University of Huddersfield

University of Huddersfield Repository

Sacks, Rafael, Ma, Ling, Yosef, Raz, Borrmann, Andre, Daum, Simon and Kattell, Uri

Semantic Enrichment for Building Information Modeling: Procedure for Compiling Inference Rules and Operators for Complex Geometry

Original Citation

Sacks, Rafael, Ma, Ling, Yosef, Raz, Borrmann, Andre, Daum, Simon and Kattell, Uri (2017) Semantic Enrichment for Building Information Modeling: Procedure for Compiling Inference Rules and Operators for Complex Geometry. Journal of Computing in Civil Engineering, 31 (6). ISSN 0887-3801

This version is available at http://eprints.hud.ac.uk/id/eprint/32807/

The University Repository is a digital collection of the research output of the University, available on Open Access. Copyright and Moral Rights for the items on this site are retained by the individual author and/or other copyright owners. Users may access full items free of charge; copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational or not-for-profit purposes without prior permission or charge, provided:

- The authors, title and full bibliographic details is credited in any copy;
- A hyperlink and/or URL is included for the original metadata page; and
- The content is not changed in any way.

For more information, including our policy and submission procedure, please contact the Repository Team at: E.mailbox@hud.ac.uk.

http://eprints.hud.ac.uk/

Semantic Enrichment for Building Information Modeling: Procedure for Compiling Inference Rules and Operators for Complex Geometry

Rafael Sacks[1]; Ling Ma[2]; Raz Yosef[3]; Andre Borrmann[4]; Simon Daum[5]; and Uri Kattel[6]

Abstract: Semantic enrichment of building models adds meaningful domain-specific or application-specific information to a digital building model. It is applicable to solving interoperability problems and to compilation of models from point cloud data. The *SeeBIM* (Semantic Enrichment Engine for BIM) prototype software encapsulates domain expert knowledge in computer readable rules for inference of object types, identity and aggregation of systems. However, it is limited to axis-aligned bounding box geometry and the adequacy of its rule-sets cannot be guaranteed. This paper solves these drawbacks by (1) devising a new procedure for compiling inference rule sets that are known a priori to be adequate for complete and thorough classification of model objects, and (2) enhancing the operators to compute complex geometry and enable precise topological rule processing. The procedure for compiling adequate rule sets is illustrated using a synthetic concrete highway bridge model. A real-world highway bridge model, with 333 components of 13 different types and compiled from a laser scanned point cloud, is used to validate the approach and test the enhanced *SeeBIM* system. All of the elements are classified correctly, demonstrating the efficacy of the approach to semantic enrichment. **DOI: 10.1061/(ASCE)CP.1943-5487.0000705.** *This work is made available under the terms of the Creative Commons Attribution 4.0 International license, http://creativecommons.org/licenses/by/4.0/.*

Author keywords: Building information modeling; Inference rules; Semantic enrichment; Solid geometry; Topology.

Downloaded from ascelibrary.org by 161.112.232.103 on 08/30/17. Copyright ASCE. For personal use only; all rights reserved.

Introduction

Semantic enrichment of building models refers to the automatic or semiautomatic addition of meaningful information to a digital model of a building or other structure by software that can deduce new information by processing rules (Belsky et al. 2016). The inputs are an existing building model, information about the building from other sources (such as a database), and a set of rules that encapsulate expert knowledge of the domain. The rules use the existing information and evaluate the topological, spatial, geometric, and other relationships between the model's objects. The output is a digital building model that incorporates the new information—new objects, property values, and/or relationships.

Development of semantic enrichment for models is motivated by the information interoperability problem (Eastman et al. 2011), which hampers the use of building information modeling (BIM), and by the difficulties faced by vendors of commercial BIM software in implementing the standard solution—exchanges based on the industry foundation classes (IFC) (BuildingSmart 2013). Semantic enrichment draws on the foundations laid by research of semantic query languages for BIM (Mazairac and Beetz 2013), semantic rule-checking systems for BIM (Eastman et al. 2009; Pauwels et al. 2011), and BIM model query using spatial and topological relationships (Borrmann and Rank 2009; Daum and Borrmann 2014).

Although semantic enrichment generally is considered to be applied to add missing information to building model instance files, it also has been applied to extend the schema of building information models. Zhang and El-Gohary (2016), for example, identified missing concepts in the IFC schema that were needed to express building code requirements.

Semantic enrichment also is useful for compilation of as-is or as-built BIM models from spatial point cloud data (PCD) collected on site through state-of-the-art surveying technologies, such as laser scanning and photo/videogrammetry (Brilakis et al. 2010; Zeibak-Shini et al. 2016). These large data sets must be converted into three-dimensional (3D) primitives and then identified as context-specific objects. Current practice requires intensive operations by experienced BIM modelers, and the problem has attracted many research efforts to automate the procedure (Bosche and Haas 2008; Kashani et al. 2014). However, the outputs of these systems are not semantically rich BIM models. Information regarding the objects' identification, relationships, and other alphanumerical data typically are missing.

[1]Associate Professor, Faculty of Civil and Environmental Engineering, Technion, Haifa 3200003, Israel. E-mail: cvsacks@technion.ac.il

[2]Lecturer, School of Art, Design and Architecture, Univ. of Huddersfield, Huddersfield HD1 3DH, U.K. (corresponding author). E-mail: l.ma@hud.ac.uk

[3]Lab Assistant, National Building Research Institute, Technion, Haifa 3200003, Israel. E-mail: razyos@gmail.com

[4]Professor, Chair of Computational Modeling and Simulation, Technische Universitat Munchen, 80333 München, Germany. E-mail: andre.borrmann@tum.de

[5]Research Assistant, Chair of Computational Modeling and Simulation, Technische Universitat Munchen, 80333 München, Germany. E-mail: simon.daum@tum.de

[6]Graduate Student, Faculty of Civil and Environmental Engineering, Technion, Haifa 3200003, Israel. E-mail: uri.kattel@gmail.com

Note. This manuscript was submitted on December 9, 2016; approved on April 20, 2017; published online on August 28, 2017. Discussion period open until January 28, 2018; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, © ASCE, ISSN 0887-3801.

Previous Work

SeeBIM 1.0 (Semantic Enrichment Engine for BIM) (Belsky et al. 2016) is an early software prototype whose primary aim was to establish the feasibility of the approach. As depicted in Fig. 1, the tool parses an IFC file to extract objects' shapes, relationships,

Downloaded from ascelibrary.org by 161.112.232.103 on 08/30/17. Copyright ASCE. For personal use only; all rights reserved.

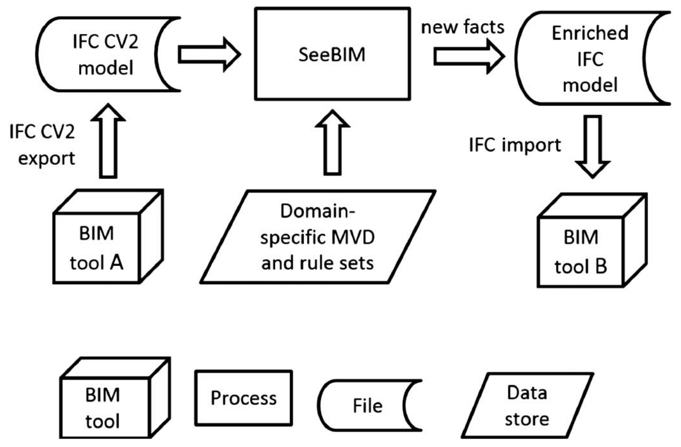


Fig. 1. *SeeBIM* process; IFC CV2 files conform to the Coordination View 2.0 model view definition

and other attributes. It then applies forward chaining to infer additional facts about the model, using sets of rules compiled in advance by experts in the domain of interest. It records the results in an enriched IFC file.

Experiments conducted using *SeeBIM* for two domains—precast concrete modeling (Belsky et al. 2016) and automated detailed design (Aram 2015)—showed how the approach could be used to add information to a model in an IFC file. The input in these efforts consisted of IFC files exported according to the Coordination View (CV) 2.0, which defines the exchange of 3D geometry data and is the only model view definition (MVD) commonly supported by BIM authoring tools (BuildingSmart 2010). The output in each case was an enriched IFC file that conforms to the MVD defined for precast concrete. More recently, Ramaji and Memari (2016) illustrated a similar idea: identification of structural features, such as beam-column joints, in a building model exported from an architectural BIM tool and enrichment of the model for import into a structural analysis tool. The common thread in these applications is that the exporting tool does not need to conform to the MVD of the importing tool, which means that export functions can remain generic. This is a major advantage for BIM software vendors because they find it difficult commercially to justify tailoring of export functions to narrow domains or specific importing software requirements.

Problem Statement

SeeBIM 1.0 has some important limitations that have become apparent in the first large-scale application of the tool, within the framework of an EU FP7 Infravation research project, *SeeBridge* (Technion 2015). The project aims to develop the ability to generate semantically rich bridge models from PCD. Computer vision technology generates 3D shapes; *SeeBIM* enriches the model by identifying bridge elements and their functional relationships. Several limitations exist in bridge model enrichment.

