CHECKING SPATIO-SEMANTIC CONSISTENCY OF BUILDING INFORMATION MODELS BY MEANS OF A QUERY LANGUAGE

Simon Daum & André Borrmann
Technische Universität München, Germany

ABSTRACT: *One of the characteristic features of object-based Building Information Models is the close integration of geometric and semantic information into one model. This concept is thoroughly implemented by the Industry Foundation Classes (IFC), a comprehensive data model designed to provide a sound foundation for complex data exchange scenarios. Besides the provision of a large variety of data types for capturing the semantics of building elements and spaces, the IFC also makes it possible to define relationships between building elements and/or spaces, respectively. In particular, a spatial aggregation hierarchy can be modeled by successively applying the relationship IfcRelAggregates to space objects. However, no validation options currently exist to check whether the semantically defined aggregation hierarchy complies with the geometric setup of the individual spaces and building elements. This lack of consistency between the semantic and the geometric part of the BIM model may lead to serious data interpretation errors at the receiving end. To prevent this, we propose a new method for validating spatio-semantic consistency based on the usage of the Query Language for Building Information Models (QL4BIM) which on the one hand provides a means of accessing the IFC object model and on the other hand provides high-level spatial operators, such as Disjoint, Touching and Containing. The formulation of corresponding queries makes it possible to verify the spatio-semantic consistency of the IFC model. The paper discusses application scenarios and provides a number of relevant examples.*

KEYWORDS: *BIM, IFC, Topology, Validation, Consistency, Spatial Relationships*

1. INTRODUCTION

A Building Information Model (BIM) is a comprehensive digital representation of a building. It provides an information base which is employed throughout its entire lifecycle – from the early phases of conceptual design, to the detailed planning phase, and the realization and operation phases (Eastman et al. 2011). To cover the different demands involved during the various phases, a BIM provides not only the precise 3D geometry of the building, but also non-geometric information, such as the type of the individual components, their attributes (material, insulation etc.) as well as the relationships between them.

Numerous specialists are involved in the design and engineering of buildings. In order to achieve interoperability between the different software products employed, the Building Information Model has to be represented by an open, neutral data model. The Industry Foundation Classes (IFC) form such a neutral data model and provide comprehensive means for the semantic and geometric description of a building and its components (buildingSMART, 2012).

The IFC model is based on a strict separation between the semantic and the geometric description. In the semantic part, the building is described as an agglomeration of semantic entities with specific properties and relationships between one another. Each of the semantic entities can be associated with one or more geometric representations. This approach is well-suited for supporting the different demands of different users and/or applications on the geometric representation. However, this separation bears the risk that inconsistencies may arise between the semantic and the geometric description.

To provide an example we refer to the relationship *IfcRelContainedInSpatialStructure*, which is used to semantically describe the association between a spatial container and a building element contained in it. When exported erroneously by the authoring application, the resulting IFC model may contain space-element-pairs for which this semantic relationship is set, while the geometric representations associated with them do not actually fulfill the containment property. These kinds of inconsistencies are hard to detect and may lead to serious misinterpretations by the receiving application.

Citation: Daum, S. & Borrmann, A. (2013). Checking spatio-semantic consistency of building information models by means of a query language. In: N. Dawood and M. Kassem (Eds.), Proceedings of the 13th International Conference on Construction Applications of Virtual Reality, 30-31 October 2013, London, UK.

In this paper, we introduce an approach for automatically checking the spatio-semantic consistency of IFC models. The proposed method is based on a query language which (1) provides access to the semantic part of the IFC model and (2) supplies spatial operators that make it possible to undertake a formal analysis of the geometric model.

2. RELATED WORK

Stadler & Kolbe (2007) discuss the problem of spatio-semantic coherence in the context of 3D city models and the associated standard CityGML. Similar to the IFC, CityGML also uses a dual data structure with a semantic and a geometric part. The main difference is that CityGML also provides ways of describing aggregation relationships on the geometric side, which is not the case for the IFC model. Accordingly the discussed approach focuses on aligning two aggregation hierarchies, while the approach presented here utilizes formal spatial analysis functionality for identifying qualitative spatial relationships between the geometric objects represented in IFC models.

