RESEARCH ARTICLE

Rules and validation processes for interoperable BIM data exchange

Yong-Cheol Lee1,*, Charles M. Eastman2 and Wawan Solihin3

1Department of Construction Management, Louisiana State University, Baton Rouge, LA 70803, USA; 2College of Design, Georgia Institute of Technology, Atlanta, GA, USA and 3Research and Innovation, novaCITYNETS Pte. Ltd., Central Singapore, Singapore

*Corresponding author. E-mail: yclee@lsu.edu http://orcid.org/0000-0002-0040-0894

Abstract

Diverse industries have recognized the significance of a neutral data standard for seamless building information modeling (BIM) data exchanges and adopted commonly agreed schemes such as the industry foundation classes (IFC) schema for enhancing BIM data interoperability. To selectively employ domain-specific data exchange requirements, the AEC-FM (the Architecture, Engineering, Construction, and Facility Management) industries have developed their own schema with a subset of the IFC schema, named a model view definition (MVD). However, because of latent human errors, inadequate MVD definitions, and error-prone data mapping problems, the adoption of IFC for exchanging and sharing BIM data is frequently limited with severe data integrity issues. This challenging situation requires the detailed examination of the limitations of the heterogeneous IFC translation processes of the current BIM authoring and application tools. To accomplish this objective, the authors thoroughly investigated the two MVDs, the Coordination View Version 2.0 and the Construction Operations Building Information Exchange, broadly used as an international standard for developing the IFC translation processes, and then identified their intrinsic requirements/rules and developed rule-based data validation processes. These new intrinsic knowledge of the two MVD specifications will be fundamental foundations to create coherent BIM data exchange systems that are currently scattered and dispersed in heterogeneous IFC translation processes and structures. Newly revealed rules pertaining to these two MVDs have been executed with IfcDoc, which is the IFC documentation tool. This BIM data validation process is expected to not only accurately evaluate the translation processes between BIM native data and IFC ones but also help industry professionals ensure the integrity and the accuracy of their BIM data according to the BIM data exchange standards using the two MVDs. In addition, the outcomes of this research study are expected to bolster the interoperable BIM adoption and offer the values of greater consistency of BIM data exchange.

Keywords: building information modeling (BIM); industry foundation classes (IFC); model view definition (MVD); rule checking; interoperability

1. Introduction

Ensuring interoperability of data exchanges of building information models is one of the imperative pillars that exponentially enhance the interrelated and interdependent work processes of the Architecture, Engineering, Construction, and Facility Management (AEC-FM) industries. As a building project entails growing demands and complex structures that require cooperative participation of various domain experts throughout the entire project, seamless data exchanges and universal data interoperability of a building project would play a pivotal role to create a collaborative working environment and facilitate project success with commonly shared and executable project information. For pursuing the seamless BIM (building information modeling)
data exchanges, the AEC-FM industries have widely employed a neutral file format for BIM data, named the industry foundation classes (IFC) (Sun, Liu, Gao, & Han, 2015), which is standardized as ISO 16739 for enhancing BIM data interoperability among heterogeneous BIM authoring tools and applications in their disciplines. The IFC schema has been continuously revised for establishing open standards of BIM data and its latest version 4.0 has been released in 2013 (Amann & Borrmann, 2016). In addition, throughout the pilot implementations and the end-user requests, the IFC addendums involving necessary improvements and additional information have also released in 2015 and 2016, respectively (Pinheiro et al., 2018). The new version of IFC 5.0, which contains extensive updates regarding civil engineering, such as roads, railways, bridges, and tunnels, is expected to be open to the public in 2020.

With the IFC schema, diverse disciplines have defined their BIM data exchange standards. They have specified their BIM data exchange requirements and defined a subset of the IFC schema, model view definition (MVD), which reflects their distinct data exchange specifications. MVD consists of concepts that are modularized datasets subject to be reused to represent the same BIM data exchange requirements and develop diverse domains’ MVD (Hietanen & Final, 2006). In other words, by aggregating concept descriptions written in the IFC schema, MVD represents BIM data exchange specifications identified in the process of information delivery manual (IDM) (Solihin, Eastman, & Lee, 2015). One concept can be referred several times with heterogeneous settings of constraints and parameters to fulfill necessary specifications of several model views (Lee, Eastman, & Lee, 2015). However, because the development of IDM and MVD requires a consensus of relevant domain experts regarding BIM data exchange processes and their detailed requirements, it is still time consuming and labor intensive. Moreover, since most of processes for defining use cases and mapping IDM information into the IFC schema require manual configurations and translations, they frequently entail latent human errors and data consistency issues, which result in additional cost and delay for rectifying MVD and IFC import/export interfaces (Lee, Solihin, & Eastman, 2019). Even though these limitations have been identified, because of the lack of an automated checking process, industry professionals and software vendors have been devoted to the manual examination of IFC translators to figure out and fix hidden errors according to associated MVD requirements. Furthermore, domain professionals have experienced severe problems in adapting and utilizing the IFC file format for their projects because of geometrical or semantic discrepancy caused by unintended translations between a BIM native model and the IFC schema.

To adequately address these issues, we need a reliable BIM data validation approach that thoroughly evaluates an IFC instance file according to MVD specifications and assesses an MVD-based data mapping process between native BIM data and an IFC instance file. As a means of pursuing the objective, this research study aims to develop an automated BIM data checking process ensuring data interoperability and enhancing neutral format-based data exchange throughout the entire building lifecycle. For accomplishing this goal, the authors investigated a huge number of BIM data exchange processes and specifications of the two MVDs (Section 3). According to the identified data exchange requirements, the authors developed the BIM data validation features that allow experts to define the requirements as a rule set and execute them for ensuring the integrity and accuracy of BIM data according to the MVD requirements (Section 4). The BIM data validation features have been designed based on the knowledge base of the identified intrinsic requirements, encompassing their necessary rules and parameters. These validation features have been established on top of IfcDoc, which has been adopted for model view documentation and validation (buildingSMART, 2018b). To represent the types of identified rules and associated checking processes, this paper entails diverse case studies and discussions regarding the effectiveness and limitations of a proposed automated IFC validation process (Section 5).

2. Literature Review

Diverse BIM authoring tools encompass IFC importing and exporting interfaces that translate native models to IFC files and reverse in conformity with the specifications of Coordination View Version 2 (CV V2.0) and the Construction Operations Building Information Exchange (COBie). Even though industry experts, software developers, and end users have encountered problems with unexpected translations of a geometry and model data, they still do not have a solution to evaluate IFC mapping processes and instance files pertaining to these two MVDs. To ensure the syntactic and semantic accuracy of the IFC translations of BIM authoring tools, an IFC instance file exported from a native BIM model has to be meticulously validated according to the concept descriptions of a given MVD (Pazlar & Turk, 2008). The validation process must incorporate two steps that evaluate IFC instance files imported from and exported to a BIM authoring tool according to syntaxes and semantics of MVD requirements (Ma, Ha, Chung, & Amor, 2006). If any types of errors occur in these two translation processes, end users would receive unintended semantic and geometric transformation and ultimately be reluctant to use the IFC format. These translation issues, which are generally caused by inaccurate specifications of MVD or incorrect development of IFC interfaces on BIM software, have been investigated and detected by manual checking by software developers (Lee, Eastman, Solihin, & See, 2016). Because of the time-consuming and laborious debugging processes, which hinder the BIM applications and seamless BIM data exchanges, academia and industry professionals have recognized the critical importance of an automated validation process for ensuring the integrity of the IFC interfaces (Weise, Liebich, & Wix, 2009).

