	0.3125	3.375	20.250
	G4	2.25	30.0	27.5	3.4375	20.250	2.6250	0.8125	2.250	1.625	15.750	0.375	1.8750	15.730	0.625	0.3125	3.375	20.250
PIER 2	G1	2.25	23.0	23.0	3.1875	17.250	2.4375	0.6875	0.750	1.625	15.750	0.375	1.3125	15.730	0.500	---	---	---
	G2	2.25	23.0	23.0	3.1875	17.250	2.4375	0.6875	0.750	1.625	15.750	0.375	1.3125	15.730	0.500	---	---	---
	G3	2.25	23.0	23.0	3.1875	17.250	2.4375	0.6875	0.750	1.625	15.750	0.375	1.3125	15.730	0.500	---	---	---
	G4	2.25	23.0	23.0	3.1875	17.250	2.4375	0.6875	0.750	1.625	15.750	0.375	1.3125	15.730	0.500	---	---	---
ABUT 2	G1	2.25	24.0	24.0	2.5625	14.875	2.0000	0.5625	1.625	1.250	11.625	0.375	1.5000	11.605	0.500	0.2500	2.625	14.875
	G2	2.25	24.0	24.0	2.5625	14.875	2.0000	0.5625	1.625	1.250	11.625	0.375	1.5000	11.605	0.500	0.2500	2.625	14.875
	G3	2.25	24.0	24.0	2.5625	14.875	2.0000	0.5625	1.625	1.250	11.625	0.375	1.5000	11.605	0.500	0.2500	2.625	14.875
	G4	2.25	24.0	24.0	2.5625	14.875	2.0000	0.5625	1.625	1.250	11.625	0.375	1.5000	11.605	0.500	0.2500	2.625	14.875

POT BEARING DIMENSION TABLE (INCHES)

LOCATION	GIRDER MARK	BEARING TYPE	GUIDE BARS AA	BB	CC	DD	EE	FF	GG	PTFE HH	II	JJ	KK	STAINLESS STEEL LL	MM	NN	OO	BEARING HEIGHT PP
ABUT 1	G1	LONGIT. GUIDED	2.875	2.875	21.500	0.3125	0.4375	0.3125	7.684	10.125	0.1875	2.125	14.625	21.375	11.125	21.375	2.375	10.6250
	G2	LONGIT. GUIDED	2.875	2.875	21.500	0.3125	0.4375	0.3125	7.684	10.125	0.1875	2.125	14.625	21.375	11.125	21.375	2.375	10.6250
	G3	LONGIT. GUIDED	2.875	2.875	21.500	0.3125	0.4375	0.3125	7.684	10.125	0.1875	2.125	14.625	21.375	11.125	21.375	2.375	10.6250
	G4	LONGIT. GUIDED	2.875	2.875	21.500	0.3125	0.4375	0.3125	7.684	10.125	0.1875	2.125	14.625	21.375	11.125	21.375	2.375	10.6250
PIER 1	G1	TRANS. GUIDED	3.625	3.625	27.500	0.5000	0.6250	0.5000	10.371	13.750	0.1875	3.875	20.000	27.375	14.750	27.375	3.125	13.1875
	G2	TRANS. GUIDED	3.625	3.625	27.500	0.5000	0.6250	0.5000	10.371	13.750	0.1875	3.875	20.000	27.375	14.750	27.375	3.125	13.1875
	G3	TRANS. GUIDED	3.625	3.625	27.500	0.5000	0.6250	0.5000	10.371	13.750	0.1875	3.875	20.000	27.375	14.750	27.375	3.125	13.1875
	G4	TRANS. GUIDED	3.625	3.625	27.500	0.5000	0.6250	0.5000	10.371	13.750	0.1875	3.875	20.000	27.375	14.750	27.375	3.125	13.1875
PIER 2	G1	NON-GUIDED	---	---	---	---	---	---	---	15.500	---	---	---	22.750	22.750	---	---	8.6875
	G2	NON-GUIDED	---	---	---	---	---	---	---	15.500	---	---	---	22.750	22.750	---	---	8.6875
	G3	NON-GUIDED	---	---	---	---	---	---	---	15.500	---	---	---	22.750	22.750	---	---	8.6875
	G4	NON-GUIDED	---	---	---	---	---	---	---	15.500	---	---	---	22.750	22.750	---	---	8.6875
ABUT 2	G1	LONGIT. GUIDED	2.875	2.875	24.00	0.3125	0.4375	0.3125	7.684	10.125	0.1875	2.125	14.625	23.875	11.125	23.875	2.375	10.6250
	G2	LONGIT. GUIDED	2.875	2.875	24.00	0.3125	0.4375	0.3125	7.684	10.125	0.1875	2.125	14.625	23.875	11.125	23.875	2.375	10.6250
	G3	LONGIT. GUIDED	2.875	2.875	24.00	0.3125	0.4375	0.3125	7.684	10.125	0.1875	2.125	14.625	23.875	11.125	23.875	2.375	10.6250
	G4	LONGIT. GUIDED	2.875	2.875	24.00	0.3125	0.4375	0.3125	7.684	10.125	0.1875	2.125	14.625	23.875	11.125	23.875	2.375	10.6250

Figure 32 - Penn Turnpike bearing schedules

Parameters for bearings could be defined to reflect the dimension tables in the plans as shown in Figure 32 which is a more typical scenario where such dimensions vary. However, this is unnecessary for this particular bridge as all dimensions are the same for each pot bearing type. If parameters were to be defined, IfcRelAssociatesConstraint would link the IfcMechanicalFastenerType to an IfcObjective having ObjectiveQualifier set to PARAMETER. An IfcMetric may have DataValue set to IfcTable, where each IfcTableColumn corresponds to a defined metric. To link dimensions and placements of geometric shapes, IfcMetric is used for each attribute (identified by IfcMetric.ReferencePath) and calculates the value according to IfcMetric.DataValue set to an IfcAppliedValue based on operations of other values (IfcReference) and literals (IfcMeasureWithUnit).

Occurrences of bearings are captured with IfcMechanicalFastener, linked to the IfcMechanicalFastenerType definition using IfcRelDefinesByType. Each bearing occurrence is placed within the bridge structure using IfcRelContainedInSpatialStructure as shown in Figure 33.

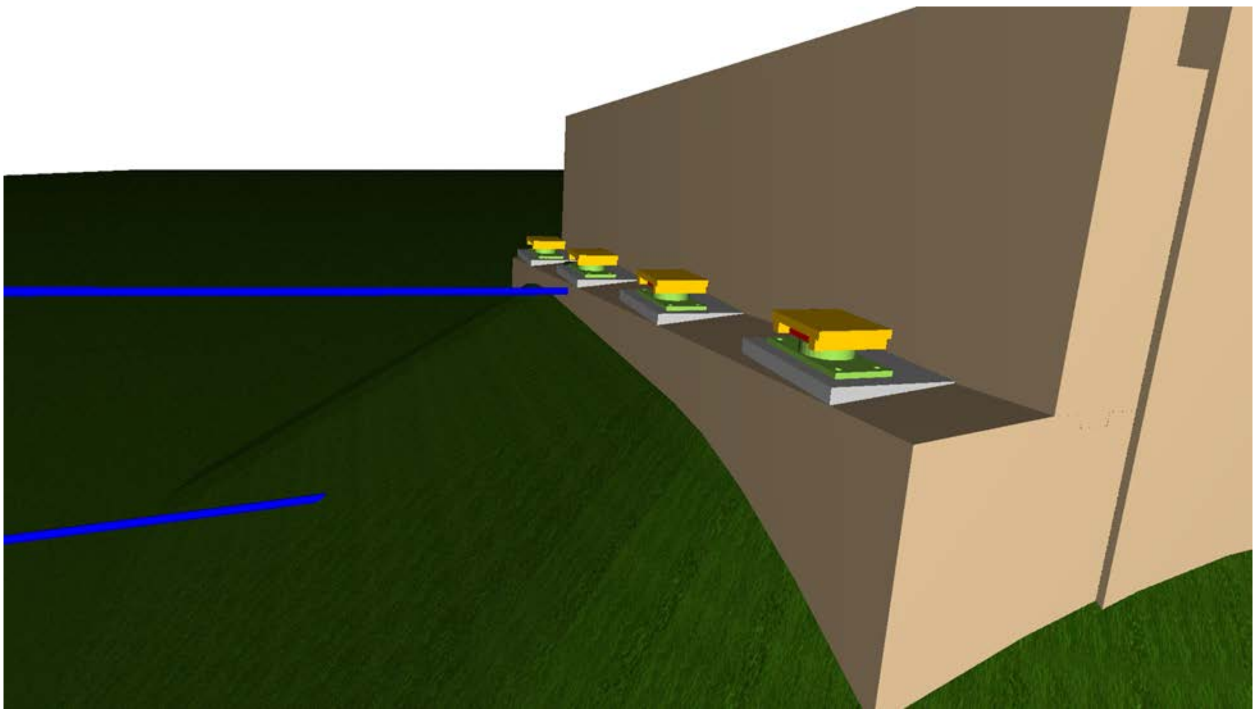


Figure 33 – Penn Turnpike Bearing occurrences model

Bearing connectivity is captured with IfcRelConnectsWithRealizingElements, where the RelatingElement refers to the bridge pier or abutment (IfcCivilElement), RelatedElement refers to the girder (IfcBeam), and RealizingElements refers to the IfcMechanicalFastener. The ConnectionType attribute may indicate the structural behavior informally.

Each bearing may be linked to a corresponding analytical element in a structural model using IfcRelAssignsToProduct where RelatingProduct refers to the IfcMechanicalFastener and RelatedObjects refers to an IfcStructuralPointConnection. The IfcStructuralPointConnection instance defines fixed or free restraints using IfcBoundaryNodeCondition, and links to analytical representations of supported girders using IfcRelConnectsStructuralMember.

1.4.3 OpenBrIM

OpenBrIM classifies a bearing having type "Bearing", and defines a set of properties having the following names:

Number of bearing
Bearing spacing
Distance from centerline of bearing to centerline of support
Centerline of bearing offset from HCL
GUID
Bearing name
Bearing support condition
Bearing type
Bearing station
Bearing bottom elevation
Elastomeric bearing type
Elastomeric bearing shape
Elastomeric pad diameter
Elastomeric pad width
Elastomeric pad height
Elastomeric pad length
Elastomeric pad material designation
Number of steel shims
Steel shim spacing
Steel shim clear cover
Steel shim width
Steel shim height
Steel shim length
Steel shim material designation
Hole diameter
Slot length
Sliding bearing shape
Sliding bearing diameter
Sliding bearing width
Sliding bearing height
Sliding bearing material designation
Sliding surface type

For such properties to be usable in software, the specific data types, syntax, and allowable values need to be defined. For some, this may be inferred by humans interpreting the names, while others are open-ended unless there is further elaboration to constrain the usage. For relating such information to plan detail and structural analysis models, also needed are relationships to geometry, applicable connection information (e.g. supporting pier, supported girder), structural mechanical behavior (e.g. which directions are fixed or free to slide), and relation to a structural analysis model for which loading is defined.

For the sample bridges encountered, geometry may be approximated for the bearings, but there is no data structure available that would make it possible to represent the fillet radius for the pot geometry.

1.5 Girders

The Penn Turnpike bridge contains four sets of steel girders, where the flange and web dimensions vary along the alignment at discrete intervals. The Van White bridge is comprised of a concrete box girder where the cross-section dimensions are uniform along spans, adapt linearly as they approach solid pier caps, and twist according to a reference super-elevation.

1.5.1 Bentley OpenBridge

The Bentley LEAP Bridge Steel application enables specification of girder lateral positioning and longitudinal partitioning as shown in Figure 34. For the sample bridge, the available parameters sufficiently described girder geometry.

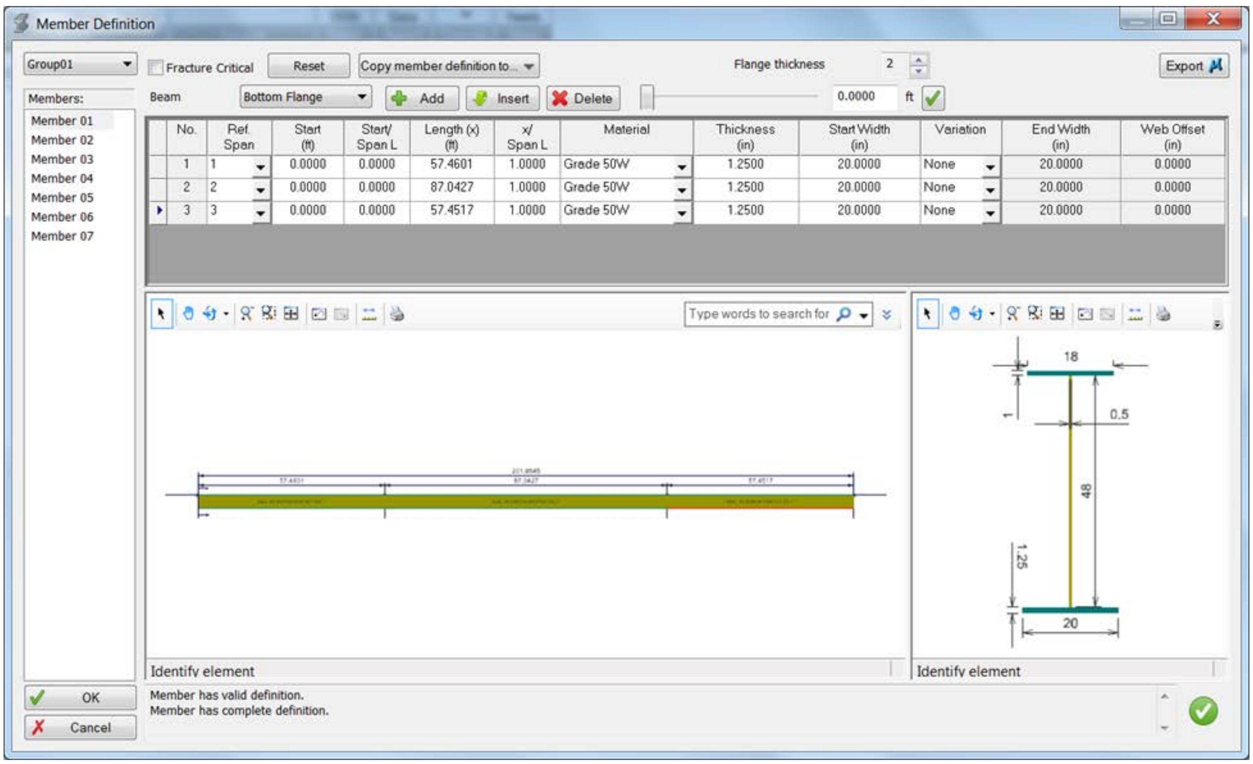


Figure 34 - Girder for Bentley

1.5.2 Industry Foundation Classes

Girders are modelled as IfcElementAssembly having PredefinedType set to GIRDER, and decomposed into girder segments modelled as IfcBeam with IfcMaterialProfileSet indicating the material and cross-section. Steel girders are commonly represented as IfcAsymmetricIShapeProfileDef, while precast concrete girders may use custom cross-sections represented as IfcArbitraryClosedProfileDef.

1.5.2.1 Penn Turnpike Bridge

Beam geometry may be generically described by applying the profile to an alignment curve using IfcFixedReferenceSweptAreaSolid or IfcSurfaceCurveSweptAreaSolid. Alternatively, straight segments may use IfcExtrudedAreaSolid, and segments with tapered sections may use IfcExtrudedAreaSolidTapered (for single taper) or IfcSectionedSpine (for multiple tapers).

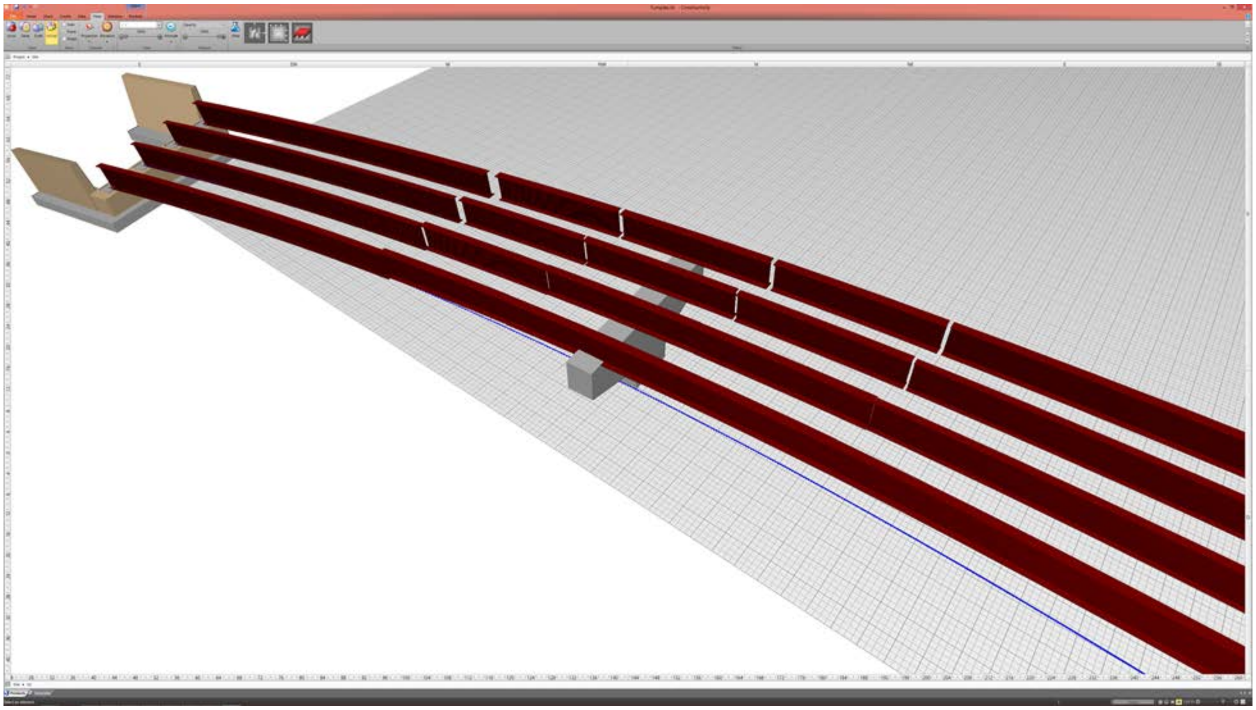


Figure 35 – Penn Turnpike girder model

As shown in Figure 35, the girders are split into segments according to defined splices. The gaps in the illustration have been added to visually differentiate (and would not be visible from a distance in the actual model).

For describing a connected series of beams along a single girder line, an IfcBeam may be used to describe each segment along each girder line. As the profiles (e.g. flange width, flange thickness) and other dimensions may vary along the alignment, such parameters may be described within a table (IfcTable) applied as a constraint (IfcRelAssociatesConstraint) at the outer IfcElementAssembly – such table usage then also provides the information in a form typically shown on plans.

Each IfcBeam is placed relative to the IfcAlignment using IfcRelPositions, where GridOrdinates identify the working point of the grid line (IfcAlignmentAxis) and the starting station offset (IfcAlignmentStation), and EndGridOrdinates identify the same working point (IfcAlignmentAxis) but a following station offset (IfcAlignmentStation). The TransformType at IfcRelPositions is set to WARP to reflect that each segment is to be fabricated according to the curve. ProfileOrdinates at IfcRelPositions are not used, as the cross-sections of the girders are independent of the alignment curve (unlike bridge deck segments).

Camber for beams is captured by traversing the IfcStructuralCurveMember assigned to each IfcBeam. Camber for specific load cases may be derived by an assigned analytical model as described further below.

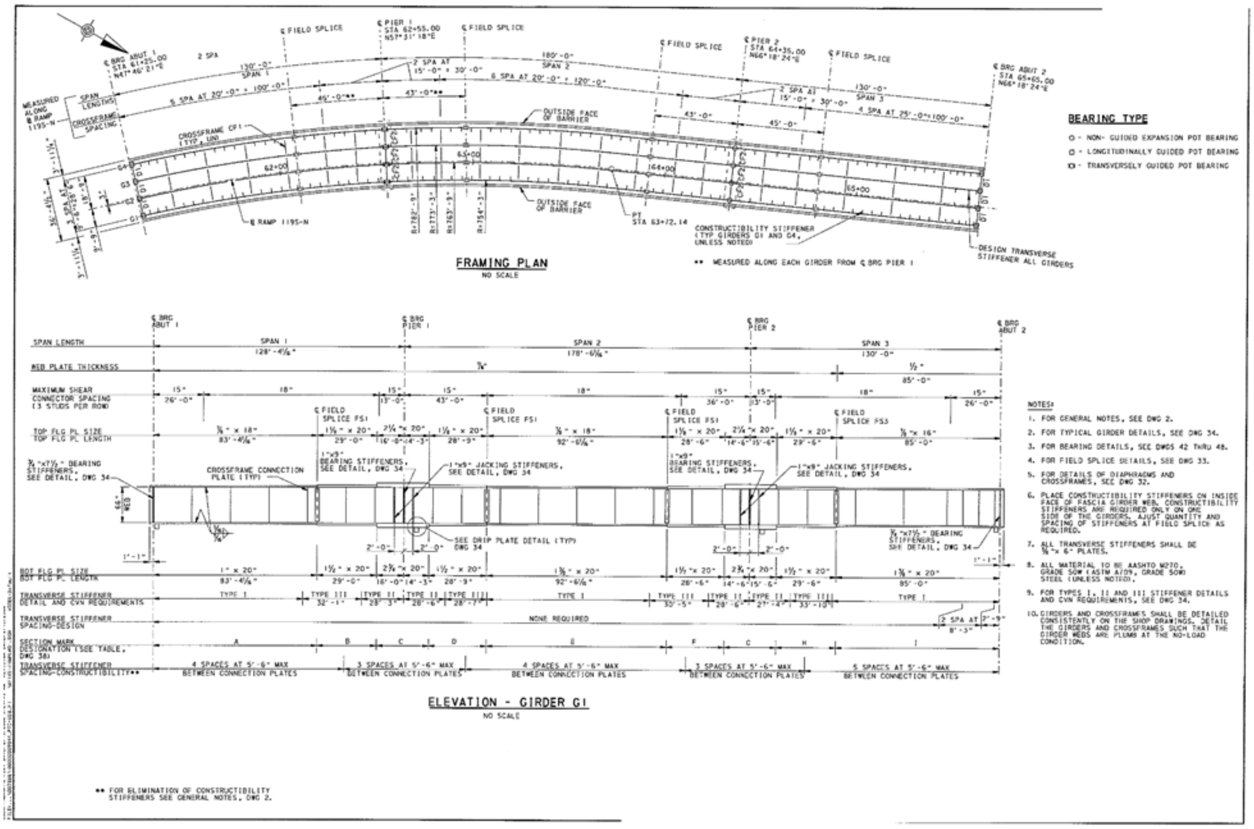
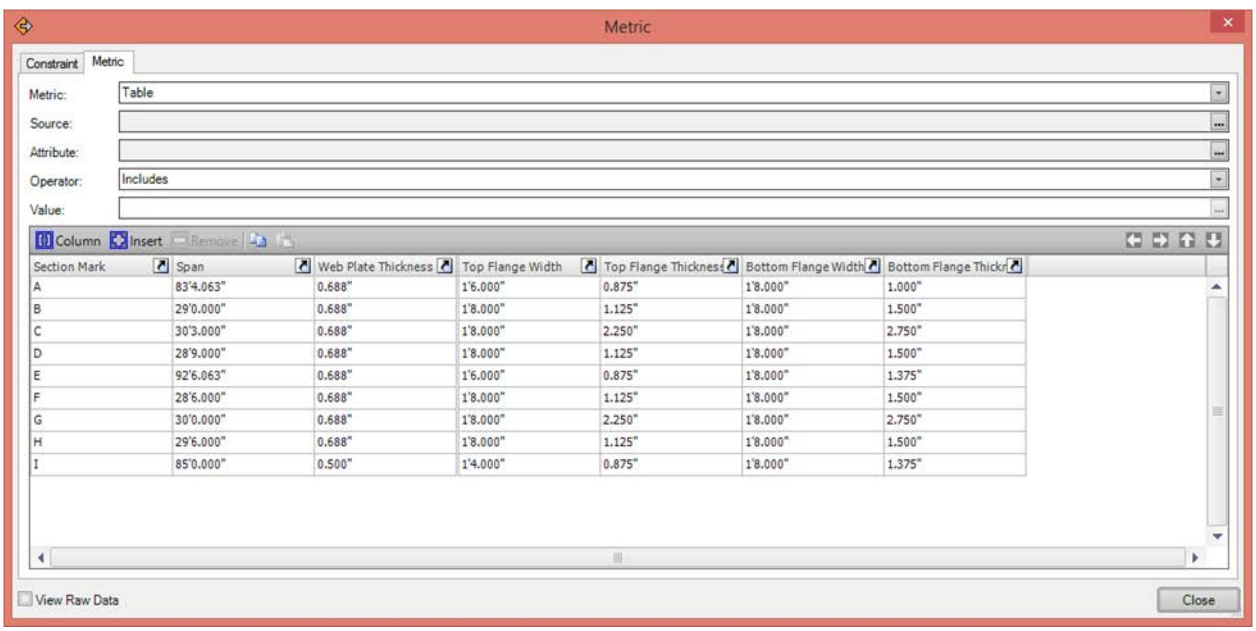


Figure 36 - Penn Turnpike Girder 1 framing plan

As shown in Figure 36, the parameters may vary within different sections along girder spans. While the underlying geometry captures the values explicitly, the girder may optionally be described parametrically with use of an IfcTable, as shown in Figure 37.



Section Mark	Span	Web Plate Thickness	Top Flange Width	Top Flange Thicknes	Bottom Flange Width	Bottom Flange Thickn
A	83'4.063"	0.688"	1'6.000"	0.875"	1'8.000"	1.000"
B	29'0.000"	0.688"	1'8.000"	1.125"	1'8.000"	1.500"
C	30'3.000"	0.688"	1'8.000"	2.250"	1'8.000"	2.750"
D	28'9.000"	0.688"	1'8.000"	1.125"	1'8.000"	1.500"
E	92'6.063"	0.688"	1'6.000"	0.875"	1'8.000"	1.375"
F	28'6.000"	0.688"	1'8.000"	1.125"	1'8.000"	1.500"
G	30'0.000"	0.688"	1'8.000"	2.250"	1'8.000"	2.750"
H	29'6.000"	0.688"	1'8.000"	1.125"	1'8.000"	1.500"
I	85'0.000"	0.500"	1'4.000"	0.875"	1'8.000"	1.375"

Figure 37 - Girder parameters

Within the IfcTable, each column is represented by IfcTableColumn and may have mappings described as shown in Figure 38.

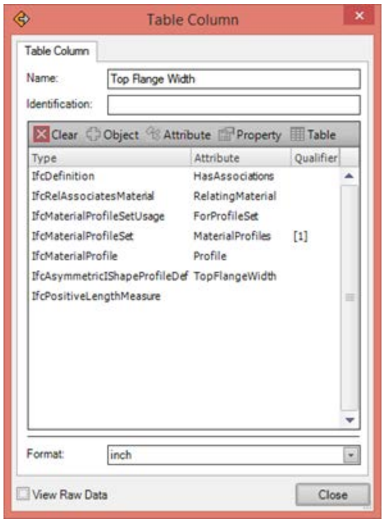


Figure 38 - Table column mapping

Such mappings are captured as a chain of IfcReference structures defined on IfcTableColumn.

The parametric constraints detailed above are considered to be entirely optional, and are only provided as a way to convey the design intent. Of note, most existing IFC software does not support constraint usage, as each have unique ways of deriving the resulting geometry. Thus, the resulting geometry of the examples should be considered as the governing definition in cases of any differences.

Stiffeners are represented using IfcPlate, where a single instance of IfcPlate may be used for each side of the web of component beams to capture all placements, where IfcMappedItem geometry indicates the positioning for each repetition. The spacing of stiffeners may also be captured parametrically within the IfcTable at IfcBeam. The IfcPlate is connected to the component IfcBeam using IfcRelConnectsElements where RelatingElement refers to the IfcBeam.

Shear studs are represented using IfcMember, where a single instance of IfcMember may be used for each IfcBeam component, where IfcMappedItem geometry indicates the positioning for each repetition. The spacing of shear studs may also be captured parametrically within the IfcTable at IfcBeam. The IfcMember is connected to the component IfcBeam using IfcRelConnectsElements where RelatingElement refers to the IfcBeam.

The connection between beams is represented using IfcRelConnectsWithRealizingElements, where the RealizingElements refers to IfcPlate elements for fastening plates on each side, IfcFastener for bolts, and IfcPlate for any flange transition plates. The reason for using this connection relationship specifically (as opposed to just placing the elements) is to be able to derive an IfcStructuralAnalysisModel that captures the beam connectivity.