Firstly, the compilation of rule sets is at present essentially a social exercise that entails interviewing domain experts to elicit their knowledge and compiling it in the form of IF-THEN rules. The process depends on intuition and subjective judgment, and neither the completeness nor the precision of rule sets can be guaranteed. Because the success or failure of the approach is dependent on the robustness of the tools, a rigorous method is needed for compiling rule sets, one that allows testing for adequacy.

Secondly, the input is restricted to the IFC model file. In the worst case, this contains only the geometry, location, and orientation of the 3D shapes. However, alphanumeric information, such as the year of construction or a building's location, can be vital in supporting semantic enrichment, providing essential clues to support inference rule processing. Such information often is available in some other data source, such as a highway agency's bridge management system (BMS), and should be imported with the model.

Finally, the prototype uses axis-aligned bounding boxes (AABBs) to approximate a model's geometry. This results in errors in many cases where objects have a nonconvex shape or they are not axis-aligned. A shape's boundary and dimensions are inappropriately enlarged when it is non-axis aligned, with the result that many spatial topology operators return incorrect results. For example, a false positive result that two objects are in contact may be obtained if the first object is partially overlapped by the second object's AABB. *SeeBIM* depends heavily on the ability to process geometric and topological information, because geometry and placement are the only guaranteed information presented in all input models. This handicap therefore severely limits the tool's application for domains such as highway bridges which commonly include many nonconvex shapes (e.g., concrete girders).

The research presented in this paper focused on resolving these limitations, all of which must be resolved before semantic enrichment can become practical. Thus the goals were (1) to devise a new procedure for compiling inference rule sets that are known a priori to be adequate for complete and thorough classification of model objects, (2) to provide an interface to incorporate alphanumeric data from external databases with the information from the building model, and (3) to enhance the operators used in the rules to compute complex geometry and enable precise topological rule processing. This paper presents and discusses the procedure that has been devised for rule compilation, the facility for integrating external information and the operators needed to support it. Some of the solutions were implemented in software and they are used throughout the paper to illustrate the concepts.

Methodology

Research and development of the *SeeBIM* system follows a standard design science approach as defined specifically for the context of information science (Peffers et al. 2007). The methodology has the following six basic steps: identify problem and motivate, define objectives of a solution, design and develop a prototype software, demonstrate, evaluate, and communicate. This paper focuses on the iteration of the design and development, demonstration, and evaluation steps. The designed artifact is the *SeeBIM* 1.0 prototype which was outlined and reported in detail by Belsky et al. (2016). The current paper enhances the prototype based on the requirements for a specific application domain, that of inspection of reinforced concrete highway bridges.

The need to use data from an alphanumeric database (the BMS) as well as the 3D geometry model (compiled based on the PCD) was identified through compilation of a formal information delivery manual (IDM) for the domain of interest in the *SeeBridge* project (Sacks et al. 2016). The need to use explicit boundary representation (BREP) geometry for correctly processing topological queries also arises from the IDM in that it identifies bridge elements that have concave geometry features and are not aligned with the major bridge axes. The third requirement—the need for a rigorous method to compile inference rules—is the result of consideration of the complexity that results in a real-world case, with large numbers of bridge component types, which renders informal rule compilation error-prone and inadequate.

In full-scale implementation for the use-case of compilation of BIM models from PCD (Scan-to-BIM), the semantic enrichment step begins once a 3D solid geometry model has been prepared. This paper compiled the 3D geometry models using BIM authoring tools. The models were exported to IFC without any of the semantic information, so that they could serve as input for the semantic enrichment process. At the same time, the BIM models provided the ground truth for validation of results.

Rule-Based Inference

The success of model enrichment depends on the completeness and effectiveness of the inference rules used. Rule sets for expert system applications are commonly derived from knowledge-acquisition interviews with domain experts (Hayes-Roth 1985). The procedural knowledge acquired is expressed in the form of IF-THEN rule clauses that form logical chains of inference. The complexity of the rules increases with the number of object types and features, and developers have limited ability to evaluate the process logic inherent in systems with large numbers of inference rules. In the case of rule sets for semantic enrichment of BIM models, the approach does not guarantee the completeness or adequacy of a rule set nor the reliability of the results.

The approach defined herein is a procedure for deriving rule sets for identification of BIM object types (classification). Classification rules use two types of IF clauses: clauses that test for features of a single object, and clauses that test for topological relationships between pairs of objects. Rules used to identify object types therefore often depend on the prior identification of other relevant, related objects. If the rule set is inadequate, some objects cannot be identified and enrichment will be partial, and in some cases interdependency within the rules can result in infinite loops. A rigorous and robust approach to compiling rules sets is preferable. Ideally, developers should be able to guarantee that if enough evidence is available in the data, the set of rules will be adequate to identify all objects in the domain and the rule set will not be redundant. This is the goal of the procedure developed and described herein.

This approach compiles rules for identifying BIM object types into seven steps, as shown in Fig. 2:

1. A set of pairwise topological relationships that are most apparently relevant for object identification is defined in consultation with domain experts.
2. The experts are asked to express their knowledge in the form of matrixes, one for each of the relationships. Each matrix represents a pairwise relationship that can be applied to all the object pairs. The values in the cells are the logical results of the relationship for each pair.
3. The values for each cell in the resulting set of matrixes are strung together to generate a string in each corresponding cell of a composite pairwise spatial/topological relationship matrix. This is an $N \times N$ matrix (where N is the number of possible object types).
4. Each string is then compared with all the other strings. Any string that is unique implies that if the set of relationship result values that string represents is found to hold for any pair of object instances in a BIM model that is being enriched, then the identity of both objects can be determined.
5. If any object type does not have at least one unique string, then additional pairwise relationships must be added, repeating the process from Step 2. This is done repeatedly, if necessary, until all object types have at least one unique string.
6. A subset of unique rule strings is selected from the whole set of unique strings, such that each object type is represented in at least one rule.
7. A *SeeBIM* rule is compiled directly from each unique string in the subset.

To illustrate this procedure, this paper presents an application to a small-scale synthetic bridge model (Fig. 3). This model consists of eight typical types of bridge elements (A–H), as shown in Table 1, so that any kind of pairwise relationship can be represented as an 8×8 matrix. For example, Table 1 shows a matrix for the contact relationships involved in this model. The relations are expressed as "IF Object 1 is of type A and Object 2 is of type B," then there are three possible values:

- y ≔ Object 1 is always in contact with Object 2;
- n ≔ Object 1 is never in contact with Object 2; and
- x ≔ Object 1 may or may not be in contact with Object 2.

As shown in Table 1, a column will always be in direct contact with a capping beam (y); a primary girder will never be in direct contact with a column (n); a primary girder may or may not be in contact with another primary girder (x).

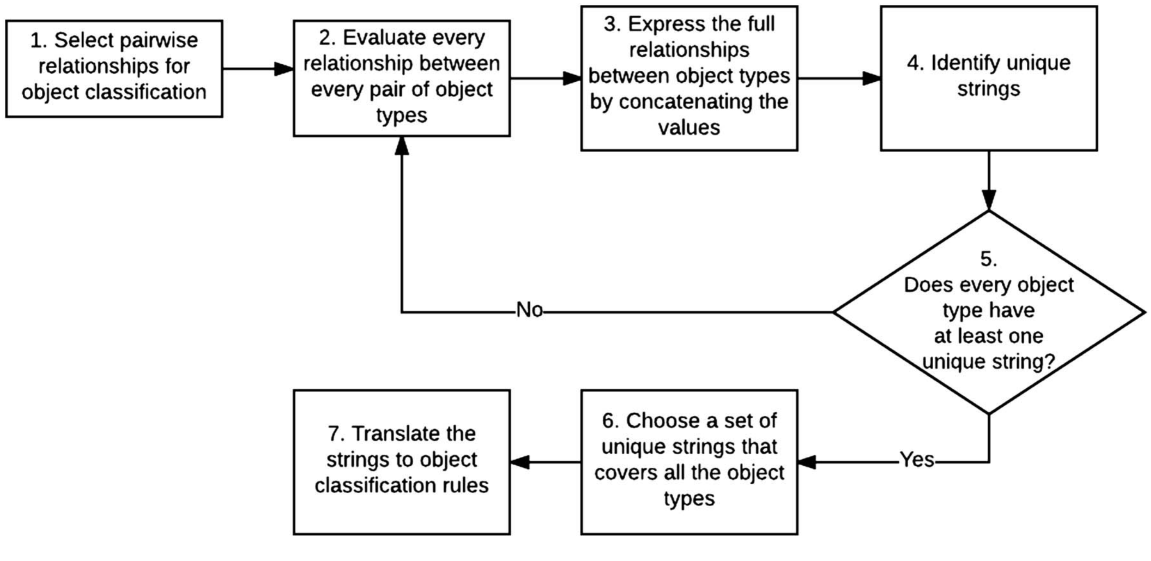

1. Select pairwise relationships for object classification
2. Evaluate every relationship between every pair of object types
3. Express the full relationships between object types by concatenating the values
4. Identify unique strings
5. Does every object type have at least one unique string?
No
Yes
6. Choose a set of unique strings that covers all the object types
7. Translate the strings to object classification rules

Fig. 2. Procedure of object identification

Downloaded from ascelibrary.org by 161.112.232.103 on 08/30/17. Copyright ASCE. For personal use only; all rights reserved.