2.1 Automated checking processes with MVD specifications

There have been several efforts to establish automated checking of the IFC translation processes. One of the validation methods for confirming the IFC mapping processes of the IFC interfaces is Global Testing and Documentation Server (GTDS) provided by buildingSMART International (bsi) (buildingSMART, 2010). As shown in Fig. 1, the web-based validation and certification program helps software developers evaluate an IFC instance file exported from their IFC interfaces according to CV V2.0 (Lee, 2015) and provides a buildingSMART certification showing an approved and compliant IFC interface. This GTDS system, however, consists of complex and hidden processes with semi-automated checking functions and supports the evaluation of an IFC instance file only with regard to CV V2.0 not dealing with other MVDs. In addition, this web-based validation platform has limitations in manipulating intrinsic rules and processes. It also requires the manual management of the certification process. To support certification, software developers and end users must submit IFC instance files to be tested, resulting in a complex process managed by the service bureau. The GTDS-based
certification is thus considerably time consuming and expensive for the approval of IFC interfaces (Solihin et al., 2015), not allowing people to iteratively assess the updated IFC translations and the exported IFC instance files pertaining to an imported MVD (Solihin, Dimyadi, Lee, Eastman, & Amor, 2017). Another feasible application for IFC validation is IFC server ActiveX Component (IfcSvr). IfcSvr-based checking entails the validation datasets of the IFC schema, including the reference checking features (SECOM & VIT, 2000). In addition, this approach supports a user-defined query that allows users to retrieve accurate entities, attributes, and relationships of an IFC instance file, assisting diverse types of rules including inheritance, data accuracy, cardinality, and relational structure (Lee et al., 2015). Even though this IfcSvr-based method has the simple user interface for selectively choosing an IFC instance file, concepts, and exchange models, it has considerable limitations for users in editing rule definitions and implementation processes (Lee et al., 2015). Specifically, to update and edit rules and execution processes for supporting other MVDs with existing concepts, it requires hard coding for extending validation features (Lee, 2015).

2.2 Tools for IFC instance file validation

Weise et al. mentioned XBIM-based BIM model checking using mvdXML. The study demonstrates how the requirements of BIM data exchange are used for automated data checking, but has insufficient information about their rules and validation processes (Weise, Liebich, Nisbet, & Benghi, 2016). Zhang et al. proposed a new MVD checking method using mvdXML and BIM collaboration format (Zhang, Beetz, & Weisen, 2015). The authors adopted mvdXML released by buildingSMART for executing IFC instance files according to MVD. The four use cases of the Rdg BIM Norm and Statsbygg BIM Manual were illustrated in the study to help identify the scope of rule checking, which is readily implementable to support validation without coding. However, as this paper explains, the limitations on implementable agreements of model views should be addressed to formalize checking types and evaluation processes (Lee, Eastman, & Solihin, 2016). The mvdXML document involves the specifications and the relevant rule sets of mvdXML, which can be stepping stones for establishing a robust validation process (Chipman, Liebich, & Matthias, 2012; Weise, 2014). Lee et al. developed the extended process to product modeling, which offers an approach for the integrated development of IDM and MVD (Lee, Park, & Ham, 2013). Even though extended process to product modeling covers the evaluation of MVD itself according to defined data requirements, it cannot validate an IFC instance file against the specifications of an imported MVD. In addition, for the syntax checking of an IFC instance file regarding the IFC schema, several tools such as the Express Engine (Lanning, 2003) and the EXPRESS Data Manager™ (Eastman, Jeong, Sacks, & Kaner, 2009) have been introduced and employed. These applications allow users to validate an IFC instance file according to the syntactic requirements of the IFC schema, not to the semantics of it. Furthermore, the JSDA™, which is a set of libraries, is one possible solution for validating an IFC instance file according to the IFC schema defined in the EXPRESS language (Fowler, 1995; Lee et al., 2015).

2.3 Current MVDs and their applications for ensuring IFC data integrity

Lee et al. proposed the semantic validation process using the precast concrete industry (PCI) MVD in order to ensure the interoperability of BIM data exchange (Lee et al., 2016). The proposed MVD validation process provides modularized checking features that address diverse types of MVD rule sets retrieved from the PCI MVD specifications. The limitation of this application resides in adding checking rules and developing execution processes. They are currently available by software coding. Furthermore, Lee et al. indicated that rules and their execution processes should be formally defined using first-order logic to ensure the consistent environment of BIM data exchange and its validation process (Lee, 2015). With the formalized rule logic and implementation processes, this paper revealed intrinsic rules and formalized processes of MVD validation and showed several examples of IfcDoc-based validation. Hjelseth and Nisbet also indicated the importance of semantic-based model checking, suggesting the examples and potentials in their paper (Hjelseth & Nisbet, 2010).

3. Coordination View and Construction Operations Building Information Exchange

IFC has been widely applied to provide interoperable BIM data environment between heterogeneous BIM authoring and application tools of the AEC-FM industries. To execute the IFC schema, BIM software developers and industry experts have collaborated to develop various subsets of the IFC schema that reflect unique requirements of their BIM data exchanges. The development process of the IFC sub-schema consists of a series of steps for selecting and assembling parts of the specification of the IFC schema, which are required for building IFC-mapping processes with native data of BIM authoring and application software. This IFC sub-schema is referred to as MVD that entails the particular requirements of BIM data exchanges associated with one particular domain. In other words, the specifications of an MVD should be developed to sufficiently translate domain knowledge and BIM data into the IFC schema throughout IFC import and export features of BIM authoring tools. Each exchange model (EM) entails syntactic and semantic requirements needed for its distinct BIM data exchange process. Thus, the scope of MVD specifications depends on the type of industry domains, requiring common agreement of pertinent professionals in a domain (Lee, 2015). BIM authoring tools generally adopt one EM to develop its IFC translation process that binds...
Table 1: Concepts of IFC CV 2.0 (buildingSMART, 2018b).

