Advanced Engineering Informatics 30 (2016) 354–367

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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

Full length article

An ontology-based approach for developing data exchange requirements and model views of building information modeling

Yong-Cheol Lee *, Charles M. Eastman, Wawan Solihin

College of Architecture, Georgia Institute of Technology, Atlanta, GA, USA

A R T I C L E I N F O

Article history:
Received 31 July 2015
Received in revised form 1 April 2016
Accepted 20 April 2016

Keywords:
Building Information Modeling (BIM)
Interoperability
Ontology
Information Delivery Manual (IDM)
Model View Definition (MVD)
Industry Foundation Classes (IFC)
Information technology
Rule checking

A B S T R A C T

Each domain industry requires the detailed specifications for sharing and exchanging building information models throughout the design, construction, and operation phases. Industry Foundation Classes (IFC) Model View Definitions (MVDs) specify required information for exchanges of building model data among building project experts. The data involves the identification of model semantics shared by two or more applications. However, since no robust standard for defining building semantics and requirements for data exchange has been agreed upon, information embedded in domain-specific MVDs are generated separately and are vague in scope, which results in a lack of consistency. In addition, the Information Delivery Manual (IDM) that includes exchange specifications needed for each exchange process of a product model is manually defined in a paper-based document. Because there is no clear logical link between the units of information in the exchange requirements of an IDM, and those of MVDs, the mapping that translates requirements of an IDM into ones of an MVD is open to various interpretation, without semantic and logical consistency. Such challenges might result in redundant requirements and rules for data exchange that are not supposed to be handled in the process of MVDs. To ameliorate this situation, this research proposes the new approach of formalizing domain knowledge and defining accurate data modules for model views. To achieve this goal, the authors employed ontological principles for generating an IDM for the precast concrete domain and for linking its MVD with formal information models. The formalized structure of domain knowledge is expected to support defining explicit data modules and developing manageable relationships among entities using semantic reasoning so that domain professionals and software vendors can identify the intents of the requirements of mapped MVDs and keep track of mapping problems. Moreover, to integrate IDM and MVD development processes, the ontology-based IDM is parsed and translated from OWL/XML to mvdXML, which automatically generates MVD documentation in the IfcDoc tool.

1. Introduction

Domain industries have a need of a neutral file format that encompasses its requirements and design information for numerous exchanges supporting full use of a building model. Industry Foundation Classes (IFC), widely used in architectural, engineering, construction, and facility management (AEC–FM) and registered with the International Standardization Organization as ISO 16739, is the standard data schema provided by buildingSMART International (bSI). The objective of this schema is to allow the definitions of explicit and semantic building data for supporting their interoperability. Using this schema, each discipline defines an Model View Definition (MVD) that describes the subsets of the schema required to transmit product and project information supporting the needs of particular data exchanges [25]. In addition, an MVD includes the model specifications of data exchanges so that software vendors and domain professionals can apply the binding process of an MVD to the IFC interface of their software products [12,17]. An MVD consists of modularized concepts with predefined specifications and rule sets to be reused [10,17,19]. A concept depicts required entities, attributes, properties, and relationships using a relational structure and a constraint. A detailed description of these requirements appears in the Information Delivery Manual (IDM), translated into supposedly implementable specifications for software vendors [12]. However, since experts developing an IDM and an MVD have no robust standard that either specifies the requirements of domain knowledge or categorizes the definitions of concept modules, the documentation of an MVD includes redundant and ambiguous specifications and lacks consistency in its implementation. In addition, a

---

* Corresponding author.
E-mail address: yclee.it@gmail.com (Y.-C. Lee).

http://dx.doi.org/10.1016/j.aei.2016.04.008
1474-0346/© 2016 Published by Elsevier Ltd.

manually developed IDM in a paper-based document such as an EXCEL table hinders the sharing of information among relevant industries. In particular, this impediment results in a discontinuous and error-prone data interpretation and translation from an IDM to an MVD, which eventually delays the development of an MVD. This procedure does not allow users to confirm the composition or the definitions of an MVD pertaining to the contexts defined in an IDM because no common set of terms and their meaning exists. Furthermore, inconsistent definitions of concept documents often lead to confusion among software vendors developing IFC interfaces on their Building Information Modeling (BIM) tools. For example, while every precast slab can be defined initially as the object of a single building element with attributes associated with a slab object, as it evolves toward fabrication, the slab becomes an assembly made up of structural elements similar to hollow core, reinforcing, and pre-stressed tendons and topping, with their own shapes and attributes. This assembly has a many-to-many mapping between components, with the topping shared among hollow core beams. It is the responsibility of the detailer to validate structural, spatial, and other design equivalences between their representations. Several parts of a product model have multiple representations and attributes and the distinction between these representations must be distinguished to properly exchange the desired data needed by downstream users. To achieve these goals, this paper proposes an approach to developing an ontology-based IDM and MVD that can provide a method of formalizing domain knowledge and integrating the processes of IDM and MVD using ontological principles.