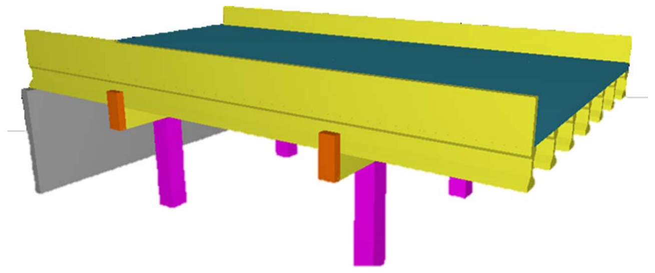


Fig. 3. Synthetic bridge model

Table 1. Matrix for Conditions of the Contact Relationship between Bridge Objects

		Type of Object 1							
Type of Object 2		A	B	C	D	E	F	G	H
A	Primary girder	x	y	y	n	n	n	x	x
B	Capping beam	—	n	n	x	x	y	n	n
C	Deck slab	—	—	x	n	n	n	n	x
D	Shear key	—	—	—	n	n	n	n	n
E	Abutment	—	—	—	—	n	n	y	n
F	Column	—	—	—	—	—	n	n	n
G	Bearing	—	—	—	—	—	—	n	n
H	Safety barrier	—	—	—	—	—	—	—	n

Table 2. Conditional Pairwise Relationships between Concrete Girder Bridges Object Types

Number	Conditional relation
1	Which axis of the bridge is Object 2 parallel to?
2	Which axis of the bridge is Object 1 parallel to?
3	Which object is larger in volume?
4	Which object's extrusion axis is longer?
5	Which object is closer to the lateral axis of the bridge?
6	Which object is closer to the longitudinal axis of the bridge?
7	Which object's bounding box is higher?
8	Which object's centroid is higher?
9	Do both objects have the same extrusion direction?
10	Are the objects in contact?

Table 3. Additional Possible Result Values Used in the Relationship Matrices

Value	Meaning
e	Equal
1	Element type 1
2	Element type 2
i	Bridge longitudinal axis
j	Bridge lateral axis
k	Bridge vertical axis

Table 2 lists 10 different spatial features and pairwise relationships used to test the approach for the synthetic bridge model, and additional possible result values used across all the features and relationships are shown in Table 3. The relationship results are compiled in an 8×8 matrix with 10-digit strings in each cell, i.e., one digit for each relationship, as shown in Table 4. Note that the first two relationships (the first two digits in each cell of Table 4) reflect the objects' relative orientation to the bridge, so that the first digit in each row is the same and the second digit in every column is the same.

All the strings then are compared with one another to identify unique string values. When comparing strings, any relationships that have an x value for either or both object types are ignored, because their values are obviously different. If a relationship's string is unique, it will be evaluated as true only in those instances for which the pair of objects being tested are of the types to which the cell belongs. This means that if the relationship evaluates as true, the pair of objects being compared can be classified with full confidence. It is this property which allows users to compile a set of rules that can be considered a priori to be adequate.

The theoretical minimum number of unique pairwise relationships needed for adequacy of a rule set—i.e., the ability to classify all the objects in a model correctly and confidently—is half the number of object types. More may be needed if some of the object types occur in more than one unique relationship. Furthermore, in cases where the model itself has inaccuracies or is incomplete, any rule derived from a unique relationship string may not evaluate as true for all the object pairs, and so the objects concerned may not be classified. In such cases, having additional unique rules beyond the theoretical minimum is useful. Some redundancy can improve the rate of success of classification.

For the case of the synthetic bridge, the four cells in Table 4 highlighted with bold text are unique and form an adequate set of unique relationship strings for classifying all the objects in a model of a bridge of this type, because they cover all the bridge element types (i.e., A-B, D-E, F-G, and C-H) in pairs.

Finally, Step 7 translates these four relationship strings into inference rules. For example, rules for identifying columns and bearings are translated from string kk222x11yn in cell F-G as follows:

- IF Object 1 is parallel to bridge vertical axis (z)
- & Object 2 is parallel to bridge vertical axis (z)
- & Object 2 has larger volume than Object 1
- & Object 2 has longer extrusion axis than Object 1
- & Object 2 is closer to lateral axis of the bridge (y)
- & Object 1 bounding box is absolutely higher than Object 2
- & Object 1 centroid is absolutely higher than Object 2
- & Object 1 extrusion axis is parallel to Object 2 extrusion axis
- & Object 1 is not in contact with Object 2
- THEN Object 1 is a bearing and Object 2 is a column

This process results in rule sets that contain sufficient tests to identify all possible object types in the domain. This ability to ensure adequacy is an important enhancement of the *SeeBIM* approach to semantic enrichment.

Merging BIM Model Data with Information from External Sources

The minimal starting point for semantic enrichment of building models is an IFC file containing building entities with solid geometry. However, most of the organizations that manage constructed facilities use databases of one form or another to describe their assets. These systems generally contain useful data that can and should be used to support semantic enrichment of BIM models. For example, state DOTs use BMS to manage their bridge networks. The BMS data shown in Table 5 identify a bridge with characteristic data in tabulated formats that can be helpful for semantic enrichment of a model of the bridge.

These data can provide prior information that is valuable for inference of bridge object types and relationships. For example, the prestressed concrete superstructure type suggests the presence and possible types of girders, and the bridge span length provides a

Downloaded from ascelibrary.org by 161.112.232.103 on 08/30/17. Copyright ASCE. For personal use only; all rights reserved.

Table 4. Conditional Relationship Matrix for the Eight Types of Bridge Elements

Type		A	B	C	D	E	F	G	H
Primary girders	A	iixxxxneyx	**ij11xx22ny**	ii1xxx11yy	ik22x2n2nn	ij112x22nn	ikx2xx22nn	ik222x22nx	ii11x211yx
Transverse beam	B	—	jjeeeeneyn	jix2xx11nn	jk22x211nx	jj1e2xn2yn	jk22x222nx	jk222xn1nn	ji211211nn
Deck slab	C	—	—	Iixxxxnexx	ik22x222nn	ij112x22nn	ik22xx22nn	ik222x22nn	**iix1x2n1yx**
Shear keys	D	—	—	—	kkeeeeneyn	**kj112122nn**	kk11x122yn	kk222122yn	ki111111nn
Abutments	E	—	—	—	—	jjxxxenxyn	jk2212n2nn	jk22ex11ny	ji211211nn
Column	F	—	—	—	—	—	kkeeeeneyn	**kk222x11yn**	ki111211nn
Bearings	G	—	—	—	—	—	—	kkeexxneyn	ki111211nn
Safety barriers	H	—	—	—	—	—	—	—	iieeeeneyn

Note: Bold font indicates an adequate set of unique relationship strings for classifying all the objects in a model of a bridge of this type.

Table 5. Examples of Bridge Data in a BMS Database

Attribute	Data example	IFC property type
Bridge span	17.6 m	IFC positive length measure
Superstructure type	Prestressed concrete	IFC property enumerated value
Year of construction	1993	IFC date
Location	Afek Road bridge above Route 79 in Kiryat Bialik	IFC text
Ownership	National roads company	IFC property enumerated value

Downloaded from ascelibrary.org by 161.112.232.103 on 08/30/17. Copyright ASCE. For personal use only; all rights reserved.

candidate measure for identifying the bridge girders. The year of construction and the location further constrain the type of bridge elements (e.g., AASHTO girders were not available in this location until the 1960s).

SeeBIM imports standard IFC files which must have building entities with BREP or extruded solid geometry. At a minimum, the entities will be IfcBuildingElementProxy entities. *SeeBIM* uses a late-binding method (RDF 2015; STEP Tools 2016) to parse IFC files on the fly through the ISO standard data access interface (SDAI) (ISO 1998), which means that it can import models from any IFC version provided that the EXPRESS schema definition files are available.

SeeBIM incorporates the data from external databases by appending them to the appropriate IFC entities. First, the properties, their data types, and their values are imported into the run-time internal database of the application. This makes them available for testing within the IF clauses of the rules. During rule-processing, rules may add additional alphanumeric data to any of the model's entities. Finally, once rule-processing is complete, the data are exported in the form of IFC property sets. The IFC property value entities are collected in IFC property set entities, which are associated with building entities using IfcRel defines by properties entities.

According to the IFC schema, entities and property sets have a many-to-many relationship. Each entity can have more than one property set and each property set can be assigned to more than one entity. For example, many prefabricated components of a concrete bridge will share the same property sets and property values. However, many BIM authoring tools duplicate the same property set for each entity, creating unnecessarily large files. *SeeBIM* 2.0 identifies, resolves, and removes these duplications, so that the IFC file size is reduced.

Enhanced Geometric and Topological Operators

An object's classification is related to its geometry, functions and other properties. In the worst case, only the geometry is guaranteed to be provided in a BIM model. Hence the deduction of other information depends on unique model features, and the success of semantic enrichment depends on the ability to identify these features, including (1) objects' shape features and (2) pairwise topological and spatial relationships.

Enhancement of the semantic enrichment engine required removing the restrictions imposed by the prototype's axis-aligned bounding box representation of the geometry by using a minimal volume bounding box (MVBB) representation, in the first instance, and implementation of more sophisticated spatial and topological operators to account for explicit and potentially concave geometry representation, in the second instance.