The domain specific, open query language BIMQL (Mazairac & Beetz, 2013) can be used to select and update subsets in a building model. Although BIMQL does not provide an interface for checking spatio-semantic coherence, it provides powerful query functionality. The idea of query shortcuts, which is integrated in BIMQL, is revisited in QL4BIM. These shortcuts are partly static so that the IFC object model is simplified e.g. to pick up geometry representations. Secondly, QL4BIM also provides dynamic shortcut creation by end users to unburden the handling of complex traversals in the object model.

3. REPRESENTATION OF SEMANTICS AND GEOMETRY IN THE IFC

The IFC Model provides a comprehensive set of entities to describe the semantics and the geometry of a digital building model. It is maintained by the international organization buildingSmart and has been implemented by a large number of AEC software vendors. The currently release, version 4, has been published as ISO standard 16739. The majority of the ongoing governmental activities for promoting BIM for the public construction sector, such as the UK Government BIM Strategy or the US National BIM standard, heavily enforce the usage of this open data format for data exchange scenarios (bimtaskgroup 2013, NBIMS-US 2013).

The IFC model is defined by means of the data modeling language EXPRESS which forms part of the ISO standard STEP – STandard for the Exchange of Product model data (SCRA, 2006). The model is strongly object-oriented, providing a large number of classes (called entities) arranged in an extensive inheritance hierarchy. In addition, the IFC model applies the concept of objectified relationships, i.e. there are specific classes which need to be instantiated for representing relationships between entities. In this paper, we specifically focus on relationships with spatial semantics.

The IFC model follows a strict separation of the semantic description and the geometric representation (Fig. 1). The semantic part is the leading information structure in the IFC, proving main access to the model and all associated information.

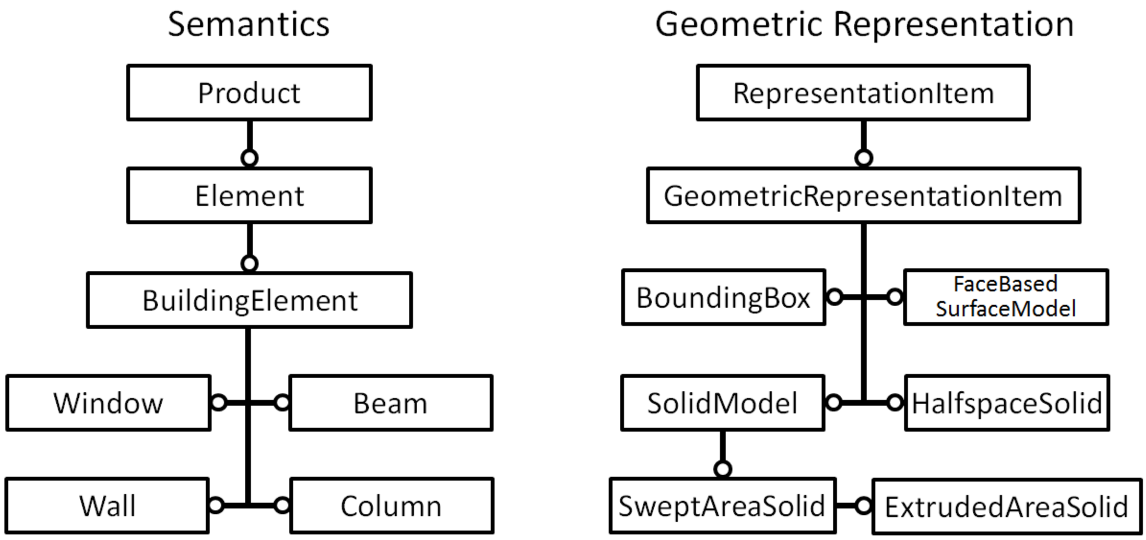


Fig. 1: Separation of semantics and geometric representation in the IFC (EXPRESS-G diagram)

The root object of an IFC model is an instance of *IfcProject*. Starting from this object, multiple *IfcRelAggregate* relationship instances are successively employed to create an aggregation hierarchy comprising the site, the building(s), the building part(s) and the building storey(s). The corresponding classes are sub-classes of *IfcSpatialStructureElement*. The actual building elements (wall, columns, etc.) are subsequently associated with one or more stories by means of the relationship *IfcRelContainedInSpatialStructure* (Fig. 2).

