Automation in Construction 58 (2015) 176–195

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Validations for ensuring the interoperability of data exchange of a building information model

Yong-Cheol Lee [a,*], Charles M. Eastman [a], Jin-Kook Lee [b]

[a] College of Architecture, Georgia Institute of Technology, Atlanta, GA, USA
[b] Department of Interior Design, Hanyang University, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 23 August 2014
Received in revised form 13 April 2015
Accepted 13 July 2015
Available online 10 August 2015

Keywords:
Building information modeling (BIM)
Design model validation
IFC schema checking
Model view definitions (MVD)
Rule-based checking

ABSTRACT

During the design and construction phases of building projects, domain experts iteratively exchange building information models. One of their goals is to ensure that the requirements and objectives of a proposed project are satisfied. In addition, most building information modeling software currently implements heterogeneous mapping processes in their IFC interfaces that bind their native models to the IFC format. However, such exchanges frequently do not realize intended geometric transformations, project requirements, and required syntactic and semantic conditions in building model data, exacerbating the problem of model integrity and resulting in expensive changes during the construction and operation phases. These problematic issues have been addressed by the development of solid frameworks for validating a building design. This paper surveys six currently available applications for validating building design data and identifies their strengths and weaknesses: The Express Engine's EXPRESSO, the JDSAI™, the EXPRESS Data Manager™, the IFC server ActiveX Component, the IfcDoc, and the Solibri Model Checker®. We also structured the validation processes into three types of tasks: 1) a syntax check of the assurance of compliance with the IFC schema defined by the EXPRESS language, 2) semantic and syntactic assessment in terms of conformity to model view definitions, which consist of concept modularizations, and 3) the validation of design programming requirements that evaluates regulations, project criteria, owner requirements, and functional performance. The purpose of this survey, based on available software that supports the validation of building model data for these three types of interoperability issues, is to integrate diverse checking approaches, as a basis for improving what are now widely distributed efforts.

1. Introduction

With the growing number of client-defined, regulatory, and subordinate codes and standards, the program requirements of a client-provided building have increased in scope and complexity. The proper management of these requirements, satisfied by diverse domain professionals in the design and construction industries [21–23], ensures the successful performance of subsequent procedures and enhances productivity on a project [20]. The program requirements, however, may not be satisfied when domain professionals exchange a building design multiple times. For example, one study showed that a precast concrete domain identified 47 distinct exchanges of a precast concrete building model from the concept design to the fabrication phases, many of which are iterated multiple times [25]. Such exchanges must satisfy the diverse requirements of domain specifications, or they can lead to syntactic problems or programmatic errors in a building design that can hinder model-based interoperability, resulting in considerable overhead [24]. In addition, because of the inconsistent translators of building information modeling (BIM) software solutions, the imported and exported design data are not reliable [10,25–27]. Even though many countries such as the United States, the United Kingdom, Korea, and Singapore specify official BIM policies, they do not yet have standard specifications for the validation of BIM data, which raises the following question: How do government officials evaluate BIM data with regard to its correctness and well-formedness? To validate whether BIM data satisfy their requirements, we need well-organized checking specifications. That is, when project participants iteratively exchange model-based design data [28], they must evaluate the data within the framework of a programmed procedure in which a receiver confirms whether the building information model is satisfied both syntactically and semantically and whether the updated building design conforms to the agreed upon specifications and requirements [25,26]. Despite the various sources of data exchanges and their checking rules, this paper proposes three broad and indispensable validation types: 1) syntactic compliance with the Industry Foundation Classes (IFC) schema defined in the EXPRESS language, 2) semantic and syntactic conformity to model view definitions (MVD) composed of modularized concept descriptions, and 3) design programming requirement checking using an automated rule-based checking application. By following these validation processes, project participants can streamline fragmented design processes and enhance the quality of proposed building designs [1,3,5].

* Corresponding author.
E-mail address: yongcheol@gatech.edu (Y.-C. Lee).

http://dx.doi.org/10.1016/j.autcon.2015.07.010
0926-5805/

In addition, the application of these tasks should improve the accuracy of building information models and the interoperability of BIM data. This paper, the initial phase of our work, will be extended to automate and integrate three types of validation processes.

2. Background

Governments, colleges, and software companies that employ diverse types of BIM authoring tools have spearheaded a strategic initiative to develop a neutral format for the interoperability of a building data model [21,27,29–32]. For building-related interoperability, an IFC data model, accepted as the industrial international standard [33], is used to exchange BIM data among diverse software applications without the supplementary process of supporting an intermediate transition between the native formats of a building design. With this neutral format, each project participant validates a building design iteratively by regarding a number of requirements and compares it with an originally proposed design when sending and receiving a building model from other participants. Even though each professional applies the neutral format, the exchange processes of design data still cause numerous errors pertaining to syntactic, semantic, and design programming requirements [20,26,34]. A 2004 National Institute of Standards and Technology study estimated that inadequate interoperability led to $15.8 billion in annual unnecessary costs [15].