Shape Representation

Objects' shape features include the shape extents and orientation, which can be derived from the MVBB of the object. Toussaint (1983) first proposed the rotating caliper algorithm that can be used to construct the smallest-area enclosing rectangle in two-dimensional (2D). O'Rourke (1985) extended the algorithm to 3D such that the MVBB has at least two adjacent faces flush with edges of the 3D shape, and presented an algorithm for generating the MVBB in a brute-force way. Based on O'Rourke's findings, Jylanki (2015) developed a more efficient algorithm to generate the MVBB, and this algorithm is used in *SeeBIM* 2.0.

The generated MVBB can be represented by three components: orientation (**axis[0]**, **axis[1]**, and **axis[2]**, each of which is a 3D vector), coordinates of the centroid point (pos[0], pos[1], and pos[2]), and the extent of the box in the local axes ($r.x$, $r.y$ and $r.z$), as shown in Fig. 4.

The local coordinates of the eight vertices of the MVBB, P_1 to P_8, can be derived as

$$\mathbf{P} = \begin{bmatrix} r.x & r.x & r.x & r.x & -r.x & -r.x & -r.x & -r.x \\ r.y & r.y & -r.y & -r.y & r.y & r.y & -r.y & -r.y \\ r.z & -r.z & r.z & -r.z & r.z & -r.z & r.z & -r.z \end{bmatrix} \tag{1}$$

The transformation from the local coordinate system to the global coordinate system can be represented as a 4×4 matrix in the homogeneous space

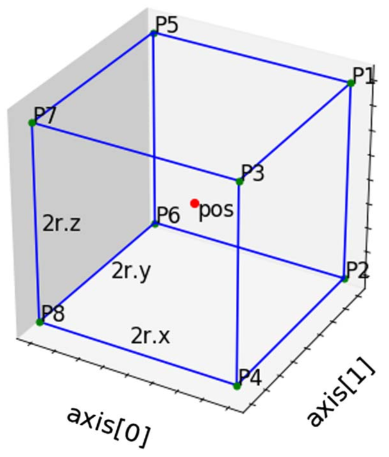


Fig. 4. Properties of a MVBB

$$
\mathbf{T} = \begin{bmatrix} \text{axis}[0][0] & \text{axis}[1][0] & \text{axis}[2][0] & \text{pos}[0] \\ \text{axis}[0][1] & \text{axis}[1][1] & \text{axis}[2][1] & \text{pos}[1] \\ \text{axis}[0][2] & \text{axis}[1][2] & \text{axis}[2][2] & \text{pos}[2] \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)
$$

To simplify the mathematical operation for computing the global coordinates of all the vertices, the vectors in $\mathbf{P}$ were augmented to the homogeneous space by increasing their dimensionality. For example: $\mathbf{P}_1' = [\, r.x \quad r.y \quad r.z \quad 1 \,]$, so that $\mathbf{P}'$ is a 4×8 matrix. The global coordinates of the vertices in the homogeneous space then can be derived as

$$
\mathbf{V}' = \mathbf{T} \times \mathbf{P}' = \begin{bmatrix} x_1 & x_2 & \dots & x_8 \\ y_1 & y_2 & \dots & y_8 \\ z_1 & z_2 & \dots & z_8 \\ 1 & 1 & \dots & 1 \end{bmatrix} \quad (3)
$$

The actual global coordinates of each vertex V_1 to V_8 can be derived by reducing the dimensionality of each vector, for example, $\mathbf{V}_1 = [\, x_1 \quad y_1 \quad z_1 \,]$. In addition, the six faces F_1 to F_6 can be derived as follows:

- F_1 $(\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_4 \quad \mathbf{V}_3)$ and F_2 $(\mathbf{V}_5 \quad \mathbf{V}_6 \quad \mathbf{V}_8 \quad \mathbf{V}_7)$ are faces whose normal direction is axis[0]
- F_3 $(\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_6 \quad \mathbf{V}_5)$ and F_4 $(\mathbf{V}_3 \quad \mathbf{V}_4 \quad \mathbf{V}_8 \quad \mathbf{V}_7)$ are faces whose normal direction is axis[1]
- F_5 $(\mathbf{V}_1 \quad \mathbf{V}_5 \quad \mathbf{V}_7 \quad \mathbf{V}_3)$ and F_6 $(\mathbf{V}_2 \quad \mathbf{V}_6 \quad \mathbf{V}_8 \quad \mathbf{V}_4)$ are faces whose normal direction is axis[2]

Spatial and Topological Relationships and Operators

Extensive data sets can be precisely analyzed, explored, and processed by a formal query language. To handle spatial data, languages such as Spatial SQL and GeoSPARQL are used in geographical information systems (GIS) (Egenhofer 1994; Perry and Herring 2012). There also have been attempts to facilitate query languages in the architecture/engineering/construction (A/E/C) domain (Borrmann 2010; Mazairac and Beetz 2013). However, none of these methods could process the 3D representations used in civil engineering in an adequate way, especially with respect to qualitative spatial predicates. This was a major deficiency, because spatial relations between building elements play a significant role in most of the design and engineering tasks of the A/E/C domain. To close this gap, a BIM query language called Query Language for 4D Building Information Models (QL4BIM) was developed (Daum and Borrmann 2013).

Among other features, QL4BIM makes it possible to select specific building elements by applying qualitative spatial predicates as part of filter expressions. These relationships provide a high level of abstraction between the technological view on building geometry using numerical coordinates, and the way humans reason about spatial entities and the relations between them. Typical examples of queries concerned with spatial semantics are:

- Which columns *touch* Slab 34?
- Get all walls which *are contained in* the first story.
- Does the space representation of Room 107 *intersect with* any heating equipment?
- Get all objects *within 1.5 m* from Wall 232.

The ability to identify and compute spatial relationships between building objects is also essential for a semantical enrichment. For that reason, the enrichment process of *SeeBIM* facilitates QL4BIM operators.

As a query language, QL4BIM was designed to be employed by domain experts and offers a carefully selected vocabulary to formulate queries at a high level of abstraction (Daum and Borrmann 2015). In *SeeBIM* 2.0, there is no need to incorporate this kind of end-user interface. Instead, the QL4BIM operators are introduced as library functions which can be invoked directly from the *SeeBIM* rule composition interface. The functionality offered includes metric, directional, and topological operators (Fig. 5).

In contrast to several applications in the GIS–A/E/C domain, the QL4BIM operators are not restricted to 2D geometry or bounding box abstractions (ISO/OGC; Nepal et al. 2012). Instead, 3D geometry is processed as triangulated boundary representation, and operators evaluate correctly with convex and nonconvex shapes.

The topological and directional operators are based on the mathematical definitions stated by Egenhofer (1989) and Borrmann (2006). In the first case, the approach is called the 9-Intersection Model (9IM) and facilitates theories of algebraic topology and set

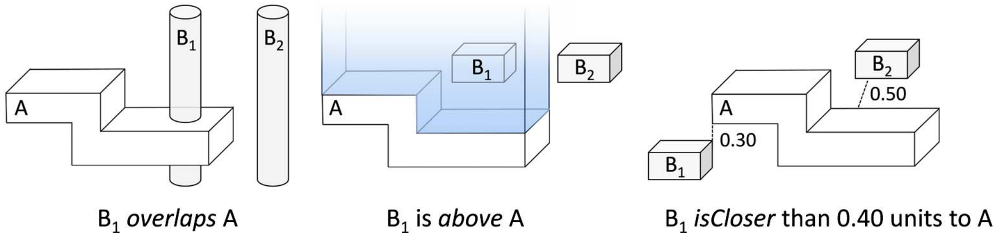
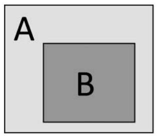


Fig. 5. Examples of the three classes of spatial operators provided by QL4BIM; in all cases Object B1 passes the predicate whereas Object B2 is rejected

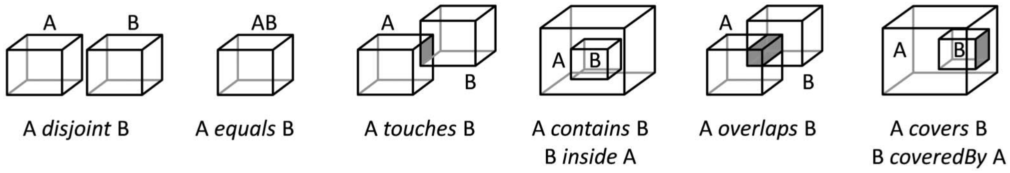
Downloaded from ascelibrary.org by 161.112.232.103 on 08/30/17. Copyright ASCE. For personal use only; all rights reserved.