Camber ordinates may be derived from structural load results related to total dead load. In addition, or in the absence of assigned structural analysis models, camber ordinates to be used for fabrication may be provided in a custom shape representation where IfcShapeRepresentation.Identification="Camber", and IfcShapeRepresentation.Items contains a single IfcIndexedPolyCurve containing coordinates at intervals, where the Z dimension varies. 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 modelled 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). Such relationship usage is already well-defined within the IFC specification.

1.5.2.2 Van White Bridge

Each of the girder spans is encapsulated within an IfcBeam component and linked to an IfcBeamType which defines the composition.

As shown in Figure 39, the IfcBeamType uses Cartesian coordinates and assumes extrusion in a constant direction. The IfcBeam occurrence is placed along an IfcAlignment using the IfcRelPositions relationship with TransformType set to WARP, where the reflected geometry at the occurrence, including all components, is then transformed into the alignment-based coordinate system, such that +X indicate the longitudinal offset along the alignment curve, +Y indicates the lateral offset to the left with rotation as facing in the direction of the alignment curve, and +Z indicates the vertical offset above the alignment curve with rotation. Each aggregated element at IfcBeamType is reflected underneath the IfcBeam occurrence, where IfcMappedItem of each reflected component is adjusted according to the alignment curve.

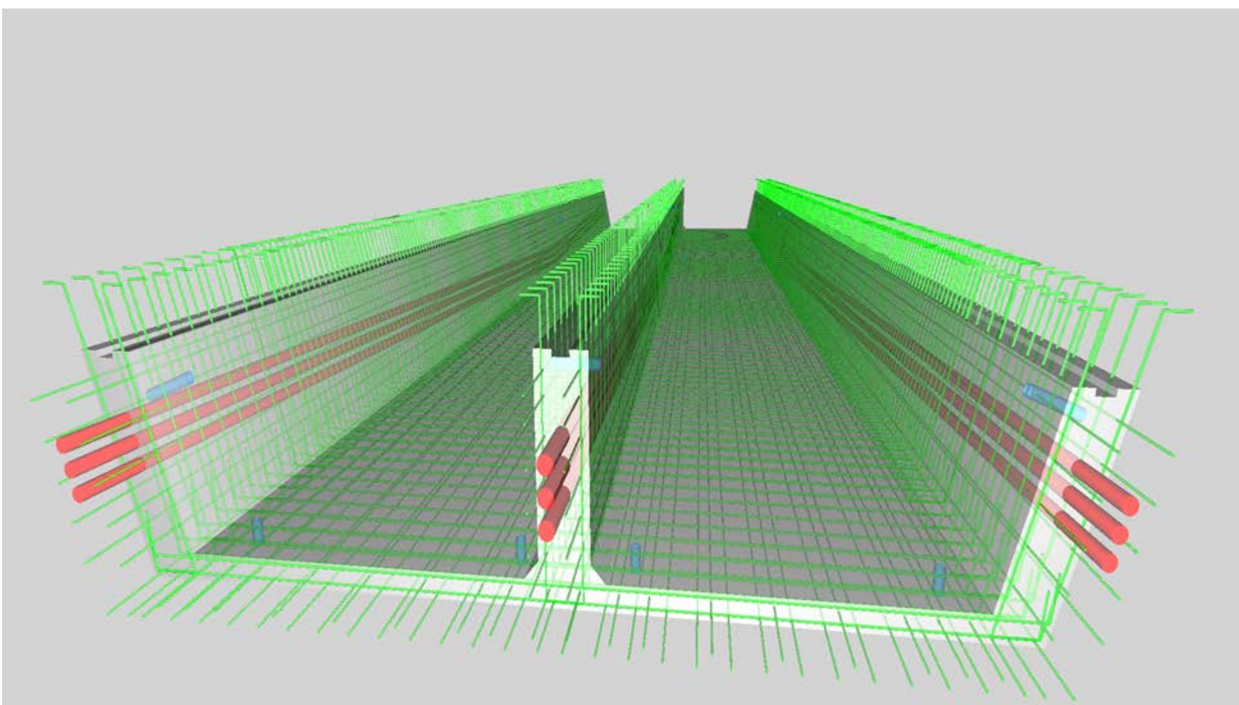


Figure 39 - Van White girder definition model

As shown in Figure 40, each IfcBeam occurrence has transformed geometry by applying the IfcBeamType geometry to the alignment coordinate system. All longitudinal swept segments (specifically IfcExtrudedAreaSolid and IfcExtrudedAreaSolidTapered) are converted to IfcSectionedSpine.

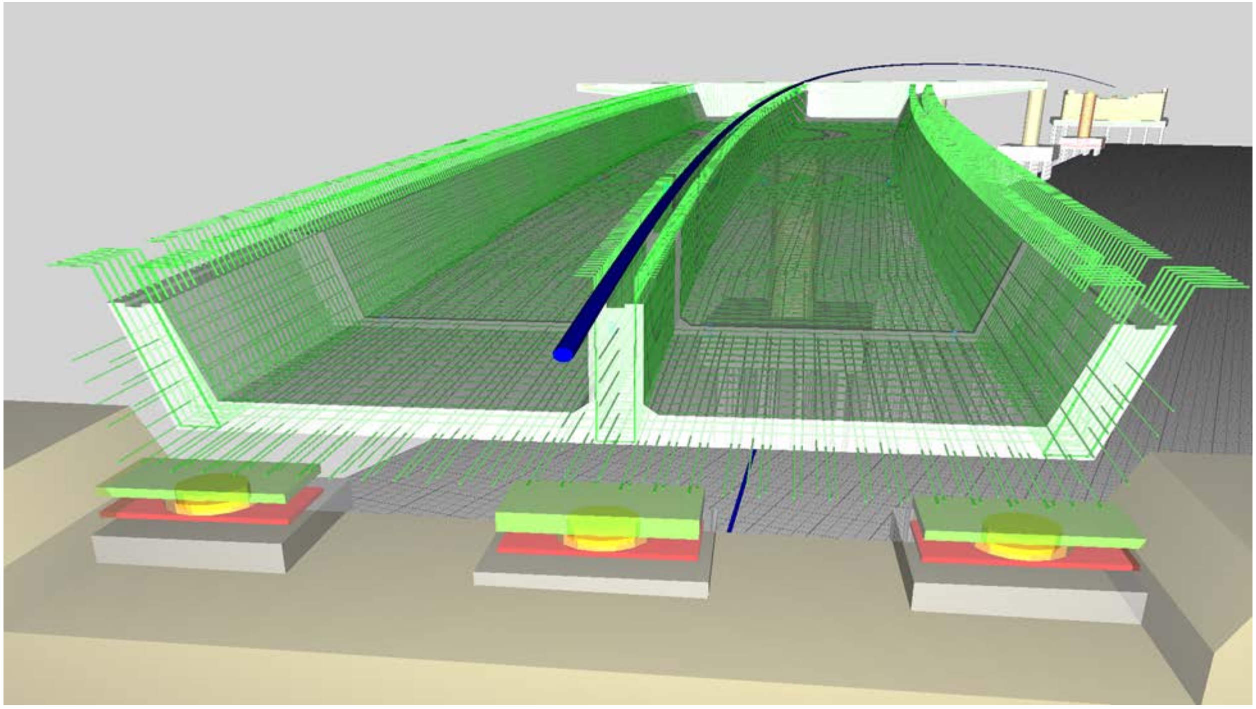


Figure 40 - Van White girder transformation model

The IfcBeam occurrence defining Span 1 is connected from the South Diaphragm component and connected to the Pier 1 Cap Beam, each using the IfcRelConnectsPathElements relationship, where RelatingConnectionType is set to ATEND and RelatedConnectionType is set to ATSTART. This relationship indicates that the elements are joined end-to-start along the longitudinal axis (X) according to the extents of the direct "Body" geometry of the IfcBeam. Note that such extents are NOT impacted by embedded components such as rebar or tendons, which may intersect with connected components. The IfcBeam occurrence is connected to the bridge deck above using the IfcRelConnectsPathElements relationship with both RelatingConnectionType and RelatedConnectionType set to ATPATH.

The IfcBeam defines camber ordinates within its "Camber" representation using IfcRationalBSpline instances corresponding to each girder. Each curve contains three control points, as defined on plans: the head and tail having Z=0, and the point of maximum camber where X indicates the longitudinal offset and Z indicates the vertical camber where positive is up.

The IfcBeamType defines geometry within its "Body" representation using two shapes: an IfcExtrudedAreaSolid for the constant section using IfcArbitraryClosedProfileDef, and an IfcExtrudedAreaSolidTapered for the section near the pier where the soffit slab thickens, also using IfcArbitraryClosedProfileDef. The IfcBeamType is decomposed into components using the IfcRelAggregates relationship, where rebar uses IfcReinforcingBar, tendons use IfcTendon, vent pipes use IfcPipeSegment, drain pipes use IfcPipeSegment, and access panels use IfcDistributionChamberElement. Each tendon group (3 groups of 3) is captured using IfcTendon linked to an IfcTendonType defining geometry using IfcSweptDiskSolid based on IfcRationalBSplineCurve consisting of parabolic segments. Placement of each tendon within the group is indicated using

IfcMappedItem. Vent pipes and drain pipes are captured using IfcPipeSegment linked to IfcPipeSegmentType defining geometry using IfcExtrudedAreaSolid with cross-section of IfcCircleHollowProfileDef. Placement of each pipe along its respective span is indicated using IfcMappedItem.

1.5.3 OpenBrIM

Girders in OpenBrIM may be defined using explicit members for each girder alignment and each girder segment of a constant cross-section. Geometry of each girder segment may be defined according to a polyline cross-section, which is then swept along the alignment curve where the girder is positioned.

Connectivity of girders is not modeled, which would be required to derive structural analysis models and erection sequences.

Representation of camber ordinates is listed as a property, however the specific data types, syntax, and meanings remain to be defined.

- Top line dimension (TL)
- Bottom line dimension (BL)
- Left end cut (LE)
- Right end cut (RE)
- WD1
- WX1
- TBL
- TBR
- BBL
- BBR
- LD
- Camber ordinate

1.6 Framing

Cross-framing between girders may be described using templates of member configurations. Such cross framing is captured within components, using AISC shapes where applicable. For curved alignments where girders are placed at different elevations, members must be placed relative to the girders at each side, for which positioning is defined relative to alignment curves.

1.6.1 Bentley OpenBridge

As shown in Figure 40, Bentley LEAP Bridge Steel contains a configuration wizard for cross-frames/diaphragms, where the placement and types of components may be specified. For the sample bridge, the parameters within this wizard were sufficient to describe the resulting detail.

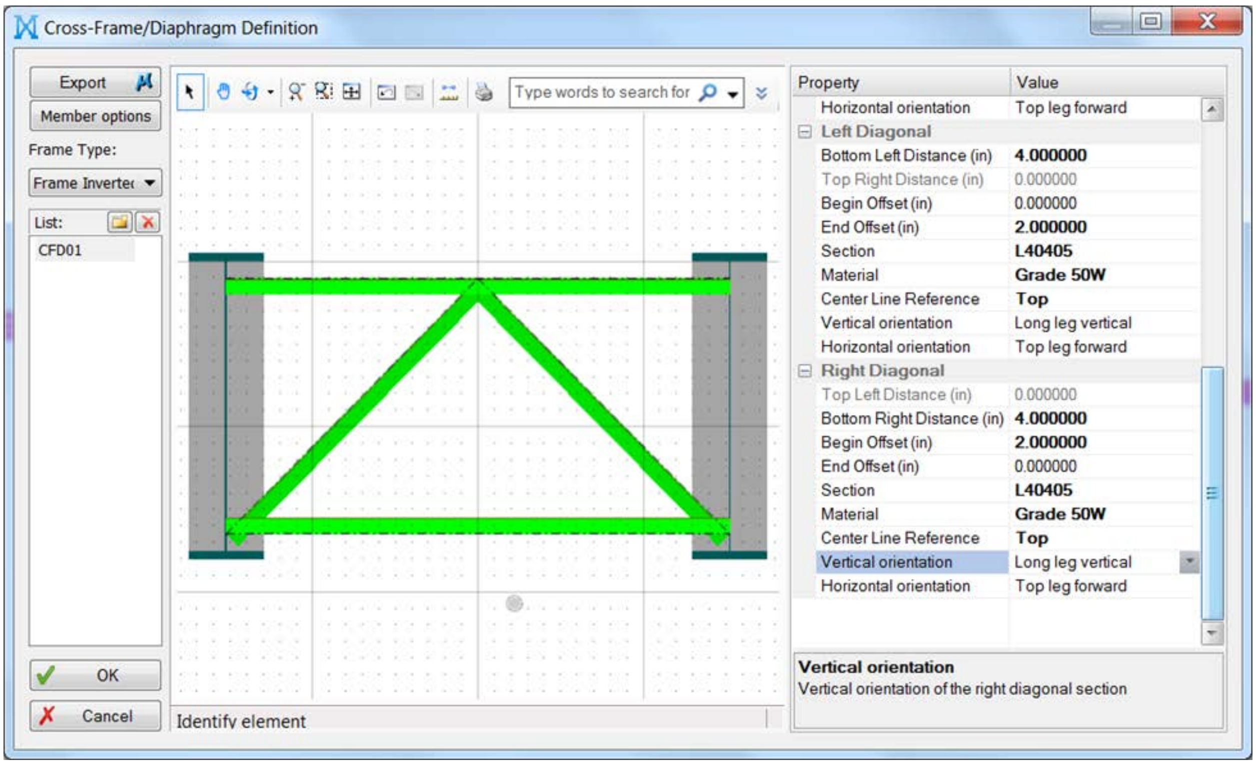


Figure 41 - Cross frame with Bentley

As shown in Figure 41, Each of the members used for the Penn Turnpike bridge sample may be referenced by AISC shape designation, where definitions of all such shapes are included within the software.

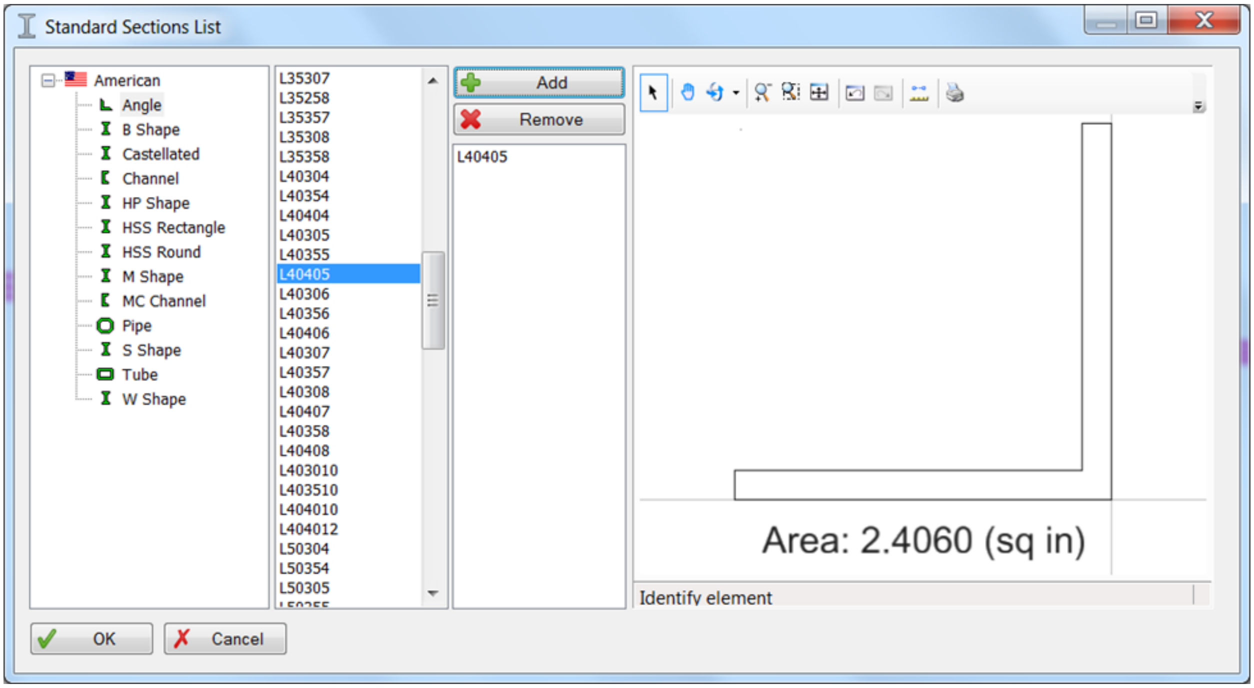


Figure 42 - Member section with Bentley

As shown in Figure 42, specifications of cross-framing may be indicated at intervals along the bridge alignment, where points of interest (POI) are defined with varying positions.

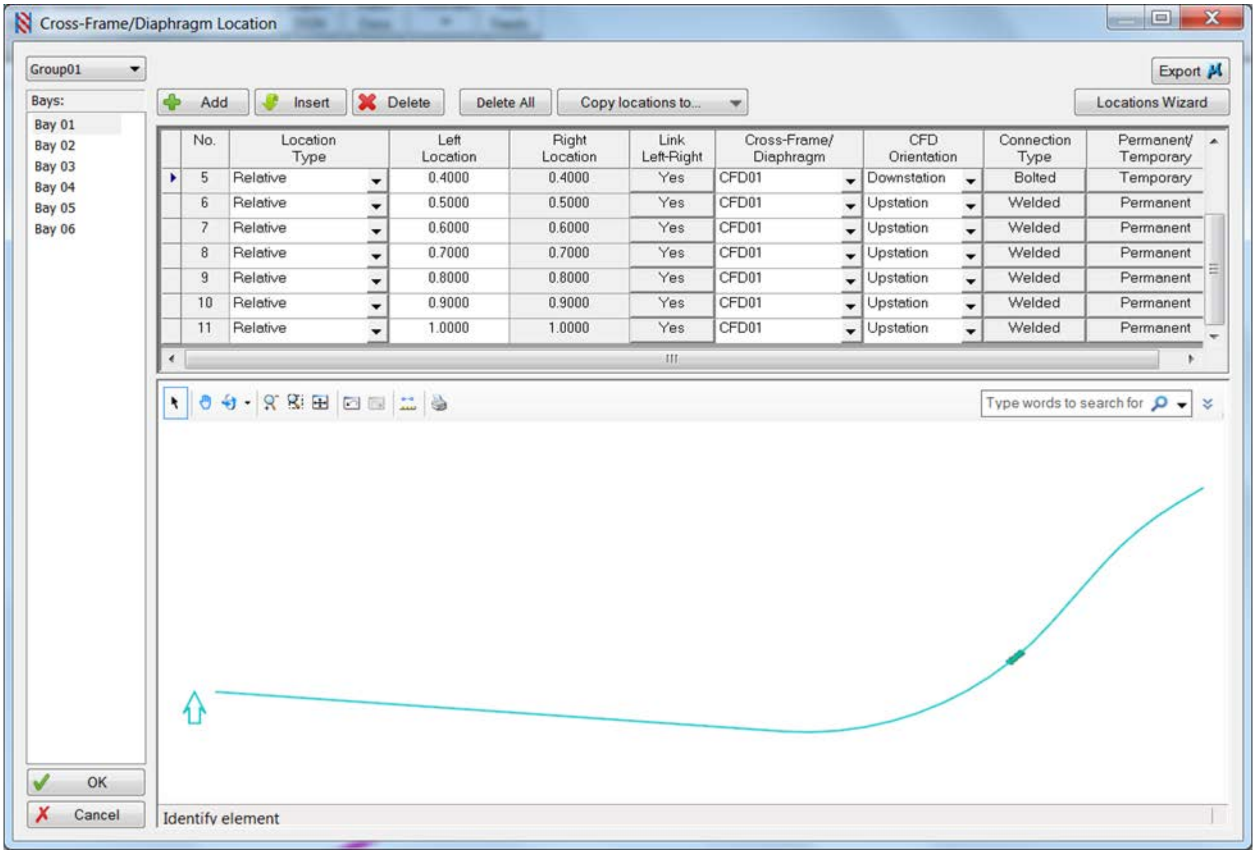


Figure 43 - Framing working points with Bentley

1.6.2 Industry Foundation Classes

Cross-frame definitions are modelled as IfcElementAssemblyType having PredefinedType set to BRACED_FRAME. The IfcElementAssemblyType is aggregated into IfcMember and IfcPlate elements.

Steel angles (AISC L shapes) are captured using IfcMember having geometry represented by IfcExtrudedAreaSolid with IfcLShapeProfileDef. Steel plates are captured using IfcPlate having geometry represented by IfcExtrudedAreaSolid with IfcRectangleProfileDef.

Members and plates within the cross-frame are connected using IfcRelConnectsElements, where the connection geometry is indicated using IfcConnectionSurfaceGeometry, which enables analysis from derived structural analysis models (IfcStructuralAnalysisModel), defines geometric constraints for skewing along the alignment curve, and describes required welding for fabrication.

Each occurrence of a cross-frame is modelled as IfcElementAssembly, reflecting the same composition as at the IfcElementAssemblyType definition, but with geometry skewed according to the alignment positioning. IfcRelPositions is used to uniquely position each cross-frame, where GridOrdinates refers to the IfcAlignmentAxis to the left, and EndGridOrdinates refers to the IfcAlignmentAxis to the right (facing in the increasing direction of the IfcAlignment). Both GridOrdinates and EndGridOrdinates refer to the same IfcAlignmentStation. ProfileOrdinates are not used, as cross-framing is lateral and the geometry cross-sections are independent of such working points.

Such cross-framing is then instantiated as object occurrences according to repetition intervals, where each occurrence has unique connectivity relationships with corresponding girder segments. While such discrete modeling results in larger files sizes, individual elements need to be captured for any structural analysis usage, and pragmatically for compatibility and for defining exact geometry to avoid any software implementation differences in calculations or round-off.

Parameterization of such framing requires referencing other instances according to connectivity, particularly for bridges with super-elevations where adjacent girders are at different elevations. To reference values at connected elements, IfcReference expressions must traverse connectivity relationships such as:
"\IfcMember.ConnectedFrom[1]\IfcRelConnectsElements.RelatingElement\IfcBeam.Representation\IfcProductDefinitionShape.Representation['Axis']\IfcShapeRepresentation.Items[1]\IfcIndexedPolyCurve.PointOnCurve[@Offset].Z".

To adjust for offsets from web thickness or flange thickness, the beam profile may be referenced such as: "\IfcMember.ConnectedFrom[1]\IfcRelConnectsElements.RelatingElement\IfcBeam.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles[1]\IfcIShapeProfileDef.WebThickness.

Performing arithmetic operations on multiple references may be accomplished using IfcAppliedValue, where ArithmeticOperation may indicate ADD or SUBTRACT, and Components includes the list of referenced parameters.

To define arrays of framing at offsets along curves, each framing set (between two specific girders) must be defined explicitly once, and then may be duplicated along a repetition path using a constraint relationship, where an IfcTable may be used for specific offsets along the alignment, or IfcAppliedValue may be used to indicate positioning of each instance relative to a constant spacing using an ArithmeticOperation of MULTIPLY.

1.6.2.1 Penn Turnpike Bridge

Figure 43 illustrates a sample cross-frame from the Penn Turnpike bridge model.

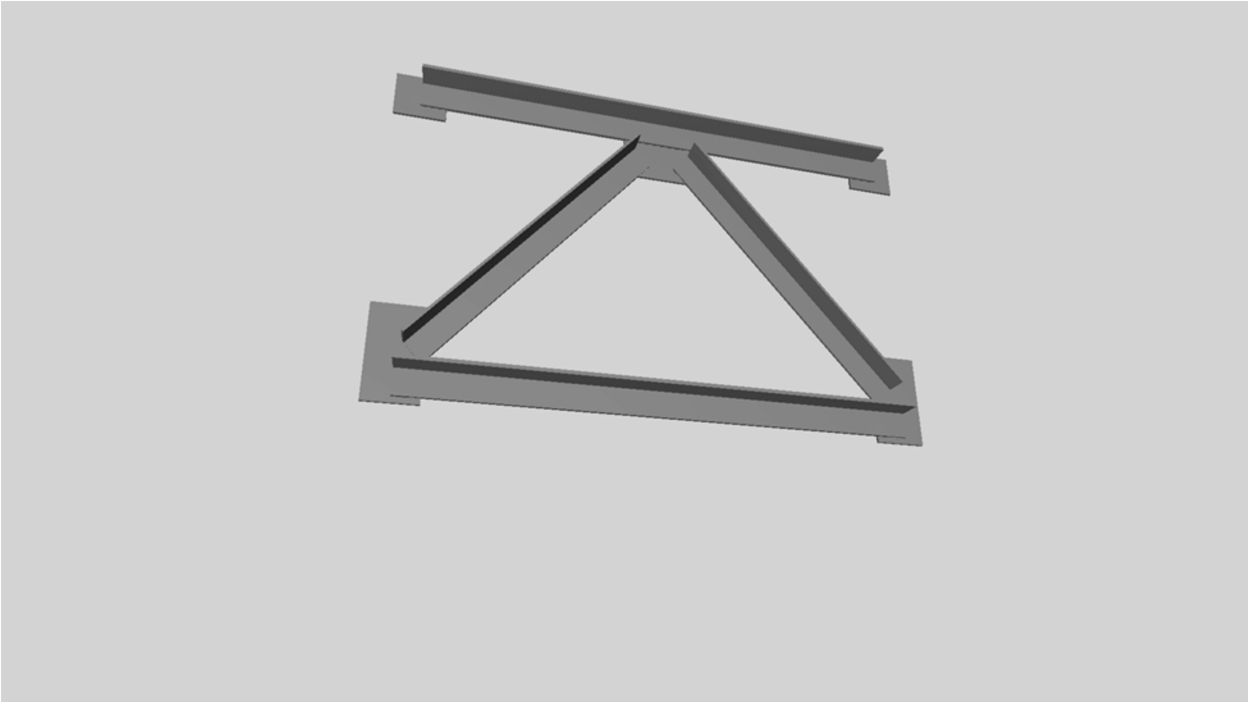


Figure 44 - Penn Turnpike framing model

1.6.3 OpenBrIM

Member components may be modeled as extruded area solids using polylines. For L-Shape profiles (AISC angle shapes), the fillet and chamfer radii cannot be explicitly modeled, but approximated as polylines.

For this particular example where adjacent girders are be at different elevations of variable offsets along the alignment curve, positioning of members of the cross-section must reference the connecting girders. Such referencing must also take into consideration the cross-sections of connecting girders to ensure clearance for the variations of web thickness and flange thicknesses. It is not clear how such referencing may be done in the absence of connection relationships – the syntax would need to be defined to indicate how to reference parameters at connecting elements to identify points of interest (POI).

1.7 Deck

Bridge deck design is driven by the bridge alignment curve, design lanes (to support traffic lanes, shoulders, sidewalks, etc.) with super-elevations accommodating design speeds, and live loads from traffic using AASHTO design standards. In addition, the underlying girders and spacing also influence the deck design, as the concrete may protrude to provide a contact area at constant elevation. Figure 44 illustrates the plans of the bridge deck for the Penn Turnpike sample bridge.

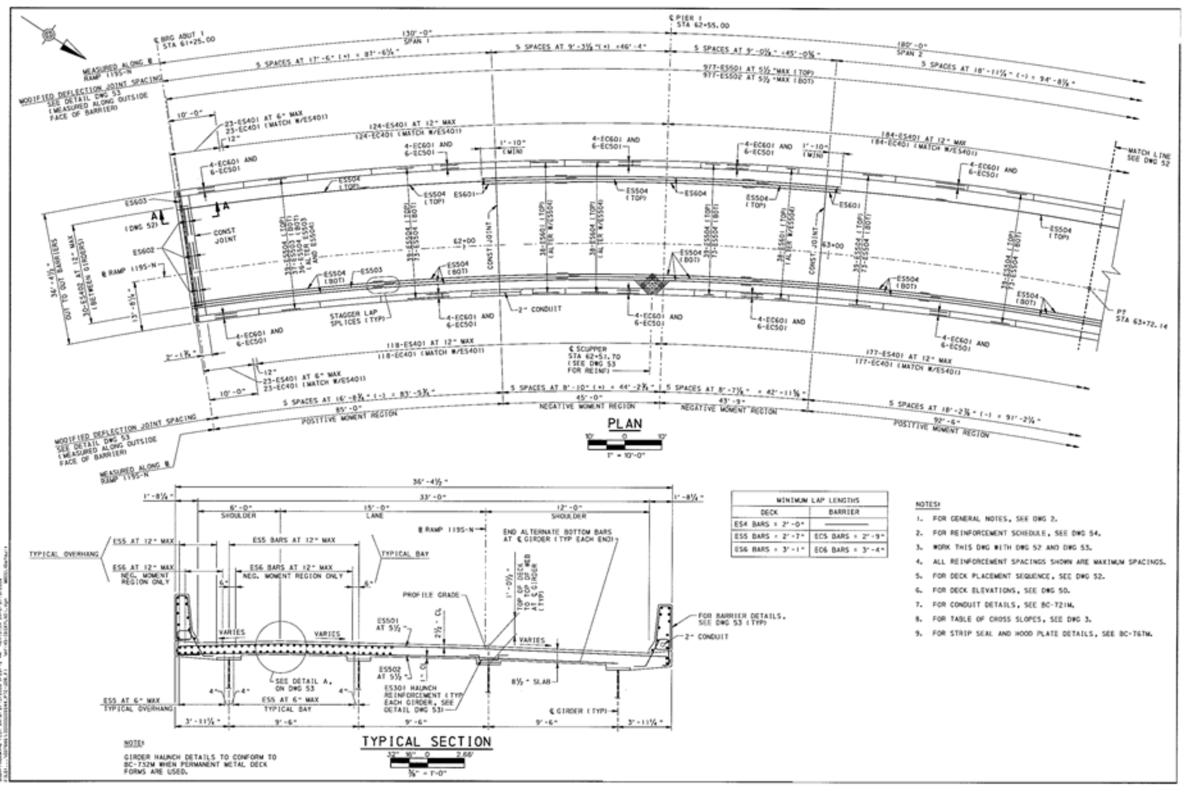


Figure 45 - Deck plan example

1.7.1 Bentley OpenBridge

Bentley LEAP Bridge Steel models bridge decks starting with lane widths and angles, as shown in Figure 45. If the vertical embankment varies along the alignment, separate cross-sections must be defined separately at intervals, where linear interpolation is presumed in between.

