

# Representation and Integration of As-Built Information to IFC Based Product and Process Models for Automated Assessment of As-Built Conditions

Burcu Akinci, Frank Boukamp

**ABSTRACT:** Frequent assessment of as-built conditions is necessary for active project control during construction. Assessment of as-built conditions involves collecting accurate as-built information and comparing the as-built information collected with given design and schedule information. Currently, different technologies exist for collecting accurate and comprehensive as-built information. The as-built information generated through these technologies is then manually evaluated by comparing it to the design information. This process is time-consuming and error prone. Therefore, there is a need for an automated assessment of as-built conditions. This paper discusses the applicability of the Industry Foundation Classes (IFC) in their current version 2x to support the assessment of as-built conditions of construction projects. The goal is to create a project model that represents and integrates as-built and design information. In this paper, we state different types of information that need to be represented in this integrated model, describe common ways of representing this information in IFC Rel.2x, and highlight the limitations of current IFC Rel.2x specifications in representing an integrated design and as-built model. The paper concludes with a proposal for extending the IFC specifications to enable the automated assessment of as-built conditions.

**KEYWORDS:** Industry Foundation Classes, integrated design and as-built models, construction

## 1. INTRODUCTION

Frequent and comprehensive assessment of as-built conditions is necessary for project control and to minimize delays caused by late detection of defects at construction sites. A comprehensive assessment of as-built conditions involves frequent, complete and accurate collection and storing of data on as-built conditions, and a formalized approach for comparing the as-built conditions with the as-designed requirements.

Different technologies can now enable the frequent, complete and accurate collection of as-built conditions. For example, laser-scanning technologies are gaining acceptance in the A/E/C (Architecture/Engineering/Construction) industry because of their ability to create complete and accurate 3D as-built environment models based on spatial information [1]. However, currently, even if a laser scanner is used at a construction site, the comparison of as-built models with existing design information is still being performed manually by visually inspecting both design and as-built information according to a construction schedule and design specifications [2]. This process is time-consuming and error prone.

Therefore, there is a need for an automated assessment of as-built conditions.

To automate the assessment of as-built conditions, the as-built information, consisting of as-built product model and schedule, needs to be integrated with the as-designed product and process information. Both as-built and as-designed models need to be represented in a semantically rich way and the necessary relationships between these two models need to be created and maintained throughout construction.

This paper discusses possible ways to integrate as-built information within IFC (Industry Foundation Classes) based product and process models, and evaluates the IFC Rel.2x specifications in terms of its ability to represent integrated design and as-built information. The discussions in this paper are based on the implementation of a small portion of a warehouse construction case.

## 2. BACKGROUND

Currently different industry efforts exist for modeling and merging design, construction and FM (facility management) information. For example, the Steel Construction Institute

is developing CIMsteel Integration Standards (CIS), a framework for modeling construction projects that are mainly based on steel structures. The goal of this effort is to “allow software vendors to make their engineering applications mutually compatible” [3]. As another example, the International Alliance for Interoperability (IAI) is developing the Industry Foundation Classes (IFC), a similar framework that shall allow making A/E/C and FM related information exchangeable between different software programs by standardizing the representation of A/E/C and FM related information within project model descriptions. In our research, we decided to use the IFC representation since it does not focus on only one specific industry and hence it is more generally applicable.

Currently IFC models are either used to store design or as-built information. Often an IFC-based design model is created and later on updated with as-built information, overwriting the designed information. To allow the storage of different versions of a project, like ‘Schematic Design’, ‘Detailed Design’, ‘As-Built’, etc. these models are usually stored into different files, which then are associated with the description of the version. For example Metracker, a performance metric tracking tool, uses this approach [4].

To be able to automate the assessment of as-built conditions efficiently, we need to represent detailed design information and as-built information in one IFC model. An integrated representation of design and as-built models allows to identify and reason about the relationships between design model and as-built model efficiently, and hence minimizes the computational efforts associated with checking one model with the other to assess as-built conditions.

Section 4 of this paper describes how we build on and extend the current IFC Rel.2x specifications to develop an integrated design and as-built model for the case that we implemented.

### 3. CASE STUDY DESCRIPTION

In November 2001, we performed a case study at a warehouse construction project in Pittsburgh, PA, where we collected design and schedule information through conventional methods and as-built information through laser scanners. The main structure of the warehouse consists of steel frames. We modeled the design and as-built information of one of the steel frames using IFC Rel.2x to automate the assessment of as-built conditions for that frame.