A contains B
B is inside A

$$I = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B & A^\circ \cap B^- \\ \partial A \cap B^\circ & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^\circ & A^- \cap \partial B & A^- \cap B^- \end{pmatrix} = \begin{pmatrix} \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \end{pmatrix}$$

Fig. 6. Deducing the topological relationship between two regions by the 9IM (*containment/inside* case)



$$\begin{pmatrix} \emptyset & \emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix} \begin{pmatrix} \neg\emptyset & \emptyset & \emptyset \\ \emptyset & \neg\emptyset & \emptyset \\ \emptyset & \emptyset & \neg\emptyset \end{pmatrix} \begin{pmatrix} \emptyset & \emptyset & \neg\emptyset \\ \emptyset & \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix} \begin{pmatrix} \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \end{pmatrix} \begin{pmatrix} \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix} \begin{pmatrix} \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \emptyset & \neg\emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \end{pmatrix}$$

Fig. 7. Topological templates for solids introduced by the 9IM; six are shown; the two not shown are the inverse templates *B contains A/A inside B* and *B covers A/A coveredby B*

topology (Gaal 1964). The 9IM applies the notion of the neighborhood of a point to describe topological concepts such as the interior A°, the boundary δA, and the exterior A^- of a point set A. Topological predicates are defined by the set-oriented intersections of the interior, the boundary, and the exterior of two operands. Here, an intersection can yield an empty (Ø) or a nonempty set (¬Ø). Fig. 6 shows the 9IM matrix for a 2D scenario with two regions A and B.

A 9IM matrix represents the topological invariants of the topological relations, reflecting that the set oriented intersection results remain constant under transformations. Theoretically, there are $2^9 = 512$ possible configurations, but only eight are encountered when closed regions are examined in 2D. The same number of configurations arises for closed solids in 3D. Fig. 7 shows six of the eight possible topological constellations of two solids and the corresponding matrixes.

The definition of the directional operators uses a projection-based model. There is a strict and a relaxed version of each directional predicate. In both cases, reference object A and target object B are spatial objects and $a \in A$, $b \in B$. Fig. 8 shows the formal definitions of the *above* operators, where the indexes of a and b denote the respective dimensions. Fig. 8 includes an example with five object pairs A–B_1 to A–B_5. Table 6 shows the results of

aboverelaxed(A,B) ⇔
$\exists a,b : a_x = b_x \wedge a_z < b_z$

abovestrict(A,B) ⇔
$\forall b : (\exists b : a_x = b_x \wedge a_z < b_z) \wedge$
$(\nexists b : a_x = b_x \wedge a_z \geq b_z)$

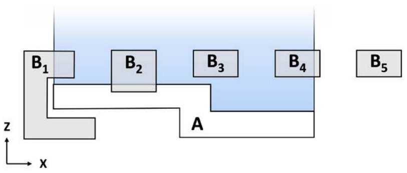


Fig. 8. Mathematical definition for of the projection-based directional predicates and an example for the *above* case (2D case for clearness)

Table 6. Results of the Two *above* Operators for the Spatial Constellation in Fig. 8

Operator	B_1	B_2	B_3	B_4	B_5
above relaxed (A, B_i)	true	true	true	true	false
above strict (A, B_i)	false	false	true	false	false

the relaxed and the strict version of the *above* operator used on these pairs.

From the mathematical definitions, algorithms are deduced that process triangulated meshes and determine the spatial predicate between two spatial objects. In the case of the topological operators, triangle intersection and inside/outside tests are applied. For example, the *touch* predicate is verified if at least two triangles meet, no triangles intersect, and B is located outside of A.

The directional functionality is realized by triangle extrusions, prism/triangle tests, and ray tests. To be *above* in the relaxed version, at least one triangle of B must intersect with a prism of A. These prisms are created by extruding the triangles of A in the correlated direction of the predicate. In the case of the *above* predicate, this is the positive z-direction.

To deal with the emerging computational complexity that arises if extensive data sets and detailed geometry representations are handed to the operators, the spatial indexing structure R*-Tree is incorporated (Beckmann et al. 1990).

Support for Tolerances in Spatial Operators

For imperfect data sets, the support of user-defined tolerances in the processing of directional and topological predicates is needed. This is especially the case if geometry is reconstructed from a laser-scanned point cloud. In addition to the existence of numerical discrepancies, parts of the objects' surfaces may be obscured in these data sets.

In addition to imperfect geometry reconstruction, the need for tolerances also derives from the practical design and construction considerations. Here, minimal gaps and intersections of a specific extent must be approved, whereas the applied tolerances depend on the modeling domain and the actual use case.

To support semantic tolerances and to yield robust results despite numerical imprecisions, a mesh handling was developed for QL4BIM within the *SeeBridge* project. The approach was based on the use of an inner and an outer mesh. These meshes were created via shifting original triangles by a user defined amount. Here, the inner boundary was given the index i, and the outer boundary was given the index o. The developed tolerance supporting topological operators were denoted as TST operators, and the tolerance supporting directional operators were denoted as TSD operators.

Downloaded from ascelibrary.org by 161.112.232.103 on 08/30/17. Copyright ASCE. For personal use only; all rights reserved.

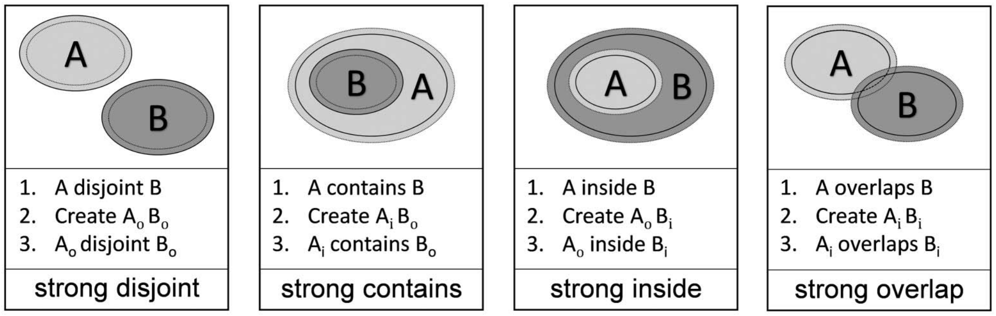


Fig. 9. Three-stage definitions of the strong TST operators in QL4BIM

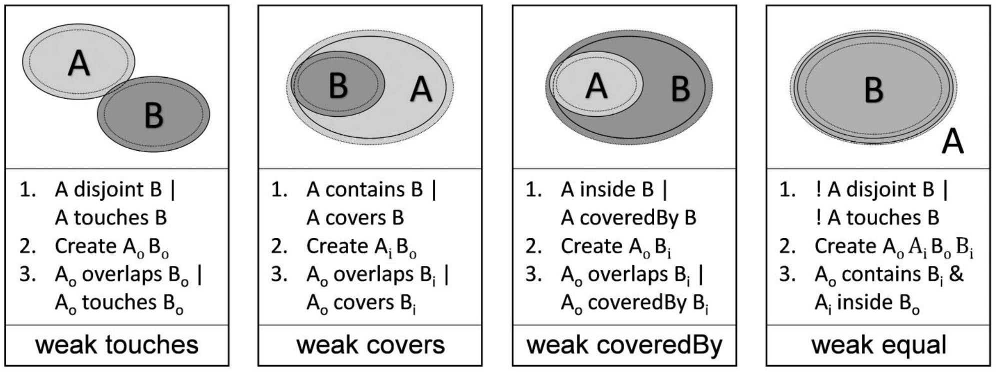

A
B
1. A disjoint B | A touches B
2. Create $A_o B_o$
3. A_o overlaps B_o | A_o touches B_o
weak touches
1. A contains B | A covers B
2. Create $A_i B_o$
3. A_o overlaps B_i | A_o covers B_i
weak covers
1. A inside B | A coveredBy B
2. Create $A_o B_i$
3. A_o overlaps B_i | A_o coveredBy B_i
weak coveredBy
1. ! A disjoint B | ! A touches B
2. Create $A_o A_i B_o B_i$
3. A_o contains B_i & A_i inside B_o
weak equal

Fig. 10. Three-stage definitions of the weak TST operators in QL4BIM

Downloaded from ascelibrary.org by 161.112.232.103 on 08/30/17. Copyright ASCE. For personal use only; all rights reserved.

The algorithms for the TST operators begin with the original geometry representations and check for several topological predicates. If none of the permissible predicates is valid, the investigation is aborted and the predicate is rejected. If the first condition is met, a modified boundary is created by triangle shifting. The decision to move inside and/or outside depends on the actual predicate and the operand. In the last step of the TST processing, one of several predicates should be confirmed with the changed geometry. Figs. 9 and 10 show (1) the predicates allowed at the beginning, (2) the geometry modification per operand, and (3) the predicates that should be finally checked.

As indicated by the figures, the operators can be divided into two groups, strong and weak. In case of the strong operators, the topological predicate returns true independent of the application tolerances. In the case of the weak operators, the predicate is confirmed only according to the tolerances applied. Thus the operators of the first group are denoted as strong, and the operators of the second group are denoted as weak variants of their originals. The strong group includes the predicates *disjoint*, *contains*, *inside*, and *overlaps*, and the weak group includes *touches*, *covers*, *coveredby*, and *equals*.

The definition of the TSD operators is equal for all directions and includes only a geometry modification and a subsequent directional analysis. In the modification step, A_i is produced and B is not altered. The original directional predicate then is executed. The TSD operators are weak variants of their originals.

The necessary offset meshes for this approach can be generated by several approaches (Egenhofer et al. 1989; Rossignac and Requicha 1986). In QL4BIM, the multiple normal vectors of a vertex method (MNVM) method is applied to balance between the geometric accuracy of the created boundaries and the computational costs (Kim et al. 2004).