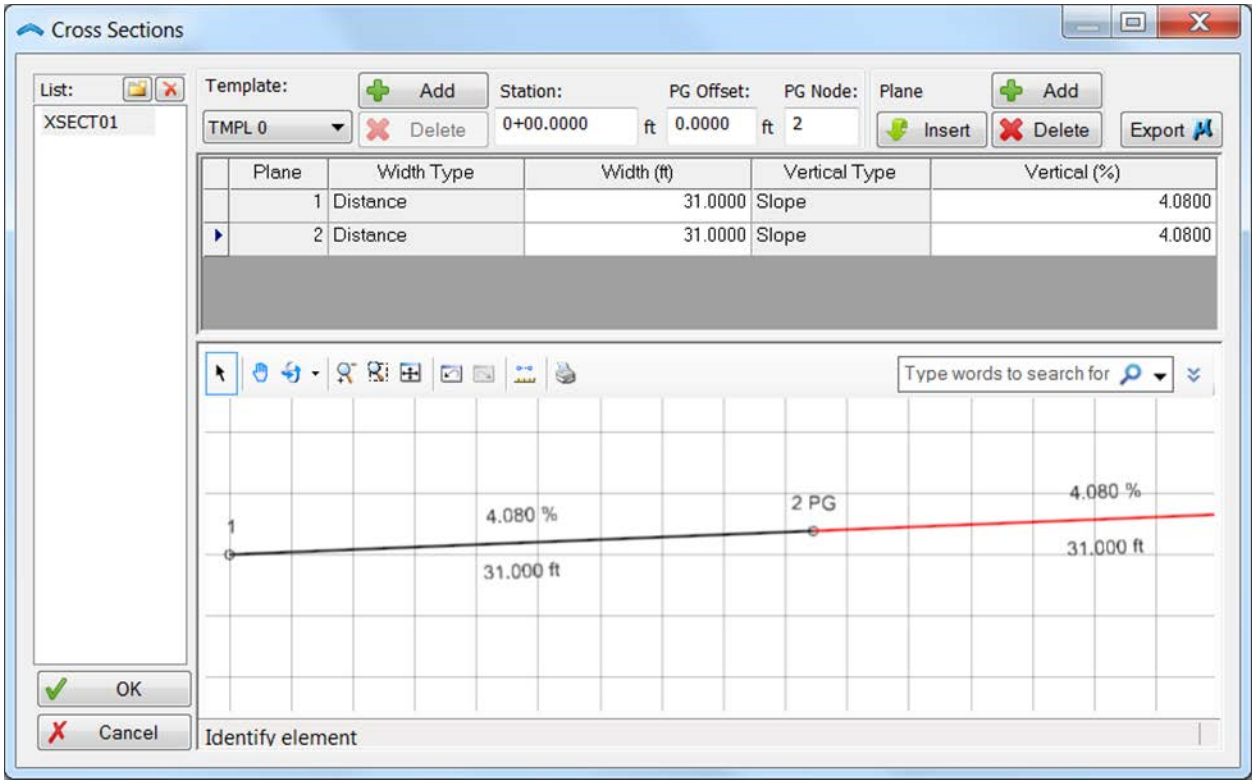


Figure 46 - Deck for Bentley

As shown in Figure 46, slab segments along the alignment are defined according to offsets relative to alignment positioning of connecting elements such as piers.

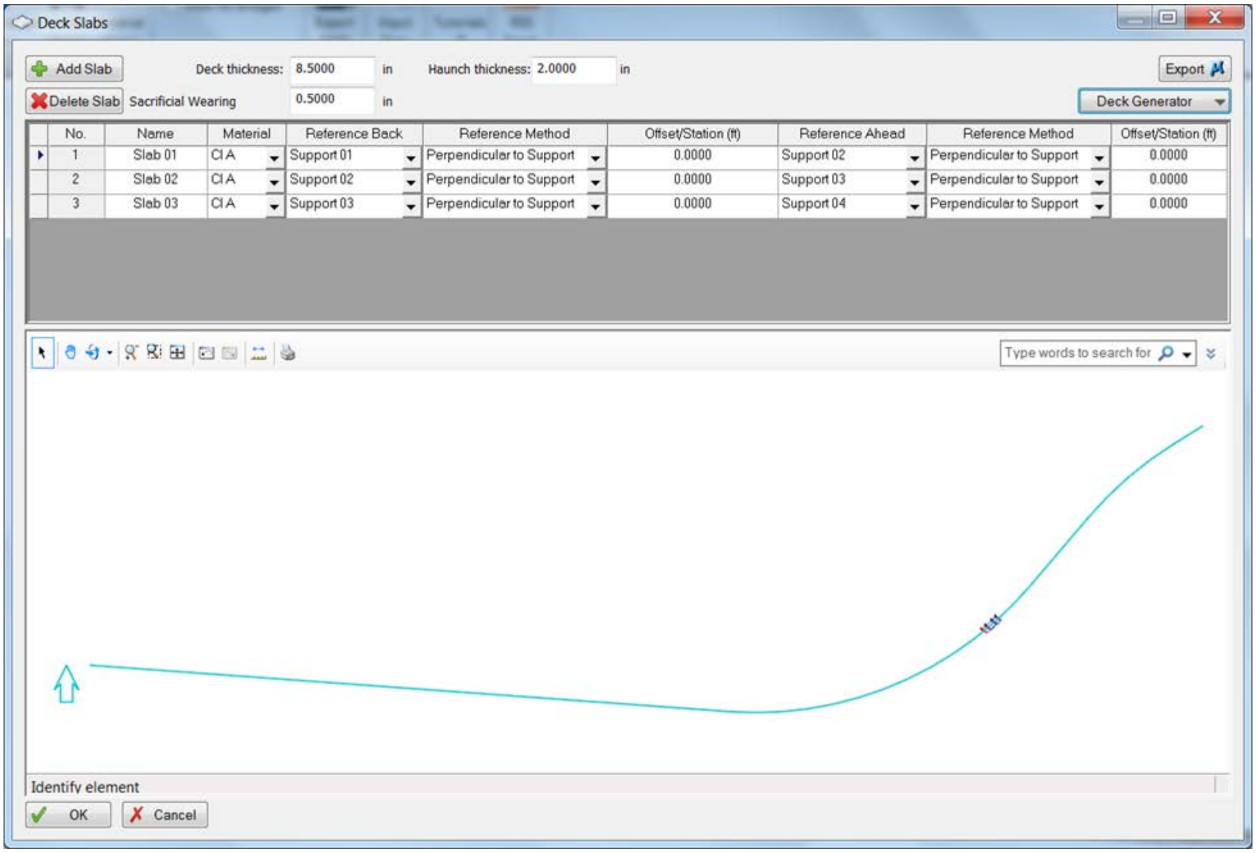


Figure 47 - Deck slabs for Bentley

1.7.2 Industry Foundation Classes

Bridge decks are modelled as IfcElementAssembly having PredefinedType set to SLAB_FIELD. This IfcElementAssembly is aggregated into IfcSlab components for each slab segment (separated by construction joint).

The overall deck cross-section is defined using IfcMaterialProfileSet where the profile of the bridge deck is described using IfcArbitraryClosedProfileDef referring to an IfcPolyline.

Each IfcSlab of the bridge deck is positioned along the alignment curve using IfcRelPositions, where GridOrdinates indicates the starting position, EndGridOrdinates indicates the ending position, and ProfileOrdinates may optionally be used for bridge deck cross-sections that can be described by working points that vary along the alignment.

1.7.2.1 Penn Turnpike Bridge

As shown in Figure 47, the horizontal X coordinates are fixed, while the vertical Y coordinates vary along the alignment. Each row in the schedule corresponds to an *IfcAlignmentStation*. The first two columns correspond to *Tag* and *DistanceAlong* attributes respectively, while each column thereafter corresponds to an *IfcAlignmentAxis* with cells populated from corresponding list positions at *AxisElevations* at *IfcAlignmentStation*.

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

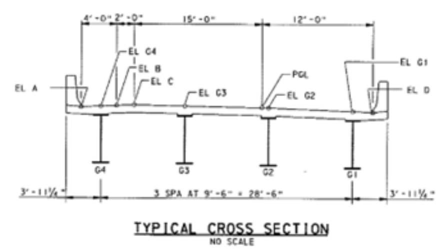
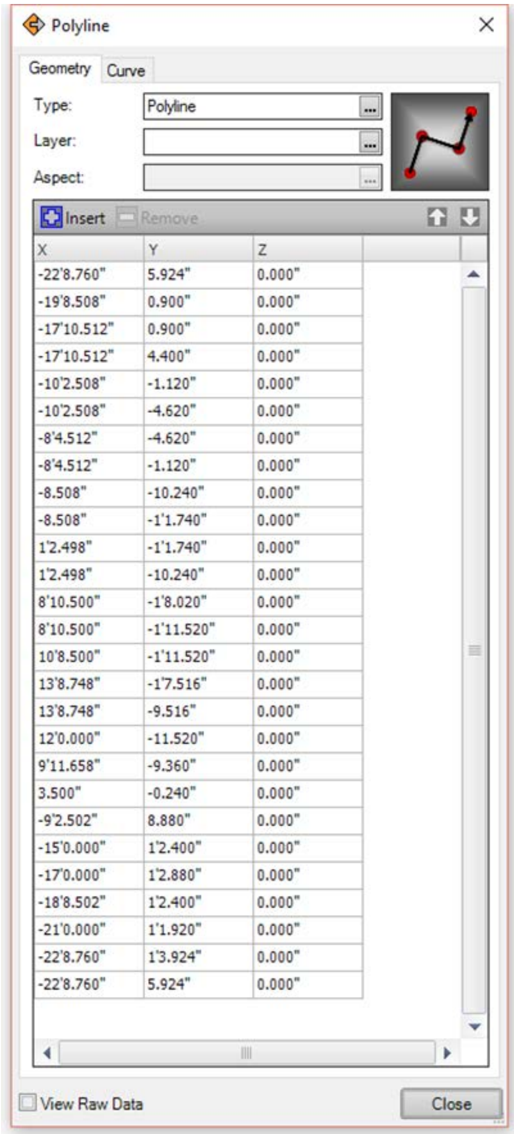


Figure 48 - Penn Turnpike deck profile

For curved alignments, the dimensions of the cross-section may vary along the alignment. To map the vertical elevations of cross-section points to working points of the alignment, the ProfileOrdinates attribute is used at IfcRelPositions, using the values 6, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 9, 9, 9, 5, 4, 1, 3, 8, 7, 2, 6, 6. These one-based indices correspond 1:1 to each IfcCartesianPoint within the IfcPolyline at IfcArbitraryClosedProfileDef as shown in Figure 48, where the value identifies the IfcAlignmentAxis by index at IfcAlignment.Axes. The elevation of each axis is defined at IfcAlignmentStation.AxisElevations corresponding to the same index value.



X	Y	Z
-22'8.760"	5.924"	0.000"
-19'8.508"	0.900"	0.000"
-17'10.512"	0.900"	0.000"
-17'10.512"	4.400"	0.000"
-10'2.508"	-1.120"	0.000"
-10'2.508"	-4.620"	0.000"
-8'4.512"	-4.620"	0.000"
-8'4.512"	-1.120"	0.000"
-8.508"	-10.240"	0.000"
-8.508"	-1'1.740"	0.000"
1'2.498"	-1'1.740"	0.000"
1'2.498"	-10.240"	0.000"
8'10.500"	-1'8.020"	0.000"
8'10.500"	-1'11.520"	0.000"
10'8.500"	-1'11.520"	0.000"
13'8.748"	-1'7.516"	0.000"
13'8.748"	-9.516"	0.000"
12'0.000"	-11.520"	0.000"
9'11.658"	-9.360"	0.000"
3.500"	-0.240"	0.000"
-9'2.502"	8.880"	0.000"
-15'0.000"	1'2.400"	0.000"
-17'0.000"	1'2.880"	0.000"
-18'8.502"	1'2.400"	0.000"
-21'0.000"	1'1.920"	0.000"
-22'8.760"	1'3.924"	0.000"
-22'8.760"	5.924"	0.000"

Figure 49 - Deck dimension table

As the bridge deck of the Penn Turnpike example varies between points along a vertical alignment curve such that the profiles between such points have variable dimensions, the IfcSectionedSpine entity is required to precisely describe such geometry.

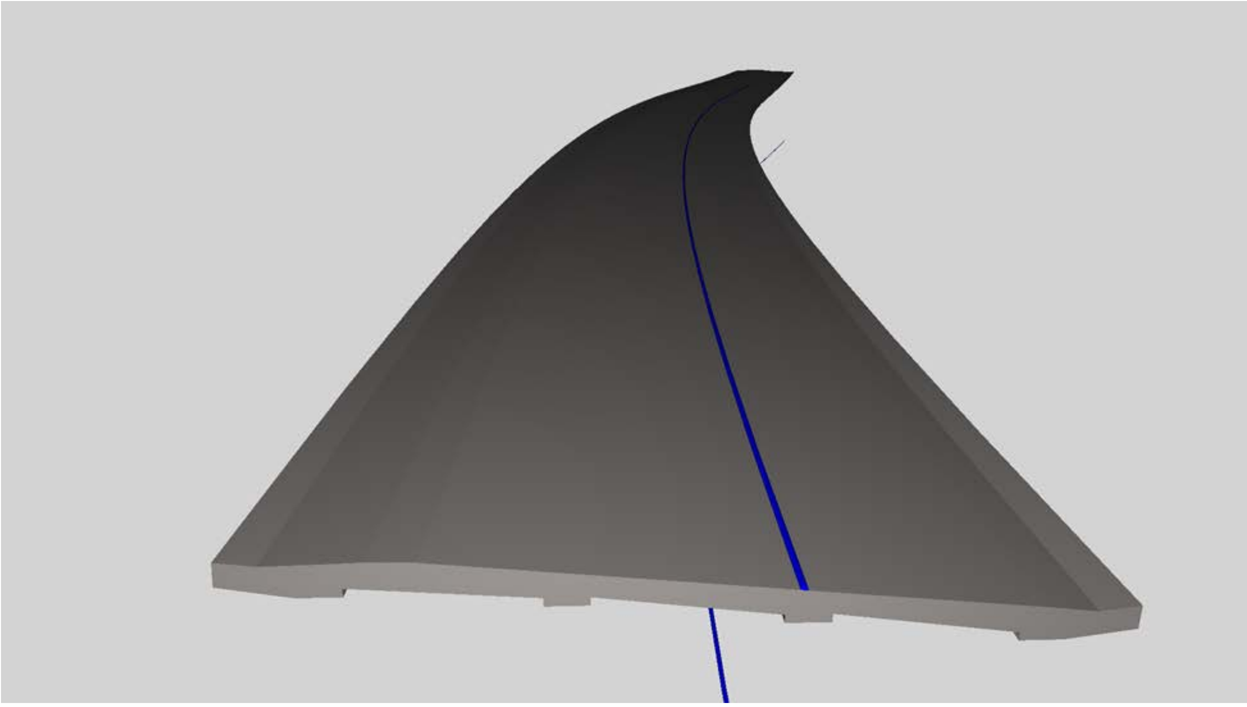


Figure 50 - Penn Turnpike deck model

As IfcSectionedSpine requires usage of IfcCompositeCurve, this cannot directly reference vertical alignment curves; rather such curves must be converted to use IfcPolyline or IfcBSplineCurve subtypes. As documentation of IfcSectionedSpine indicates that the shape between profile segments is undefined – for use in bridges, this would need to be clarified as strictly linear interpolation according to orientations of sequential profiles, where such intervals are defined at granularity required of construction.

Each of the deck segments (IfcSlab) are interconnected using IfcRelConnectsPathElements to indicate sequencing along the alignment axis where the ConnectionGeometry attribute describes construction joint separation.

1.7.2.2 Van White Bridge

For the Van White Memorial Bridge example, a constant profile is used for the span sections within the bridge superstructure, and is rotated along the alignment. The soffit slab thickens at a constant rate near the piers, and the overall profile is solid concrete at the cap beams over the bridge piers.

To accommodate such variation, the superstructure is subdivided into segments as shown in Table 1.

.

Table 1 - Van White deck geometry

Name	Type	Profile Geometry	Solid Geometry
South abutment diaphragm	IfcBeam	IfcArbitraryClosedProfileDef	IfcExtrudedAreaSolidTapered
Span 1	IfcSlab	IfcArbitraryClosedProfileDef	IfcSurfaceCurveSweptAreaSolid
Pier 1 cap beam	IfcBeam	IfcTShapeProfileDef	IfcExtrudedAreaSolidTapered
Span 2	IfcSlab	IfcArbitraryClosedProfileDef	IfcSurfaceCurveSweptAreaSolid
Pier 2 cap beam	IfcBeam	IfcTShapeProfileDef	IfcExtrudedAreaSolidTapered
Span 3	IfcSlab	IfcArbitraryClosedProfileDef	IfcSurfaceCurveSweptAreaSolid
Pier 3 cap beam	IfcBeam	IfcTShapeProfileDef	IfcExtrudedAreaSolidTapered
Span 4	IfcSlab	IfcArbitraryClosedProfileDef	IfcSurfaceCurveSweptAreaSolid
Pier 4 cap beam	IfcBeam	IfcTShapeProfileDef	IfcExtrudedAreaSolidTapered
Span 5	IfcSlab	IfcArbitraryClosedProfileDef	IfcSurfaceCurveSweptAreaSolid
North abutment diaphragm	IfcBeam	IfcArbitraryClosedProfileDef	IfcExtrudedAreaSolidTapered

Each of these sections are connected sequentially using the IfcRelConnectsPathElements relationship to indicate sequencing along the alignment axis where the ConnectionGeometry attribute is set to NULL to indicate continuity of concrete (no construction joint).

As shown in Figure 50, the slab segments (IfcSlab) use a constant cross-section that rotates. The separation between cap beams (IfcBeam) and spans (IfcSlab) occurs where the cross-section begins to taper (soffit slab thickens) near the piers and abutments.

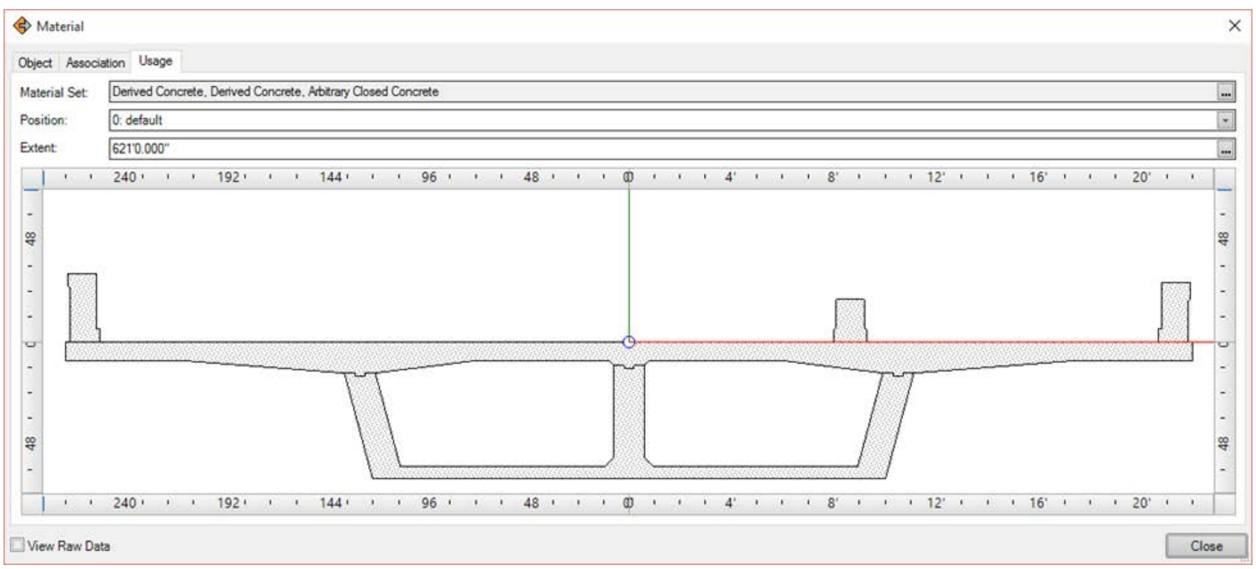


Figure 51 - Van White deck profile

While the deck span segments could also be modelled using IfcSectionedSpine at intervals acceptable for construction, the curvature may be exactly described using IfcSurfaceCurveSweptAreaSolid. This geometry defines both a directrix based on any IfcCurve (including IfcBSplineCurve for precise definition for any arbitrary curve), and a reference IfcSurface (including IfcBSplineSurface for precise definition of any arbitrary surface). For this particular example, an IfcBSplineSurface may be constructed consisting of a ruled surface having two arrays of points along the bridge deck alignment.

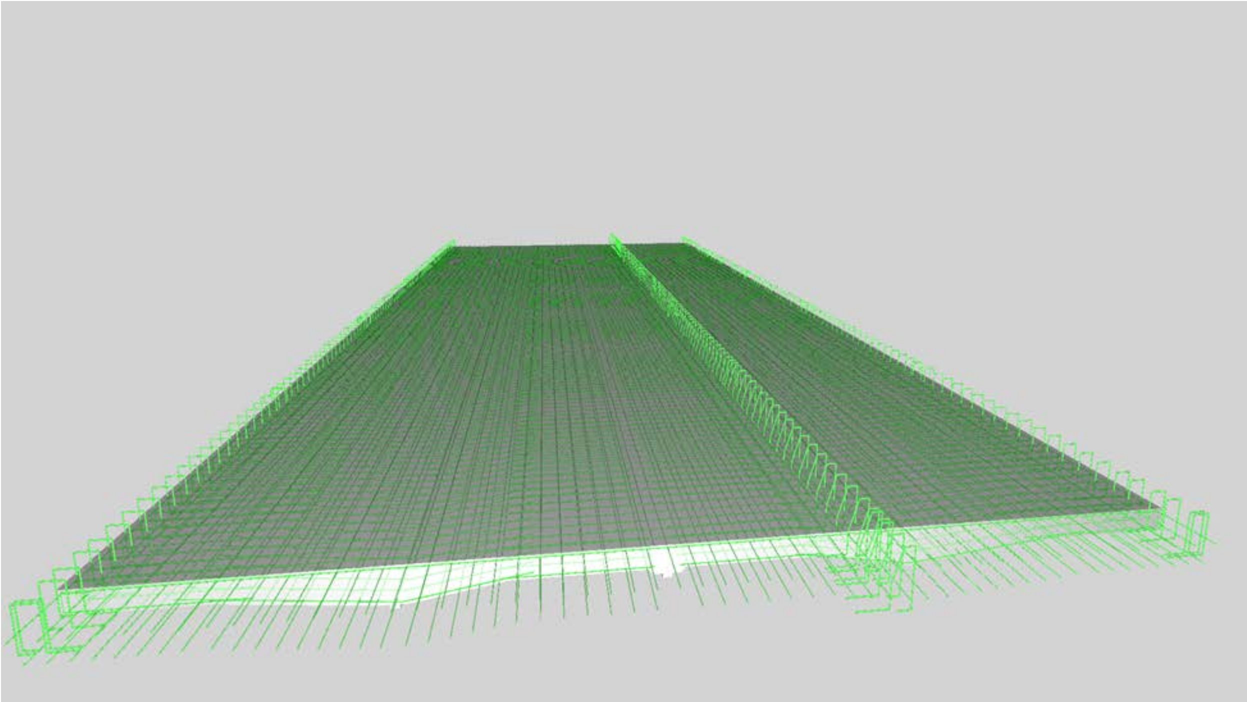


Figure 52 - Van White deck model

The cap beams are represented by IfcBeam, and make use of a template using IfcBeamType. At the IfcBeamType, geometry for the concrete is captured using IfcExtrudedAreaSolidTapered based on an IfcTShapeProfileDef with values as shown in Figure 52.

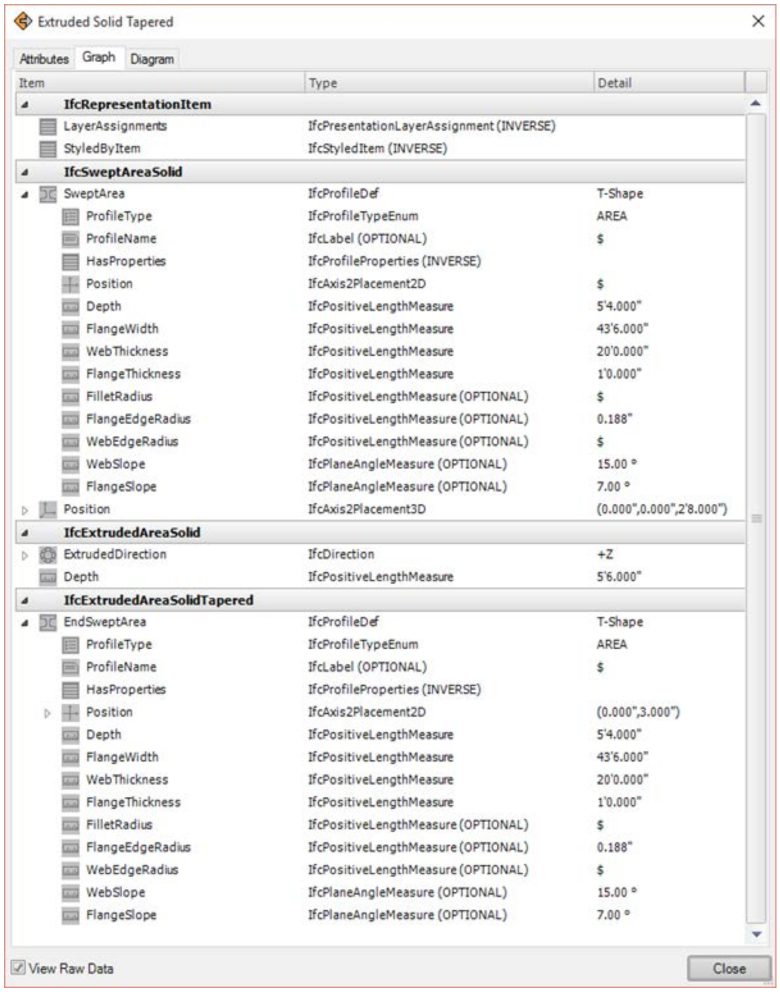


Figure 53 - Van White pier cap geometry values

As illustrated in Figure 53, The IfcBeamType is aggregated into components using the IfcRelAggregates relationship, and contains rebar (IfcReinforcingBar), tendons (IfcTendon), tendon anchors (IfcTendonAnchor), and sleeves for conduit (IfcCableCarrierSegment).