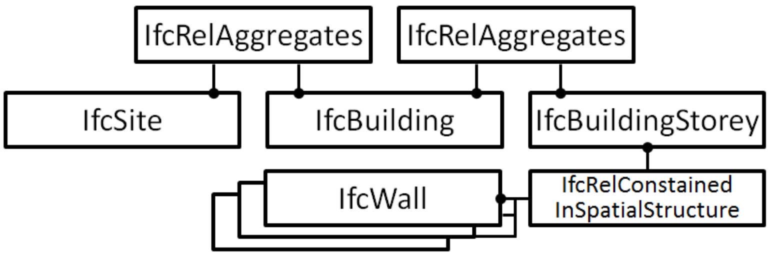


Fig. 2: The spatial aggregation hierarchy provided by the semantic parts of the IFC (EXPRESS-G diagram)

In addition, space objects can be included in the model to represents rooms. They should be associated with the surrounding walls by means of the *IfcRelSpaceBoundary* relationship. Each semantic object representing a building element or a space can be associated with one or more geometric representations. This is realized by associating the *IfcProduct* instance with an *IfcProductRepresentation* instance which in turn may refer to a number of instances of *IfcRepresentation* (Fig. 3). Possible options for representing geometry in IFC are Boundary Representation, Constructive Solid Geometry, and Swept Solid, among others.

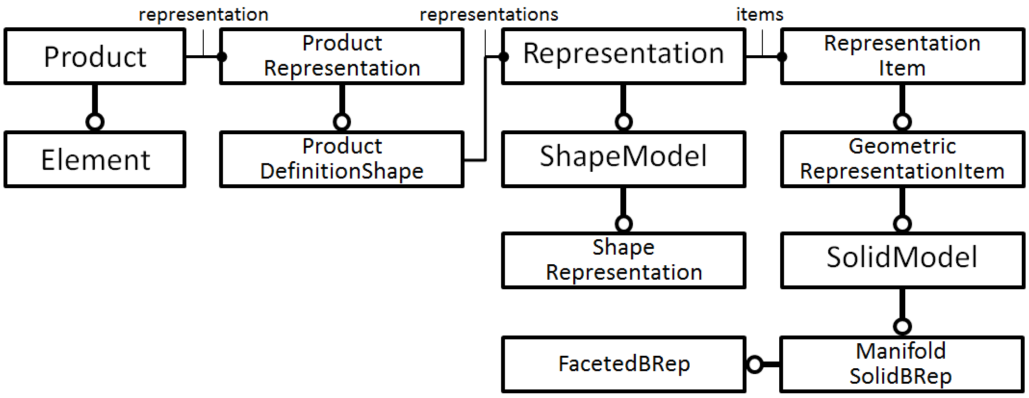
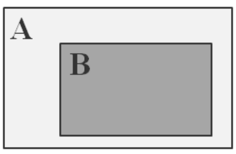


Fig. 3: Association of a semantic object with geometric representations (EXPRESS-G diagram)

If the IFC model is exported correctly by the BIM authoring application, the aforementioned relationships are set such that they comply with the geometric representation. For example, a building element and a space are associated via the *IfcRelContainedInSpatialStructure* relationship, if, and only if, the corresponding geometric objects do fulfill the containment relationship. However, due to the sheer complexity of the IFC model, in many cases erroneous models are created. While geometry is often handled correctly, particularly critical is the accurate use of the relationships with spatial semantics. It is here where inconsistencies between the geometric and semantic representation can easily arise. In the next section we present an approach for checking the consistency by means of a query language.