Full-Scale Test of a Real-World Bridge

The enhanced semantic enrichment tool was tested using a model of a concrete girder highway bridge on Route 79 in Haifa, Israel. The bridge was scanned using a terrestrial laser scanner. To obtain a panoramic view of the entire bridge, several scans were taken from different positions and a complete point cloud was derived by registration of the collected PCD sets. A 3D model of the bridge geometry was compiled manually in a BIM authoring tool from the PCD. The model, shown in Fig. 11, contains 333 bridge elements of 10 different types (Table 7).

The bridge had 13 object types, whereas the synthetic bridge previously used to explain the process had only eight. Although the original eight object types were unchanged, the five new object types essentially made this a new case. Therefore, a new set of rules was needed, and the set required at least seven rules, whereas the synthetic bridge required only four. To obtain a sufficient set of unique rules in this case required 19 conditional relations (Table 8).

Any ambiguity in understanding and modeling the bridge objects will affect the classification result. Fig. 12 shows an example of this: in the test, the outer concrete columns were modeled such that the shear key was on top of the column, whereas the capping beam did not rest on the topmost face of the column. However, in the initial compilation of Conditional relation 5 for the case of the capping beam and the column, the logical assumption was "the capping beam (Object 1) is always in contact with the top face of the column (Object 2)." Similarly, the inverse Condition 4 was assumed to be always true (y value). This would be true for all cases only if the capping beam were modeled as an extrusion extending to both ends, such that the shear key would be above the capping beam and the capping beam would be in contact with the top face of all the columns (as it was with the two middle columns), but this

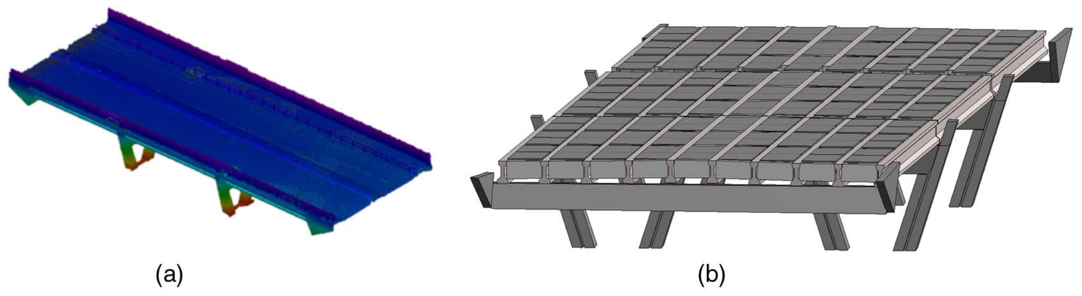
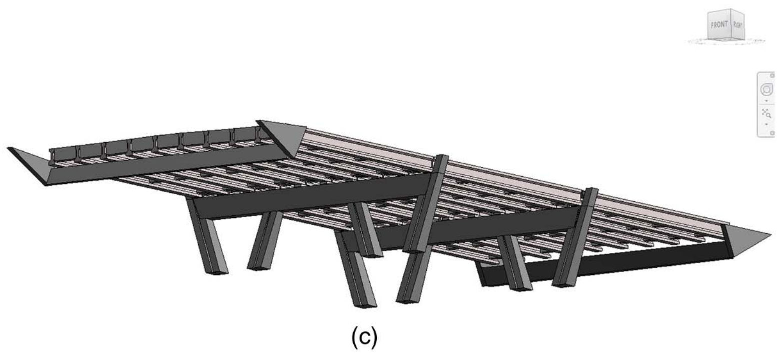



Fig. 11. Manually prepared bridge model geometry: (a) point cloud; (b) BIM model top view; (c) IFC model bottom view

Table 7. Bridge Elements in the Bridge Model

Concrete girder bridge object types		Number of visible objects
Deck/superstructure	Primary girder	30
	Capping beam	2
	Transverse beam	99
	Partial depth precast deck panel	162
Substructure	Bearing	None
	Plinth on capping beam	20
	Plinth on abutment	None
	Shear key on capping beam	4
	Shear key on abutment (wing wall)	4
	Abutment	2
	Column	8
Nonstructural	Lamp posts	2
	Safety barriers	None

Note: The bearings and abutment plinths were not visible in the PCD due to occlusion.

Table 8. Conditional Pairwise Relationships between Concrete Girder Bridge Object Types

Number	Conditional relation
1	Are the two objects in contact?
2	Is Object 1 in contact with the Object 2's side face?
3	Is Object 1 in contact with the Object 2's front/back face?
4	Is Object 1 in contact with the Object 2's bottom face?
5	Is Object 1 in contact with the Object 2's top face?
6	Are the two objects in parallel along their extrusion direction?
7	Are the two objects in parallel along their long edges?
8	Is Object 1's centroid higher than Object 2's?
9	Is Object 1's extrusion longer than Object 2's?
10	Is Object 1's volume greater than Object 2's?
11	Is Object 1 vertically extruded?
12	Is Object 1's extrusion direction parallel to the road axis?
13	Is Object 1's extrusion direction parallel to the skew angle of the bridge supports?
14	Is Object 1 horizontal?
15	Is Object 1 the bridge?
16	Is Object 2 the bridge?
17	Is Object 1 wider than Object 2?
18	Is Object 1 taller than Object 2?
19	Is Object 2 a capping beam?

bridge illustrated that the condition is not in fact always true. Therefore, to cope with the more general case, the conditional value for this pair of objects for Conditions 4 and 5 had to be relaxed and given an x value (i.e., not always).

The IDM prepared for the *SeeBridge* project (Sacks et al. 2016) lists thirteen relevant object types for a concrete girder bridge such as this one. Of those, none of the bearings or the plinths on the abutments were visible in the PCD for the test bridge because they were occluded by other objects. Nevertheless, they were included in the rule sets to ensure that the rules were valid for the general case of concrete girder bridges. The entire bridge also was considered as an object because its boundary and orientation were of great importance for inference of other objects. Therefore the examination used a 14×14 matrix with 19-digit strings in each cell.

The result values in all the matrixes were filled in consultation with three experts: a senior partner in a bridge structural design

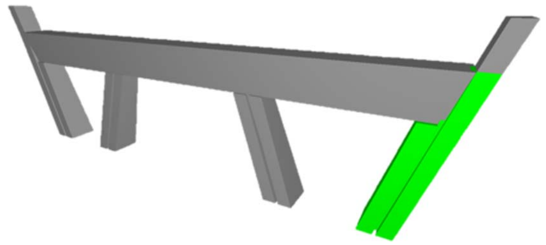
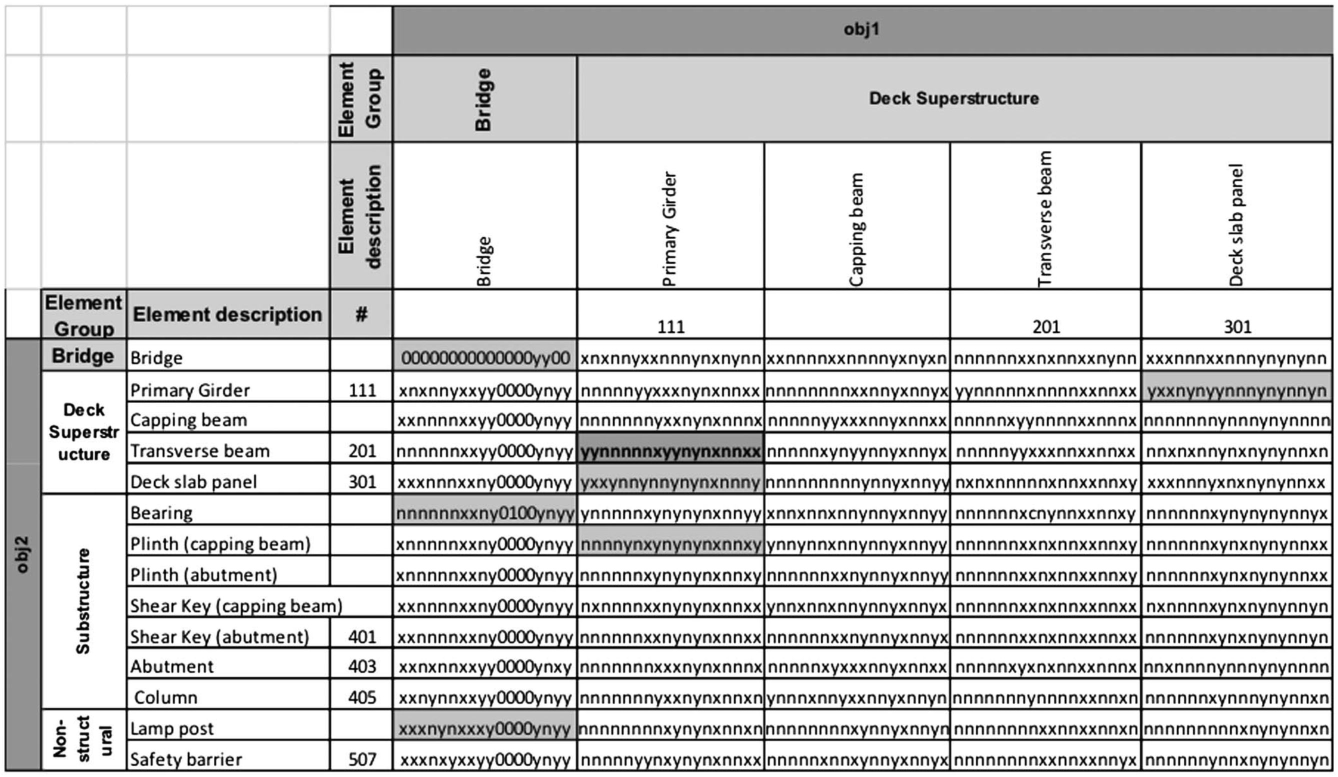


Fig. 12. Part of the substructure in the bridge model

Downloaded from ascelibrary.org by 161.112.232.103 on 08/30/17. Copyright ASCE. For personal use only; all rights reserved.