Assessment of as-built conditions in the frame example involves comparing the as-designed and the as-built geometric, material and schedule information. Figure 1 shows the design and as-built information of the frame that we modeled. In this case, the contractor used the right material and built it in the right location. However, the as-built schedule deviated from the as-designed. Hence, an automated assessment system should be able to compare the different types of information represented in as-design and as-built model and identify the schedule deviation in this case.

### 4. INTEGRATING DESIGN AND AS-BUILT INFORMATION USING IFC

Since the assessment of as-built conditions requires reasoning about material information, building element type information, geometric information, scheduling information and relationships between building elements, we need to explicitly represent this information. In IFC Rel.2x it is usually represented as follows:

MATERIAL INFORMATION is usually assigned to the respective *IfcObject*-object, through a relationship called *IfcRelAssociatesMaterial* (Figure 2).

BUILDING ELEMENT TYPE INFORMATION is usually modeled through properties or type libraries defined by relationships of type *IfcRelDefinesByProperties* (Figure 3) or *IfcRelDefinesByType* (Figure 4) respectively.

GEOMETRIC INFORMATION is modeled by grouping *IfcRepresentation*-objects in an *IfcProductRepresentation*-object that is then assigned to a building object (Figure 5) or through properties providing geometric information being assigned to an *IfcObject*-object by an *IfcRelDefinesByProperties* relationship (Figure 6).

SCHEDULING INFORMATION is represented in IFC by assigning an *IfcTask*-object, representing a task in a schedule, to an *IfcObject*-object, via a relationship of type *IfcRelAssignsToProcess* (Figure 7).

CONNECTIONS BETWEEN BUILDING ELEMENTS are modeled by using *IfcRelConnectsElements* relationships.

As stated before, IFCs were developed to allow the representation of either design or as-built information in an interoperable product model. When trying to incorporate both into one model, three scenarios might be observed [5]:

1. The current IFC model representations should be used if possible.
2. The current IFC model representations can be extended if necessary.
3. New concepts are developed only if a sufficient representation cannot be created using scenarios 1 and 2.

Since minimal changes to the IFC model are desirable, first we tried to use existing representations to integrate the as-built information into the IFC-based model and then suggested extension to the current representations before we develop any new representation schema.

#### 4.1 Utilization of current IFC Rel.2x Representations

Three approaches seemed promising in representing integrated as-designed and as-built information using the current IFC Rel.2x specifications:

#### (1) Utilization of representation context concept defined through *IfcRepresentationContext*.

*IfcRepresentationContext* objects, to which product representations can be assigned, allow products to have different representations, e.g. one that represents a sketch of the product and one that represents the detailed design of the product. To do this, the *ContextType* attribute of this object is being proposed to be used to store different context information such as 'Sketch' and 'Design' to distinguish between the different contexts. By using 'Design' and 'AsBuilt' as values for this attribute we would be able to create two different representation contexts for a building element to which we can assign design and as-built information respectively. This would allow having the design and as-built information incorporated into one IFC-based product model. Currently, only *IfcRepresentation* objects can be assigned to an *IfcRepresentationContext*. As a result, only geometric or topologic information can be assigned to these different contexts. However, as we discussed in the case study, other information, such as scheduling information, will also have to be kept in distinguishable design and as-built contexts. The usage of different *IfcRepresentationContexts* thus becomes insufficient for our purposes.

#### (2) Utilization of logical grouping concept to group as-designed and as-built information.

IAI provides the possibility to group information by using *IfcGroup*-objects. Based on this, we can group design information and as-built information in a design-group or an as-built-group through the assignment of the information to the respective *IfcGroup*-object (Figure 8). The design and as-built information can be assigned to the respective groups by using *IfcRelAssignsToGroup* relationships. The *IfcRelAssignsToGroup*-relationship is a subtype of *IfcRelAssigns*-relationships, which can only link subtypes of *IfcObject* among each other. This creates a problem

for representing designed and as-built material information using *IfcMaterialProperties*-objects which are not derived from *IfcObject*. Since it is important to represent and reason about the material information in design and as-built models, the *IfcGroup*-based approach becomes inadequate for our purposes.

### (3) Creation of new properties of entities using *IfcProperty* object.

This would involve the creation of *IfcProperty*-objects having a specific value in its *Name*-attribute which tells whether it describes design or as-built information. These *IfcProperty*-objects then will be assigned to the related entities through an *IfcRelDefinesByProperties*-relationship. The usage of properties for extending IFC concepts is the most common technique used. Currently properties can only be assigned to objects derived from the class *IfcObject*. This again creates a limitation of the usability of this concept for our purposes, since in IFC for example material information or relationships in general are not derived from *IfcObject*.

#### 4.2. Extended Concepts

Since the current IFC Rel.2x specifications do not allow us to model the as-built conditions completely we investigated other ways to incorporate design and as-built information into one project model with minimal extension of existing IFC concepts.