4. QL4BIM – A QUERY LANGUAGE FOR BUILDING INFORMATION MODELS

To realize the checking functionality described above, the Query Language for Building Information Models (QL4BIM) presented in (Borrmann & Rank, 2009a, 2009b, Daum & Borrmann 2013a, 2013b) is utilized. On the one hand, the query language provides an object-oriented access to the IFC model (Daum & Borrmann, 2013b). The main feature, however, is its strong support for temporal and spatial operators which allows users to operate on a more abstract level and formulate high-level queries such as "Select all walls located above slab 1 but constructed earlier". The spatial operators comprise metric, directional and topological operators (Borrmann & Rank, 2009a, 2009b). The topological operators – which are of particular interest here – make it possible to analyse topological relationships between objects in three-dimensional space. The defined predicates correlate two spatial entities and can be described by the 9-Intersection Model (9-IM) introduced in (Egenhofer, 1991). The 9-IM calculus is based on the mathematical theory of Point Set Topology (Gaal, 1964) which applies the notion of the neighbourhood of a point to describe topological concepts such as the interior $A°$, the boundary δA and the exterior A^{-} of a point set A.



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

A contains B

Fig. 4: Contain relationship described by the 9-Intersection Model

The intersections of interior, boundary and exterior of two entities result in a 3 × 3 matrix. The individual entries indicate if there is an empty or a non-empty set for the particular intersection. Fig. 4 shows the 9-IM matrix for a simplified 2D constellation where object A is *inside* object B and object B *contains* object A, respectively.

The 9-IM matrix can be used to define the topological predicates *Disjoint, Equal, Touching, Containing, Inside-of, Overlapping, Covering* and *Covered-by* in 3D space as depicted in Fig. 5. The algorithms developed to implement the topological operators populate a 9-IM matrix by performing tests on the operands' geometry. Two different approaches have been developed: one operating on an octree representation (Daum & Borrmann, 2012) and another operating on the boundary representation (Daum & Borrmann, 2013a).

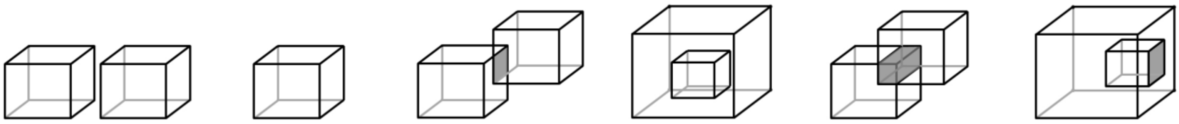


Fig. 5: Available topological predicates provided by the query language (Borrmann & Rank, 2009b)

QL4BIM make use of the LINQ technology as it provides powerful query mechanisms for in-memory collections and object networks. LINQ is neatly integrated into the .NET framework and queries can be formulized in C# syntax. The queries are type safe and attributes and methods of involved objects can be used. For the definition of a query, an anonymous function, called a Lambda expression is defined. QL4BIM acts directly on the IFC object model and is thus well suited for queries with semantic subparts. For more information concerning the query system, see (Daum & Borrmann, 2013b).

5. CHECKING THE SPATIO-SEMANTIC CONSISTENCY OF IFC MODELS USING QL4BIM

In this contribution we present a concept for the validation of spatio-semantic consistency of IFC models using QL4BIM. The spatial structure described in the semantic part is validated by means of the available geometry representations. Spatio-semantic inconsistencies typically arise as a product of modeling mistakes by the user. Additionally, the complexity of the IFC model contributes to the erroneous import or export functionality of the BIM authoring application, which may also result in corrupted building models.

The developed approach comprises two parts: first, the model's spatial hierarchy built up by *IfcSite*, *IfcBuilding*, *IfcBuildingStorey* and *IfcSpace* entities is inspected; and second, the topological relationships between *IfcProducts* with their superior spatial structure, e.g. an *IfcBuildingStorey*, are evaluated.

5.1 Spatio-semantic consistency of the IfcRelAggregates relationship

In the first part of processing, the entities that define the spatial structure are fetched from the IFC model. In a plain configuration, a hierarchical structure similar to Fig. 6 should be found. When iterating over all *IfcRelAggregates* relationships, *IfcsSites* and related entities are topologically examined. The geometric representation of *IfcSite* and *IfcBuilding* are described by *IfcProductDefinitionShape* and *IfcLocalPlacement* objects. As a general concept of the IFC, it is possible to associate several geometry representations with one entity if this is required in a particular context.