**Architectural and structural elements (shell and core structure)**
- Wall Standard Case, Wall, Beam, Column, Slab, Stair, Ramp, Railing, Curtain Wall, Roof
- Building Element Proxy

**Building service elements (predominately)**
- Distribution Element, Distribution Flow Element, Distribution Control Element
- Energy Conversion Device, Flow Controller, Flow Fitting, Flow Moving Device
- Flow Segment, Flow Storage Device, Flow Terminal, Flow Treatment Device
- Transport Element, Distribution Chamber Element

**Architectural elements (predominately)**
- Window, Door, Covering, Furnishing Element

**Structural elements (predominately)**
- Member, Plate, Footing, Pile, Fastener, Mechanical Fastener, Discrete Accessory,
  Element Assembly, Reinforcing Bar, Reinforcing Mesh, Tendon, Tendon Anchor
- Project, spatial structure and system elements
- Project, Site, Building, Building Storey, Space, Group, System, Zone, Grid

IFC and native data. Since one industry domain requires numerous and iterative BIM data exchange processes, one MVD consists of several sets of EMs that reflect the interoperable BIM data of each process. EMs are subject to repetitively share same data throughout the design, construction, and facility management processes. For efficient configuration of EMs’ data, a modularized dataset referred to as a concept has been developed and utilized for MVD development. An MVD consists of a series of concepts that involve the specifications and implementation agreements of BIM data exchanges pertaining to one or more entities, their attributes, and relationships. The concept provides the opportunities that diverse industry domains share predefined data exchange specifications and reuse modularized units of EMs. Furthermore, the specifications for implementation written in a document are supposed to be used as a guideline for developing a mapping process between IFC and native objects. Industry domain can define its MVD to reflect unique BIM data exchange processes. For example, the industry-defined MVD such as the precast concrete MVD includes data exchange specifications reflecting what BIM data should be exchanged with what project participants during what phases. This industry-defined MVD is supposed to be used within one industry. However, CV 2.0 and the COBie, which this study targets at, are international standard MVDs that cover various domains. These two MVDs have been defined by bSI to be broadly utilized and referred to as a standard baseline in diverse areas for IFC translation. For example, as shown on the buildingSMART website (buildingSMART, 2018a), “the Coordination View has been the first view definition developed by buildingSMART International and is currently the most implemented view of the IFC schema.” It also says that the primary objective of CV 2.0 is to “allow sharing of building information models between the major disciplines of architecture, structural engineering, and building services (mechanical).” These two MVDs also entail their particular requirements for IFC translation. In this study, the authors aim to reveal their intrinsic requirements and validation rules/processes for providing intellectual knowledge and a significant contribution to diverse areas including each domain that developed, has developed, or will develop their MVD. Because of the significance and unique implications of CV 2.0 and the COBie, the authors conducted the in-depth investigation of a huge number of specifications inside of these two MVDs, which have been used and referred to as an international standard. The detailed information of these two MVDs is described in the following sections.

3.1 Coordination view V2.0

According to the bSI website, the Coordination View has been developed by bSI as a first model view and has been widely used in the AEC-FM industries (buildingSMART, 2018b). CV 2.0 has a primary goal to develop a set of standardized specifications that can be commonly adopted for BIM data exchanges among heterogeneous BIM authoring tools and applications in the architectural, structural engineering, and mechanical industries (buildingSMART, 2018b). To provide the common specifications of BIM data exchanges to several disciplines, CV 2.0 involves various domains’ knowledge and its mapping information such as architectural, structural, building service, and project spatial structure system elements. These specifications are defined in a set of modularized concepts: the total number of concepts for applicable entities and relationships of CV 2.0 is 49 (buildingSMART, 2018b). Each concept encompasses IFC translation definitions of a name, an aggregation, a material, a shape representation, a connection, an assignment, a property set, and others. Even though the bSI provides the baseline schema IfcDoc file including CV 2.0, which can be used for the starting point of MVD development (http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool/ifcdoc-baselines), it does not entail correct concept structure and data for defining identified intrinsic rules and implement evaluation processes. Thus, according to CV 2.0 specifications shown in the following link (www.buildingsmart-tech.org/specifications/ifc-view-definition/coordination-view-v2.0/concepts), the authors redefined most of concepts and their attribute definitions for adding proper rule definitions so that the set of concept templates in IFCDoc V9.4 reflects 49 concept descriptions as shown in Table 1. In addition, since each concept entails requirements for IFC data exchanges, the associated rules that implement the requirements were added into the IfcDoc file. In the IFCDoc application, concepts that define data exchange requirements based on an entity, an attribute, a relationship, or a property set, consist of concept templates and concept blocks. Concept templates that provide a frame and a structure of an associated IFC entity, generally reflect the IFC schema. MVD developers employ the templates to consistently define MVD with predefined IFC schema specifications and structure. The templates can be assigned into an associated entity so that MVD developers can define the detailed data exchange requirements. For building the concept block, underneath the particular entity structure brought from the concept template, MVD developers can
specify unique attributes and their corresponding values according to IDM. Adopting the modularized concept templates facilitates the reuse of concept definitions multiple times for defining numerous entities.

### 3.2 COBie

Since numerous building projects require that facility managers be participated in the early design phase to discuss requirements of end users and operators, the demands and needs regarding facility management should be explicitly identified, reflected, and maintained throughout the design, construction, and facility management (FM) phases. The specifications of FM are defined in COBie as one of the model views for providing the requirements of data exchanges regarding building asset information and FM (East, Nisbet, & Liebich, 2012). COBie is an international standard for data exchanges regarding facility management and project handover information. From the design and construction phases to facility maintenance and operations, all available and relevant data of a facility must be delivered to an owner, a facility manager, or a facility operator. With the agreed specifications of industry knowledge and practices, COBie provides the BIM data exchange requirements defined in a spreadsheet and an MVD. Various BIM authoring tools or applications, such as the Autodesk Revit COBie Toolkit or the FM Systems, support features for exporting their native model information to IFC files or documents according to the specifications of COBie. Such IFC instance files translated pertaining to the COBie specifications ensure that a BIM model contains necessary handover information demanded by facility managers or owners. Thus, iterative evaluation of FM information embedded in BIM models is imperative for domain experts. To validate the IFC instance files according to the specifications of COBie, this paper involves the definition and development of intrinsic rules for COBie on top of the IfcDoc platform.

In this study, the authors redefined the COBie specifications as a format of an IfcDoc file by using the developed rule-checking features of IfcDoc so it can be utilized as validation criteria for assessing an IFC instance file according to the COBie specifications. Figure 2 shows three concept templates of this model view. These concept templates are iteratively referred and composed by 28 EMs represented in Fig. 3 in order to avoid any duplicated definitions of concepts and leverage the efficiency of MVD development.