Researchers have devoted considerable effort to using a validation framework for checking the data integrity and accuracy of a building design to ensure its interoperability. For syntactic checking, several tools such as the Express Engine and the EXPRESS Data Manager™ have been developed concurrently with the evolution of the IFC schema [6,17]. Since the syntactic checking area is uncluttered, present tools provide fundamental capabilities to validate IFC instance files according to the EXPRESS language and the IFC schema. For semantic validation, given a reliable MVD, users must implement the semantic validation of an IFC instance file to assure conformance to MVD specifications. The Global Testing and Documentation Server (GTDS), which is a web-documentation platform, provides a service for evaluating an IFC interface and instance file [35]. Using this checking platform, managed by buildingSMART International, software vendors and end-users can evaluate an IFC instance file according to the IFC Coordination View, an agreed subset of the IFC 2 × 3 schema [35]. In addition, they can obtain certification provided by buildingSMART International for facilitating interoperable data exchanges in IFC implementations on software platforms. This MVD checking process, however, is hidden and limited to evaluating an IFC instance file against the IFC Coordination View. That is, software vendors and end-users can neither define requirements for exchanges of their domains nor manually validate IFC instance files against the subsets of MVD. To improve semantic validation, eXtended Process to Product Modeling (xPPM) integrates the development processes of a model view and allows users to check defined model views according to an internal schema such as UNIQUE or WHERE rules [36]. xPPM, however, does not validate a building design itself against model view specifications. Thus, to validate the conformity of an IFC instance file to diverse requirements in a predefined MVD and concepts, each domain expert requires access to an automated rule-checking system. For automated rule-based checking pertaining to design requirements, Eastman et al. surveyed efforts at automated rule-based checking including comparisons of five systems [1] such as FORNAX, developed by the novaCITYNETS Pte. Ltd [38] and the SMARTcodes, developed by the AEC3 and the Digital Alchemy [39]. These checking platforms primarily focus on building codes and laws. The authors have recognized that the efforts devoted to checking a building design are too fragmented for use in checking its accuracy and interoperability. Thus, this paper frames the diverse validation issues of building model data in a uniform format. That is, to identify whether a building model is correctly created and whether it includes required data, domain experts should validate it according to syntactic, semantic, and design programming requirements. This section outlines the causes of three types of evolving interoperability errors in a building information model and their differences.

2.1. Syntactic problems: the IFC schema based on the EXPRESS language

The specifications of the IFC schema are defined using the EXPRESS language (ISO 10303-11), which outlines the data models of STEP (STandard for the Exchange of Product Model Data) [13,14]. The IFC schema must be compliant with the specifications of the EXPRESS language. Fig. 1 describes the specifications of the IfcProject entity of the IFC 2 × 3 TC1 schema. Since the IfcProject entity is one subtype of the IfcObject entity, it must exist in the schema definitions. Moreover, GLOBAL, UNIQUE, and WHERE, which are formal EXPRESS propositions, are properly specified within the descriptions of the IFC schema. In the example of Fig. 1, the statement, "WR31: EXISTS(SELF\IfcRoot.Name)" demonstrates that the IfcProject entity must contain the Name attribute of the IfcRoot entity. If the schema file lacks any of these definitions, the file cannot be correctly parsed and read when importing it into IFC compliant applications.

In addition, a STEP ISO 10303 Part 21 physical file (P21 file) must comply with the IFC schema pertaining to the types and the relationships of entities and attributes [12]. Fig. 2 shows the attributes and the relationships of the IfcProject entity that a P21 file must provide: GlobalId, OwnerHistory, RepresentationContexts, and UnitsInContext. The file can also optionally contain the attributes of Name, Description, ObjectType, LongName, and Phase. Moreover, the IfcProject entity can be inversely related using the attributes of HasAssignments, IsDecomposedBy, Decomposes, HasAssociations, and IsDefinedBy. The #5 instance for the OwnerHistory attribute must be an entity of IfcOwnerHistory, the #22 instance for the RepresentationContexts attribute must be an entity of IfcRepresentationContext, and the #23 instance for the IfcUnitAssignment attribute must be an entity of IfcUnitAssignment. If a P21 file does not satisfy these predefined requirements in the IFC schema, it will have syntactic problems that can cause translation errors, omissions, and other primary defects.

Syntactic errors are generally caused by the heterogeneous binding structures of BIM authoring tools and the mistaken interpretation of IFC requirements. Because the specifications of a neutral format standard are open to interpretation [19], each BIM authoring tool can have the varied binding processes of its own import and export features [10,25], including a heterogeneous mapping procedure that binds its native model into the IFC format. Furthermore, each building modeling software generally supports the heterogeneous subsets of a neutral format standard applicable to its own software solution [16,19]. For example, Fig. 3 represents the potentially heterogeneous IFC binding assignments of the Revit Architecture and the Vectorworks. If a column object built in the Revit Architecture is exported to the IFC format and imported to the Vectorworks, its attributes and relations are redefined by the heterogeneous IFC mapping process of the Vectorworks. In a comparison of two IfcColumn instances exported from two BIM authoring tools, as shown in Fig. 4, their IFC interfaces changed or removed the values of attributes of GUID, Name, ObjectType, and Tag. With regard to INVERSE relationships, IfcColomn in the P21 file exported from the Vectorworks lacks references for both IsTypedBy and IsDefinedBy attributes. Thus, the potentially heterogeneous binding assignments of each BIM software program might result in such changes, which can interrupt the exchange of BIM data and syntactically restrict a P21 file from accurately defining required design information.

2.2. Semantic and syntactic problems: MVD and concept descriptions

MVD, documentation specifying an intermediate data model required for realizing a particular exchange [11,25], consists of concepts, modularized descriptions that are aggregations of one or more associated entities and their attributes [25]. To modularize the development