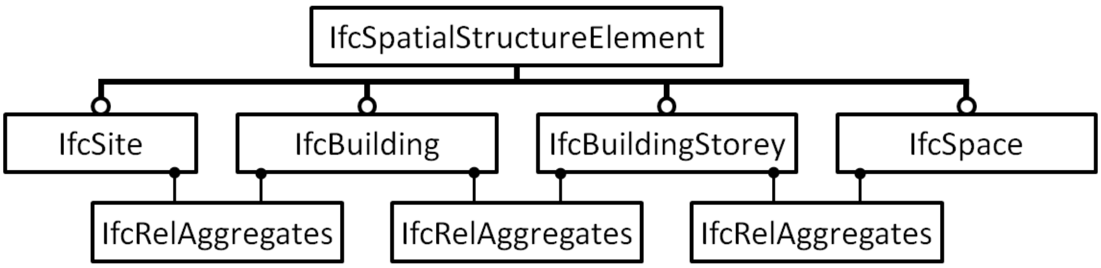


Fig. 6: Exemplary IFC spatial structure established by IfcRelAggregates relationships (EXPRESS-G diagram)

In the developed prototype system for all instances of *IfcProduct* an explicit geometry representation is generated and made available as *IfcFacetedBrep* via the Shape attribute of the *IfcProduct* object. The complete query formulated in QL4BIM is shown in Fig. 7. It returns all non-conforming *IfcBuilding* objects combined with their hosting *IfcSite* for further manual review.

```
IfcRelAggregates.Select(a =>               //simplified version without handling of
                                           //unsupported types
{
    var site = a.RelatingObject As IfcSite;
    var shapeSite = site.Shape;
    var nonconfirmingBuildings = new List<IfcBuilding>();

    foreach relatedObject in a.RelatedObjects
    {
        var building = relatedObject As IfcBuilding;
        var shapeBuilding = building.Shape;

        var allowed =   shapeSite.Contain(shapeBuilding) ||
                        shapeSite.CoveredBy(shapeBuilding);
        if(!allowed)
            nonconfirmingBuildings.Add(building);
    }
    return new Tuple<IfcSite, List<IfcBuilding>>(site, nonconfirmingBuildings);
}
```

Fig. 7: Query returning IfcSite objects and related, topological non-conforming IfcBuilding objects

In QL4BIM all objects from a given set, e.g. *IfcRelAggregates*, are examined in the query expression. In the developed algorithm, the types of the related and relating objects are first checked. For reasons of clarity, this is not shown in the depictured code in Fig. 7. If the appropriate types are present (e.g. an *IfcSite* and *IfcBuildings*), topological processing is executed by calling the *Containing* and *Covered-by* predicates.

These are the topological allowable attributes of aggregated *IfcSite* and *IfcBuilding* objects as demonstrated in Fig. 8. Buildings that do not conform to these topological predicates indicate that there is an error in either the topological definition or in the used geometry representations. Therefore, a list of buildings is linked with each site and erroneous buildings are added. Finally, the query yields an enumeration of all tuples, each containing one site object and its topologically non-confirming building objects.

The same approach can be applied for checking the spatio-semantic consistency of the remaining aggregation relationships, e.g. *IfcBuilding* objects related to *IfcBuildingsStorey* objects and *IfcStorey* objects related to *IfcSpace* objects. Here, the type selection has to be adapted accordingly.