### 4. Research Method for Validation Processes and Reports

#### 4.1 Primary research questions and methodology

After conducting several research studies regarding BIM data interoperability and doing the thorough literature review, the authors have recognized that even though almost all of BIM authoring tools have adopted the two MVDs to import and export BIM data, there is no reliable and automated process for evaluating their data exchange accuracy and interoperability. To resolve this demanding problem affecting inconsistent BIM data exchanges and considerable project time/schedule overruns, the authors had the following two primary research questions: (i) what are the intrinsic requirements of the most broadly used two MVDs, CV V2.0 and the COBie and (ii) what are the formalized BIM data validation requirements and processes for ensuring the integrity and the interoperability of BIM data according to the identified intrinsic requirements of the two MVDs. The authors believed that the two MVDs entail their intrinsic data exchange requirements that should be satisfied and followed by IFC translation interfaces of BIM authoring tools.

To answer the research questions, the authors conducted an extensive investigation of a large number of the specifications of CV V2.0 and COBie posted in bSI. After having a thorough examination, the authors were able to identify their intrinsic requirements and categorize them into types of rules. The four primary requirements of the two MVDs are (i) uniqueness of an attribute value, (ii) relational reference, (iii) semantic accuracy, and (iv) general syntax checking. These requirements are described in detail in the following sections including case studies of BIM data checking. According to the identified categorization of the requirements, the authors designed checking scenarios and developed validation processes. The implementation on IfcDoc is one of the possible applications utilizing the outcomes of this research study. The authors adopted the checking features of IfcDoc to validate the methodology and the outcomes. These features have been added in IfcDoc and it is open to the public through the bSI website: http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool/ifcdoc-download-page.
4.2 Research processes and implementation

The processes of this study consist of (i) investigation of the two MVDs, (ii) identification and definitions of rules, (iii) development of rules in IfcDoc, and (iv) validation with case studies. To provide an automated checking process and investigate the current IFC translation processes, this research study defines the specifications and associated rule-sets of CV V2.0 and COBie in the IfcDoc tool. This section introduces the validation process and the evaluation report of IfcDoc. The bSI website provides the IfcDoc tool downloadable as a Windows binary file. Constructivity™ has developed the IfcDoc tool for facilitating the development process of MVD using concept templates. In addition to the MVD documentation features, IfcDoc had limited functions for evaluating IFC files according to the EM specifications of user-defined or imported MVDs. To improve its validation features and adopt them to industry data exchange processes, this research team has collaborated with Constructivity™ to develop new rule-checking features on IfcDoc that operate on the diverse entities, attributes, relationships, and data types in IFC. The types of rules were identified from the PCI MVD, which has 94 exchange models. The IfcDoc tool can import and export an IfcDoc file, which is generated based on the given version of IFC to be represented such as IFC Release 2 × 3 or Release 4 baselines. According to concept templates generated, concept definitions and associated rule sets can be organized and assigned to fulfill the specifications of each EM. Thus, IfcDoc allows users to automatically evaluate an imported IFC instance file according to a given EM. The current IfcDoc can support automated rule checking pertaining to 12 EMs of the PCI MVD to ensure the interoperability of the data exchange of precast concrete BIM models. IfcDoc supports diverse types of rules that can be used for evaluating an IFC instance file according to MVD. The checking areas of rule types involve: a data value, an arithmetic constraint, an enumeration value, a value of an attribute (a defined type), data existence, the number of attributes, global/local uniqueness, a data type/subtype, a referential relationship, a conditional constraint, and a fundamental syntax (Lee, Eastman, & Solihin, 2018). Figure 4 shows an example of a precast concrete BIM model manually developed for this study that consists of beams, columns, slabs, topplings, reinforcing bars, meshes, and tendons. Figure 5 represents the primary parts of components and assemblies in a transparent view, which clearly illustrates how the components and assemblies are organized and related: beam details, including a rebar into stirrups into a beam and a lifting anchor in beams, assembly details about a slab with connection joints, a slab assembly with reinforcing bars, and corbel components.

The IfcDoc tool allows to import an IFC instance file for validating it according to the specifications of an EM. In other words, the automated rule-based checking of IfcDoc facilitates the process ensuring the interoperability of BIM data. The following two types of validation reports can be generated to illustrate checking results: color-coded report represented in a user interface (UI) and an HTML report. Figure 6 shows the UI report showing a list of concepts, a relational diagram of each concept shown in the main window, and a validation result of an instance file shown in the separate pane. Instances represented in the separate pane are interactively evaluated according to updates of concepts and instance files for an efficient debugging process. A color-coding method showing a pass in green and a fail in red helps users intuitively recognize validation results.

Figure 7 is an example of the UI validation report showing the detailed information of instances that did not pass the validation. When the “2240 IfcRelAssociateMaterial” instance is selected in the right-hand side pane in Fig. 7, its relational structure is visualized according to a defined concept and color-coded according to validation results. Since this tool shows the detailed information of checking errors for each instance, a user can intuitively keep track of causes and locations of errors pertaining to specific exchange requirements of an EM.

Figure 8 is the validation example of the IfcIdentifier attribute of IfcElementAssembly. According to the specifications of the PCI MVD, the concept of IfcElementAssembly should include one value of Rebar Assembly, Mesh Assembly, Tendon Assembly, or Custom Assembly for the attribute of IfcIdentifier. In Fig. 8, one constraint having Rebar Assembly is flagged in green. This representation indicates the IfcElementAssembly instance of the imported IFC instance file fulfills this condition.

The HTML format validation report is generated for showing detailed results that can be readily shared with other project participants. Figure 9 shows a validation report in the HTML format that shows the FAIL results of PCI-061 validation. The

| Object Type                  | Number |
|------------------------------|--------|
| Beam                         | 13     |
| Column                       | 8      |
| Topping (BUILDINGELEMENTPART)| 2      |
| Assembly                     | 21     |
| Rebar                        | 50     |
| Mesh                         | 20     |
| Tendon                       | 8      |

Since these two MVDs have been utilized by the existing BIM authoring and application tools as a baseline covering broad IFC translation practices, unveiling their intrinsic rules and data validation scope of BIM data exchange specifications is expected to provide a crucial knowledge foundation for building seamless BIM data exchange environment. Specifically, the revealed intrinsic requirements and their rules would be utilized and referenced for MVD developers, industry experts, and software developers, helping to ensure integrity and interoperability of BIM data. Furthermore, revealing data checking scenarios extracted from discovered rule types, developing the rule-checking features, and providing an automated BIM data validation approach would considerably improve BIM data interoperability and facilitate BIM applications in the AEC-FM industries.
PCI-061 concept covers associated materials with an element. In the report, PCI-061: Associate Material to Piece contains the plus (+) sign, which indicates PASS. Other notes attached to this table indicate FAIL. Because of the incorrect references of RelatedObjects of the IFCRELASSOCIATESMATERIAL entity, #5438 instance was reported as FAIL. According to the PCI-061 concept, RelatedObjects of the IFCRELASSOCIATESMATERIAL entity must be IfcDefinitionSelect. Figure 10 illustrates that the #5438 instance, which entails IFCGRIDAXIS, caused the reported error. In this case, IfcGridAxis was incorrectly assigned to the list receiving the material.