2. Background

The National Building Information Modeling Standard (NBIMS) defines the procedural steps for developing an MVD as shown in Fig. 1. The NBIMS defines standard semantics and processes for building information exchanges, which supports providing interoperable domain contexts [5]. This specified process of MVD development consists of three core steps: an IDM, an MVD, and implementation. Domain professionals in an industry define an information exchange template that contains functional specifications for each exchange process. This template is applied to an IDM that describes required entities and their attributes. This IDM is translated into an MVD using a subset of the IFC schema. An MVD has the set of specifications for data exchange implementation and supports the IFC binding process of the native objects of BIM authoring tools [10,17]. The structure of an MVD consists of reusable concept modules for representing domain knowledge efficiently and consistently [10,17,27]. The concept is a modular-based knowledge unit that helps software vendors properly encode a building model in the IFC translation interfaces of their BIM authoring tools [11]. For example, concepts in a precast concrete domain include global properties such as cardinal line, seam connectors, cambered elements, editable parametric elements, and spatially more complex elements using boundary representation (B-rep). Concepts of a product model are composed iteratively for fulfilling diverse requirements of an MVD so that the exchange requirements of a particular domain can be represented accurately and consistently by a modularized context. In addition, a modular framework encourages the re-usability of existing specifications for the model view development of other disciplines. A concept that describes the required contexts for exchanges of a building model consists of the instructions of implementation, the requirements of attributes, and the shared structure of the IFC schema.

Fig. 2 depicts a diagram and an implementation agreement of a concept pertaining to a precast slab aggregation. This concept shows an aggregation of multiple parts including double tees, hollow core, or precast concrete slabs. The diagram illustrates how entities and their attributes are connected and what values and types are required for them. In other words, this relational data structure in Fig. 2 requires that the aggregation of a precast concrete slab and beam should be represented by IfcRelAggregates using two attributes RelatingObject and RelatedObjects connected to IfcSlab and IfcBeam. The IfcSlab entity has two types of attributes: explicit and those associated through Express's inverse attributes. The explicit attributes such as GlobalId, OwnerHistory, ObjectType, and Representation exist in an IFC instance file called Part 21 physical (P21) file and the inverse attributes such as HasAssociations, IsDecomposedBy, and Decomposes are identified relationally across multiple entities. The binding document also includes implementation agreements that define specific types and values according to the attributes of IfcSlab and IfcRelAggregates entities. These specifications define that an IfcSlab instance of a P21 file must contain GlobalId, OwnerHistory, ObjectType, ObjectPlacement, and Representation: Particularly, ObjectType must be Slab or Precast Slab; ObjectPlacement must be defined by IfcLocalPlacement using the PlacementRelTo attribute, which refers the relative location of a site and a building; and Representation should be Boundary Representation using IfcManifoldSolidBrep as a reference of an Item attribute of IfcShapeRepresentation. With regard to IfcRelAggregates, RelatingObject should refer to a slab entity and RelatedObjects should connect to components in the slab. These specifications are supposed to be used to transform native objects of BIM authoring tools into a P21 file.

3. Problems in current practice

IDM data, generally configured in a spreadsheet in current practice, are too fragmented and vague to define the requirements of an MVD. In particular, IDM developers have no explicit baseline for defining data exchange requirements pertaining to applicability of IDM data to an MVD and a binding process for IFC interfaces. From the initial phase of defining the scope and business rules for specific domain knowledge, more explicit criteria and formal specifications based on the domain knowledge of a field are needed for the development of an IDM and an MVD. Even though we have a buildingSmart International (BSI) IDM process and a NBIM MVD development process, we still lack their base definitions and detailed processes to exchange specifications. One way of considering the issues is to associate baseline model views as derived from the LOD (Level of Development) for each of the exchanged objects. The absence of a standard causes several problems such as inconsistency in developing an IDM and an MVD. Several efforts to define an MVD have been devoted to using a manually defined data table for generating an IDM and an MVD. Table 1 represents an EXCEL sheet for the IDM of the precast concrete domain. The information for precast concrete in the table is categorized into an information group, an information item, an attribute set, an attribute, and availability. The example defines various attributes and their use of slab systems for exchange processes. These slab system data are translated into an MVD to be represented in the IFC schema. However, a current method that collects exchange requirements in an EXCEL table and generates an MVD based on table-based information raises several concerns pertaining to knowledge formalization for IDM and process integration for the development of an IFC MVD. This section illustrates four concerns about a current method for the development of an IDM and an MVD.