Extending the concept of reified relationships looked most promising for our purposes. Most of the information that we need to model to support automated assessment of as-built conditions is assigned through relationships to the objects (Figures 2-4, 6, 7). All of these relationships are represented as instances of classes that are derived from the superclass *IfcRelationship*. Thus, if we add a new attribute called *Context* to the *IfcRelationship*-class, then we can use that attribute to determine whether the information is related to the design or the as-built context. This *Context*-attribute should be of type *IfcLabel* and have a

descriptive name, e.g. 'Design' or 'AsBuilt', assigned as value.

To enable the interoperability of the model, it is necessary to standardize these values that will be assigned to the context-distinguishing attributes. Otherwise, software developers using IFCs will have to agree on how to describe the different contexts to be able to interpret the models that are interchanged using their software.

Enabling the class *IfcRelationship* to describe the context it is assigned to through the new *Context*-attribute allows us to keep nearly all the information that we need for automated defect detection in separate contexts, i.e. a design and an as-built context. This is because all the information is assigned through subtypes of *IfcRelationship* to the entities in the IFC model (Figures 2-4, 6, 7). With this extension we can provide a 'Design'-relationship to the design information of an element and an 'AsBuilt'-relationship to the respective as-built information of an element.

However, this extension of the IFC concept of objectified relationships is not entirely sufficient, since the geometric information is the only information that may not be assigned to an object through a relationship but may also be described through *IfcRepresentation*-objects that are directly assigned to the element not using *IfcRelationship*-objects.

#### 4.3. Proposed Concept

In the previous two sections, we described four different approaches to incorporate design and as-built information into one integrated model:

- using *IfcRepresentationContext*-objects
- using *IfcGroup*-objects
- using new *IfcProperty*-objects
- using an extended version of *IfcRelationship*

All of these approaches, when used alone, are insufficient for the purpose of automated as-built assessment. Instead of creating a completely new concept for the integration of design and as-built information into one IFC-based project model, we propose combining two of the approaches described above as the proposed way of modeling as-designed and as-built information in an integrated way.

Our proposed approach combines utilization of different *IfcRepresentationContext*-objects and extending the class *IfcRelationship* as explained before (see Figure 9).

The weakness of the concept of extending the class *IfcRelationship* was that the part of the geometric information that is described through *IfcRepresentation*-objects cannot be distinguished by using 'Design'- and 'AsBuilt'-relationships. This weakness can be overcome by the usage of different *IfcRepresentationContext*-objects. The *IfcRepresentation*-objects, representing the geometric information of the entities in the IFC model, are assigned to a 'Designed'- or 'AsBuilt'-representation context. All the other information will be assigned to the entities through objectified relationships that are defined through our extended version of the *IfcRelationship*-class, which makes the information distinguishable.

## 5. CONCLUSIONS

The current release 2x of the IFC specifications have limitations in modeling as-designed and as-built information into one project model to support the automation of as-built conditions. It is necessary to extend the current IFC representation without increasing the complexity unnecessarily from both an understandability and a processability point of view.

For the purpose of allowing IFCs to represent both design and as-built information in one project model simultaneously, we propose a solution that allows for fast processing of the IFC model

without increasing the complexity of the IFCs. This was accomplished through the utilization of the *IfcRepresentationContext* concept in IFCs and adding a new attribute to the *IfcRelationship* class.

We have tested this approach in automating the as-built assessment of the construction of a steel frame that we observed on a recent project. We will further test the scalability of this concept by adding different types of design, schedule and as-built information that we are going to collect in an upcoming construction project.

## 6. ACKNOWLEDGEMENTS

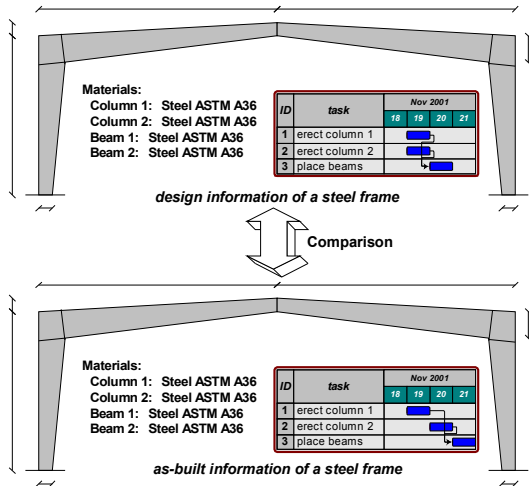
The project is funded by a grant from the National Science Foundation, CMS #0121549. NSF's support is gratefully acknowledged. Any opinions, findings, conclusions or recommendations presented in this paper are those of authors and do not necessarily reflect the views of the National Science Foundation.