To efficiently investigate and track errors and their locations of instances, both UI and HTML checking reports can be referred simultaneously. Figure 11 shows one example of validation pertaining to the IFC spatial structure. The IFC schema must satisfy the following spatial hierarchy structure of a building model: Project > Site > Building > Storey. However, the example in Fig. 11 indicates that the #5456 instance does not comply with the spatial hierarchy defined in the PCI-062 concept. Both reports explicitly show that the element is wrongly assigned to the IFC spatial structure.

5. Rules and Validation Processes of CV V2.0 and COBie MVDs

5.1 Checking rule types

This study has an objective to investigate the types of rules of the two MVDs, CV V2.0 and COBie. An MVD entails the requirements of BIM data exchange that can be utilized as validation rules. To establish concrete and consistent BIM data validation frameworks, rule types embedded in MVD specifications should be explicitly investigated. Validation of an IFC instance file pertaining to MVD requirements covers functional semantics of BIM data exchanges and syntactic restrictions of the IFC schema. To identify the underlying rules of these two MVDs, the authors examined all concepts of CV V2.0 and COBie, which contain requirement specifications. According to given entities, attributes, relationships, properties, and other requirements of the two model views, the authors defined the following four types of rules: (i) uniqueness checking, (ii) reference existence checking, (iii) semantic accuracy checking, and (iv) general syntax checking. The following sections describe validation rule types and their examples.
5.1.1 Uniqueness checking rule
This uniqueness checking type includes the two subtypes of rules: global uniqueness checking and internal data uniqueness checking. First, the all object instances of an IFC instance file must have a globally unique identifier (GUID) including a 22-character length string. The GUID is supposed to be used for conserving a fixed space and reducing overhead needed by IFC object instances during BIM data exchange processes. To evaluate this uniqueness of the GUIDs, it requires to check the following two rules:

- **Uniqueness 1.1**: The 22 number of characters, the string length of the GUID attribute – TYPE IfcGloballyUniqueId = 22
- **Uniqueness 1.2**: The uniqueness of the GUID attribute of each object pertaining to all GUID attributes of all objects of an IFC instance file – A GUID attribute value ≠ one of all GUID values of all object entities

Second, as the IFC schema declares, an attribute can encompass a SET data type in a cardinality setting that requires data uniqueness of each value at a logic syntax level. In other words, enumerated values in one attribute must be unique within the attribute level. Duplicate values are not allowed in the SET data type attribute. This syntactic requirement must be satisfied by all IFC instance files to be seamlessly exchanged and utilized by project participants. This rule can be represented as follows:

- **Uniqueness 2**: If a cardinality setting of an attribute is a SET data type,
  - An attribute value ≠ one of enumerated values within the attribute

5.1.2 Reference checking rule
To represent an BIM object and its relevant information, the IFC schema encompasses diverse entities and their relational structures. CV V2.0 and COBie also entail references and inverse relationships defined within available constraints of the IFC schema. This checking rule requires an attribute to have a reference value as well as refer a correct entity according to the specifications of the two MVDs. In addition, an attribute must be referred by a correct entity by an inverse relationship. Thus, this reference checking rule is defined in the following two ways:

- **Reference 1.1**: An attribute of an entity must have a reference value and refer a correct entity – Attribute:Referencing to → Correct Relationship and Entity
- **Reference 1.2**: An attribute of an entity must be referred by a correct inverse relationship and entity – Attribute:Referenced by ← Correct Inverse Relationship and Entity

5.1.3 Semantic accuracy checking rule
The CV V2.0 and COBie MVDs contain various semantic requirements for BIM data exchange by defining an existence requirement and a particular semantic value for an attribute. Since an MVD generally encompasses semantic specifications for BIM data exchange including an object name, a description, and an object type, this semantic accuracy checking rule plays a pivotal role in executing most of MVD compliance validation. First, according to mandatory or optional specifications of an attribute, this checking rule evaluate the existence of the attribute value.
Second, a type of an attribute value must follow a defined data type including STRING, BINARY, BOOLEAN, and LOGICAL. In addition, this checking rule evaluates an attribute value regarding a semantic accuracy defined in the CV V2.0 and COBie MVDs. Thus, this semantic accuracy checking rule can be defined in the following three ways:

- **Semantic accuracy 1.1**: An attribute declared as mandatory must contain a value
  - Mandatory attribute → A value existence
- **Semantic accuracy 1.2**: A type of an attribute value must comply with one defined in CV V2.0 and COBie
  - A type of an attribute value = A value type in STRING, BINARY, BOOLEAN, and LOGICAL as defined CV V2.0 and COBie
- **Semantic accuracy 1.3**: A value of an attribute must be the same defined in CV V2.0 and COBie
  - A value of an attribute = A value defined CV V2.0 and COBie

### 5.1.4 Syntax checking rule

This rule type deals with syntactic requirements and structural specifications defined in the IFC schema and the EXPRESS language. Since an IFC instance file exported from BIM applications must comply with specifications of the IFC schema, this validation is critical in checking any syntactic or structural errors of an instance file. In addition, because the CV V2.0 and COBie MVDs are the subsets of the IFC schema, an IFC instance file translated based on their data exchange requirement must comply with syntactic constraints of the IFC schema. This rule type is defined as follows:

- **Syntax 1.1**: An IFC instance file must follow the syntactic requirements of the IFC schema
  - An IFC instance file including entities, attributes, properties, relationships → Syntactic requirements of the IFC schema
Validation Results

| Instance File | C:\YongChool\Project_Work\IfcDoc_Extension\Sample files\garage_sym_4.ifc |
|---------------|-----------------------------------------------------------------------|
| Project File  | C:\YongChool\Dropbox\Dropbox\PCI-NBIMS\ifcDOC\051815_IFC2x3TC1_Properties_Baseline.ifc   |
| Model View    | PCI Model View Definition                                               |
| Exchange      | [EM1] Building Concept                                                 |
| Tests Executed| 52                                                                     |
| Tests Passed  | 30                                                                     |
| Tests Ignored | 0                                                                      |
| Tests Percentage| 57%                                                                   |

IfcSlab (31)
- PCI-040: Building Element Aggregation
- PCI-053: Building Element Attributes - [FAIL]
- PCI-054: Building Element Type Assignment
- PCI-063: Placement of Pieces to Building Element
- PCI-068: Extruded Shape Geometry - [FAIL]

IfcSite (1)
- PCI-042: Site Contained in Project
- PCI-096: Site Geometric Curve Representation - [FAIL]
- MVC-880: Site Attributes - [FAIL]

IfcBuilding (1)
- PCI-043: Building Contained in Site
- MVC-893: Building Attribute - [FAIL]

IfcRelAssociatesMaterial (53)
- PCI-061: Associate Material to Piece - [FAIL]

| Instance | Structure       | Constraints |
|----------|-----------------|-------------|
| #5438    | RelatedObjects/IfcElement | +           |
| #5439    |                 | +           |
| #5440    |                 | +           |
| #5441    |                 | +           |

Figure 9: HTML validation report showing FAIL for the concept regarding Associate Material to Piece.