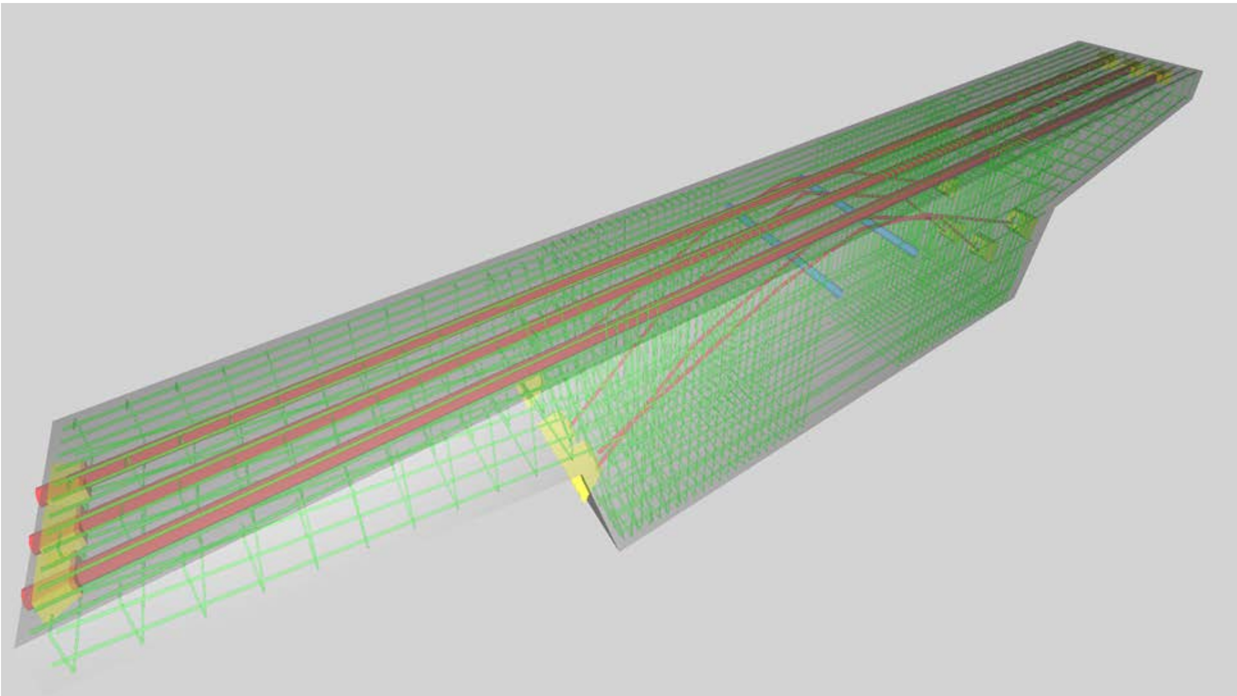


Figure 54 - Van White pier cap model

Each reinforcing group is captured using IfcReinforcingBar, and links to the backing rebar shape exactly as detailed in the plans (the “Superstructure Bar Lists” pages) linking to IfcReinforcingBarType with the IfcRelDefinesByType relationship. Placement of each bar within the group is indicated using IfcMappedItem. Several repetition scenarios are encountered with rebar in the pier cap beams, as described in Table 2, where “Occurrences” indicates the number of IfcReinforcingBar instances and “Items” indicates the number of IfcMappedItem’s within each IfcReinforcingBar, such that the total bars within the cap beam are equal to multiplying Occurrences by Items and totaling each type.

While indicating each bar placement specifically is required for interoperability with existing IFC applications, indicating placement patterns is entirely optional, and is described in Volume II of this Report.

Table 2 - Van White rebar patterns

Rebar Type	Location	Occurrences	Items	Pattern
P1601E	Stirrup	4	16	Offset
P1602E	Stirrup	4	6	Offset
P1603E	Inclined Stirrup	4	5	Offset with rotation
P1304E	Cantilever Stirrup	4	11	Offset with scale
P1605E	Vertical at column	1	1	None
P1606E	Fascia	2	5	Offset
P1907E	Horizontal	1	7	Offset variable
P1908E	Horizontal	1	4	Offset
P1909E	Horizontal	1	2	Offset
P1910E	Horizontal	1	2	Offset
P1911E	Horizontal	1	2	Offset
P1912E	Horizontal	1	2	Offset
P1913E	Horizontal	1	2	Offset
P1914E	Horizontal	1	13	Offset
P1315E	Horizontal	2	9	Offset

As shown in Figure 54, The IfcReinforcingBarType describes fixed geometry of bars using IfcSweptDiskSolidPolygonal, and contains parameters for driving the geometry. The NominalDiameter attribute indicates the nominal bar size- for example, 0.625 is equivalent to #5 Imperial and #16 Soft Metric. The BendingCode attribute indicates the standard rebar bending shape according to ACI-318, for example “102” for hooked-U shape stirrups. The BendingParameters attribute indicates parameters applied to the applicable shape (e.g. width, height).

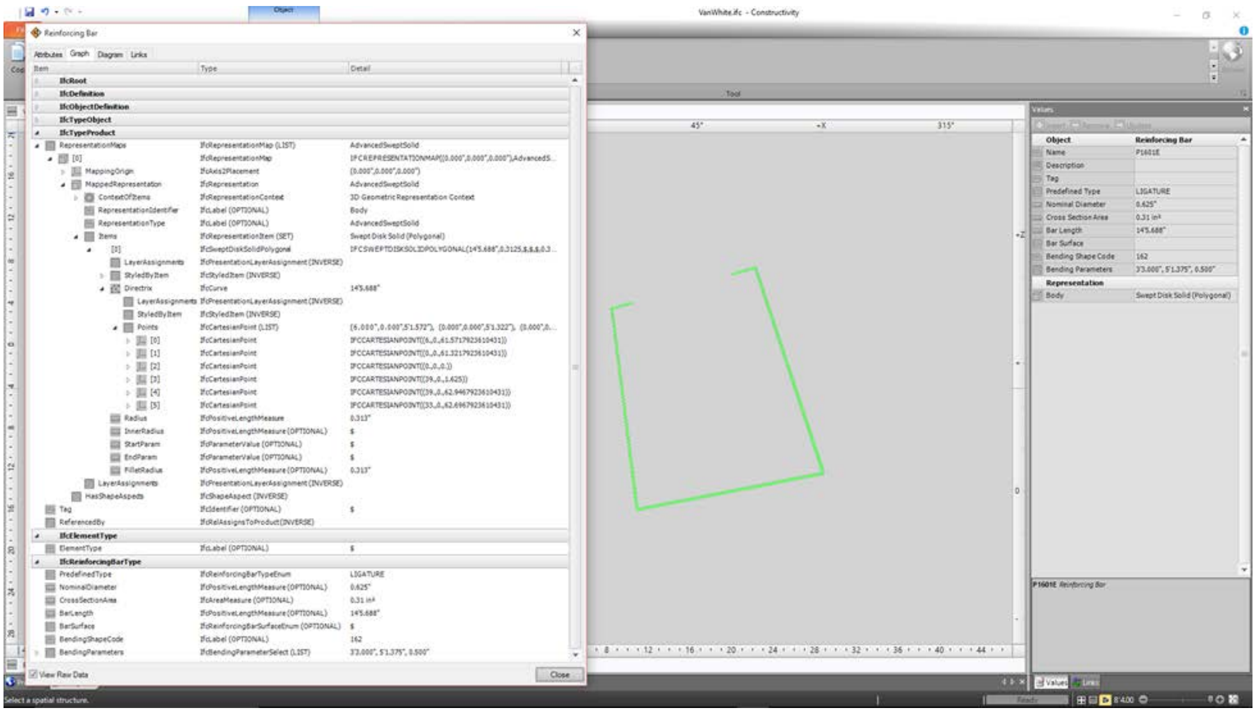


Figure 55 - Van White rebar definition

Each tendon is captured using IfcTendon, and defines geometry using IfcSweptDiskSolid based on IfcIndexedCurve consisting of linear and circular segments. Tendon anchors are captured using IfcTendonAnchor, linking to common geometry defined at IfcTendonAnchorType. Each tendon is connected to two anchors using IfcRelConnectsPathElements at each end. The sequence of the relationships is significant, such that the head anchor is fixed and the tail anchor is adjustable – specifically, IfcRelConnectsPathElements.RelatingElement refers to the "head" element and IfcRelConnectsPathElements.RelatedElement refers to the "tail" element.

Sleeves for conduit are captured using IfcCableCarrierSegment, where geometry is defined using IfcExtrudedAreaSolid based on IfcCircleHollowProfileDef.

The diaphragms above the north and south abutments are each represented by IfcBeam, and make use of a template using IfcBeamType. As shown in Figure 55, such diaphragm includes holes modeled as IfcCircle within IfcArbitraryProfileWithVoids, which is extruded within IfcExtrudedAreaSolidTapered. Note that some rebar is exposed outside of the concrete – this is intentional, as the plans indicate that such exposed region is to be poured separately after initial placement, where such adjoining concrete is defined within the abutment component having later sequence.

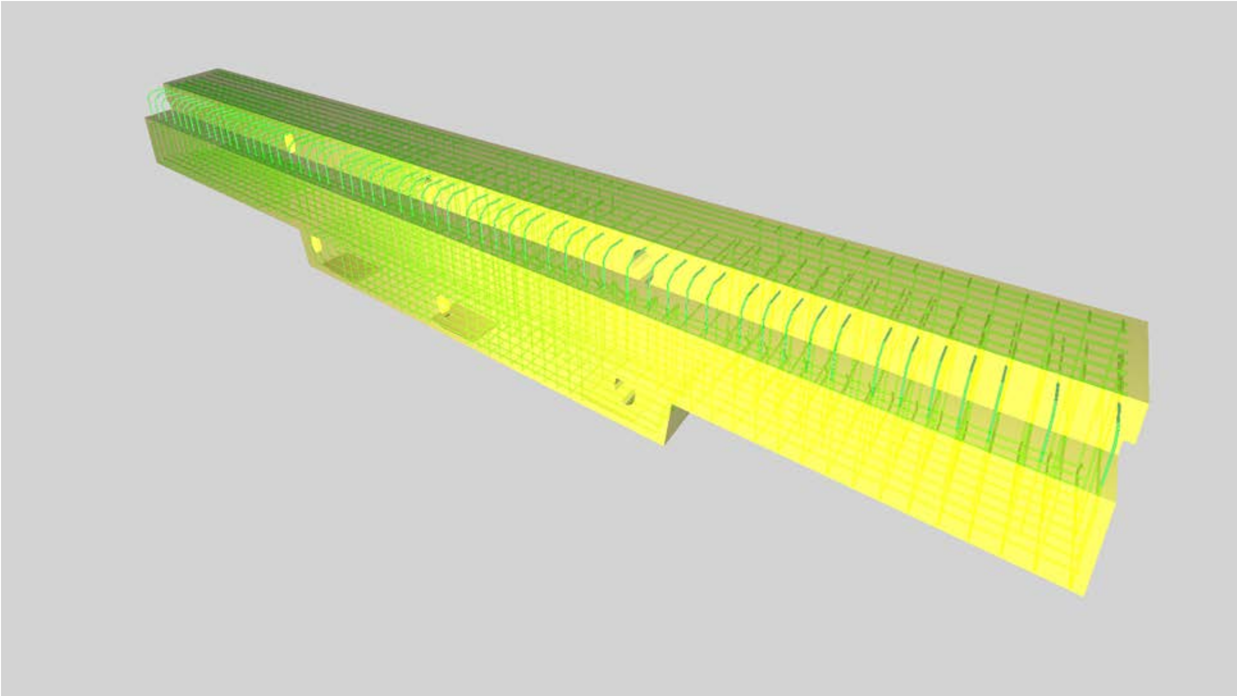


Figure 56 - Van White diaphragm model

1.7.3 OpenBrIM

Modelling the bridge deck in OpenBrIM involves defining formula expressions for the bridge deck profile. Current OpenBrIM-based bridge models observed use a constant cross-section that is swept along the alignment curve.

For the Penn Turnpike steel bridge, the cross-section of the deck varies, such that there would need to be a way to define this variation along the alignment. For the cross-framing, there would need to be a way to relate the vertical positions of plates and members relative to connected elements.

For the Van White concrete box girder bridge, the cross-section of span segments is constant but rotates, however transitions to solid segments at piers and abutments. It would seem possible to model this as separate segments, however there would need to be a way to indicate connectivity between such segments, and indicate connectivity to adjoining elements in common to all segments (e.g. guard rails). In capturing plan-level detail, the concrete shells are rather trivial compared to the detail required for rebar, tendons, conduit, and drainage structures – how to capture such geometry and semantics would also need to be elaborated. As including rebar can inflate the size of a file by orders of magnitude, it may be worthwhile revisiting the late-bound nature of ParamML and defining built-in data structures that can be more efficiently processed, stored, and retrieved.

1.8 Barriers

Barriers may be defined as a constant cross-section placed along the alignment at either edge or anywhere in between.

1.8.1 Bentley OpenBridge

Bentley LEAP Bridge Steel defines the placement relative to the left or right edge of the bridge deck slab laterally, and relative to a connecting element such as an abutment longitudinally, as shown in Figure 56.

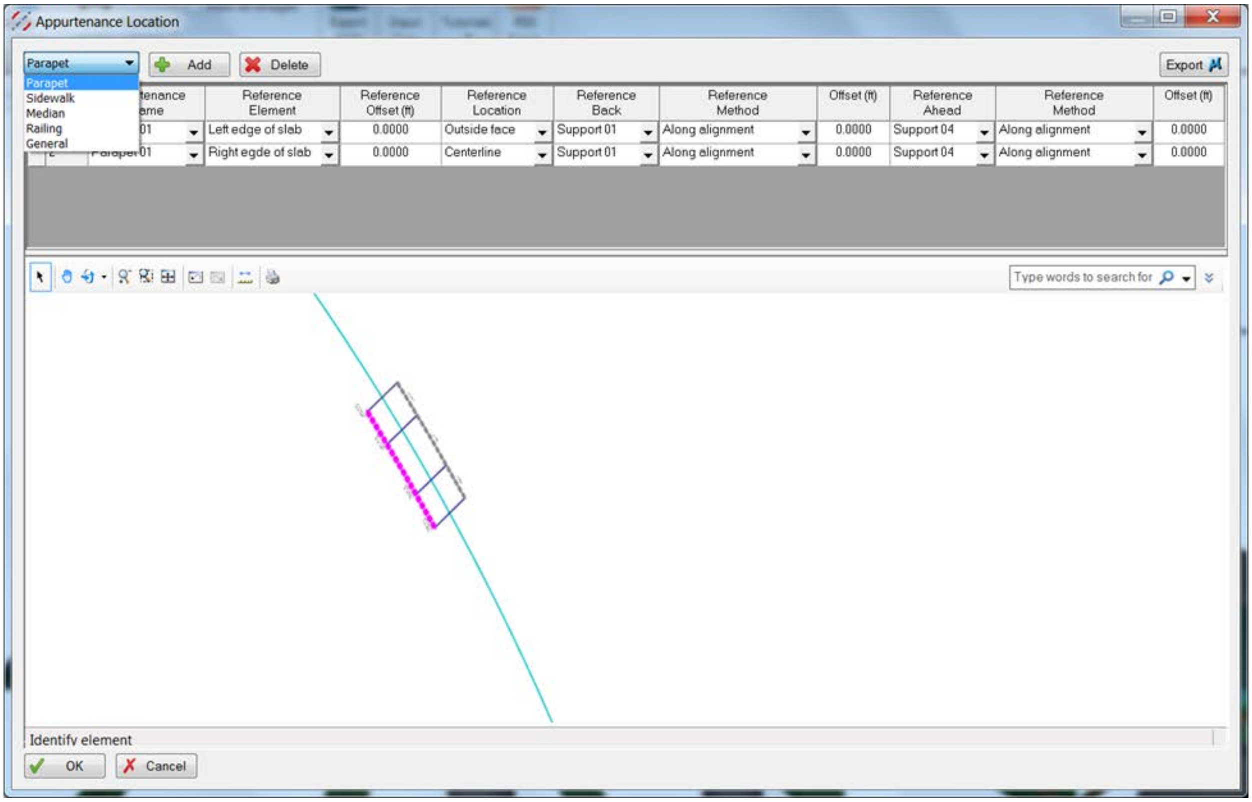


Figure 57 - Barriers with Bentley

1.8.2 Industry Foundation Classes

Guard rails and barriers are modelled as IfcRailing making use of custom profiles. Such barriers are modelled very similarly to girders, where an IfcArbitraryClosedProfileDef describes the cross-section, an IfcAlignment2DVertical describes the path, and an IfcFixedReferenceSweptAreaSolid captures the 3D shape.

1.8.2.1 Penn Turnpike Bridge

Figure 57 illustrates guardrails modeled for the Penn Turnpike bridge.

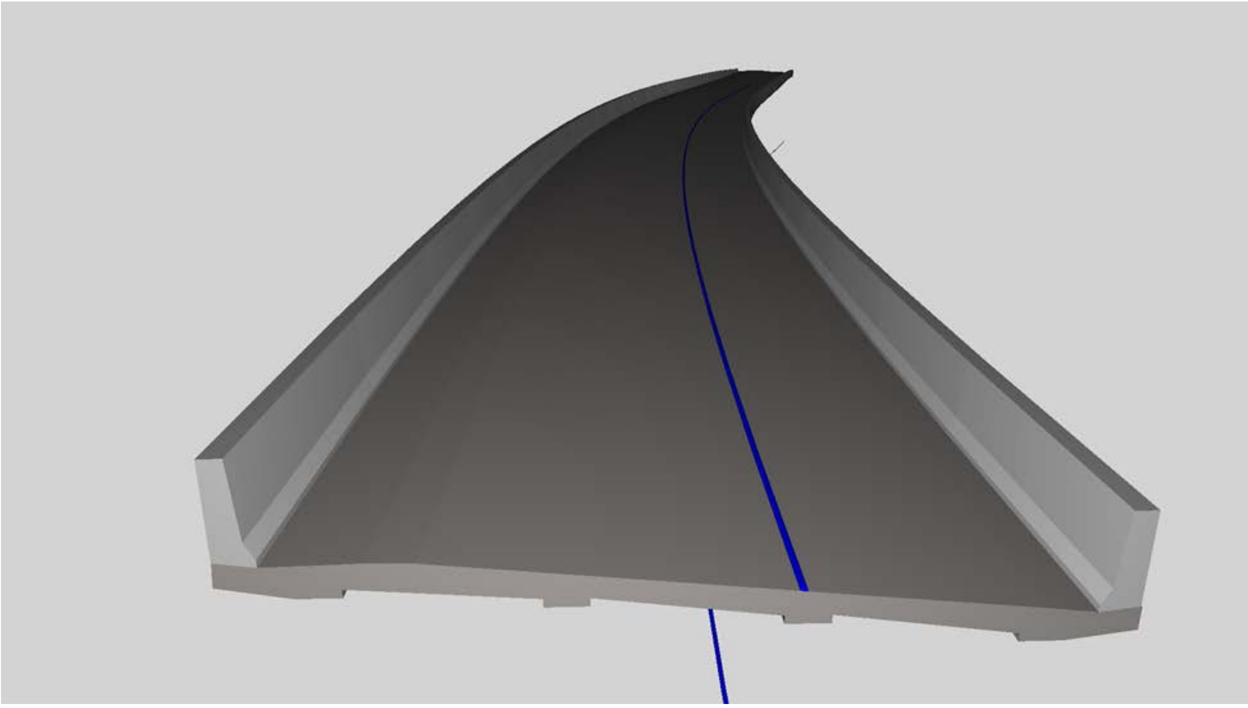


Figure 58 - Penn Turnpike barrier model

Each IfcRailing is positioned along the IfcAlignment using IfcRelPositions, where GridOrdinates refers to the corresponding working point along the left or right edge. As shown in Figure 58, the profile is defined using IfcArbitraryClosedProfileDef. The solid is constructed by sweeping the profile along the alignment using IfcSectionedSpine.

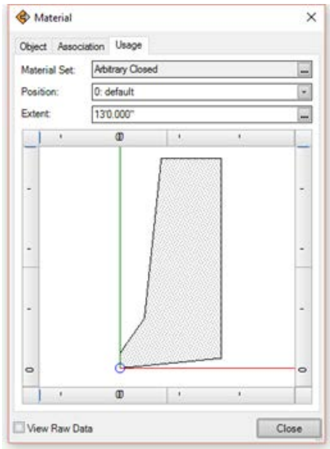


Figure 59 - Penn Turnpike barrier profile

Rebar (IfcReinforcingBar) and embedded conduit (IfcCableCarrierSegment) are linked using IfcRelAggregates where RelatingElement refers to the IfcRailing. Rebar that protrudes into the barrier but is first poured as part of the bridge deck is aggregated within the element where it is poured (e.g. IfcSlab, IfcBeam) and is NOT aggregated within the IfcRailing.

1.8.2.2 Van White Bridge

For the Van White Memorial bridge, in addition to guard rails along the edges, there is also an intermediate guard rail separating vehicles from pedestrians, and each guard rail has railings and light fixtures attached.

While such detail may be considered secondary, such parts can already be fabricated based on digital definitions of IFC, such that savings may be achieved by eliminating manual takeoff, layout, and programming – i.e. steel fabrication equipment may use the IFC file directly as input. Such detail also has particular implications for bridges following curved alignments, where components may require transformation.

Railings are modeled using IfcRailing, with type definitions based on IfcRailingType. The railing for this particular example is aggregated into components of IfcMember corresponding to posts, rails, and spindles. Figure 59 illustrates the railing at the edge of the walkway on one side of the bridge, as modelled linearly.

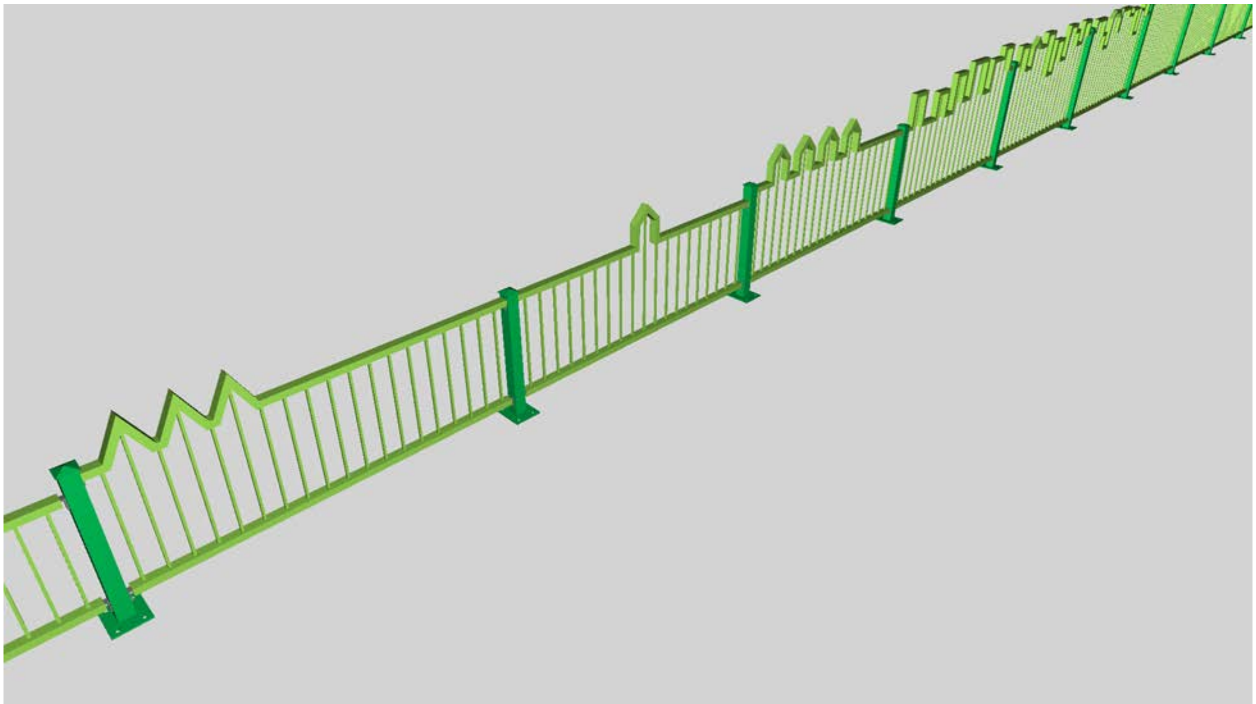


Figure 60 - Van White railings for walkway

Figure 60 illustrates the railing dividing vehicles from pedestrian traffic.

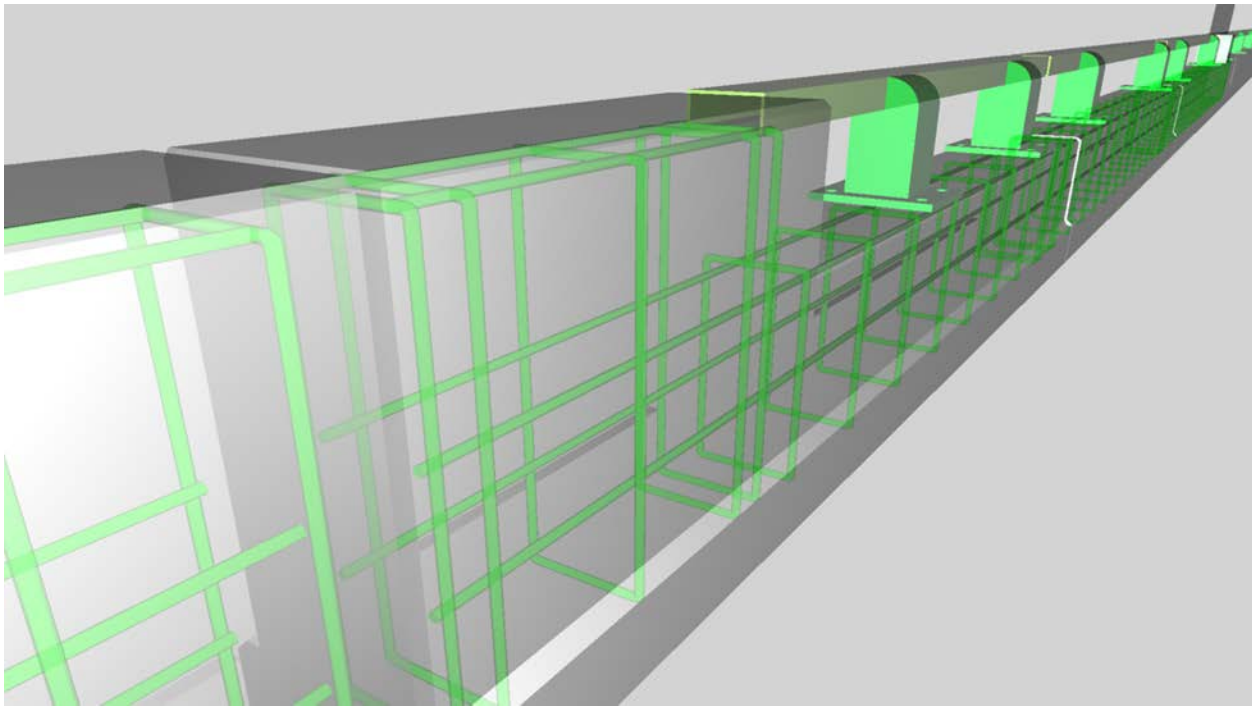


Figure 61 - Van White railings for divider

Each post is defined by IfcMember having PredefinedType=POST. While such IfcMember could be aggregated further into plates and other elements, such composition isn't warranted, as it is assumed that they will be fabricated into solid components before construction. The geometry for plates is described using IfcExtrudedAreaSolid based on IfcArbitraryProfileDefWithVoids, where the outer curve is based on IfcPolyline and the inner holes are defined using IfcCircle. The geometry for hollow sections is described using IfcExtrudedAreaSolid based on IfcRectangleHollowProfileDef. The geometry for angles connecting to railings is described using IfcExtrudedAreaSolid based on IfcLShapeProfileDef. The caps for the dividing railing posts are described using IfcRevolvedAreaSolid based on IfcCShapeProfileDef. The caps for the outer railing posts are described using IfcCsgSolid based on IfcRectangularPyramid, or alternatively reduced to IfcFacetedBrep. Such representation is consistent with existing usage of IFC for fabrication, such as supported by Tekla Structures and SDS/2.

Figure 61 illustrates railing posts for the divider.

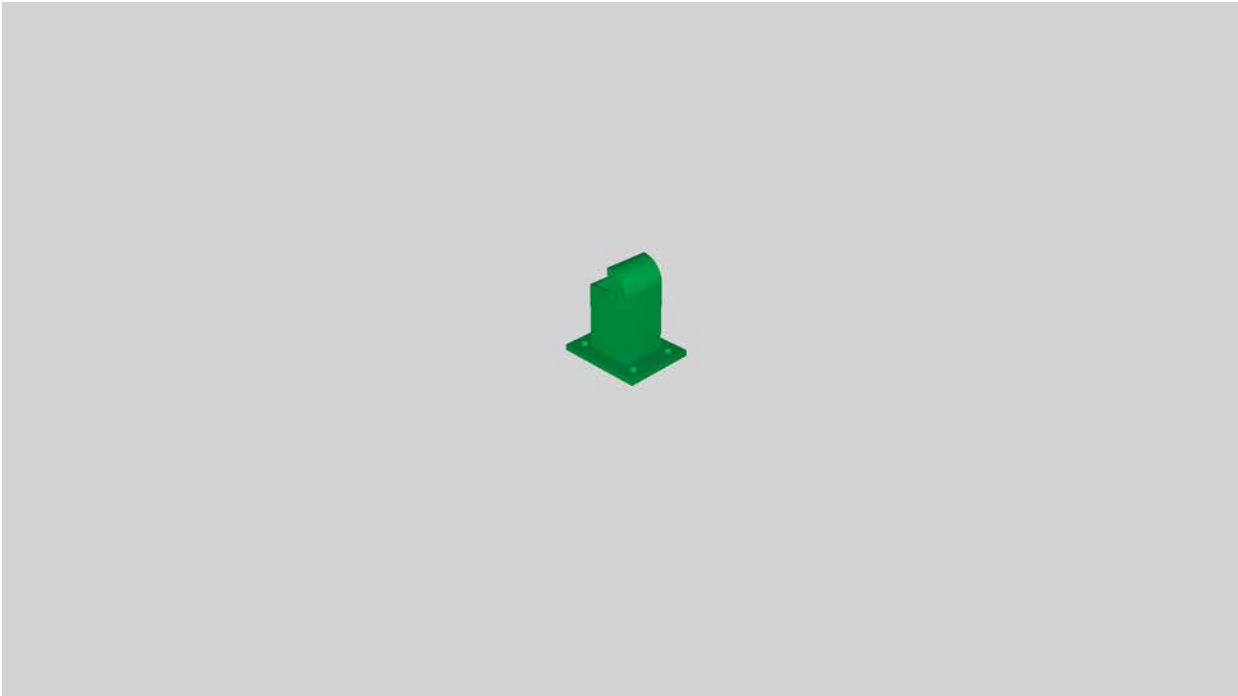


Figure 62 - Van White railing post for divider

Figure 62 illustrates railing posts for the walkway.