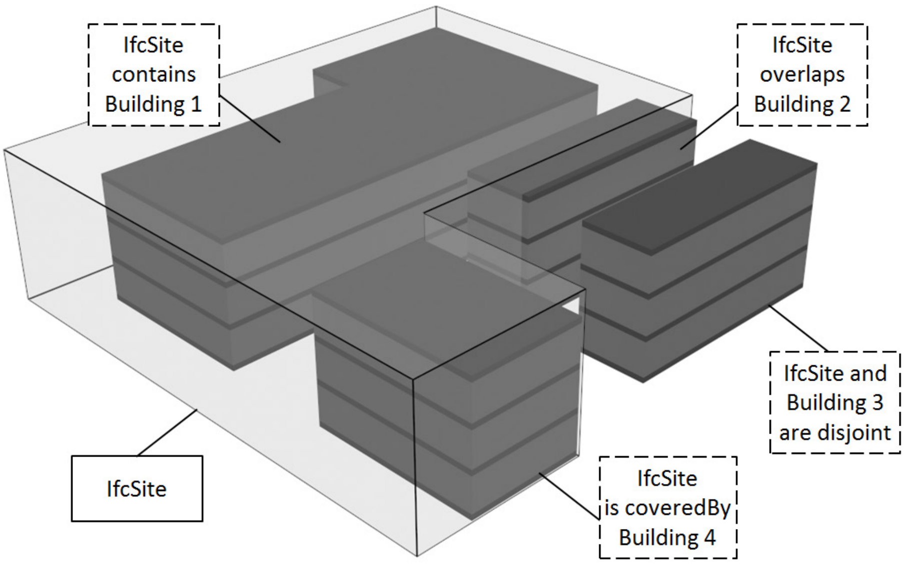


Fig. 8: The geometry representation of an IfcSite and the topological classification of its IfcBuilding objects

Additionally, the IFC model makes it possible to group entities used in the project's spatial structure. As an example, an *IfcBuildingStorey* object can be associated with its child storeys. In this case, the parent storey reflects its grouping semantic by a *CompositionType* attribute of the value *COMPLEX*. In the nested children this attribute is set to *PARTIAL* as shown in Fig. 9. The *CompositionType* member variable is available in all subtypes of *IfcSpatialStructureElement*. If such nesting relationships are present in the model, the spatio-semantic consistency can also be verified using the same processing method except that only one type is involved in the query, e.g. *IfcBuildingStorey* or *IfcSpace*.

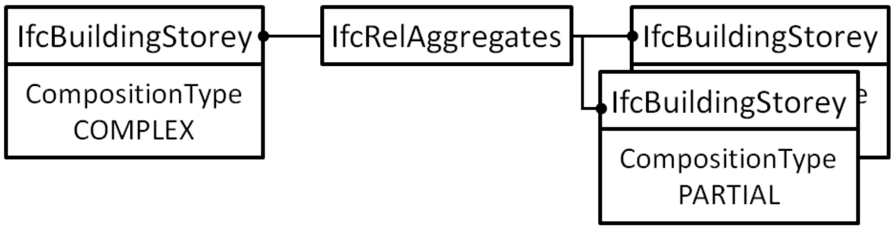


Fig. 9: Example of a grouping established by nested IfcSpatialStructureElement objects (EXPRESS-G diagram)

5.2 Spatio-semantic consistency of the IfcRelContainedInSpatialStructure relationship

As described above, it is possible to establish a spatial structure semantically by relating *IfcSpatial-StructureElements* like *IfcSite*, *IfcBuilding* and *IfcBuildingStorey*. Furthermore, the IFC model is able to reflect a containment relationship of products and a superior *IfcSpatialStructureElement*. This important semantic information is used frequently in downstream processes like resource management and construction scheduling. For example, equally leveled columns are connected to the storey they are on. This is realized using the *IfcRelContainedInSpatialStructure* relationship. An *IfcElement*, subtype of *IfcProduct*, can only be assigned once to one *IfcSpatialStructureElement*. Typically, the Brep geometry of the spatial structure contains the element's geometry. On rare occasions, the contained element overlaps the spatial structure to which it is related. As an example, a lift shaft might be modeled as contained by the ground level storey. The other storeys connect to the shaft via *IfcRelAggregates* objects. This means that elements with geometry representations that do overlap *IfcSpatialStructureElements* cannot be generally declared as false. The decision as to how to handle overlapping situations must be made on project level. The exemplary query for the verification of containment relationships of *IfcElements* and *IfcSpatialStructureElements* is shown in Fig. 10.

```
IfcRelContainedInSpatialStructures.Select(a =>
{
    var spatialStructure = a.RelatingObject As IfcSpatialStructureElement;
    var spatialStructureShape = spatialStructure.Shape;
    var nonconfirmingElements = new List<IfcElement>();

    foreach relatedObject in a.RelatedObjects
    {
        var element =   relatedObject As IfcElement;
        var elementShape = element.Shape;

        var allowed =   spatialStructureShape.Contain(elementShape) ||
                        spatialStructureShape.CoveredBy(elementShape) ||
                        spatialStructureShape.Overlap(elementShape);

        if(!allowed)
            nonconfirmingElements.Add(element);
    }
    return new Tuple<IfcSpatialStructureElement, List<IfcElement>>
                    (spatialStructure, nonconfirmingElements);
}
```