Figure 10: Error reported instance of an Ifc instance model.

- Syntax 1.2: An IFC instance file must follow the syntactic requirements of the EXPRESS language
  - An IFC instance file including entities, attributes, properties, relationships → Syntactic requirements of the EXPRESS language

5.2 Validation processes and IfcDoc implementation

Based on the identified four types of rules, the validation process can be executed and utilized for evaluating an IFC instance file according to the compliance of the CV V2.0 and COBie MVDs and assessing an IFC exporter of a BIM authoring tool pertaining to its translation performance with the two MVDs. Because the authors have identified the unique types of rule-checking processes for PCI specifications and developed the logical checking processes of each checking type, the current IfcDoc tool has been equipped to assess diverse types of syntactic and semantic requirements of MVD. For disseminating the robust automated rule-checking features to academia and industries, the identification of requisite checking types and processes for broadly used MVDs are critically imperative.

Figure 12 indicates one concept and its specifications of CV V2.0 regarding the requirements of IfcWallStandardCase. To adopt the current rule-checking features into CV V2.0, the authors have updated its structure and defined in IfcDoc, but its contents have the same data exchange requirements of the one published by the buildingSmart. The table involves BIM data exchange information regarding entities, attributes, entity
Figure 11: PCI-062 validation reports: the IFC spatial hierarchy structure.

Figure 12: Specifications of IfcWallStandardCase.
relationships, checking required/optimal statuses, rule explanations, and adaptable rule logic. The values in the second column indicate the associated rule logic identified in the PCI validation study. As Fig. 12 shows the specifications of IfcWallStandardCase, which are defined and implemented as the list of rules, IfcWallStandardCase must entail the values for the following attributes for GlobalId, OwnerHistory, RelatingGroup, RelatingMaterial, and MaterialName. The following are the detailed explanations about the items in Fig. 12:

- **eq_cardinality**: It indicates the number of values for an attribute. `asSchema means that an attribute should have the number of values as defined in the IFC schema. `reflective means that the number of values is flexible based on the relationships assigned in an attribute. Zero and One means that the number of a value must be zero and one, respectively.
- **Schema**: It defines whether a value for an attribute is mandatory or not. It has R for required and O for optional.
- **Object breakdown**: This section shows the breakdown structure of an associated entity and the relationships of low-level entities.
- **Concepts**: Each concept template having a vacant information form for an entity, an attribute, a relationship, or a property is assigned into an associated item to define specific requirements.

With the categorized information, this research team has redefined current CV V2.0 and defined the CV V2.0 IfcDoc file, which can be used by various end users and software vendors iteratively evaluate their IFC instance files and IFC translation features. Particularly, BIM software companies that have already obtained CV V2.0 certification from bSI, such as Autodesk, Tekla, and Bentley Systems, are able to constantly utilize this validation process for ensuring IFC translation features.

To examine the reliability and the consistency of the revised CV V2.0 IfcDoc file and its rules, several IFC models including a precast concrete garage, a hospital, and a commercial building were evaluated with regard to the specifications of CV V2.0. The rules of IfcWallStandard in Fig. 12 are defined by concepts on IfcDoc, including rule functions pertaining to the given specifications as shown in Fig. 13 and implemented for checking an IFC instance file to evaluate whether IFC importing/exporting features correctly transform native BIM data into/from IFC ones. Figure 13 shows the validation result represented on the UI and Fig. 14 is the HTML format report showing the checking details.

### 5.2.1 CV V2.0 validation processes

Figure 15 represents the validation report about the reference relationship, which evaluates that an IFC instance has accurate attributes and relationships referring correct entities as defined in CV V2.0. The evaluation results in Fig. 15 show that #62551 instance, IfcStair, lacks the required reference, IfcMaterial, for a material definition. Because CV V2.0 requires the definition of a material relationship, the #62551 IFC instance, which lacks the material relationship, is categorized as an error and represented in red on IfcDoc.

The second type of error is the lack of an object type reference. CV V2.0 requires IfcObject to entail a reference indicating its object type. IfcRelDefinesByType plays a role in assigning an object type to single or more objects. However, as shown in Fig. 16, because the IFC sample model lacks this reference with IfcRelDefinesByType for IfcBeam, the validation report represents several IfcBeam instances in red.

The third type of error is about the semantic accuracy of attributes associated with the lack of an object type reference. CV V2.0 requires IfcObject to contain a reference indicating its
### IfcWallStandardCase (17)

- GUIDs (Operator: And)
- History (Operator: And) [OPTIONAL]
- Naming (Operator: And)
- CAD Layer (Operator: And)
- Spatial Containment (Operator: And)
- Classification (Operator: And) [OPTIONAL]
- Objects Assignment (Operator: And) - **FAIL**
- RelatingGroup Assignment (Operator: And) - **FAIL**
- Material (Operator: And)
- Association Type (Operator: And)
- Product Local Placement (Operator: And)
- Path Connectivity (Operator: And) - **FAIL**

| Instance | Structure                  | Constraints |
|----------|----------------------------|-------------|
| #1082    | IfcRelConnectsPathElements | +           |
| #1107    | *                          | +           |
| #1131    | *                          | +           |
| #1156    | *                          | +           |
| #52935   | IfcRelConnectsPathElements | +           |
| #52960   | *                          | +           |
| #52984   | *                          | +           |
| #53142   | *                          | +           |
| #87917   | IfcRelConnectsPathElements | +           |
| #87942   | *                          | +           |
| #87966   | *                          | +           |
| #88040   | *                          | +           |
| #118752  | IfcRelConnectsPathElements | +           |
| #118777  | *                          | +           |
| #118801  | *                          | +           |
| #118825  | IfcRelConnectsPathElements | +           |
| #118899  | *                          | +           |

- Spatial Structure (Operator: And)

### Slab Type (Operator: And) - **FAIL**

| Instance | Structure | Constraints |
|----------|-----------|-------------|
| #111     | *         | +           |
| #188     | *         | +           |
| #237     | *         | +           |
| #286     | *         | +           |
| #335     | *         | +           |
| #384     | *         | +           |
| #433     | *         | +           |
| #482     | *         | +           |
| #531     | *         | +           |
| #580     | *         | +           |
| #629     | *         | +           |
| #678     | *         | +           |
| #727     | *         | +           |
| #776     | *         | +           |
| #825     | *         | +           |
| #874     | *         | +           |
| #923     | *         | +           |
| #972     | *         | +           |
| #1021    | *         | +           |
| #1070    | *         | +           |
| #1235    | *         | +           |
| #3335    | *         | +           |
| #52930   | IfcRelDefinesByType | +           |
| #53409   | *         | +           |
| #62550   | *         | +           |
| #85834   | IfcRelDefinesByType | +           |

**Figure 14:** Validation report in the HTML format.

**Figure 15:** Lack of the material relationship.
object type. Since MVD is generally developed to reflect the requirements of BIM data exchanges in one particular domain, it is supposed to contain the specifications that need agreed semantics for IFC instances. Thus, CV V2.0 has various semantic requirements that an IFC translator should map accurate semantics into attributes of relevant IFC instances. Figure 17 shows the validation report containing the semantic errors regarding a representation type of IfcBeam. According to CV V2.0, IfcBeam should only adopt one of Curve2D or SweptSolid. Curve2D can be represented by IfcTrimmedCurve or IfcPolyline and SweptSolid can be represented by IfcExtrudedAreaSolid or IfcRevolvedAreaSolid. The IfcBeam instances in the test IFC file are represented in red because of the lack of the correct semantics for ContextOfItems of IfcShapeRepresentation.