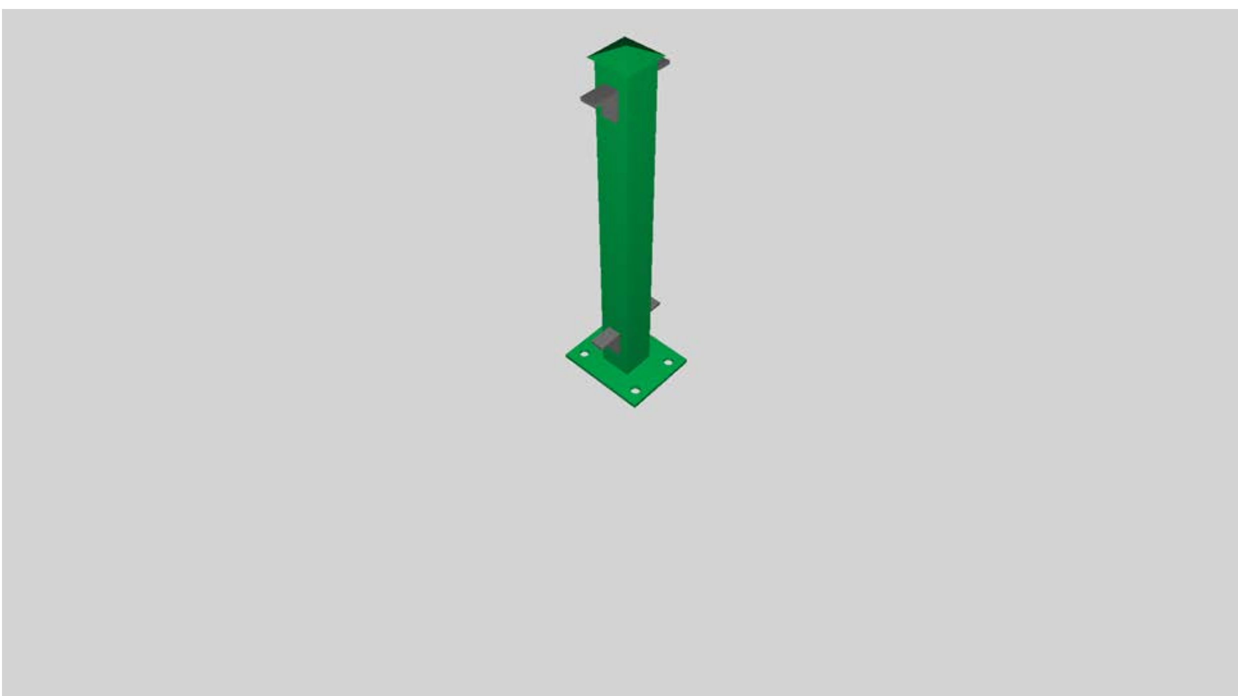


Figure 63 - Van White railing post for walkway

The rails are modeled using IfcMember with PredefinedType=CHORD, where straight segments are modeled using IfcExtrudedAreaSolid based on IfcRectangleHollowProfileDef (as shown in Figure 63), and decorative segments are modeled using IfcFixedReferenceSweptAreaSolid based on IfcRectangleHollowProfileDef (as shown in Figure 64).

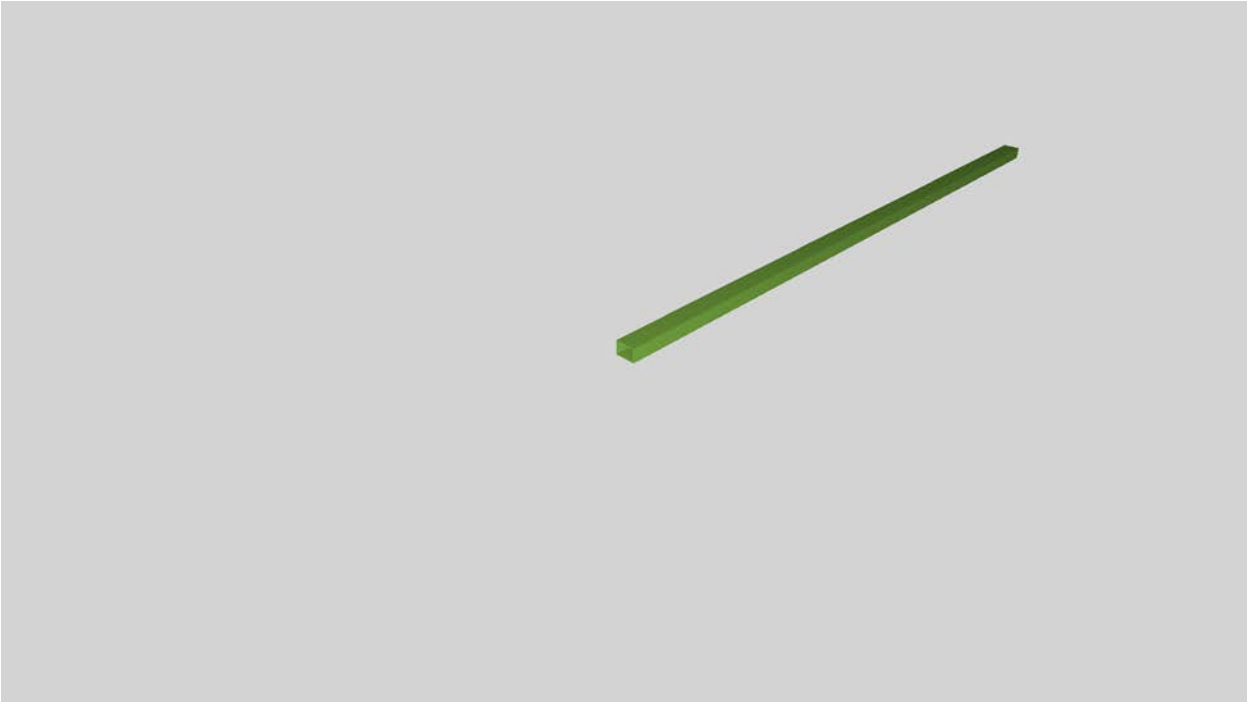


Figure 64 - Van White railing rails straight

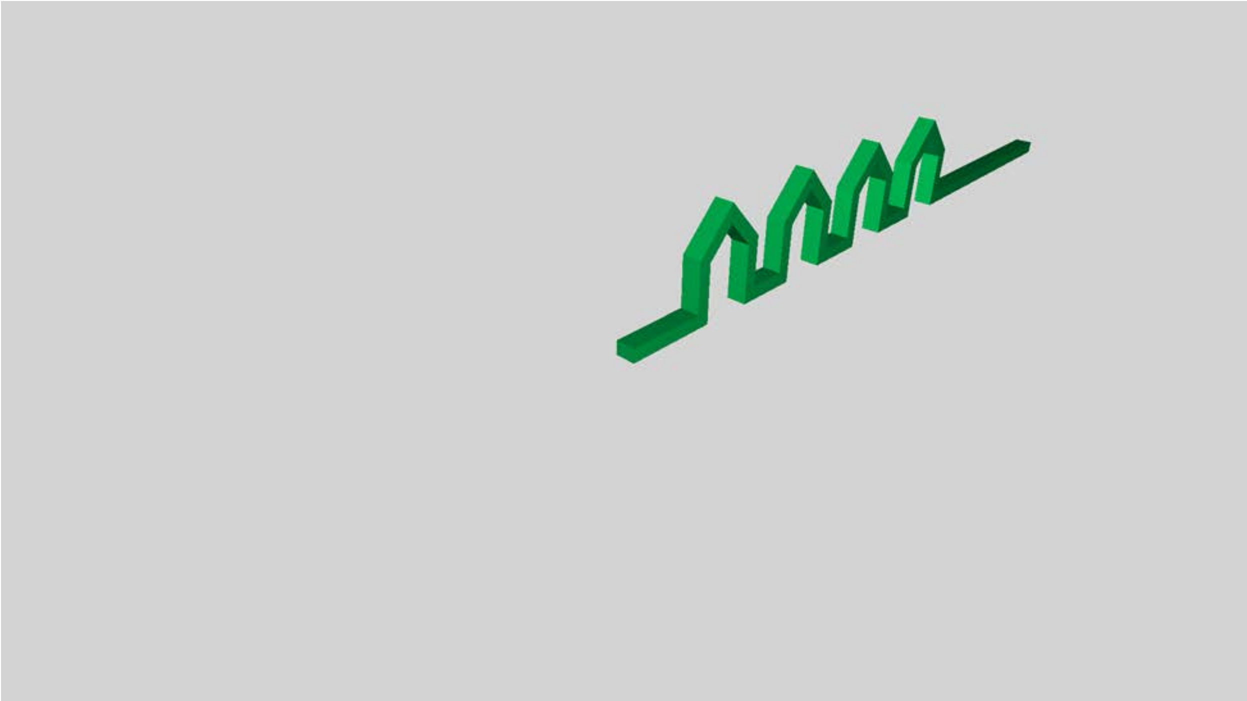


Figure 65 - Van White railing rails decorative

As shown in Figure 65, railing spindles are modeled using IfcMember with PredefinedType=STRUT, where geometry is represented by IfcExtrudedAreaSolid based on a solid IfcRectangleProfileDef.

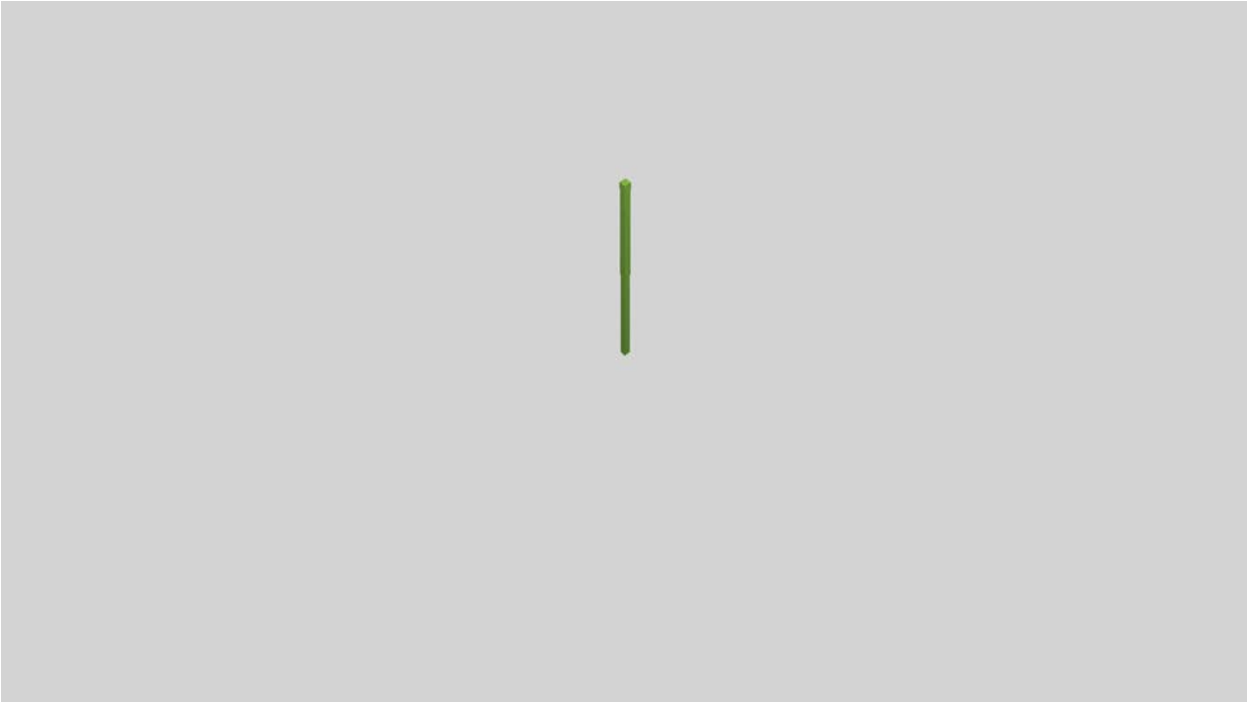


Figure 66 - Van White railing spindles

Where such railing type definitions are placed relative to the alignment curve, specific transformation is required to address variable usage.

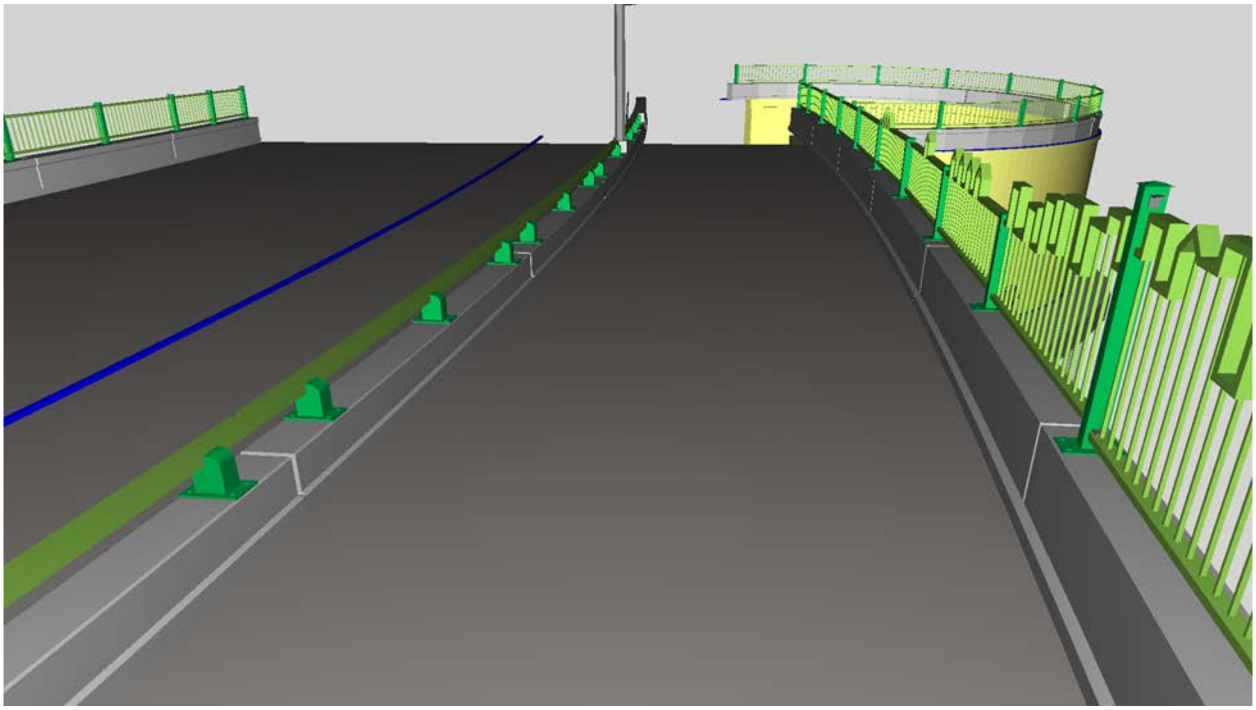


Figure 67 –Van White railing chain transform

As shown in Figure 66, railing sections may be defined to use CHAIN transformation (described at IfcRelPositions in Volume II), such that each component is positioned separately along the alignment, but no geometry is transformed. For sections where curvature is significant such that railing geometry should also be curved (or the economics of fabrication allow), such components may use WARP transformation as shown above the retaining wall in Figure 67.

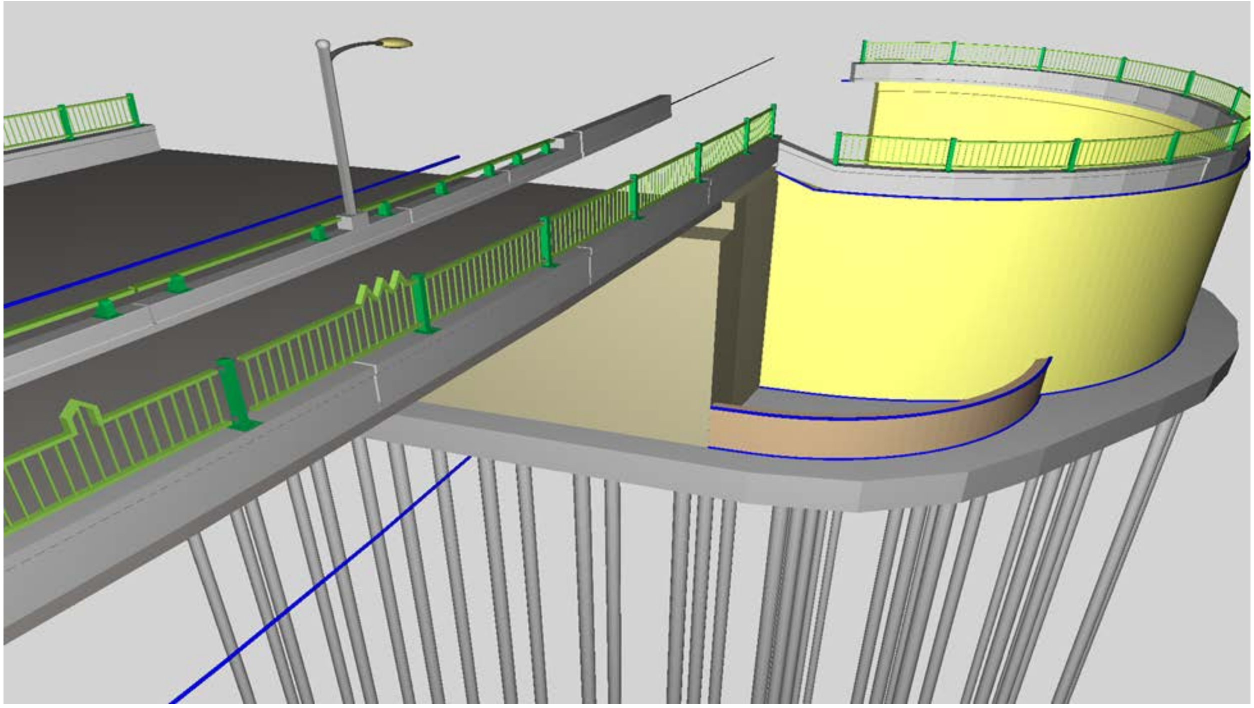


Figure 68 - Van White railing warp transform

1.8.3 OpenBrIM

Barriers are modelled in OpenBrIM as constant polyline-based cross-sections which are swept along an underlying alignment curve at relative offsets. Detail requiring elaboration includes how to model joints between segments of barriers, rebar, and embeddings. Transformation of detailed components along curved alignments such as architectural railings requires specification. OpenBrIM does not define any standardized cross-sections that would be used for fabrication such as hollow rectangle, I-Shape, etc. – such definitions would need to be formalized in order for fabrication software (e.g. Ficep, Peddinghaus, Voortman) to correlate cross-sections with stock materials, or for converting to formats used by fabrication software that require this information such as DSTV/NC1.

1.9 Lighting

As light fixtures, conduit, and junction boxes are often included in bridge plans and in most cases also impact the structural components (e.g. conduit and junction boxes embedded within guardrails), they must be detailed within any digital representation serving the same purpose.

1.9.1 Bentley OpenBridge

Bentley LEAP Bridge Steel does not model lighting or electrical components, however downstream applications may do so.

1.9.2 Industry Foundation Classes

IFC defines structures for physical electrical elements such as electrical conduit, cables, and fixtures. It also defines functional behavior of circuits and systems, and electrical connectivity of all elements. For purposes of bridge construction, the behavior of such electrical systems are out-of-scope, however the impact to structural elements is certainly in scope.

Light fixtures are defined using IfcLightFixture, with type definitions modeled using IfcLightFixtureType. As light fixtures are always fabricated off-site, the geometry is not of importance such that tessellated shapes are sufficient using IfcTriangulatedFaceSet. Pragmatically, manufacturer product catalogs for light fixtures typically provide geometry in such form, rather than design primitives used for manufacturing. The actual lighting required of fixtures may be defined using the custom 'Lighting' representation, where for street (and bridge) lighting uses IfcLightSourceDirectional.

Connectivity between light fixtures and circuits is captured by IfcDistributionPort, where a light fixture (as an end-device) consists of a single port having FlowDirection=SINK, where electrical requirements are indicated. This IfcDistributionPort is connected to electrical wiring (IfcCableSegment) using the relationship IfcRelConnectsPorts, where the circuit may be traversed by following such relationships between cables and devices.

1.9.2.1 Penn Turnpike Bridge

Lighting is not described within the plans, however conduit is detailed within each of the abutments.

1.9.2.2 Van White Bridge

Embedded electrical conduit is captured using IfcCableCarrierSegment having PredefinedType=CONDUIT. Junction boxes are captured using IfcJunctionBox, each of which have ports (IfcDistributionPort) connected to one or more IfcCableCarrierSegment elements using IfcRelConnectsPorts. The relationship IfcRelAggregates is used to embed conduit and junction boxes into the enclosing element (e.g. IfcWall, IfcRailing). If conduit spans multiple elements, it may only be embedded within of them and is presumed to interfere with connecting elements, where the embedded element may serve as an anchor for design purposes but otherwise carries no meaning.

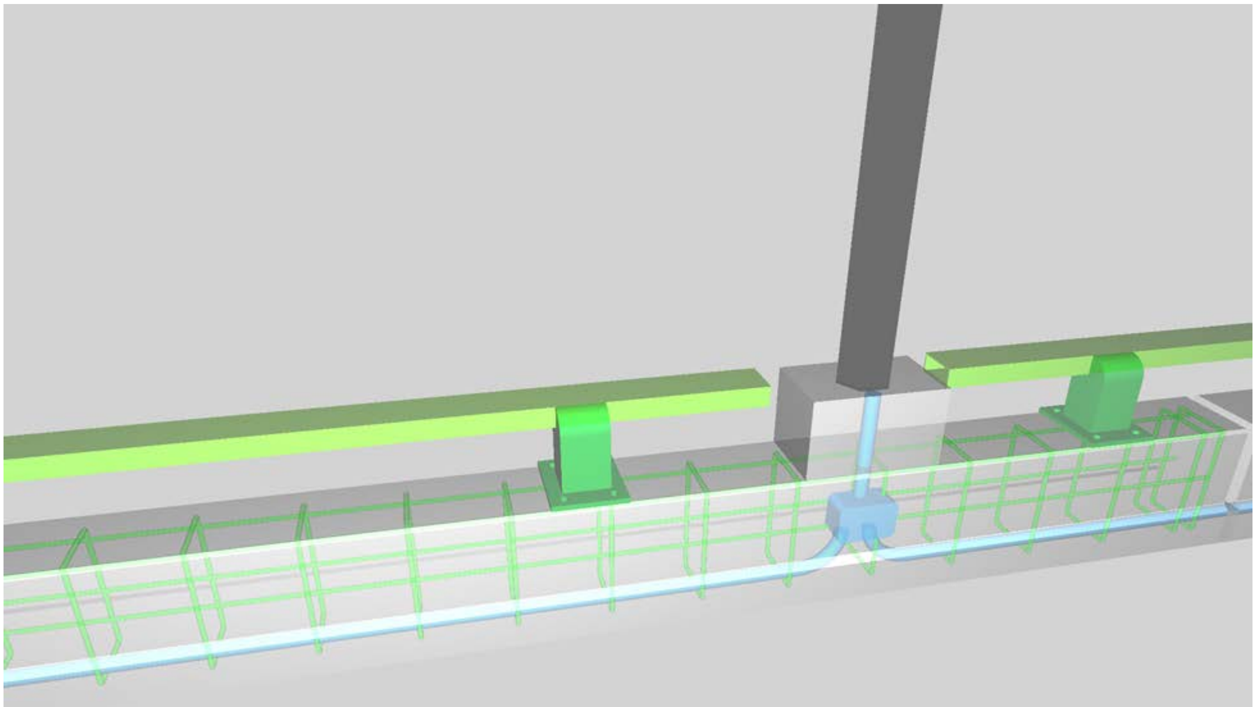


Figure 69 - Van White lighting

1.9.3 OpenBrIM

OpenBrIM does not define modeling of electrical systems.

1.10 Sitework

Site grading is indicated using several geometric structures, for which contour lines or the elevation at any point may be derived.

1.10.1 Bentley OpenBridge

Bentley LEAP Bridge Steel does not collect input for terrain, however may import this information from other files including LandXML. The program supports defining the vertical alignment of the terrain relative to the alignment of the bridge as shown in Figure 69.

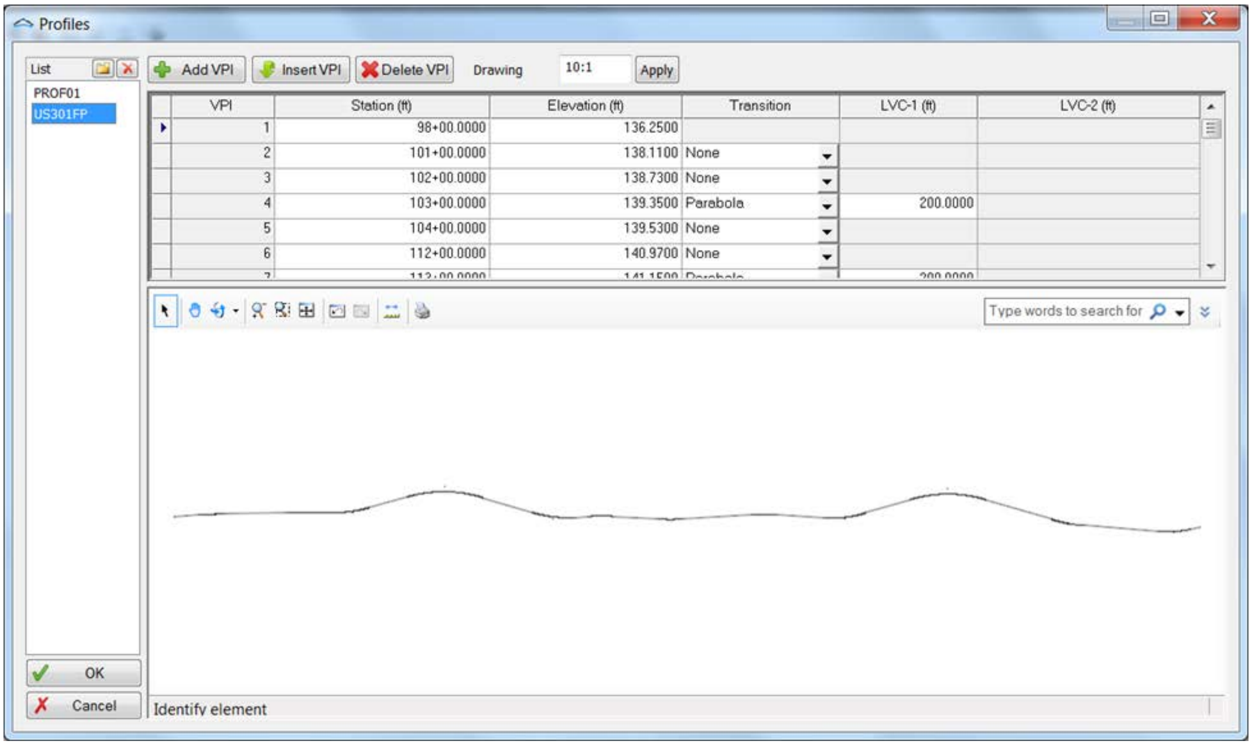


Figure 70 - Sitework with Bentley

1.10.2 Industry Foundation Classes

IFC captures terrain information using several representations. The “SurveyPoints” representation captures a set of points indicated by IfcCartesianPoint for elevations at arbitrary locations, which may form a rectangular grid or no predefined pattern. This representation also captures breaklines in the form of IfcPolyline indicating elevations that are to be interpolated between points. The “Surface” representation captures the surface using any surface model such as non-uniform rational B-Spline surfaces (NURBS) using IfcBSplineSurface subtypes, a tessellated model using IfcTriangulatedFaceSet, or other geometric constructs.