Fig. 10: Query returning IfcSpatialStructureElement objects and associated non-conforming IfcElement objects

For each *IfcRelContainedInSpatialStructure*, its related object of type *IfcSpatialStructureElement* is fetched to receive the referenced Brep geometry. The return type is an enumeration of tuples in which the first element is a spatial structure. The tuple's second element represents a list of *IfcElement* objects. The query thereby yields *IfcSpatialStructureElements* that are connected to topologically defective *IfcElements*. The returned tuples then have to be revised and the user needs to determine whether the failure arises due to errors in the defined geometry representations or topological deficits in the building information model.

6. EXAMPLE

In the following, we consider the scenario of a structural model of an office building. A 3D view of the building is depictured in Fig. 11.

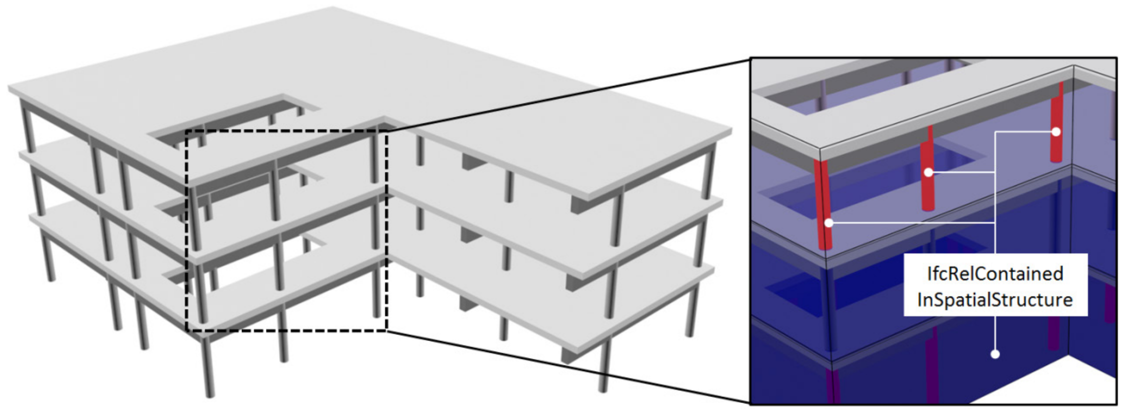


Fig. 11: Structural model of an office building with topologically erroneous containment relations

For the end user, the scene seems to be accurate because defects in the topological definitions of the model cannot be recognized without undertaking a formal validation. This checks the geometry information available in the model against the topological relationships. As shown on the right side of Fig. 11, three columns of the second storey are erroneous related to the ground level's spatial structure, an *IfcBuildingStorey*.

In the IFC model, the error is encoded in the XML markup in the *IfcRelContainedInSpatialStructure* elements concerning the three columns. For the first column with id=i34066 this XML element is shown in Fig. 12. In the *RelatingStructure* element, an incorrect reference to the *IfcBuildingStorey* id=i1595 is established. This leads to a configuration in which the column is modeled as spatially contained by the ground level storey.

```
<IfcRelContainedInSpatialStructure id="i34102">
    <GlobalId>35r2O_4kTAf8AbY</GlobalId>
    <RelatedElements exp:cType="set">
        <IfcColumn ref="i34066"/>
    </RelatedElements>
    <RelatingStructure>
        <IfcBuildingStorey ref="i1595"/>
    </RelatingStructure>
</IfcRelContainedInSpatialStructure>
```

```
<IfcBuildingStorey id="i1595">
    <GlobalId>0hozoFnxj9leO</GlobalId>
    <Name>01 - Entry Level</Name>
    <ObjectPlacement>
        <IfcLocalPlacement ref="i1594"/>
    </ObjectPlacement>
    <LongName>01 - Entry Level</LongName>
    <CompositionType>element</CompositionType>
    <Elevation>0.</Elevation>
</IfcBuildingStorey>
```