The general syntax validation is also one type of error identified in this experiment. Since this syntactic integrity of an IFC instance model pertaining to the targeted IFC schema is mandatory, this validation plays a pivotal role to ensure fundamental interoperability of an IFC instance file. IfcDoc utilizes the IFC 2×3 or IFC 4 baselines that consist of the specifications of the corresponding IFC schema and the reusable concept templates of MVDs. Thus, when IfcDoc validates an IFC instance file, it is automatically evaluated pertaining to the underlying IFC schema specifications.
5.2.2 COBie validation processes

The existing rule-checking features were adopted to define the specifications of COBie. Without any extensions of new rule types, COBie and its associated rules were fully defined in IfcDoc. The underlying rule types identified for evaluating an IFC instance file according to the specifications of COBie are uniqueness, semantic accuracy, relational references, and general syntax checking.

Figure 18 shows the definition of uniqueness rule. The metadata concept embedded in COBie requires IfcRoot to have a unique value for the name attribute. This unique value in a BIM model should be maintained throughout an entire project so that it can help establish distinct data references of internal and external data. Thus, this rule-checking feature helps a project participant maintain unique names of objects and ensures them while iteratively exchanging BIM data.

Figure 19 shows the accuracy checking of semantics of attributes. Since COBie refers to the taxonomy of the OmniClass classification, IfcClassification should entail associated values for the following attributes: Source, Name, and Tokens parameters (East, 2007). According to the Omniclass classification and the relevant resources, diverse values can be used for evaluating the IFC instance file.

A relational reference checking shown in Fig. 20 is one of the primary rule types required to define and evaluate the specifications of COBie. This example represents that a building level has relationships with assigned spaces. IfcObject referring spaces can be inversely related to IfcRelAggregates that has a parameter for RelatingObject referencing IfcBuildingStorey in this concept. As a part of a hierarchical composition in the IFC schema, the relationship between IfcSpace and IfcBuildingStorey is defined by this concept description. In addition, the general syntax checking is also imperative to comply with the fundamental syntactic requirements of the IFC schema.

6. Conclusion

Interoperability has been critically recognized as one of the critical pillars for establishing a robust foundation of BIM-based design, construction, and facility management. Even though the IFC schema allows industry experts and researchers to flexibly define their own MVDs for representing BIM data exchange of their domain knowledge, the AEC-FM industries have been struggling to adopt and utilize the IFC neutral format because of latent data exchange issues and unexpected problems. For supporting an interoperable BIM data environment in the AEC-FM industries, ensuring translated IFC data is imperative. In this study, the new processes of automated MVD validation pertaining to CV V2.0 and COBie were proposed and evaluated with the various case studies. In addition, the outcomes of this study
reveal the intrinsic rules for defining the requirements of concepts embedded in MVDs and for implementing the automated validation process. Since these two MVDs provide the standard reference baselines of BIM data exchange requirements for major disciplines, identified intrinsic rules and developed validation processes will be critical knowledge foundation for them to build their IFC translation interface according to these two standard MVDs and develop their own industry MVDs on top of these two MVDs. Since diverse BIM applications and authoring software use these two MVDs, the redefined CV V2.0 and COBie in IfcDoc are expected to be used for accelerating the BIM adoption and offer the values of greater regulatory consistency. Particularly, the proposed validation method and the redefined IfcDoc files can significantly enhance the current time-consuming certification process that evaluates the IFC translation features according to CV V2.0. In terms of COBie, ensuring the accuracy of handover specification data aggregated throughout the entire project is invaluable for facility managers who manage the complete building over the decades (Roper & Payant, 2009). According to the demands of facility managers, the automated evaluation method for checking the well-formedness of BIM models should be implemented to pass the collected handover information over to facility managers. Thus, the revealed rule sets and checking processes for executing COBie in the IfcDoc platform are expected to be greatly helpful for diverse domain professionals to evaluate semantic errors, technical problems, and translation issues.

The authors believe that unraveling intrinsic rules of CV V2.0 and COBie and establishing the robust validation process with the identified rules and processes are expected to bring a significant impact on researchers and industries around the world that study, develop, and use diverse MVDs. The redefined IfcDoc files for CV V2.0 and COBie specifications can be easily shared with the public and repetitively executed by industry experts and software companies. In specific, as the IFC schema has developed to encompass the newly added information and to rectify the errors reported from the public, MVD should also be updated and revised according to the needs or demands of new data exchange processes or industry uses. Thus, the veiled rule types and automated checking method can be flexibly edited and reused by end users to update the embedded rules and implement them accordingly. Hence, the outcomes of this research study are expected to bolster the adoption of BIM and its accessories and facilitate collaboration between practitioners at the government organization and the industry.

Currently, the limitations reside in that users who are not familiar with IFC can have learning curve to edit and implement this validation method. In addition, the IfcDoc-based evaluation of an IFC instance file pertaining to the requirements of MVD has the following challenges that should be overcome: checking sequence management of multiple rules, mandatory/optional evaluation of an attribute, and inconsistent concept organizations of MVD. In addition, one of the critical issues is the continuous revision of the IFC schema according to the new business and the requests of domain industries, which requires updating corresponding MVDs and their validation processes accordingly. It is expected for relevant professionals to manually reflect the changes of the new or revised versions of the IFC schema on the MVDs and validation processes. Another main challenge of the MVD-based IFC validation approach is its extensibility. Even though MVD is the subset of the IFC schema, the scope of MVD definitions and validation is broad and sometimes specific, causing multiple interpretations of data exchange requirements and diverse ways of references. This unclear MVD scope can lead to prevent reusing predeveloped MVD concepts and ultimately their validation processes. In particular, the main issue of the MVD validation is the heterogeneous structure and inconsistent definitions of diverse MVDs of industry domains, making the broad and abstract scope of evaluation of an IFC instance file. This demanding situation brings open and various interpretations in defining MVD rules and implementing validation processes according to the defined MVD specifications. To ameliorate these limitations and challenges, the development of a robust rule definition, classification, and formal structure for MVD definitions are inevitable to academia and industries.