1.10.2.1Penn Turnpike Bridge

Figure 70 illustrates terrain for the Penn Turnpike sample bridge.

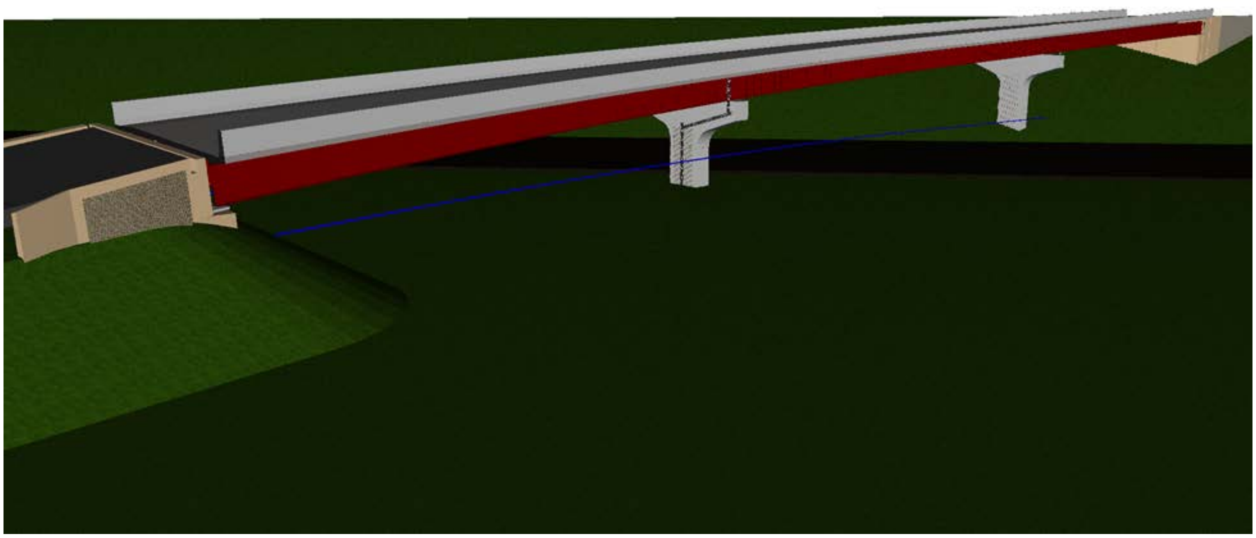


Figure 71 - Penn Turnpike terrain

For the Penn Turnpike bridge, the overall terrain is modeled as a tapered extruded profile (IfcExtrudedAreaSolidTapered with IfcArbitraryClosedProfileDef), which corresponds to the conditions as shown in the plans.

The terrain approach for Abutment 1 is modeled using IfcGeographicElement consisting of geometry using IfcExtrudedAreaSolidTapered and IfcRoundedRectangleProfileDef, positioned according to the alignment curve which gives it the slope.

For modeling arbitrary terrain, such surface may be represented by IfcTriangulatedFaceSet. For purposes of construction, only the surface needs to be captured, as soil conditions are indicated at specific boring locations as described in the next section. Thus, there is no need to capture terrain as a solid model, and doing so would be misleading as it would describe gross assumptions made according to application-specific calculations made according to borings, rather than just the data that is known. However, for geotechnical analysis, such solid modeling could be used, though that is out of scope for the design-to-construction exchange.

1.10.3 OpenBrIM

OpenBrim does not define any specific structures for terrain, though it is conceivable that such information may be located according to a well-known identifier with terrain indicated by a collection of surfaces and points corresponding to tessellated representations in LandXML and IFC.

1.11 Soil Borings

Soil boring information indicates the position of the test boring with longitudinal and lateral offsets relative to the alignment curve, and classification of soil between elevations for the specified depth of each boring. It is also possible to capture such information using the DIGGS schema, though it may also be beneficial to capture all information within a single file based on a single schema.

1.11.1 Bentley OpenBridge

The Bentley LEAP Bridge applications reviewed did not describe terrain information, however such information would be captured in downstream detailing in Microstation.

1.11.2 Industry Foundation Classes

Soil test borings are captured as IfcGeographicElement objects with a particular predefined type yet to be introduced. The location of the boring is defined relative to the horizontal alignment curve using IfcAlignmentPlacement, where the vertical elevation is at the surface of the ground where the sample was taken.

As shown in Figure 71, the soil layers are captured using IfcMaterialLayerSet, where the lower and upper elevations of each layer may be determined according to the thickness of each IfcMaterialLayer. Each layer references the corresponding soil type using IfcMaterial, where additional descriptions, properties, and material composition may be defined.

As described earlier, solid modeling of soil is out-of-scope for the design-to-construction exchange, however could be considered in-scope for a future geotechnical or structural analysis exchange.

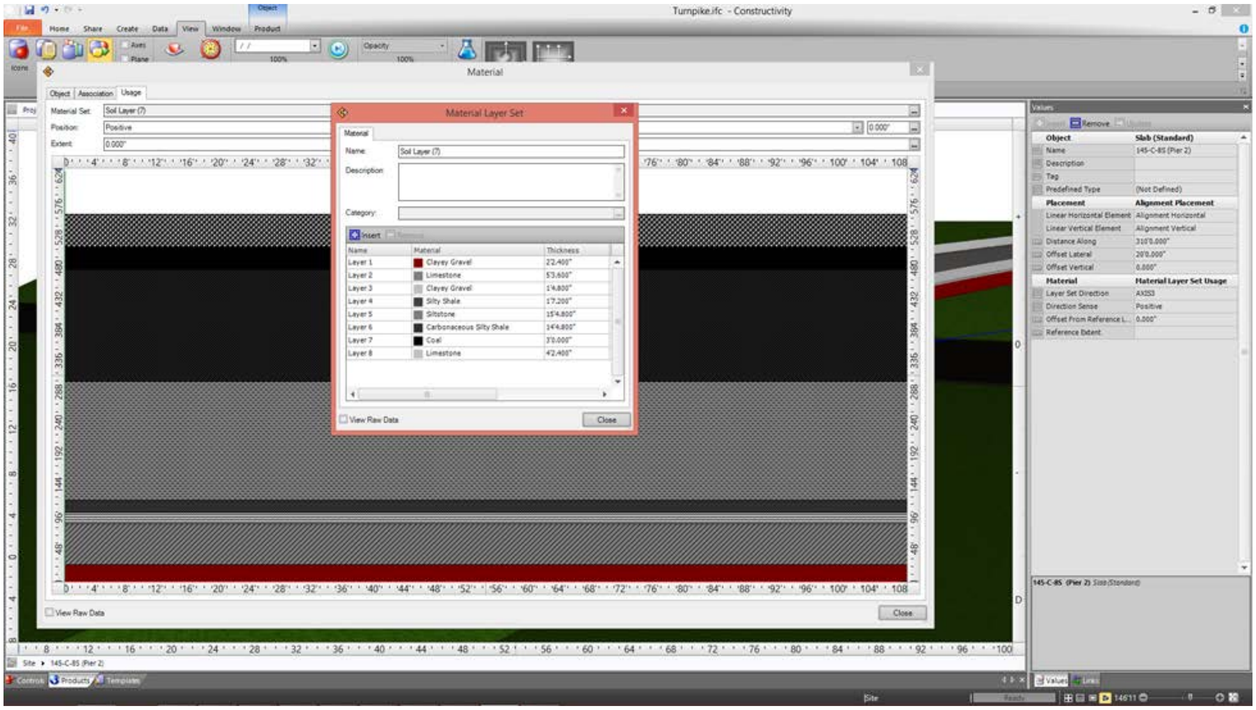


Figure 72 - Soil layers

1.11.3 OpenBrIM

OpenBrIM does not contain any specific static data structures for soil borings, however it is conceivable that reserved identifiers could be used for this purpose, where each layer could be defined as geometry spanning two elevation points.

1.12 Quantities

Quantity information may be included in plans for several reasons: as a shortcut for contractors considering bidding to quickly determine whether the scope of a project is compatible with resource availability, and as verification in preparing a bid to determine whether everything has been captured in quantity takeoff. If geometry is of sufficient detail, then quantities may be more easily derived automatically, i.e. parameterized solid modeling rather than boundary representations or tessellations.

Quantities are detailed here because they are specifically included in the plans for the sample bridge. No recommendation is made as to whether they should be included for a particular bridge; rather that is left for the contracting parties to determine.

1.12.1 Bentley OpenBridge

The Bentley LEAP Bridge applications reviewed did not describe quantity information.

1.12.2 Industry Foundation Classes

As shown in Figure 72, schedules of quantities are captured as IfcCostSchedule with PredefinedType of BILLOFQUANTITIES. Each item in the schedule is captured as IfcCostItem, with individual costs captured as IfcCostValue. Physical elements that comprise quantities are related using the relationship IfcRelAssignsToControl.

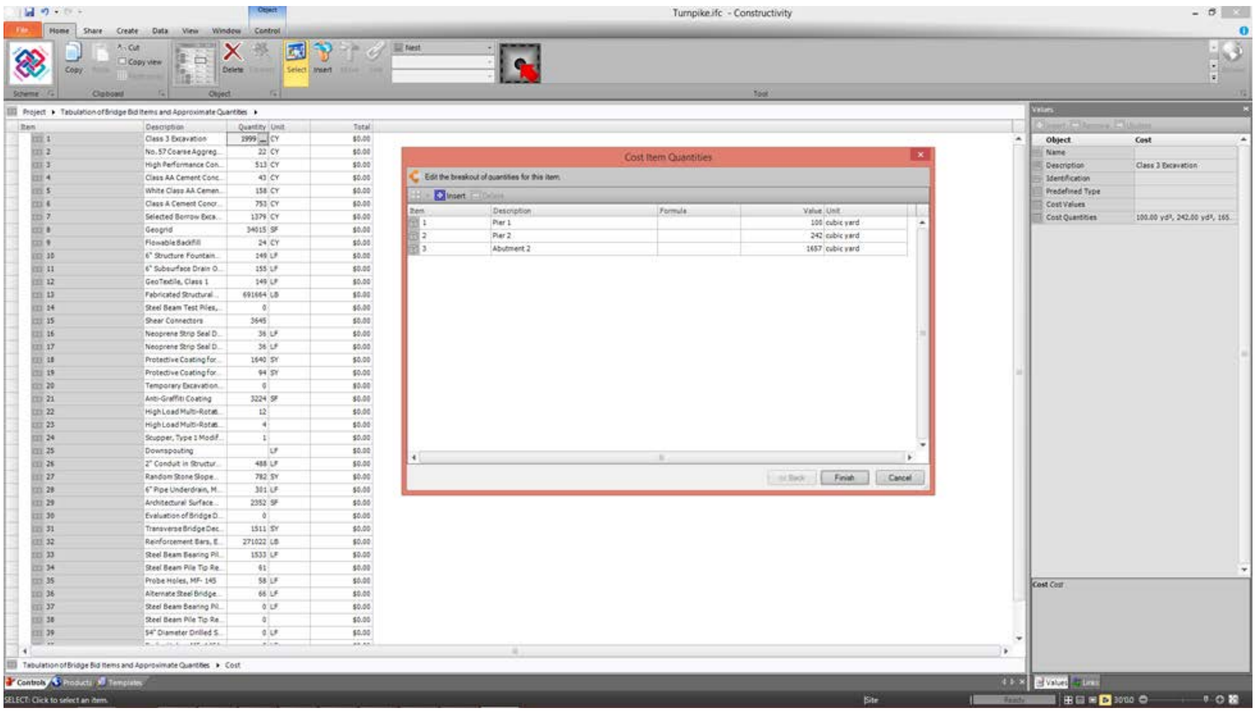


Figure 73 - Quantity schedule

1.12.3 OpenBrIM

OpenBrIM does not define any specific structures for quantities, though it is conceivable that such information could be provided as parameters of reserved object types.

1.13 Construction Staging

The sequencing of construction may be specified for various reasons: to maintain a level of service of existing roadways during construction; to coordinate timing of a project with other related projects; and to define expected structural loads during construction. Most commonly, and for this particular bridge, the sequencing of bridge girders is indicated for structural design purposes. There may also be use cases for exporting to other applications supporting construction scheduling or erection plans and procedures.

1.13.1 Bentley OpenBridge

As shown in Figure 73, Bentley LEAP Bridge Steel allows specification of construction stages of the bridge deck, for which elements are associated with each stage.

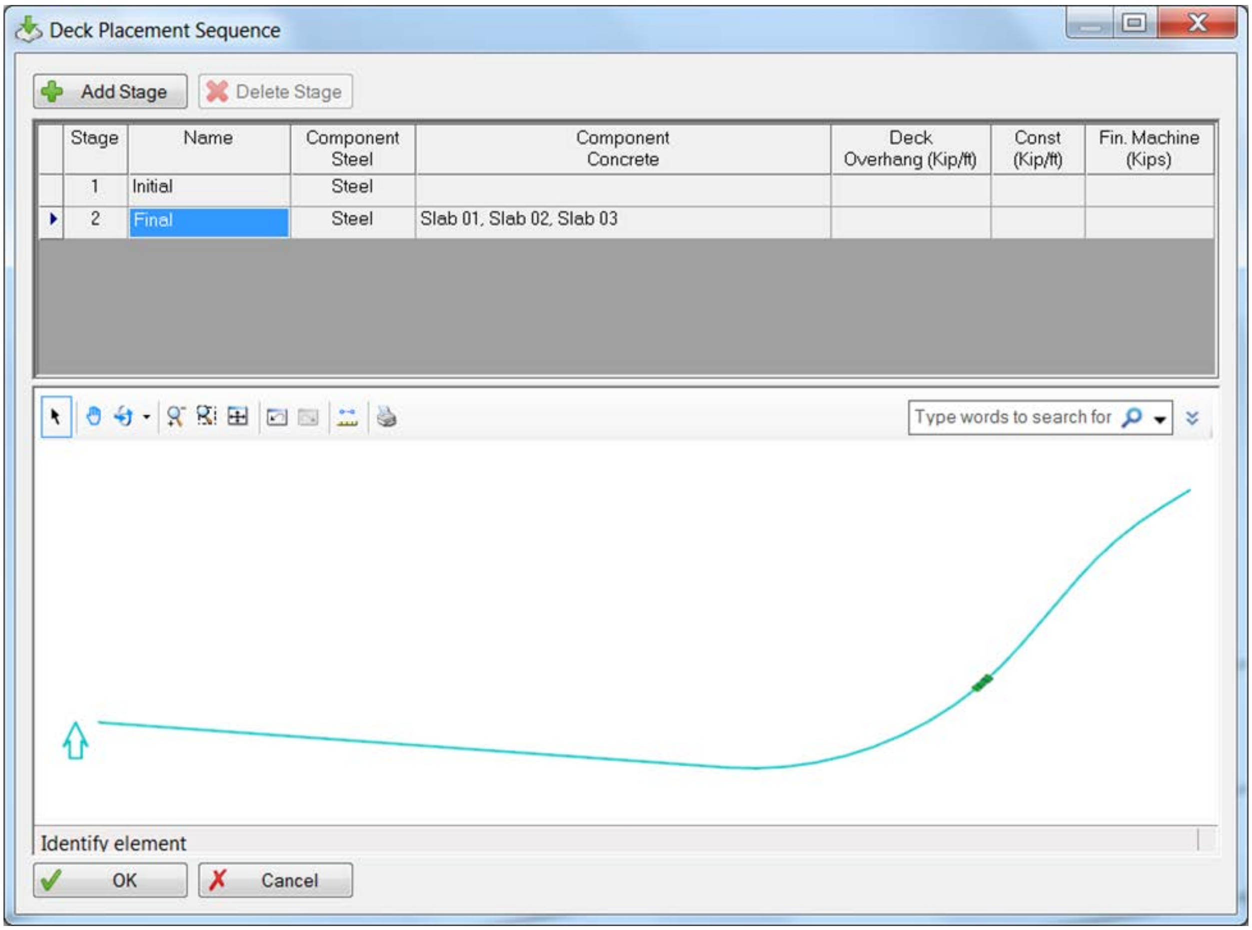


Figure 74 - Sequencing with Bentley

1.13.2 Industry Foundation Classes

IFC defines process models in parallel with product models, where each phase of a project may be represented by IfcTask. The IfcTask entity defines anything occurring in time over a time span and may correspond to a high-level summary task such as a construction stage, very-low level tasks such as placing a specific beam, or anything in between and a combination of all. Tasks may be nested into subtasks for drilling into detail using the IfcRelNests relationship. An IfcTask instance may be assigned to elements affected by the task using IfcRelAssignsToProduct. The meaning of a task may be further defined by IfcTask PredefinedType, e.g. CONSTRUCT means to fabricate the element, INSTALL means to place the element, DEMOLISH means to destroy the element, REMOVE means to uninstall the element. In addition to specific meaning for construction, applications that support 4D animation of construction schedules may also make use of such designations for presentation purposes.

Figure 74 illustrates tasks assigned to bridge elements.

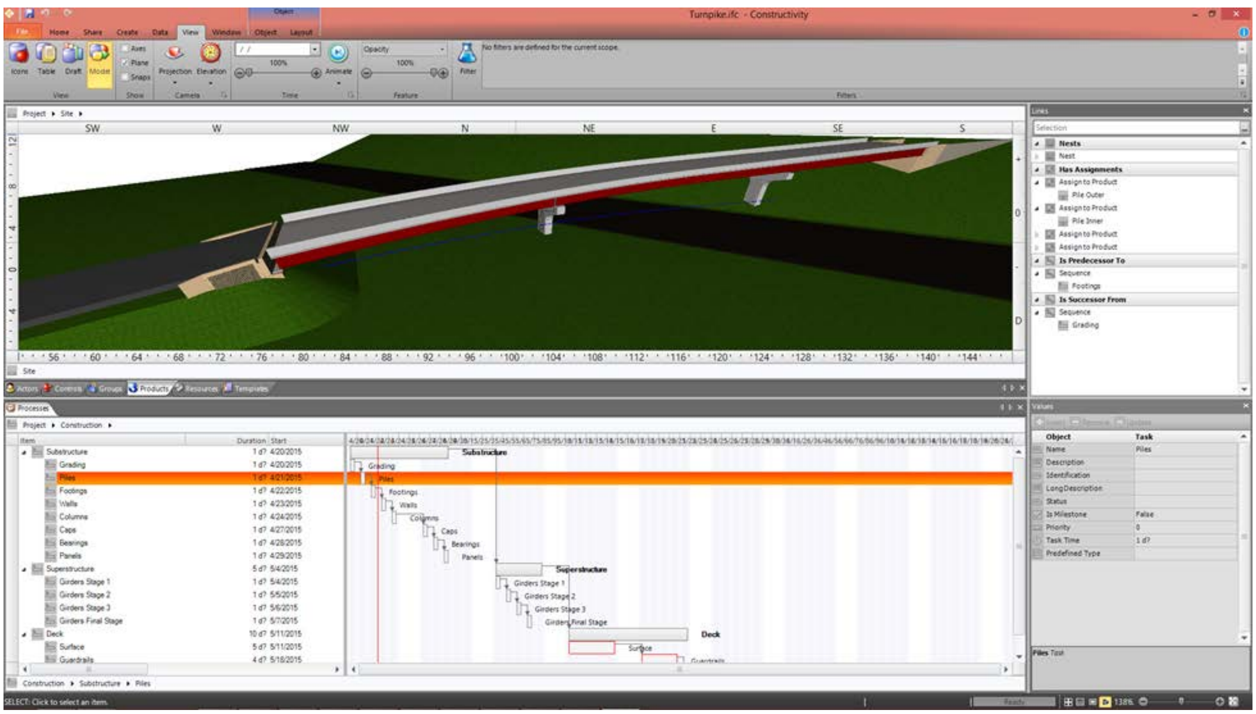


Figure 75 - Sequencing tasks

1.13.3 OpenBrIM

OpenBrIM 2.0 defines a named parameter called “Sequence” which is defined periodically for various elements. Presumably this is an integer number. To preserve sequencing in a collaborative environment as elements may be inserted and deleted, it may be beneficial to use object relationships in addition to (or instead of) sequence numbers.

1.14 Structural Analysis

Structural analysis of bridges involves analytical models that approximate the physical behavior of specific bridge geometry. Such analytical models may make use of curve members, surface members, and volumetric members. Finite element analysis may be applied, effectively reducing everything into a mesh of nodes and linear members. Within such structural model, load cases and combinations may be defined, with resulting stresses and displacements captured.

For the purposes of design plans, inclusion of full structural analysis information may not be necessary, however the resulting stresses and deflections at each member according to each load combination must be captured, to determine allowable loading during construction.

As shown in Figure 75, the specific plans for the Penn Turnpike example contain resulting shear and moments at locations along girder spans, and displacements from dead loads used to calculate camber ordinates. Additionally, the plans indicate the load cases and combinations.

GIRDER 1 - MOMENTS

LOC	DL1	DL2	DL1+DL2	HS25 + IMP		P-82 + IMP	
				POS	NEG	POS	NEG
0.0	0	0	0	0	0	0	0
0.1	675	294	969	892	-213	1483	-300
0.2	1147	488	1635	1468	-416	2465	-550
0.3	1355	555	1910	1812	-609	3020	-778
0.4	1328	549	1877	1938	-780	3157	-968
0.5	1049	440	1489	1878	-941	3038	-1175
0.6	546	261	807	1673	-1081	2745	-1392
0.65	171	119	290	1480	-1144	2420	-1528
0.7	-205	-23	-228	1286	-1206	2095	-1663
0.8	-1227	-410	-1637	834	-1367	1292	-2012
0.9	-2550	-940	-3490	484	-1905	542	-2455
1.0	-4351	-1767	-6118	537	-3132	638	-3318
1.0	-4351	-1767	-6118	537	-3132	638	-3318
1.1	-1939	-700	-2639	511	-1565	747	-1930
1.2	-289	-40	-329	1111	-912	1762	-1403
1.24	156	122	278	1340	-922	2184	-1337
1.3	823	366	1189	1684	-937	2818	-1237
1.4	1477	601	2078	2082	-1038	3493	-1170
1.5	1687	678	2365	2229	-1072	3715	-1163
1.6	1467	603	2070	2113	-1073	3577	-1238
1.7	777	351	1128	1811	-1086	3008	-1390
1.76	75	91	166	1466	-1053	2375	-1558
1.8	-393	-83	-476	1236	-1031	1953	-1670
1.9	-2085	-775	-2860	513	-1655	813	-2163
2.0	-4414	-1834	-6248	529	-3162	628	-3280
2.0	-4414	-1834	-6248	529	-3162	628	-3280
2.1	-2640	-996	-3636	495	-1956	552	-2538
2.2	-1270	-426	-1696	935	-1438	1458	-2188
2.3	-174	36	-138	1475	-1274	2387	-1920
2.35	243	198	440	1715	-1211	2798	-1792
2.4	659	359	1018	1954	-1148	3208	-1663
2.5	1209	573	1782	2222	-1001	3575	-1398
2.6	1519	689	2208	2284	-830	3705	-1118
2.7	1545	712	2257	2103	-640	3493	-842
2.8	1295	598	1893	1701	-439	2842	-560
2.9	742	349	1091	994	-226	1637	-280
3.0	0	0	0	0	0	0	0

Figure 76 - Structural analysis girder moments

1.14.1 Bentley OpenBridge

Bentley LEAP Steel software allows specification of support conditions underneath girders along with load cases and combinations. Material and profile information is obtained from physical elements modeled accordingly.

Figure 76 illustrates definition of support types at each bearing.

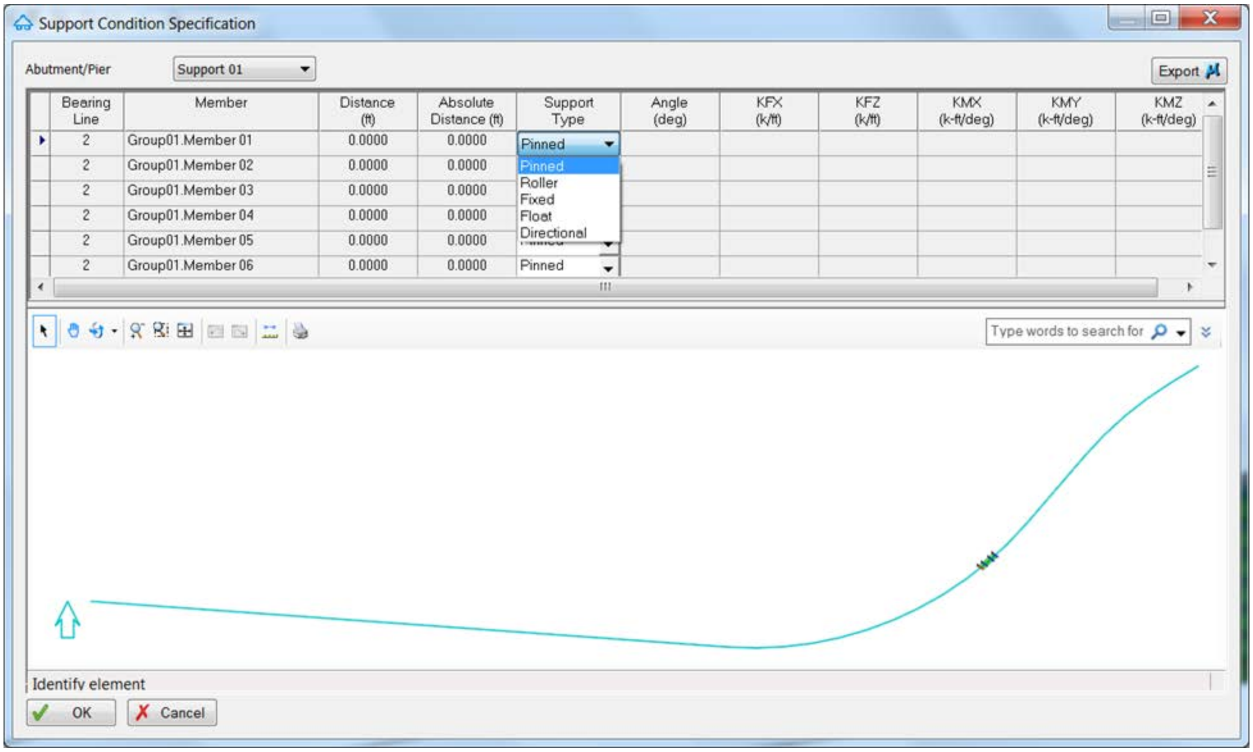


Figure 77 - Structural support condition with Bentley

As shown in Figure 77, Dead loads may be defined along girders using a combination of point or curved forces and moments.

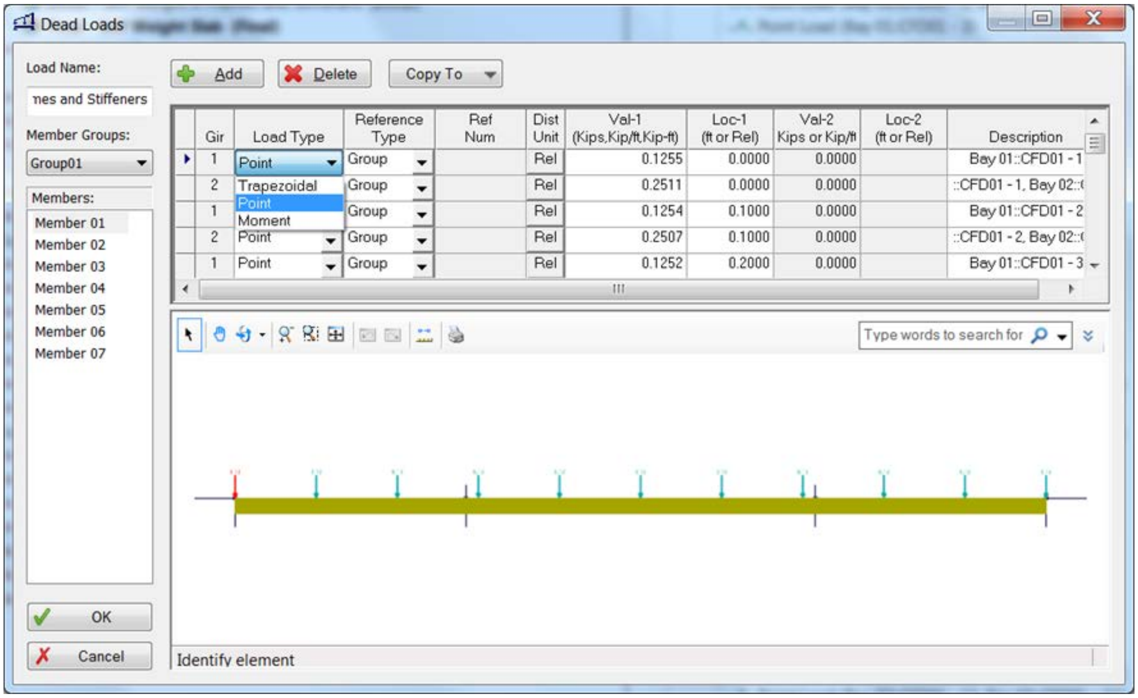


Figure 78 - Structural loads with Bentley

As shown in Figure 78, load cases and combinations may be defined for partial framing at each stage (as defined for construction stages).