## 7. REFERENCES

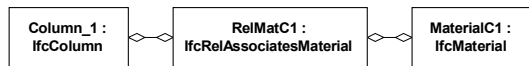
- [1] Cyra Technologies, Inc. *Case Studies*; [http://www.cyra.com/case\\_studies/case\\_studies.html](http://www.cyra.com/case_studies/case_studies.html) (accessed: 06/2002)
- [2] Cyra Technologies, Inc. *Cyra Case Studies: Condition Assessment of Old Suspension Pipeline Bridge*; 1999; <http://www.cyra.com/PDFs/PipelineBridge.pdf> (downloaded: 06/2002)
- [3] The Steel Construction Inst. *CIMsteel Integration Standards, Rel. 2, Vol. 1, Overview*; <http://www.cis2.org/documents/Vol1/SCI-P265on.pdf> (06/2002)
- [4] R. J. Hitchcock, M. A. Piette, S. E. Selkowitz: *Documenting Performance Metrics in a Building Life-cycle Information System*; Building Technologies Depart., Lawrence Berkeley National Lab., Univ. of California, Berkeley, CA, August 1998; <http://eande.lbl.gov/BTP/papers/41942.pdf> (downloaded: 06/2002)
- [5] IAI: *IFC 2x Extension Modeling Guide*; [http://cig.bre.co.uk/iai\\_uk/documentation/Ifc2x\\_EMG/EMG-Base.htm](http://cig.bre.co.uk/iai_uk/documentation/Ifc2x_EMG/EMG-Base.htm) (06/2002)

[6] T. Froese, M. Fischer, F. Goebler, J. Ritzenthaler, K. Yu, S. Sutherland, S. Staub, B. Akinci, R. Akbas, B. Koo, A. Barron, J. Kunz: *Industry Foundation Classes for Project Management - A Trial Implementation*; ITcon Vol. 4 (1999); <http://www.itcon.org/1999/2/paper.pdf> (downloaded: 06/2002)

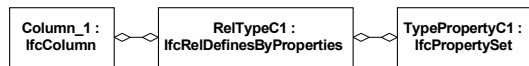
**FIGURES:**



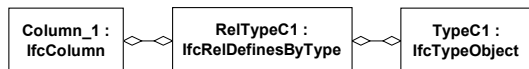
**Figure 1: The design and as-built information of a steel frame.**



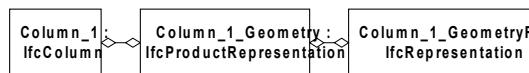
**Figure 2: Representation of material information of a column.**



**Figure 3: Representation of element type information through property sets.**



**Figure 4: Representation of element type information through predefined types.**



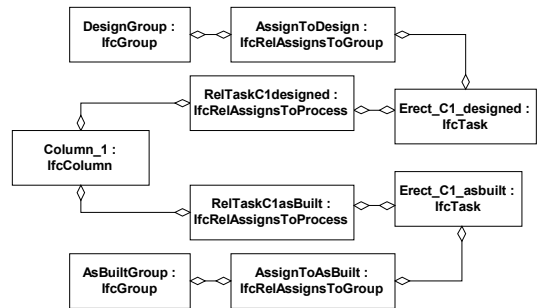
**Figure 5: Representation of geometric information as a group of representation items.**



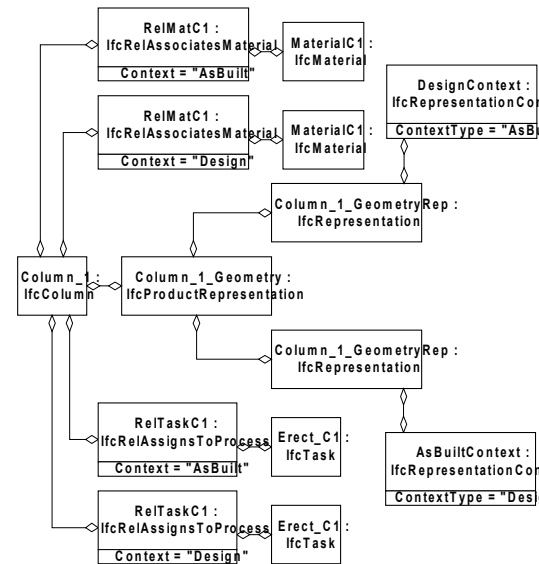
**Figure 6: Representation of geometric information through property sets.**



**Figure 7: Representation of scheduling information to a column.**



**Figure 8: Representation of design and as-built information through groups.**



**Figure 9: The proposed concept.**