Downloaded from ascelibrary.org by 161.112.232.103 on 08/30/17. Copyright ASCE. For personal use only; all rights reserved.



				obj1				
			Element Group	Bridge	Deck Superstructure			
			Element description	Bridge	Primary Girder	Capping beam	Transverse beam	Deck slab panel
	Element Group	Element description	#		111		201	301
obj2	Bridge	Bridge		00000000000000yy00	xnxnnyxxnnnynxnynn	xxnnnnxxnnnnyxnyxn	nnnnnnxxnxnnxxnynn	xxxnnnxxnnnynynynn
	Deck Superstructure	Primary Girder	111	xnxnnyxxyy0000ynyy	nnnnnyyxxxnynxnnxx	nnnnnnnnxxnnyxnnyx	yynnnnnxnnnnxxnnxx	yxxnynyynnnynynnyn
		Capping beam		xxnnnnxxyy0000ynyy	nnnnnnnyxxnynxnnnx	nnnnnyyxxxnnyxnnxx	nnnnnxyynnnnxxnnnx	nnnnnnnynnnynynnnn
		Transverse beam	201	nnnnnnxxyy0000ynyy	**yynnnnnxyynynxnnxx**	nnnnnxynyynnyxnnyx	nnnnnyyxxxnnxxnnxx	nnxnxnnynxnynynnxn
		Deck slab panel	301	xxxnnnxxny0000ynyy	yxxynnynnynynxnnny	nnnnnnnnnynnyxnnyy	nxnxnnnnnxnnxxnnxy	xxxnnnyxnxnynynnxx
	Substructure	Bearing		nnnnnnxxny0100ynyy	ynnnnnxynynynxnnyy	xnnxnnxnnynnyxnnyy	nnnnnnxcnynnxxnnxy	nnnnnnxynynynynnyx
		Plinth (capping beam)		xnnnnnxxny0000ynyy	nnnnynxynynynxnnxy	ynnynnxnnynnyxnnyy	nnnnnnxxnxnnxxnnxy	nnnnnnxynxnynynnxx
		Plinth (abutment)		xnnnnnxxny0000ynyy	nnnnnnxynynynxnnxy	nnnnnnxxnynnyxnnyy	nnnnnnxxnxnnxxnnxy	nnnnnnxynxnynynnxx
		Shear Key (capping beam)		xxnnnnxxny0000ynyy	nxnnnnxxnynynxnnxx	ynnxnnxnnynnyxnnyx	nnnnnnxxnxnnxxnnxx	nxnnnnxynxnynynnyn
		Shear Key (abutment)	401	xxnnnnxxny0000ynyy	nnnnnnxxnynynxnnxx	nnnnnnxxnynnyxnnyx	nnnnnnxxnxnnxxnnxx	nnnnnnxynxnynynnyn
		Abutment	403	xxnxnnxxyy0000ynxy	nnnnnnnxxxnynxnnnx	nnnnnxyxxxnnyxnnxx	nnnnnxyxnxnnxxnnnx	nnxnnnnynnnynynnnn
		Column	405	xxnynnxxyy0000ynyy	nnnnnnnyxxnynxnnxn	ynnnxnnyxxnnyxnnyn	nnnnnnnynnnnxxnnxn	nnnnnnxynnnynynnxn
	Non-structural	Lamp post		xxxnynxxxy0000ynyy	nnnnnnnnxynynxnnxn	nnnnnnnnxynnyxnnyn	nnnnnnnnxxnnxxnnxn	nnnnnnnnnxnynynnxn
		Safety barrier	507	xxxnxyxxyy0000ynyy	nnnnnyynxynynxnnxx	nnnnnxnnxynnyxnnyx	nnnnnnnnxxnnxxnnyx	nnnnnnynnxnynynnyn

Fig. 13. Conditional relation strings for the 13 bridge element types

practice with over 15 years of bridge design experience, a professor of structural engineering and construction management, and a structural engineer with 5 years experience in reinforced-concrete design.

Fig. 13 shows a part of the matrix, with unique strings identified in the shaded cells. Each string represents the 19 conditions of a rule for object classification of the two elements in the row and column to which the cell belongs. The string in the primary girder–transverse beam cell, shown in bold text, is an example of a unique string. It can be translated as

- IF Object 1 is in contact with Object 2
- & Object 1 is in contact with Object 2's side face
- & Object 1 is not in contact with Object 2's front or back face
- & Object 1 is not in contact with Object 2's top face
- & Object 1 is not in contact with Object 2's bottom face
- & the two objects are not parallel along their extrusion direction
- & the two objects are not parallel along their long edges
- & Object 1's extrusion axis length is longer than Object 2's extrusion axis length
- & Object 1's volume is greater than Object 2's volume
- & Object 1 is not vertical
- & Object 1's extrusion direction is parallel to the road axis
- & Object 1's extrusion direction is not parallel to the skew angle of the bridge supports
- & Object 1 is not the bridge
- & Object 2 is not the bridge,
- THEN Object 1 is a primary girder and Object 2 is a transverse beam

The matrix yielded 14 unique rule strings. Seven of these were selected and implemented in the *SeeBIM* interface. These seven rules were sufficient for classifying all 13 object types, because each pairwise rule identifies two object types. Finally, the BIM model was loaded and processed for matching and enrichment in the rule-processing engine, which iterates over the set of rules using two nested loops to process rules for every possible pair of elements and in both possible orders for each pair. It infers new information in each cycle, stopping only when no additional information can be inferred. Thus the sequence of the rules is unimportant, and each rule may be checked several times in the enrichment process. For the case of this girder bridge, with 333 elements of ten different types, it proved possible to compile a set of rules that could perform complete classification with 100% precision and recall.

Discussion

Although the specific set of rules derived and implemented for the case of reinforced-concrete highway girder bridges was sufficient for correct classification of all thirteen object types, the value of this work is in the procedure, not in the rule set. The procedure enables users to compile *SeeBIM* rule sets for classifying the objects in a building information model that is not typed, and to do so with the knowledge that the rule set is comprehensive and effective.

Two aspects could be improved: (1) sufficiency of the rule set for any given domain, and (2) redundancy of rule set for computation.

1. In theory, for n object types, $n/2$ rules should suffice to identify all object types if only pairwise rules are used. In practice, one object type A may have unique pairwise relationships with two (or more) objects, e.g., B and C, and if neither B nor C has any unique pairwise relationships with objects other than A, then identification of B and C both depend on A. In this case, some pairwise rules overlap (share an object type), and more than $n/2$ rules will be needed. Furthermore, some dependencies may be nested (e.g., B depends on A, and C depends on the fact that A and B have been classified). In this case, C will be classified in the second or later iteration of the system

(e.g., Relationship 19 in Table 8 can be true only after a first iteration in which capping beams already have been identified). This too will result in the need for more than $n/2$ rules. Finally, given possible inaccuracies of the 3D object data, some rules may not work for instances of objects that are inaccurately modeled; redundancy in the number of rules will improve the probability of classifying the objects.

2. The number of possible rule strings that could be generated is much larger than the number of rules needed. In theory, the number of rules, with redundancy, could range from $n/2$ to n^2, and for each object type there are $2n - 1$ possible cells with rule strings. If k pairwise relationships are evaluated and each relationship has two conditions, a string could have 2^k different combinations. As a result, the $2n - 1$ cells almost always will have more than one unique string that comes from the 2^k different combinations. This paper manually chose a subset of unique rules from the full set such that the subset covered all the object types.

Note that at this stage, the approach does not consider the efficiency of computation because the runtime for large models remains very short. In theory, an algorithm could be developed to select an optimal subset of unique rules. Such an algorithm should also be able to evaluate whether some substring exists for any rule string selected such that it is still unique within the set of strings (i.e., it may be possible to compile unique rules with fewer than k conditions). This would improve the efficiency of the computation

Conclusions

Semantic enrichment is an important process that can relieve the problem of information interoperability and greatly improve the functionality of BIM models throughout a facility's lifecycle. This paper presented the enhancement of a semantic enrichment tool. The enhancements include a novel and rigorous method for compilation of inference rules, adoption of external data for enrichment, and additional operators for identification of shape features and spatial relationships that are common in geometrically complex facilities like bridges. The system was validated using a 3D model of a real-world concrete girder highway bridge.

The process developed for rule definition results in rule sets that contain sufficient tests to identify all the possible object types in the domain. This is an important enhancement of the *SeeBIM* approach to semantic enrichment. Naturally, however, such a system is still subject to the quality of the input data. The objects can be completely and correctly classified only when the models have sufficiently small errors in the locations and geometry of the bridge components to allow the geometry and topological relationship operators to perform correctly with suitable tolerances. However, model deficiencies cannot be completely avoided. For example, two objects expected to be touching may be modeled as overlapped or disconnected objects; in this case, the rule checking may give a false negative error. Setting large tolerance values could avoid such results, but setting the tolerance too large is likely to result in false positive errors. Notwithstanding the robustness of the rule compilation process, success of the object classification process remains dependent on the quality of the geometric model.