References
Amann, J., & Borrmann, A. (2016). Embedding procedural knowledge into building information models: The IFC procedural language and its application for flexible transition curve representation. Journal of Computing in Civil Engineering, 30(5), https://doi.org/10.1061/(ASCE)CP.1943-5487.0000592.
East, W. (2007). Construction operations building information exchange (COBie). http://www.dtic.mil/dtic/tr/fulltext/u2/a491932.pdf, Last accessed on 4 June 2018.

East, W., Nisbet, N., & Liebich, T. (2012). Facility management handover model view. Journal of Computing in Civil Engineering, 27(1), 61–67, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000196.

Eastman, C. M., Jeong, Y. S., Sacks, R., & Kaner, I. (2009). Exchange model and exchange object concepts for implementation of national BIM standards. Journal of Computing in Civil Engineering, 24(1), 25–34, https://doi.org/10.1061/(ASCE)0887-3801(2010)24:1(25).

Fowler, J. (1995). STEP for data management, exchange and sharing. City Twickenham, UK: Ta Technology Appraisals Ltd. ISBN 1871802369, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.2934294&rep=rep1&type=pdf, Last accessed on 18 May 2018.

Hietanen, J., & Final, S. (2006). IFC model view definitions for building information modeling. Advanced Engineering Informatics, 20(1), 36–38. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.2051&rep=rep1&type=pdf, Last accessed on 18 May 2018.

Hjelseth, E., & Nisbet, N. (2010). Exploring semantic based model checking. In Proceedings of the 27th CIB W78 International Conference (Vol. 54). http://itc.scix.net/data/networks/att/w78-2010-54.pdf, Last accessed on 11 June 2018.

Lanning, C. (2003). Express engineer user guide, Express engine project. http://exp-engine.sourceforge.net/, Last accessed on 11 March 2018.

Lee, G., Park, Y. H., & Ham, S. (2013). Extended process to product modeling (xPPM) for integrated and seamless IDM and MVD development. Advanced engineering informatics, 27(4), 636–651, https://doi.org/10.1016/j.aei.2013.08.004.

Lee, Y. C. (2015). Rule logic and its validation framework of model view definitions for building information modeling. https://smartech.gatech.edu/bitstream/handle/1853/54430/lee-dissertation-2015.pdf?sequence=1&isAllowed=y, Last accessed on 4 June 2018.

Lee, Y.-C., Eastman, C. M., & Lee, J.-K. (2015). Validations for ensuring the interoperability of data exchange of a building information model. Automation in Construction, 58, 176–195, https://doi.org/10.1016/j.autcon.2015.07.010.

Lee, Y.-C., Eastman, C. M., & Solihin, W. (2016). An ontology-based approach for developing data exchange requirements and model views of building information modeling. Advanced Engineering Informatics, 30(3), 354–367, https://doi.org/10.1016/j.aei.2016.04.008.

Lee, Y. C., Eastman, C. M., & Solihin, W. (2018). Logic for ensuring the data exchange integrity of building information models. Automation in Construction, 85, 249–262, https://doi.org/10.1016/j.autcon.2013.08.004.

Lee, Y.-C., Eastman, C. M., Solihin, W., & See, R. (2016). Modularized rule-based validation of a BIM model pertaining to model views. Automation in Construction, 63, 1–11, https://doi.org/10.1016/j.autcon.2015.11.006.

Lee, Y. C., Solihin, W., & Eastman, C. M. (2019). The mechanism and challenges of validating a building information model regarding data exchange standards. Automation in Construction, 100, 118–128.

Ma, H., Ha, K. M. E., Chung, C. K. J., & Amor, R. (2006). Testing semantic interoperability. In Proceedings of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering (pp. 1216–1225), Montréal, Canada, http://ww.w.iibnet.de/daten/iconda/cib20898.pdf, Last accessed on 22 May 2018.

Paziar, T., & Turk, Ž. (2008). Interoperability in practice: Geometric data exchange using the IFC standard. Journal of Information Technology in Construction (ITcon), 13(24), 362–380. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.2934294&rep=rep1&type=pdf, Last accessed on 4 June 2018.

Pinheiro, S., Wimmer, R., O’Donnell, J., Muhic, S., Bazjanac, V., Maileb, T., Frischb, J., & Treeck, C. (2018). MVD based information exchange between BIM and building energy performance simulation. Automation in Construction, 90, 91–103, https://doi.org/10.1016/j.autcon.2018.02.009.

Roper, K., & Payant, R. (2009). The facility management handbook, Amacom, ISBN: 978–0814413807.

SECOM, and VIT. (2000). IFC model server development project. http://cic.vtt.fi/projects/icserv/download.html#IFCServ, Last accessed on 13 March 2018.

Solihin, W., Dimyadi, J., Lee, Y. C., Eastman, C., & Amor, R. (2017). The critical role of the accessible data for BIM-based automated rule checking systems. In Proceedings of the Joint Conference on Computing in Construction (JC3) (pp. 53–60), https://doi.org/10.1016/j.autcon.2018.08.004.

Solihin, W., Eastman, C. M., & Lee, Y.-C. (2015). Toward robust and quantifiable automated IFC quality validation. Advanced Engineering Informatics, 29(3), 739–756, https://doi.org/10.1016/j.aei.2015.07.006.

Sun, J., Liu, Y. S., Gao, G., & Han, X. G. (2015). IFCCompressor: A content-based compression algorithm for optimizing industry foundation classes files. Automation in Construction, 50, 1–15, https://doi.org/10.1016/j.autcon.2014.10.015.

Weise, M. (2014). mvdXML requirements and examples: Review of a standardized format to define and exchange model view definitions with exchange requirements and validation rules. https://github.com/BuildingSMART/mvdXML/tree/master/mvdXML1.1, Last accessed on 9 January 2019.

Weise, M., Liebich, T., Nisbet, N., & Benghi, C. (2016). IFC model checking based on mvdXML 1.1. eWork and eBusiness in Architecture, Engineering and Construction, 19. http://nrl.northumbria.ac.uk/30286/1/rcpmpm2016_aec3_northumbria_v20160513.pdf, Last accessed on 9 January 2019.
Weise, M., Liebich, T., & Wix, J. (2009). Integrating use case definitions for IFC developments. *eWork and eBusiness in Architecture and Construction*. London: Taylor & Francis Group, 637–645, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.467.9870&rep=rep1&type=pdf, Last accessed on 23 April 2018.

Zhang, C., Beetz, J., & Weisen, M. (2015). Interoperable validation for IFC building models using open standards. *Journal of Information Technology in Construction*, 20(ECPPM-2014), 24–39, https://pure.tue.nl/ws/files/3969218/2763599124870.pdf, Last accessed on 17 June 2018.