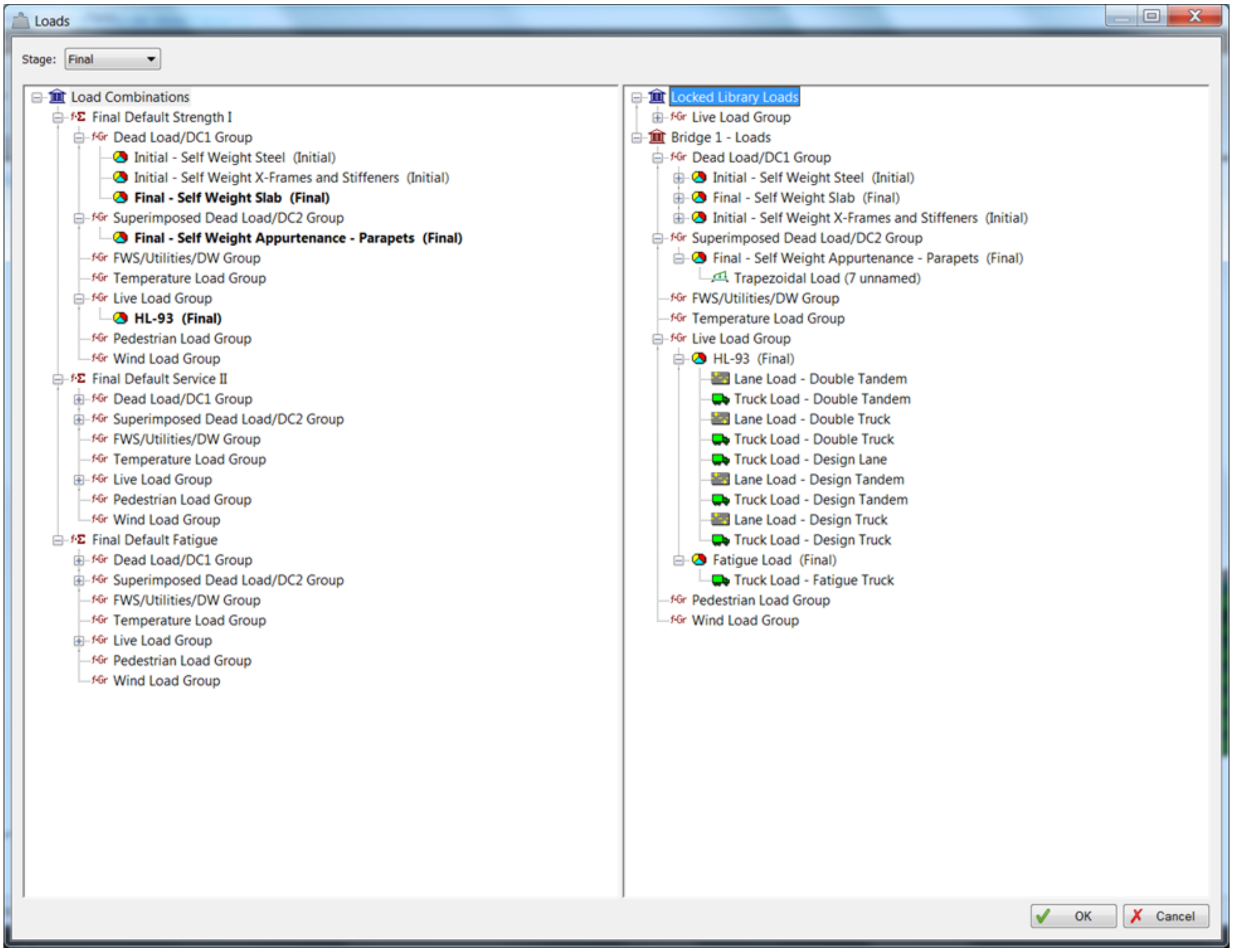


Figure 79 - Structural load cases with Bentley

1.14.2 Industry Foundation Classes

Support conditions may be derived from connectivity relationships where IfcRelConnectsWithRealizingElements may be used to specifically identify the connection geometry. The corresponding structural model may be captured using IfcAppliedCondition as shown in Figure 79.

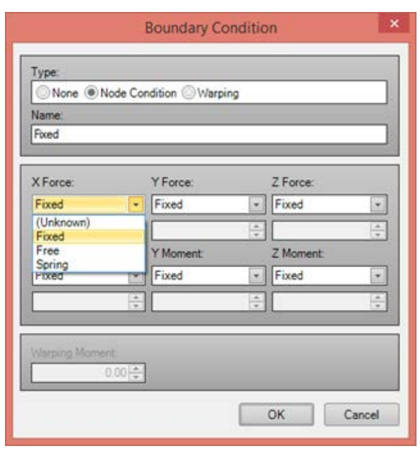


Figure 80 - Structural boundary condition

As shown in Figure 80, curved loads may be defined as constant, linear, polygonal, equidistant, sinus, parabola, or discrete values. For this particular bridge, only constant loading is used.

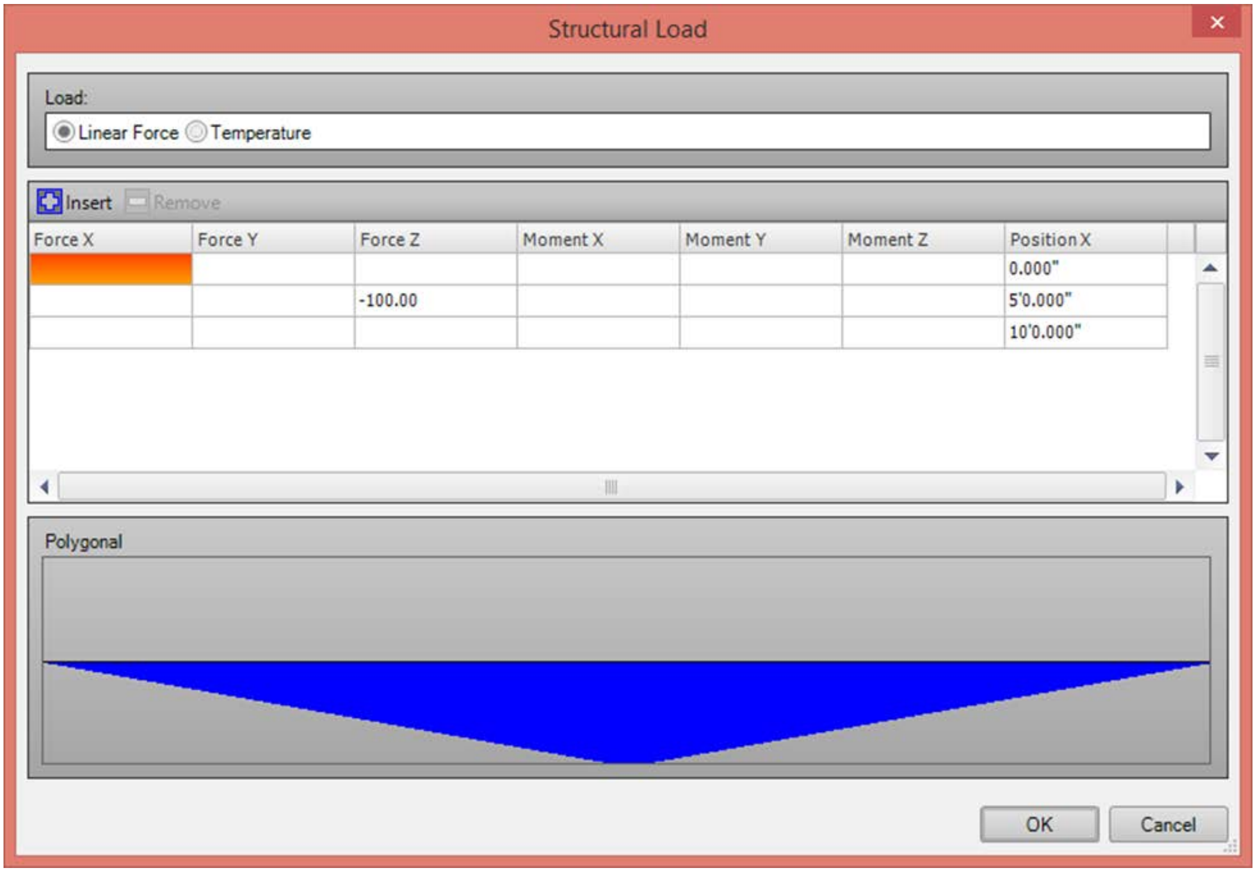


Figure 81 - Structural load

1.14.3 OpenBrIM

Structural point loads have been defined in the OpenBrIM specification. Yet to be defined are loads along paths or areas, and the enclosing load cases, combinations, and results as detailed within the plans for the sample bridge.

2 Implementation

Critical to the analysis of data schemas for bridges is to apply such schemas to a previously constructed bridge reflecting the full detail of what is described in the plans. Significant effort was applied in generating sample bridges in IFC formats that capture all the major components. As typical with many modeling efforts, the Pareto principal applies, where capturing the last 20% of what goes into design documents took approximately 80% of the time for this endeavor. The example files reflect all components in the plans as indicated.

NOTE: This Report only includes two examples, as developing such detailed information is very time consuming, particularly before software is available to automate the production of information.

2.1 Data Formats

The examples provided have been generated in multiple data schemas and multiple data formats for each schema. Table 3 summarizes the files as published.

Table 3 - Files for test cases

File	Schema	Format	Size
Bridge-Turnpike.ifc	IFC4x1	SPF	2,463 KB
Bridge-Turnpike.ifcxml	IFC4x1	XML	5,132 KB
Bridge-VanWhite.ifc	IFC4x1	SPF	5,910 KB
Bridge-VanWhite.ifcxml	IFC4x1	XML	15,945 KB

Each of these files was generated in canonical form for readability; thus, the encoding of information is less efficient than it would be if optimized for file size. Both the STEP Physical File (SPF) and eXtended Markup Language (XML) formats, as described further in Volume II of this Report, were generated to intern repetitive structures (referencing one instance multiple times).

In addition to the IFC4x1 schema, each bridge was also converted to the IFC2x3 schema to assist software developers in understanding new functionality required in IFC4x1. Data structures unsupported in IFC2x3 were either dropped or converted into alternative representations such as B-rep geometry.

As these files are relatively large, they would consume thousands of pages if included within this document; therefore, small snippets are shown for illustration.

The appendices of this document contain snippets that illustrate modeling in each of these formats.

2.2 Correlation of Plan Model to Integrated Data Model

In modeling the sample bridges, a summary was made of the IFC data structures corresponding to information found on each page of the plans.

2.2.1 Penn Turnpike Bridge

Table 4 relates the sheets of the Penn Turnpike plan to the contents of the file.

Table 4 - Penn Turnpike plan contents

#	Description	Data Structures
1	General plan and elevation	IfcAlignment, IfcGeographicElement
2	General notes	IfcProject
3	Index, typical sections	IfcSlabType, IfcMaterialProfileSet, IfcArbitraryClosedProfileDef
4	Estimate of quantities 1	IfcCostSchedule, IfcCostItem, IfcElementQuantity
5	Estimate of quantities 2	IfcCostSchedule, IfcCostItem, IfcElementQuantity
6	Stake-out plan	IfcAlignment, IfcCivilElement, IfcAlignmentPlacement
7	Abutment 1 plan	IfcWall, IfcPlate, IfcPipeSegment, IfcCableCarrierSegment
8	Abutment 1 pile, reinforcement	IfcFooting, IfcPile, IfcReinforcingBar
9	Abutment 1 elevation	IfcWall, IfcExtrudedAreaSolidTapered, IfcArbitraryClosedProfileDef
10	Abutment 1 wing walls	IfcWall, IfcExtrudedAreaSolidTapered, IfcArbitraryClosedProfileDef
11	Abutment 1 backfill and barrier	IfcRelSpaceBoundaries, IfcExternalSpatialStructure
12	Abutment 1 reinforcement	IfcReinforcingBar, IfcSweptDiskSolid, IfcIndexedPolyCurve
13	Abutment 2 plan	IfcWall, IfcPlate, IfcPipeSegment, IfcCableCarrierSegment
14	Abutment 2 pile, reinforcement	IfcFooting, IfcPile, IfcReinforcingBar
15	Abutment 2 elevation	IfcWall, IfcExtrudedAreaSolidTapered, IfcArbitraryClosedProfileDef
16	Abutment 2 sections	IfcWall, IfcExtrudedAreaSolidTapered, IfcArbitraryClosedProfileDef
17	Abutment 2 wing wall C	IfcWall, IfcExtrudedAreaSolidTapered, IfcArbitraryClosedProfileDef
18	Abutment 2 wing wall D	IfcWall, IfcExtrudedAreaSolidTapered, IfcArbitraryClosedProfileDef
19	Abutment 2 wing wall details	IfcWall, IfcExtrudedAreaSolidTapered, IfcArbitraryClosedProfileDef
20	Abutment 2 MSE wall details	IfcWall, IfcExtrudedAreaSolidTapered, IfcArbitraryClosedProfileDef
21	Abutment 2 backfill	IfcRelSpaceBoundaries, IfcExternalSpatialStructure
22	Abutment 2 reinforcement	IfcReinforcingBar, IfcSweptDiskSolid, IfcIndexedPolyCurve
23	Pier 1 geometry	IfcMember, IfcExtrudedAreaSolid, IfcArbitraryClosedProfileDef, IfcIndexedPolyCurve
24	Pier 1 column and cap details	IfcColumn, IfcMember, IfcReinforcingBar, IfcExtrudedAreaSolid
25	Pier 1 footing details	IfcFooting, IfcExtrudedAreaSolid, IfcCircleProfileDef, IfcReinforcingBar
26	Pier 2 geometry	IfcMember, IfcExtrudedAreaSolid, IfcArbitraryClosedProfileDef, IfcIndexedPolyCurve
27	Pier 2 column and cap details	IfcColumn, IfcMember, IfcReinforcingBar, IfcExtrudedAreaSolid
28	Pier 2 footing details	IfcFooting, IfcExtrudedAreaSolid, IfcCircleProfileDef, IfcReinforcingBar
29	Pier reinforcement	IfcReinforcingBar, IfcRelAssociatesConstraint, IfcTable
30	Girder framing plan	IfcBeam, IfcRelAssociatesConstraint, IfcTable, IfcIShapeProfileDef
31	Girder elevations	IfcBeam, IfcRelAssociatesConstraint, IfcTable, IfcIShapeProfileDef
32	Girder crossframe details	IfcPlate, IfcRelConnectsElements, IfcMember, IfcFastener, IfcLShapeProfileDef
33	Girder field splice details	IfcPlate, IfcRelConnectsWithRealizingElements, IfcMechanicalFastener
34	Girder details	IfcPlate, IfcRelConnectsWithRealizingElements, IfcConnectionSurfaceGeometry
35	Moment tables	IfcStructuralResultGroup, IfcStructurePointReaction, IfcStructuralPointLoad
36	Lateral moment tables	IfcStructuralResultGroup, IfcStructurePointReaction, IfcStructuralPointLoad
37	Shear tables	IfcStructuralResultGroup, IfcStructurePointReaction, IfcStructuralPointLoad
38	Girder design parameters 1,2	IfcStructuralLoadGroup, IfcStructuralCurveAction, IfcStructuralLinearLoad
39	Girder design parameters 3,4	IfcStructuralLoadGroup, IfcStructuralCurveAction, IfcStructuralLinearLoad
40	Future rating example	--
41	Camber table and diagram	IfcShapeRepresentation, IfcRelAssociatesConstraint, IfcTable
42	Bearing location plan, table	IfcMechanicalFastener, IfcRelConnectsWithRealizingElements
43	Non-guided pot bearing details	IfcMechanicalFastener, IfcExtrudedAreaSolid
44	Guided pot bearing details 1	IfcMechanicalFastener, IfcExtrudedAreaSolid
45	Guided pot bearing details 2	IfcMechanicalFastener, IfcExtrudedAreaSolid
46	Pot bearing notes and details	IfcMechanicalFastenerType
47	Misc. pot bearing details	IfcMechanicalFastenerType
48	Pot bearing dimension table	IfcMechanicalFastenerType, IfcRelAssociatesConstraint, IfcTable
49	Future jacking notes	--

#	Description	Data Structures
50	Deck elevations	IfcSlab, IfcRelAssociatesConstraint, IfcTable
51	Deck paving plan and section	IfcSlab, IfcRelAssociatesMaterial, IfcMaterialProfileSet, IfcMaterialProfile
52	Deck paving plan	IfcSlab, IfcRelAggregates
53	Deck paving details	IfcSlab, IfcRailing, IfcRelAssociatesMaterial, IfcMaterialProfileSet, IfcMaterialProfile
54	Deck reinforcement	IfcReinforcingBar, IfcRelAssociatesConstraint, IfcTable
55	Pier drainage details	IfcWasteTerminal, IfcPipeSegment, IfcValve, IfcDistributionPort, IfcRelConnectsPorts
56	Future redecking	IfcTaskType, IfcRelAssignsToProduct
57	Erection scheme	IfcTaskType, IfcRelAssignsToProduct
58	Grading, foundation window	IfcGeographicElement
59	Architectural details	IfcCovering, IfcSurfaceStyle, IfcImageTexture
60	Plan and location of borings	IfcGeographicElement, IfcAlignmentPlacement
61	Test boring logs	IfcGeographicElement, IfcRelAssociatesMaterial, IfcMaterialLayerSet
62	Test boring logs	IfcGeographicElement, IfcRelAssociatesMaterial, IfcMaterialLayerSet
63	Test boring logs	IfcGeographicElement, IfcRelAssociatesMaterial, IfcMaterialLayerSet
64	Test boring logs	IfcGeographicElement, IfcRelAssociatesMaterial, IfcMaterialLayerSet
65	Test boring logs	IfcGeographicElement, IfcRelAssociatesMaterial, IfcMaterialLayerSet
66	Test boring logs	IfcGeographicElement, IfcRelAssociatesMaterial, IfcMaterialLayerSet
67	Test boring logs	IfcGeographicElement, IfcRelAssociatesMaterial, IfcMaterialLayerSet
68	Test boring logs	IfcGeographicElement, IfcRelAssociatesMaterial, IfcMaterialLayerSet
69	Test boring logs	IfcGeographicElement, IfcRelAssociatesMaterial, IfcMaterialLayerSet

2.2.2 Van White Bridge

Table 5 relates the sheets of the Van White plan to the contents of the file.

Table 5 - Van White plan contents

#	Description	Data Structures
1	General Plan & Elevation	IfcProject, IfcAlignment
2	List of Sheets & Schedule of Quantities	IfcCostSchedule, IfcCostItem, IfcPhysicalQuantity
3	Typical Bridge Sections	IfcSlabType, IfcMaterialProfileSet, IfcArbitraryClosedProfileDef
4	General Notes & Post-Tensioning Notes	IfcStructuralLoadCase, IfcStructuralAction, IfcMaterialProperties,
5	Bridge Layout	IfcRelAssociatesConstraint, IfcMetric, IfcTable, IfcReference
6	Assembled South Abutment Plan View	IfcCivilElement
7	South Abutment Coordinate Layout	IfcFooting, IfcWall, IfcExtrudedAreaSolidTapered
8	South Abutment Plan & Elevation	IfcFooting, IfcWall, IfcExtrudedAreaSolidTapered
9	South Abutment Rustication Details	IfcCovering, IfcSurfaceStyle
10	South Abutment Footing Plan & Reinforcement	IfcPile, IfcReinforcingBar, IfcSweptDiskSolid
11	South Abutment Reinforcement	IfcReinforcingBar
12	South Abutment Reinforcement Details (1 of 2)	IfcReinforcingBarType, IfcSweptDiskSolidPolygonal
13	South Abutment Reinforcement Details (2 of 2)	IfcReinforcingBarType, IfcSweptDiskSolidPolygonal
14	Southeast Wing wall Plan & Elevation	IfcWall, IfcExtrudedAreaSolidTapered, IfcSectionedSpine
15	Southeast Wing wall Footing Plan & Reinforcement	IfcFooting, IfcPile, IfcReinforcingBar, IfcMappedItem, IfcSweptDiskSolid
16	Southeast Wing wall Reinforcement	IfcReinforcingBar
17	Southeast Bridgehead Dimensions & Coordinates	IfcWall, IfcFooting, IfcPile, IfcReinforcingBar
18	Southeast Bridgehead Footing Plan & Reinforcement (1 of 3)	IfcFooting, IfcPile, IfcReinforcingBar
19	Southeast Bridgehead Footing Plan & Reinforcement (2 of 3)	IfcFooting, IfcPile, IfcReinforcingBar
20	Southeast Bridgehead Footing Plan & Reinforcement (3 of 3)	IfcFooting, IfcPile, IfcReinforcingBar
21	Southeast Bridgehead Wall Reinforcement	IfcReinforcingBar, IfcReinforcingMesh
22	Southeast Bridgehead Elevation View	
23	Southeast Bridgehead Footing & Pedestal Details	IfcFooting, IfcPile, IfcReinforcingBar
24	Southeast Bridgehead Support Column Geometry	IfcColumn, IfcFacetedBrep
25	Southeast Bridgehead Pedestal and Column Reinforcement	IfcFooting, IfcPile, IfcReinforcingBar
26	Southeast Bridgehead Construction Details	IfcMemberType, IfcArbitraryProfileDefWithVoids

#	Description	Data Structures
27	Southeast Bridgehead Sculpted Top Portion	IfcMemberType, IfcFacetedBrep
28	South Retaining Wall Coordinate Layout	IfcWall, IfcFooting
29	South Retaining Wall Plan & Elevation	IfcWall
30	South Retaining Wall Footing Plan & Reinforcement	IfcFooting, IfcPile, IfcReinforcingBar
31	South Retaining Wall Reinforcement	IfcReinforcingBar, IfcSweptDiskSolidPolygonal
32	Assembled North Abutment Plan View	IfcCivilElement
33	North Abutment Coordinate Layout	IfcWall, IfcFooting
34	North Abutment Plan & Elevation	IfcWall, IfcFooting
35	North Abutment Rustication Details	IfcCovering, IfcSurfaceStyle
36	North Abutment Elevation Views	IfcWall, IfcFooting
37	North Abutment Footing Plan & Reinforcement	IfcFooting, IfcPile, IfcReinforcingBar
38	North Abutment Reinforcement	IfcReinforcingBar, IfcMappedItem
39	North Abutment Reinforcement Details (1 of 2)	IfcReinforcingBarType, IfcSweptDiskSolidPolygonal
40	North Abutment Reinforcement Details (2 of 2)	IfcReinforcingBarType, IfcSweptDiskSolidPolygonal
41	Northeast Wing wall Reinforcement	IfcReinforcingBar
42	Northeast Plaza Layout Dimensions	IfcWall
43	Northeast Plaza Coordinate Layout (1 of 2)	IfcWall, IfcFooting
44	Northeast Plaza Coordinate Layout (2 of 2)	IfcWall, IfcFooting
45	Northeast Plaza Wall Elevation	IfcWall
46	Northeast Plaza Footing Plan & Reinforcement	IfcFooting
47	Northeast Plaza Wall Reinforcement	IfcReinforcingBar
48	Northeast Bridgehead Sculpture Elevation	IfcFurnishingElement
49	Northeast Bridgehead Sculpture Plan & Details	IfcFurnishingElement
50	Northeast Bridgehead Sculpture Support Column Details	IfcColumn
51	Northeast Bridgehead Sculpture Ring Details	IfcMember
52	Abutment Bar Lists (1 of 2)	IfcReinforcingBarType
53	Abutment Bar Lists (2 of 2)	IfcReinforcingBarType
54	Summary of Quantities for Abutments and Plaza	IfcCostSchedule, IfcCostItem
55	Pier 1 Footing Plan & Reinforcement	IfcCivilElement, IfcFooting, IfcPile, IfcReinforcingBar
56	Pier 1 Column Details & Reinforcement	IfcColumn, IfcReinforcingBar
57	Pier 2 Footing Plan & Reinforcement	IfcCivilElement, IfcFooting, IfcPile, IfcReinforcingBar
58	Pier 2 Column Details & Reinforcement	IfcColumn, IfcReinforcingBar
59	Pier 3 Footing Plan & Reinforcement	IfcCivilElement, IfcFooting, IfcPile, IfcReinforcingBar
60	Pier 3 Column Details & Reinforcement	IfcColumn, IfcReinforcingBar
61	Pier 4 Footing Plan & Reinforcement	IfcCivilElement, IfcFooting, IfcPile, IfcReinforcingBar
62	Pier 4 Column Details & Reinforcement	IfcColumn, IfcReinforcingBar
63	Pier Quantities and Bar Lists	IfcCostSchedule, IfcCostItem
64	Framing Plan	IfcAlignment
65	Typical Cross Section	IfcCivilElementType, IfcMaterialProfileSet
66	Superstructure Details	IfcTendon
67	Geometric Data for Superstructure (1 of 2)	IfcSectionedSpine
68	Geometric Data for Superstructure (2 of 2)	IfcSectionedSpine
69	Soffit Slab Reinforcement Bottom Layer	IfcReinforcingBar, IfcReinforcingMesh
70	Soffit Slab Reinforcement Top Layer	IfcReinforcingBar, IfcReinforcingMesh
71	West Fascia Girder Reinforcement Details	IfcReinforcingBarType
72	Center Girder Reinforcement Details	IfcReinforcingBarType
73	East Fascia Girder Reinforcement Details	IfcReinforcingBarType
74	South Abutment End Diaphragm Reinforcement & Details	IfcReinforcingBar
75	Pier 1 Cap Beam Reinforcement & Details	IfcReinforcingBar
76	Pier 2 Cap Beam Reinforcement & Details	IfcReinforcingBar
77	Pier 3 Cap Beam Reinforcement & Details	IfcReinforcingBar
78	Pier 4 Cap Beam Reinforcement & Details	IfcReinforcingBar
79	North Abutment End Diaphragm Reinforcement & Details	IfcReinforcingBar
80	Deck Slab Reinforcement Bottom Layer	IfcReinforcingBar, IfcReinforcingMesh
81	Deck Slab Reinforcement Top Layer	IfcReinforcingBar, IfcReinforcingMesh
82	Miscellaneous Superstructure Details	IfcDiscreteAccessory
83	Post-Tensioning Sequence Details	IfcTendon, IfcTendonAnchor

#	Description	Data Structures
84	West Traffic Railing Elevation Views	IfcRailing, IfcMember
85	West Traffic Railing Details	IfcRailingType, IfcMemberType
86	East Traffic Railing Elevation Views	IfcRailing, IfcMember
87	East Traffic Railing Details	IfcRailingType, IfcMemberType
88	East Sidewalk Railing Elevation Views	IfcRailing, IfcMember
89	East Sidewalk Railing Details	IfcRailingType, IfcMemberType
90	Railing Details of South Common Railing	IfcMemberType, IfcFixedReferenceSweptAreaSolid
91	Railing Details of North Common Railing	IfcMemberType, IfcFixedReferenceSweptAreaSolid
92	Railing Details at Southeast Wingwall	IfcMemberType, IfcFixedReferenceSweptAreaSolid
93	Railing Details at Northeast Wingwall	IfcMemberType, IfcFixedReferenceSweptAreaSolid
94	Railing Cover Plate Details	IfcPlateType
95	Railing Details for Northeast Plaza Wall	IfcRailingType
96	Bridge Soffit Lighting Details	IfcLightFixture, IfcLightFixtureType
97	Details for Ornamental Metal Railing, Type Special (1 of 3)	IfcRailingType
98	Details for Ornamental Metal Railing, Type Special (2 of 3)	IfcRailingType
99	Details for Ornamental Metal Railing, Type Special (3 of 3)	IfcRailingType
100	Superstructure Bar Lists (1 of 2)	IfcReinforcingBarType
101	Superstructure Bar Lists (2 of 2)	IfcReinforcingBarType
102	Summary of Quantities for Superstructure	IfcCostSchedule, IfcCostItem
103	Modular Exp. Device & Conduit Systems at South Abutment	IfcJunctionBox, IfcCableCarrierSegment, IfcDistributionSystem
104	Modular Exp. Device & Conduit Systems at North Abutment	IfcJunctionBox, IfcCableCarrierSegment, IfcDistributionSystem
105	Electric Light Systems	IfcLightFixture, IfcDistributionSystem
106	Lighting Conduit Layout & Details	IfcJunctionBox, IfcCableCarrierSegment, IfcDistributionSystem
107	South Abutment Approach Panel Details	IfcFastener, IfcMechanicalFastener
108	North Abutment Approach Panel Details	IfcFastener, IfcMechanicalFastener
109	Standard Details B101 & B201	IfcDiscreteAccessoryType
110	Pot Bearing Assemblies	IfcMechanicalFastenerType
111	Standard Details B942 & B910	IfcPipeSegmentType, IfcMechanicalFastenerType
112	As-Built Bridge Data	IfcRelAssignsToProduct
113	Bridge Survey	IfcSite, IfcGeographicElement
114	Bridge Survey Plan & Profile (1 of 3)	IfcGeographicElement, IfcTriangulatedFaceSet
115	Bridge Survey Plan & Profile (2 of 3)	IfcGeographicElement, IfcTriangulatedFaceSet
116	Bridge Survey Plan & Profile (3 of 3)	IfcGeographicElement, IfcTriangulatedFaceSet
117	Boring Logs	IfcGeographicElement, IfcMaterialLayerSet, IfcMaterial, IfcMaterialProperties

3 Findings

3.1 Data Modeling

For each of the bridges, modeling surface geometry of structural components required only several days of effort, while modeling detailed aspects required for construction including rebar, tendons, railings, conduit, lighting, drainage, and site conditions required several months. It is understood that modeling in the reverse direction (from plans) does not benefit from automation capabilities found in design software (e.g. automatic rebar layout) that would be used in practice. Some of the most critical modeling features were only realized after modeling such detail in depth.