Future work will address additional aspects of semantic enrichment. For the general Scan-to-BIM use case, in addition to object classification, rules are needed for object aggregation, numbering/naming objects, generating abstract objects, and applying corrections where objects are occluded. In addition, researchers should consider attempting to apply machine-learning approaches to semantic enrichment for BIM in general and to each of these challenges.

Acknowledgments

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under Grant Agreement 31109806.0007. *SeeBridge* is cofunded by Funding Partners of the ERA-NET Plus Infravation and the European Commission. The Funding Partners of the Infravation 2014 Call are Ministerie van Infrastructuur en Milieu, Rijkswaterstaat, Bundesministerium für Verkehr, Bau und Stadtentwicklung, Danish Road Directorate, Statens Vegvesen Vegdirektoratet, Trafikverket—Trv, Vegagerðin, Ministere de L'ecologie, du Developpement Durable et de L'energie, Centro para el Desarrollo Tecnologico Industrial, Anas S.P.A., Netivei Israel–National Transport Infrastructure Company Ltd., and Federal Highway Administration USDOT. Dr. Ma was a postdoctoral researcher at the Technion when he worked on this project.

References

Aram, S. (2015). "A knowledge-based system framework for semantic enrichment and automated detailed design in the AEC projects." Ph.D. dissertation, Georgia Institute of Technology, Atlanta.

Beckmann, N., Kriegel, H.-P., Schneider, R., and Seeger, B. (1990). "The R*-tree: An efficient and robust access method for points and rectangles." *Proc. ACM Sigmod Rec.*, 19(2), 322–331.

Belsky, M., Sacks, R., and Brilakis, I. (2016). "Semantic enrichment for building information modeling." *Comput.-Aided Civ. Infrastruct. Eng.*, 31(4), 261–274.

Borrmann, A. (2006). "Extended formal specifications of 3D spatial data types." *Technical Rep.*, Technische Universität München, Munich, Germany.

Borrmann, A. (2010). "From GIS to BIM and back again—A spatial query language for 3D building models and 3D city models." *5th Int. 3D GeoInfo Conf.*, Vol. XXXVIII-4/W15, International Society for Photogrammetry and Remote Sensing, Hannover, Germany.

Borrmann, A., and Rank, E. (2009). "Specification and implementation of directional operators in a 3D spatial query language for building information models." *Adv. Eng. Inform.*, 23(1), 32–44.

Bosche, F., and Haas, C. T. (2008). "Automated retrieval of 3D CAD model objects in construction range images." *Autom. Constr.*, 17(4), 499–512.

Brilakis, I., et al. (2010). "Toward automated generation of parametric BIMs based on hybrid video and laser scanning data." *Adv. Eng. Inform.*, 24(4), 456–465.

BuildingSmart. (2010). "Coordination view version 2.0." ⟨http://www.buildingsmart-tech.org/specifications/ifc-view-definition/coordination-view-v2.0⟩ (Feb. 1, 2017).

BuildingSmart. (2013). "Industry foundation classes: IFC4 official release." ⟨http://www.buildingsmart-tech.org/ifc/IFC4/final/html/⟩ (Feb. 1, 2017).

Daum, S., and Borrmann, A. (2013). "Definition and implementation of temporal operators for a 4D query language." *Proc., Int. Workshop on Computing in Civil Engineering*, ASCE, Reston, VA.

Daum, S., and Borrmann, A. (2014). "Processing of topological BIM queries using boundary representation based methods." *Adv. Eng. Inform.*, 28(4), 272–286.

Daum, S., and Borrmann, A. (2015). "Simplifying the analysis of building information models using tQL4BIM and vQL4BIM." *Proc., EG-ICE 2015*, European Group for Intelligent Computing in Engineering, Munich, Germany.

Eastman, C., Lee, J. M., Jeong, Y. S., and Lee, J. K. (2009). "Automatic rule-based checking of building designs." *Autom. Constr.*, 18(8), 1011–1033.

Eastman, C., Teicholz, P., Sacks, R., and Liston, K. (2011). *BIM handbook*, Wiley, Hoboken, NJ.

Egenhofer, M. (1989). "A formal definition of binary topological relationships." *Proc., 3rd Int. Conf. on Foundations of Data Organization and Algorithms (FODO)*, Springer, Berlin.

Downloaded from ascelibrary.org by 161.112.232.103 on 08/30/17. Copyright ASCE. For personal use only; all rights reserved.

Egenhofer, M. (1994). “Spatial SQL: A query and presentation language.” *IEEE Trans. Knowl. Data Eng.*, 6(1), 86–95.

Egenhofer, M., Frank, A., and Jackson, J. P. (1989). “A topological data model for spatial databases.” *Proc., 1st Int. Symp. on the Design and Implementation of Large Spatial Databases*, Springer, Berlin.

Gaal, S. A. (1964). *Point set topology*, Academic Press, New York.

Hayes-Roth, F. (1985). “Rule-based systems.” *Commun. ACM*, 28(9), 921–932.

ISO. (1998). “ISO 10303-22—Standard data access interface.” ⟨https://www.iso.org/obp/ui/#iso:std:iso:10303:-22:ed-1:v1:en⟩ (Feb. 1, 2017).

ISO/OGC (International Standards Organisation/Open Geospatial Consortium). (2011). “Simple feature access—Part 1: Common architecture.” ⟨http://www.opengeospatial.org/standards/sfa⟩.

Jylanki, J. (2015). “An exact algorithm for finding minimum oriented bounding boxes.” ⟨http://clb.demon.fi/projects/an-exact-algorithm-for-finding-minimum-oriented-bounding-boxes⟩ (Feb. 1, 2017).

Kashani, A., Crawford, P., Biswas, S., Graettinger, A., and Grau, D. (2014). “Automated tornado damage assessment and wind speed estimation based on terrestrial laser scanning.” *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000389, 04014051.

Kim, S.-J., Lee, D.-Y., and Yang, M.-Y. (2004). “Offset triangular mesh using the multiple normal vectors of a vertex.” *Comput.-Aided Des. Applic.*, 1(1–4), 285–291.

Mazairac, W., and Beetz, J. (2013). “BIMQL—An open query language for building information models.” *Adv. Eng. Inform.*, 27(4), 444–456.

Nepal, M. P., Staub-French, S., Pottinger, R., and Webster, A. (2012). “Querying a building information model for construction-specific spatial information.” *Adv. Eng. Inform.*, 26(4), 904–923.

O’Rourke, J. (1985). “Finding minimal enclosing boxes.” *Int. J. Comput. Inf. Sci.*, 14(3), 183–199.

Pauwels, P., et al. (2011). “A semantic rule checking environment for building performance checking.” *Autom. Constr.*, 20(5), 506–518.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). “A design science research methodology for information systems research.” *J. Manage. Inform. Syst.*, 24(3), 45–77.

Perry, M., and Herring, J. (2012). “OGC GeoSPARQL—A geographic query language for RDF data.” *OGC implementation standard*, Oracle, Wayland, MA.

Ramaji, I. J., and Memari, A. M. (2016). “Interpreted information exchange: Systematic approach for BIM to engineering analysis information transformations.” *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000591, 04016028.

RDF. (2015). “IFC engine DLL.” ⟨http://rdf.bg/ifc-engine-dll.php⟩ (Feb. 1, 2017).

Rossignac, J. R., and Requicha, A. A. G. (1986). “Offsetting operations in solid modelling.” *Comput. Aided Geom. Des.*, 3(2), 129–148.

Sacks, R., Kedar, A., Borrmann, A., Ma, L., Singer, D., and Kattel, U. (2016). “*SeeBridge* information delivery manual (IDM) for next generation bridge inspection.” *33rd Int. Symp. on Automation and Robotics in Construction*, A. Sattineni, ed., Auburn Univ., Auburn, AL.

STEP Tools. (2016). “STEP and STEP-NC software for e-manufacturing.” ⟨http://www.step-nc.org/data/emo_fair_presentation8.pdf⟩ (Feb. 1, 2017).

Technion. (2015). “SeeBridge—Semantic enrichment engine for bridges.” ⟨http://seebridge.net.technion.ac.il/⟩ (Feb. 1, 2017).

Toussaint, G. T. (1983). “Solving geometric problems with the rotating calipers.” *Proc. IEEE Melecon, 83*, Athens, Greece.

Zeibak-Shini, R., Sacks, R., Ma, L., and Filin, S. (2016). “Towards generation of as-damaged BIM models using laser-scanning and as-built BIM: First estimate of as-damaged locations of reinforced concrete frame members in masonry infill structures.” *Adv. Eng. Inform.*, 30(3), 312–326.

Zhang, J. S., and El-Gohary, N. M. (2016). “Extending building information models semiautomatically using semantic natural language processing techniques.” *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000536, C4016004.

Downloaded from ascelibrary.org by 161.112.232.103 on 08/30/17. Copyright ASCE. For personal use only; all rights reserved.