Fig. 12: ifcXML encoding of an erroneously established containment relationship

This kind of error can arise as a result of a mistake in the editing of the model or as a product of an inaccurate export of the IFC data. Although not recognizable in the visualization, such undetected mistakes in the modeling of a building can cause problems in the downstream process. High quality results and efficient workflow in the construction phase can only be achieved if the data basis is accurate. For example, difficulties concerning material deliveries are expected to occur here. When the ground level is constructed, material for the three erroneously included columns is stored but not used. This material must be stored until it is actually needed, which is not before the third storey is built. If such errors accumulate, the construction of a building becomes more difficult and the time required and financial expenditure increases. If the query defined in Fig. 10 is used to examine all *IfcRelContainedInSpatialStructure* objects in the IFC model, it will return a tuple containing the *IfcBuildingStorey* i1595 and a list containing the three problematic *IfcColumns*.

7. CONCLUSION AND FUTURE WORK

This contribution presented a new approach for the computational validation of spatio-semantic consistency of IFC-based building information models using the query language QL4BIM. As a key aspect, the query language provides access to the semantic model of the IFC and at the same time makes it possible to apply high-level spatial operators that act directly on the geometric representations of the individual objects. Combining these features makes it possible to efficiently and flexibly formulate rules for validating spatio-semantic consistency.

The presented examples show that deficits in the established spatial structure of components and virtual containers can be reliably detected. The methods developed enable the end user of building information models to inspect even large data sets efficiently. This significantly contributes to improving the quality of IFC models. Finally, this enhances efficient workflows and cost-effectiveness in the buildings' construction phase.

In future research, we will integrate semi-automatic repair functionality into the system that automatically produces proposals for expert users to help them create the correct spatial structure of building elements and spaces.

8. REFERENCES

Borrmann, A., and Rank, E. (2009a). Specification and implementation of directional operators in a 3D spatial query language for building information models, *Advanced Engineering Informatics* AEI 23 (1), pp. 32-44. 2009.

Borrmann, A., and Rank, E. (2009b). Topological analysis of 3D building models using a spatial query language, *Advanced Engineering Informatics* AEI 23(4), 370-385.

bimtaskgroup (2013), UK Government BIM Strategy, http://www.bimtaskgroup.org/ (10. May 2013).

buildingSMART Ltd. (2013). IFC data schemas, Industry Foundation Classes http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/index.htm (10. May 2013).

Daum, S., and Borrmann, A. (2013a). Boundary Representation-Based Implementation of Spatial BIM Queries, *Proceedings of the 20th EG-ICE International Workshop* (EG-ICE 2013).

Daum, S., and Borrmann, A. (2013b). Definition and Implementation of Temporal Operators for a 4D Query Language, *Proc. of the ASCE International Workshop on Computing in Civil Engineering* (ASCE IWCCE 2013).

Daum, S., and Borrmann, A. (2012). Efficient and Robust Octree Generation for Implementing Topological Queries for Building Information Models, *Proc. of the 19th EG-ICE International Workshop* (EG-ICE 2012).

Eastman, C. M., et al. (2011): BIM handbook. A guide to building information modeling for owners, managers, designers, engineers and contractors. 2nd Edition, Hoboken, NJ,Wiley.

Egenhofer, M. and Herring, J. (1989). A mathematical framework for the definition of topological relationships, *Proceeding. of the 4th Int. Symp. on Spatial Data Handling.*

Gaal, S. A. (1964). Point set topology (Vol. 16). Academic Press.

Mazairac, W., and Beetz, J. (2013). BIMQL - An Open Query Language for Building Information Models, *Advanced Engineering Informatics*, accepted.

NBIMS-US, US National BIM standard (2013). http://www.nationalbimstandard.org/ (10. May 2013).

SCRA (2006), STEP Application Handbook, ISO 10303, Version 3, http://www.uspro.org/ (10. May 2013).

Stadler, A., and Kolbe, T. H. (2007). Spatio-semantic Coherence in the Integration of 3D City Models. *Proceedings of the 5th International Symposium on Spatial Data Quality,* Enschede.