The Penn Turnpike Bridge illustrated the need to support cross-sections that vary at intervals along the alignment curve. The IfcSectionedSpine data structure was found to be the most suitable for such scenarios.

The Van White Bridge illustrated the need to support transformed component geometry, with distinction between components that may be repositioned (such as rebar) from those that must be reshaped (such as architectural railings along sharp curves). The coordinate transformation capability in IFC using IfcCartesianTransformationOperator became integral in defining such shapes where the resulting axes are not orthogonal but skewed. A new positioning relationship (IfcRelPositions) was proposed to handle placement of bridge elements along alignment curves as well as parametric transformation of geometry.

For both bridges, it became evident that generalized approaches to sweeping cross-sections along alignment curves, while adequate for capturing design intent, are entirely insufficient to describe construction detail. For such construction detail, box girders and steel plate girders typically gain thicknesses near piers, and often use entirely different geometry for pier caps and diaphragms. Additionally, provisions are made for construction joints, expansion joints (including tapering of adjacent components), access panels, embedded conduit, drainage, rebar, tendons, and vents, all of which require specific placement that cannot be conveyed by parameterization alone. It became evident that while parameterized templates may be used for some components, many other components may lend themselves to design algorithms following structural design codes.

The need for parametric positioning of components became evident as a way to efficiently define repetitive elements. For the Van White Bridge, each cross-frame occurrence had unique dimensions determined by the positions of the connecting girders. Existing IFC connectivity relationships such as IfcRelConnectsWithRealizingElements were leveraged to capture such dependencies and provide a standard way for design software to resize components and all dependencies automatically.

After introducing positioning capabilities for bridge superstructure, it also became evident that secondary elements could also leverage such capability. The north abutment of the Van White bridge had a retaining wall for a walkout plaza that took the form of a horizontal spiral with vertical incline – such wall geometry was able to leverage the same positioning functionality intended for bridge decks and girders.

Implications for memory usage, application performance, GPU load, and bandwidth also became evident after modeling megabytes of detail. At this level, it became critical to minimize unique geometry and maximize repetition and scaling of such geometry. Usage of IfcMappedItem became critical for rebar, along with IfcSweptDiskSolidPolygonal, recently introduced in IFC4. With all critical data types built into a compiled schema (early-bound), performance was sufficient, such that the Van White bridge could be fully loaded within 5 seconds on a 64-bit i7 system with 16GB RAM.

3.2 Vendor Feedback

Feedback from bridge design software vendors indicated that existing IFC geometry is entirely sufficient to capture the data they support, such that there is no need to introduce new data structures for defining swept geometry according to alignment curves; rather CAD platforms always use STEP-compatible curve types, converting from alignment curves as necessary.

Feedback from fabrication software vendors expressed a strong desire to leverage the same version of IFC for buildings and infrastructure, as they already deal with both domains and have no business or technical reasons to distinguish between them in their operations.

Feedback from building design software vendors indicated a preference to retain compatibility as much as possible, where it is desirable to leverage existing released software for using such models.

Other feedback indicated a need to assign responsibility of particular elements within a bridge model, which resulted in inclusion of the IfcOwnerHistory data structure.

3.3 Verification

The model bridges were loaded into software that was made accessible to the project team, which included Autodesk Revit, Bentley Microstation, and Tekla Structures. All software was able to load the model bridges without error. Though certain capabilities were available in some software but not others, and no software supported everything, collectively all geometry was supported, with every component captured in at least one of the software applications.

One of the key data structures, IfcSectionedSpine, was supported in Tekla but not the others, however it has never been part of the official IFC coordination view implemented by CAD software. Several new IFC4 data structures finding common usage such as IfcExtrudedAreaSolidTapered and IfcSweptDiskSolidPolygonal (part of the IFC4 Design Transfer view) were not supported, which required downgrading to IfcFacetedBrep and IfcSweptDiskSolid respectively. Non-uniform transformation of detailed components along curved alignments was supported by Bentley, but not others.

Data structures for connectivity and parametric layout, while included, were not used within the tested software. Unsurprisingly, none of the software systems supported alignment curves, which were only introduced within the past year as part of IFC4.1. Upon discussions with vendors who already support IFC, all have committed to supporting the additional capabilities needed for bridges and some have had their software development teams engaged at their own will.

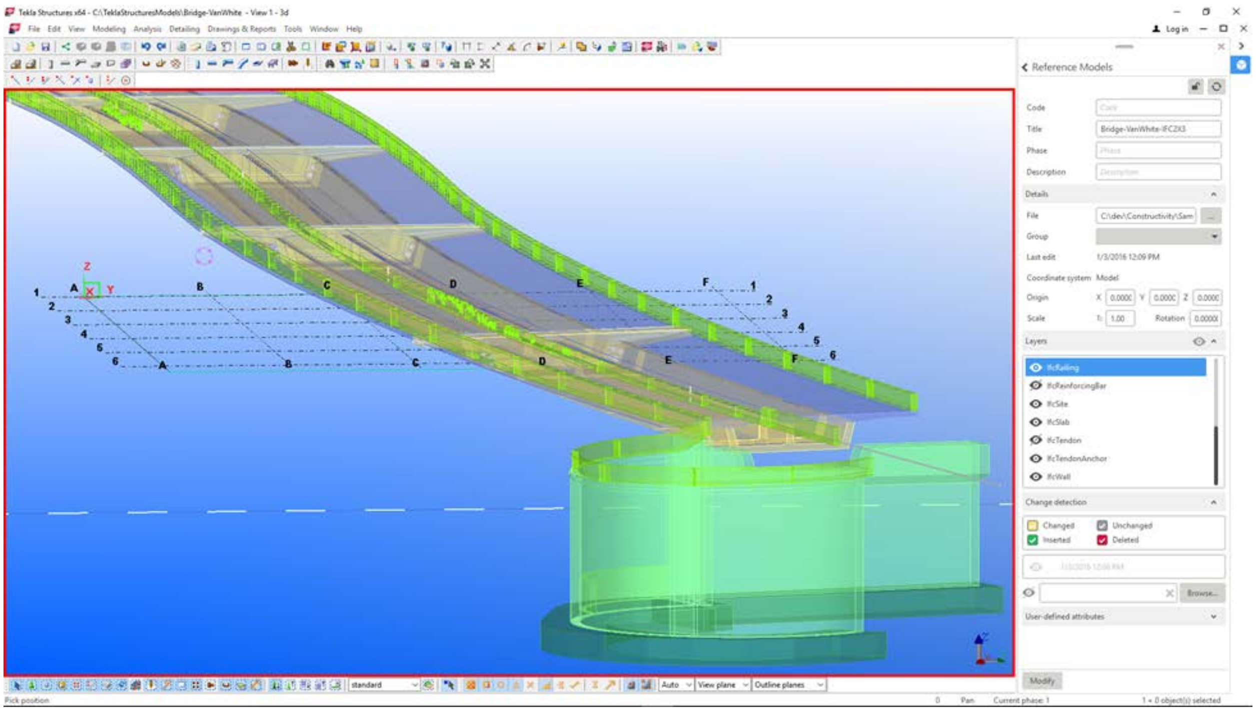


Figure 82 - Van White bridge loaded into Tekla Structures

3.4 Information Standards

Of the various data schemas evaluated, it is generally accepted that given enough time, money, and will, any such schema could sufficiently address industry requirements. Thus, a decision to leverage a particular schema isn't necessarily driven by the assumption that one is inherently better, but may involve a calculated risk in predicting which is most likely to gain the most support in industry. Such a decision is similar to choosing a phone (iPhone, Android, Windows, Blackberry, etc.), where technical merits may have mattered more in earlier stages of the marketplace, but industry support (applications, interoperability, etc.) ultimately becomes most relevant. While past and present conditions are not always predictors of the future, software companies generally favor developing towards known functionality with established user bases before investing in what is unknown.

This Report has attempted to demonstrate modeling of complete bridges using the IFC schema for purposes of bidding and construction of completed designs using representative test cases. While this Report has gone into detail describing particular data structures and physical elements of bridges, the ultimate test of whether a schema works (by nature of being documented sufficiently to work), is to move data between independent software platforms using such schema. As it was possible to load this data into three of the major software platforms without customization, the first step in attempting usage of IFC has been proven. The next step is to attempt usage on real projects with real parties involved having interest in using the data.

Appendix C:1 IFC-XML Format Sample


```
<ifc:IfcSlab id="i6039" GlobalId="2X6PApeLr3efgfjSMea_5n" Name="Deck Span 5">
    <ifc:IsDecomposedBy>
    <ifc:IfcRelAggregates GlobalId="1CUcbZR5z8jPW4AU58n$jz">
    <ifc:RelatedObjects>
    <ifc:IfcReinforcingBar id="i6040" GlobalId="25X8rS2iXAkfS7fwdYPrpW" Name="Bottom-Lateral-1-
West" NominalDiameter="0" CrossSectionArea="0" PredefinedType="notdefined">
        <ifc:ObjectPlacement xsi:type="ifc:IfcLocalPlacement" id="i6041">
        <ifc:PlacementRelTo xsi:type="ifc:IfcLocalPlacement" xsi:nil="true" ref="i6042" />
        <ifc:RelativePlacement>
        <ifc:IfcAxis2Placement3D>
            <ifc:Location Coordinates="0 0 0" />
            <ifc:Axis DirectionRatios="0 0 1" />
            <ifc:RefDirection DirectionRatios="0 -1 0" />
        </ifc:IfcAxis2Placement3D>
        </ifc:RelativePlacement>
        </ifc:ObjectPlacement>
        <ifc:Representation xsi:type="ifc:IfcProductDefinitionShape">
        <ifc:Representations>
        <ifc:IfcShapeRepresentation RepresentationIdentifier="Body"
RepresentationType="MappedRepresentation">
            <ifc:ContextOfItems xsi:type="ifc:IfcGeometricRepresentationContext" xsi:nil="true"
ref="i6043" />
            <ifc:Items>
            <ifc:IfcSweptDiskSolidPolygonal Radius="0.25" FilletRadius="0.25">
            <ifc:Directrix xsi:type="ifc:IfcPolyline">
                <ifc:Points>
                <ifc:IfcCartesianPoint Coordinates="6113.46000415686 486.180915725691
482.765236016929" />
                <ifc:IfcCartesianPoint Coordinates="6181.60477179572 502.132261053177
479.27943682823" />
                </ifc:Points>
            </ifc:Directrix>
            </ifc:IfcSweptDiskSolidPolygonal>
            </ifc:Items>
        </ifc:IfcShapeRepresentation>
        </ifc:Representations>
        </ifc:IfcRepresentation>
    </ifc:RelatedObjects>
    </ifc:IfcRelAggregates>
    </ifc:IsDecomposedBy>
    <ifc:ObjectPlacement xsi:type="ifc:IfcLocalPlacement" id="i6042">
    <ifc:RelativePlacement>
    <ifc:IfcAxis2Placement3D>
        <ifc:Location Coordinates="0 0 0" />
    </ifc:IfcAxis2Placement3D>
    </ifc:RelativePlacement>
    </ifc:ObjectPlacement>
    <ifc:Representation xsi:type="ifc:IfcProductDefinitionShape">
    <ifc:Representations>
    <ifc:IfcShapeRepresentation RepresentationIdentifier="Body"
RepresentationType="MappedRepresentation">
        <ifc:ContextOfItems xsi:type="ifc:IfcGeometricRepresentationContext" xsi:nil="true"
ref="i6134" />
        <ifc:Items>
        <ifc:IfcSectionedSpine>
        <ifc:SpineCurve>
        <ifc:IfcCompositeCurve SelfIntersect="unknown">
            <ifc:Segments>
            <ifc:IfcCompositeCurveSegment Transition="continuous" SameSense="true">
            <ifc:ParentCurve xsi:type="ifc:IfcPolyline">
                <ifc:Points>
                <ifc:IfcCartesianPoint Coordinates="6174.20629864121 235.416811275201
490.622042824639" />
                <ifc:IfcCartesianPoint Coordinates="6449.42291393482 303.269802669081
476.575256228744" />
                </ifc:Points>
            </ifc:ParentCurve>
            </ifc:IfcCompositeCurveSegment>
```

```
            <ifc:IfcCompositeCurveSegment Transition="continuous" SameSense="true">
            <ifc:ParentCurve xsi:type="ifc:IfcPolyline">
                <ifc:Points>
                <ifc:IfcCartesianPoint Coordinates="6449.42291393482 303.269802669081
476.575256228744" />
                <ifc:IfcCartesianPoint Coordinates="6722.42568726407 379.543648312938
462.401778334386" />
                </ifc:Points>
            </ifc:ParentCurve>
            </ifc:IfcCompositeCurveSegment>
            <ifc:IfcCompositeCurveSegment Transition="continuous" SameSense="true">
            <ifc:ParentCurve xsi:type="ifc:IfcPolyline">
                <ifc:Points>
                <ifc:IfcCartesianPoint Coordinates="6722.42568726407 379.543648312938
462.401778334386" />
                <ifc:IfcCartesianPoint Coordinates="6992.9570291542 464.166380660072
448.228300440029" />
                </ifc:Points>
            </ifc:ParentCurve>
            </ifc:IfcCompositeCurveSegment>
            <ifc:IfcCompositeCurveSegment Transition="continuous" SameSense="true">
            <ifc:ParentCurve xsi:type="ifc:IfcPolyline">
                <ifc:Points>
                <ifc:IfcCartesianPoint Coordinates="6992.9570291542 464.166380660072
448.228300440029" />
                <ifc:IfcCartesianPoint Coordinates="7260.7616820288 557.058154641827
434.054822545671" />
                </ifc:Points>
            </ifc:ParentCurve>
            </ifc:IfcCompositeCurveSegment>
            </ifc:Segments>
        </ifc:IfcCompositeCurve>
        </ifc:SpineCurve>
        <ifc:CrossSections>
        <ifc:IfcArbitraryClosedProfileDef ProfileType="area">
            <ifc:OuterCurve xsi:type="ifc:IfcPolyline">
            <ifc:Points>
            <ifc:IfcCartesianPoint Coordinates="0 0" />
            <ifc:IfcCartesianPoint Coordinates="-78.5 0" />
            <ifc:IfcCartesianPoint Coordinates="-263.5 0" />
            <ifc:IfcCartesianPoint Coordinates="-263.5 -9" />
            <ifc:IfcCartesianPoint Coordinates="-205 -9" />
            <ifc:IfcCartesianPoint Coordinates="-133.25 -14.5" />
            <ifc:IfcCartesianPoint Coordinates="-128.25 -14.5" />
            <ifc:IfcCartesianPoint Coordinates="-128.25 -16" />
            <ifc:IfcCartesianPoint Coordinates="-123.75 -16" />
            <ifc:IfcCartesianPoint Coordinates="-118.75 -14.5" />
            <ifc:IfcCartesianPoint Coordinates="-74.125 -9" />
            <ifc:IfcCartesianPoint Coordinates="-9 -9" />
            <ifc:IfcCartesianPoint Coordinates="-7 -11" />
            <ifc:IfcCartesianPoint Coordinates="-2.25 -11" />
            <ifc:IfcCartesianPoint Coordinates="-2.25 -12.5" />
            <ifc:IfcCartesianPoint Coordinates="2.25 -12.5" />
            <ifc:IfcCartesianPoint Coordinates="2.25 -11" />
            <ifc:IfcCartesianPoint Coordinates="7 -11" />
            <ifc:IfcCartesianPoint Coordinates="9 -9" />
            <ifc:IfcCartesianPoint Coordinates="74.125 -9" />
            <ifc:IfcCartesianPoint Coordinates="118.75 -14.5" />
            <ifc:IfcCartesianPoint Coordinates="123.75 -14.5" />
            <ifc:IfcCartesianPoint Coordinates="123.75 -16" />
            <ifc:IfcCartesianPoint Coordinates="128.25 -16" />
            <ifc:IfcCartesianPoint Coordinates="128.25 -14.5" />
            <ifc:IfcCartesianPoint Coordinates="133.25 -14.5" />
            <ifc:IfcCartesianPoint Coordinates="205 -9" />
            <ifc:IfcCartesianPoint Coordinates="263.5 -9" />
            <ifc:IfcCartesianPoint Coordinates="263.5 0" />
            <ifc:IfcCartesianPoint Coordinates="0 0" />
            </ifc:Points>
            </ifc:OuterCurve>
        </ifc:IfcArbitraryClosedProfileDef>
        </ifc:CrossSections>
```

```
        <ifc:CrossSectionPositions>
        <ifc:IfcAxis2Placement3D>
            <ifc:Location Coordinates="6174.20629864121 235.416811275201 490.622042824639" />
            <ifc:Axis DirectionRatios="0.974488858769484 0.224435879783399 0" />
            <ifc:RefDirection DirectionRatios="-0.224180337733208 0.973379308544722 -
1.73472347597681E-18" />
        </ifc:IfcAxis2Placement3D>
        <ifc:IfcAxis2Placement3D>
            <ifc:Location Coordinates="6449.42291393482 303.269802669081 476.575256228744" />
            <ifc:Axis DirectionRatios="0.967135912151161 0.254259960331038 0" />
            <ifc:RefDirection DirectionRatios="-0.253942728258777 0.965929247408627 0" />
        </ifc:IfcAxis2Placement3D>
        <ifc:IfcAxis2Placement3D>
            <ifc:Location Coordinates="6722.42568726407 379.543648312938 462.401778334386" />
            <ifc:Axis DirectionRatios="0.958870432562534 0.283844136027749 0" />
            <ifc:RefDirection DirectionRatios="-0.28348999271964 0.957674080396209
1.73472347597681E-18" />
        </ifc:IfcAxis2Placement3D>
        <ifc:IfcAxis2Placement3D>
            <ifc:Location Coordinates="6992.9570291542 464.166380660072 448.228300440029" />
            <ifc:Axis DirectionRatios="0.949700218827474 0.31316049297293 0" />
            <ifc:RefDirection DirectionRatios="-0.312769772577919 0.948515308045402 0" />
        </ifc:IfcAxis2Placement3D>
        <ifc:IfcAxis2Placement3D>
            <ifc:Location Coordinates="7260.7616820288 557.058154641827 434.054822545671" />
            <ifc:Axis DirectionRatios="0.939633923424163 0.34218136996411 0" />
            <ifc:RefDirection DirectionRatios="-0.341754441143144 0.938461572038964 0" />
        </ifc:IfcAxis2Placement3D>
        </ifc:CrossSectionPositions>
        </ifc:IfcSectionedSpine>
        </ifc:Items>
    </ifc:IfcShapeRepresentation>
    </ifc:Representations>
    </ifc:Representation>
    <ifc:PositionedFrom>
    <ifc:IfcRelPositions GlobalId="2kAHmBvxfEseb0V6qdXRZn" DistanceAlong="6159" LateralOffset="0"
VerticalOffset="0" PositionType="vertical" TransformType="warp">
    <ifc:RelatedPositioningElement xsi:type="ifc:IfcAlignment" xsi:nil="true" ref="i80" />
    <ifc:Axes>
1 1           </ifc:Axes>
    <ifc:RelativePlacement>
        <ifc:Location Coordinates="0 0 0" />
    </ifc:RelativePlacement>
    </ifc:IfcRelPositions>
    </ifc:PositionedFrom>
</ifc:IfcSlab>
```

Appendix C:2 IFC-SPF Format Sample

```
#16005= IFCSLAB('2X6PApeLr3efgfjSMea_5n',$,'Deck Span 5',$,$,#19229,#19230,$,$);
#19229= IFCLOCALPLACEMENT($,#19232);
#19230= IFCPRODUCTDEFINITIONSHAPE($,$,(#19233));
#19232= IFCAXIS2PLACEMENT3D(#19234,$,$);
#19233= IFCSHAPEREPRESENTATION(#19235,'Body','MappedRepresentation',(#19236));
#19234= IFCCARTESIANPOINT((0.,0.,0.));
#19235= IFCGEOMETRICREPRESENTATIONCONTEXT('3D','Model',3,1.0E-05,#19237,$);
#19236= IFCSECTIONEDSPINE(#19238,(#15096,#15096,#15096,#15096,#15096),
(#19239,#19240,#19241,#19242,#19243));
#19237= IFCAXIS2PLACEMENT3D(#19244,$,$);
#19238= IFCCOMPOSITECURVE((#19245,#19246,#19247,#19248),.U.);
#19239= IFCAXIS2PLACEMENT3D(#19249,#19250,#19251);
#19240= IFCAXIS2PLACEMENT3D(#19252,#19253,#19254);
#19241= IFCAXIS2PLACEMENT3D(#19255,#19256,#19257);
#19242= IFCAXIS2PLACEMENT3D(#19258,#19259,#19260);
#19243= IFCAXIS2PLACEMENT3D(#19261,#19262,#19263);
#19244= IFCCARTESIANPOINT((0.,0.,0.));
#19245= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#19264);
#19246= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#19265);
#19247= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#19266);
#19248= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#19267);
#19249= IFCCARTESIANPOINT((6174.20629864121,235.416811275201,490.622042824639));
#19250= IFCDIRECTION((0.974488858769484,0.224435879783399,0.));
#19251= IFCDIRECTION((-0.224180337733208,0.973379308544722,-1.73472347597681E-18));
#19252= IFCCARTESIANPOINT((6449.42291393482,303.269802669081,476.575256228744));
#19253= IFCDIRECTION((0.967135912151161,0.254259960331038,0.));
#19254= IFCDIRECTION((-0.253942728258777,0.965929247408627,0.));
#19255= IFCCARTESIANPOINT((6722.42568726407,379.543648312938,462.401778334386));
#19256= IFCDIRECTION((0.958870432562534,0.283844136027749,0.));
#19257= IFCDIRECTION((-0.28348999271964,0.957674080396209,1.73472347597681E-18));
#19258= IFCCARTESIANPOINT((6992.9570291542,464.166380660072,448.228300440029));
#19259= IFCDIRECTION((0.949700218827474,0.31316049297293,0.));
#19260= IFCDIRECTION((-0.312769772577919,0.948515308045402,0.));
#19261= IFCCARTESIANPOINT((7260.7616820288,557.058154641827,434.054822545671));
#19262= IFCDIRECTION((0.939633923424163,0.34218136996411,0.));
#19263= IFCDIRECTION((-0.341754441143144,0.938461572038964,0.));
#19264= IFCPOLYLINE((#19249,#19252));
#19265= IFCPOLYLINE((#19252,#19255));
#19266= IFCPOLYLINE((#19255,#19258));
#19267= IFCPOLYLINE((#19258,#19261));

#18595= IFCRELPOSITIONS('2kAHmBvxfEseb0V6qdXRZn',$,$,$,#16005,#15983,(1,1),#757435,6159.,0.,0.,
.Vertical.,.Warp.);
#757435= IFCAXIS2PLACEMENT3D(#757436,$,$);
#757436= IFCCARTESIANPOINT((0.,0.,0.));

#15983= IFCALIGNMENT('2azDUWpfr2$QxwhNaImyeq',$,'Northbound
Alignment',$,$,#18570,$,$,#18571,#18572,$);
#18570= IFCLOCALPLACEMENT($,#18596);
#18571= IFCALIGNMENT2DHORIZONTAL($,(#18597,#18598));
#18572= IFCALIGNMENT2DVERTICAL((#18599,#18600,#18601));
#18596= IFCAXIS2PLACEMENT3D(#18602,$,$);
#18597= IFCALIGNMENT2DHORIZONTALSEGMENT(.T.,'708+04.971','711+43.497',#18603);
#18598= IFCALIGNMENT2DHORIZONTALSEGMENT(.T.,'711+43.497','714+37.258',#18604);
#18599= IFCALIGNMENT2DVERSEGLINE(.T.,'708+01.571','709+27.000',0.,1505.148,429.36,0.045);
#18600= IFCALIGNMENT2DVERSEGPARABOLICARC(.T.,'709+27.000','713+27.000',
1505.148,4800.,497.09166,0.045,50526.24,.F.);
#18601= IFCALIGNMENT2DVERSEGLINE(.T.,'713+27.000','714+18.44',
6305.148,1097.28,485.091317999487,-0.05);
#18602= IFCCARTESIANPOINT((0.,0.,0.));
#18603= IFCLINESEGMENT2D(#18605,0.,4103.112);
#18604= IFCCIRCULARARCSEGMENT2D(#18606,0.,3525.132,9228.,.T.);
#18605= IFCCARTESIANPOINT((0.,0.));
#18606= IFCCARTESIANPOINT((4103.112,0.));

#15089= IFCRELDEFINESBYTYPE('3$8Jd5KxTADgvDpqtwcx8l',$,$,$,(#16005),#550);
```

```
#550= IFCSLABTYPE('149Rkrnxn2b8TAhFnxYOG2',$,'Deck Span 5','Fifth span of
deck',$,$,(#15087),$,$,.NOTDEFINED.);
#15087= IFCREPRESENTATIONMAP(#15090,#15091);
#15090= IFCAXIS2PLACEMENT3D(#15092,$,$);
#15091= IFCSHAPEREPRESENTATION(#15093,'Body','CSG',(#15094));
#15092= IFCCARTESIANPOINT((0.,0.,0.));
#15093= IFCGEOMETRICREPRESENTATIONCONTEXT('3D','Model',3,1.0E-05,#15095,$);
#15094= IFCEXTRUDEDAREASOLID(#15096,#15097,#15098,1133.875);
#15095= IFCAXIS2PLACEMENT3D(#15099,$,$);
#15096= IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,$,#15100);
#15097= IFCAXIS2PLACEMENT3D(#15101,#15102,#15103);
#15098= IFCDIRECTION((0.,0.,1.));
#15099= IFCCARTESIANPOINT((0.,0.,0.));
#15100=
IFCPOLYLINE((#15104,#15105,#15106,#15107,#15108,#15109,#15110,#15111,#15112,#15113,#15114,#15115,
#15116,#15117,#15118,#15119,#15120,#15121,#15122,#15123,#15124,#15125,#15126,#15127,#15128,#15129
,#15130,#15131,#15132,#15133));
#15101= IFCCARTESIANPOINT((33.,0.,0.));
#15102= IFCDIRECTION((1.,0.,0.));
#15103= IFCDIRECTION((0.,1.,0.));
#15104= IFCCARTESIANPOINT((0.,0.));
#15105= IFCCARTESIANPOINT((-78.5,0.));
#15106= IFCCARTESIANPOINT((-263.5,0.));
#15107= IFCCARTESIANPOINT((-263.5,-9.));
#15108= IFCCARTESIANPOINT((-205.,-9.));
#15109= IFCCARTESIANPOINT((-133.25,-14.5));
#15110= IFCCARTESIANPOINT((-128.25,-14.5));
#15111= IFCCARTESIANPOINT((-128.25,-16.));
#15112= IFCCARTESIANPOINT((-123.75,-16.));
#15113= IFCCARTESIANPOINT((-118.75,-14.5));
#15114= IFCCARTESIANPOINT((-74.125,-9.));
#15115= IFCCARTESIANPOINT((-9.,-9.));
#15116= IFCCARTESIANPOINT((-7.,-11.));
#15117= IFCCARTESIANPOINT((-2.25,-11.));
#15118= IFCCARTESIANPOINT((-2.25,-12.5));
#15119= IFCCARTESIANPOINT((2.25,-12.5));
#15120= IFCCARTESIANPOINT((2.25,-11.));
#15121= IFCCARTESIANPOINT((7.,-11.));
#15122= IFCCARTESIANPOINT((9.,-9.));
#15123= IFCCARTESIANPOINT((74.125,-9.));
#15124= IFCCARTESIANPOINT((118.75,-14.5));
#15125= IFCCARTESIANPOINT((123.75,-14.5));
#15126= IFCCARTESIANPOINT((123.75,-16.));
#15127= IFCCARTESIANPOINT((128.25,-16.));
#15128= IFCCARTESIANPOINT((128.25,-14.5));
#15129= IFCCARTESIANPOINT((133.25,-14.5));
#15130= IFCCARTESIANPOINT((205.,-9.));
#15131= IFCCARTESIANPOINT((263.5,-9.));
#15132= IFCCARTESIANPOINT((263.5,0.));
#15133= IFCCARTESIANPOINT((0.,0.));
```