3.1. Manually collected information on the table

Industry professionals determine requirements for data exchange and provide sources for their IDM. As shown in Table 1, a manually developed IDM in an EXCEL table may not be

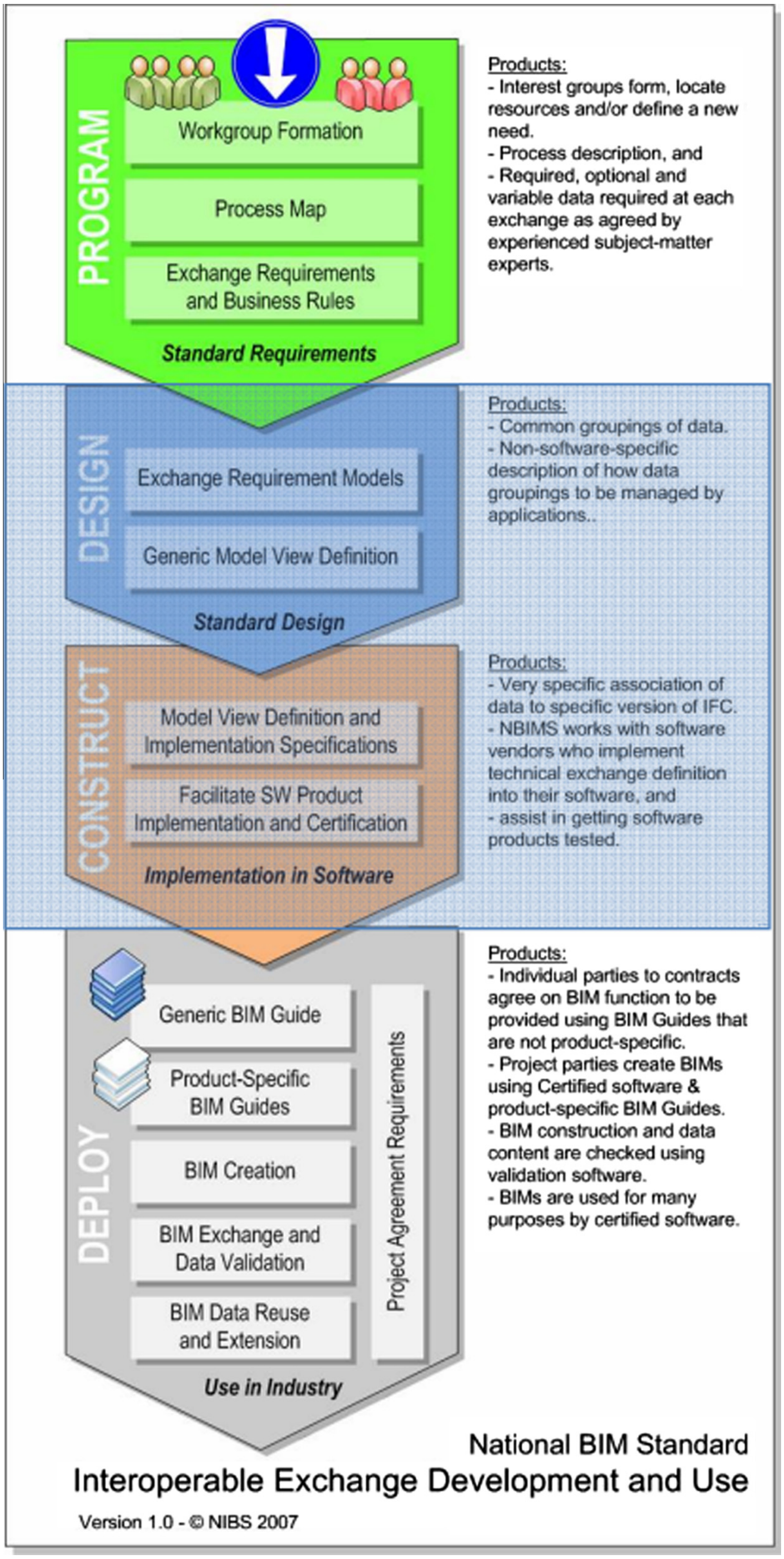
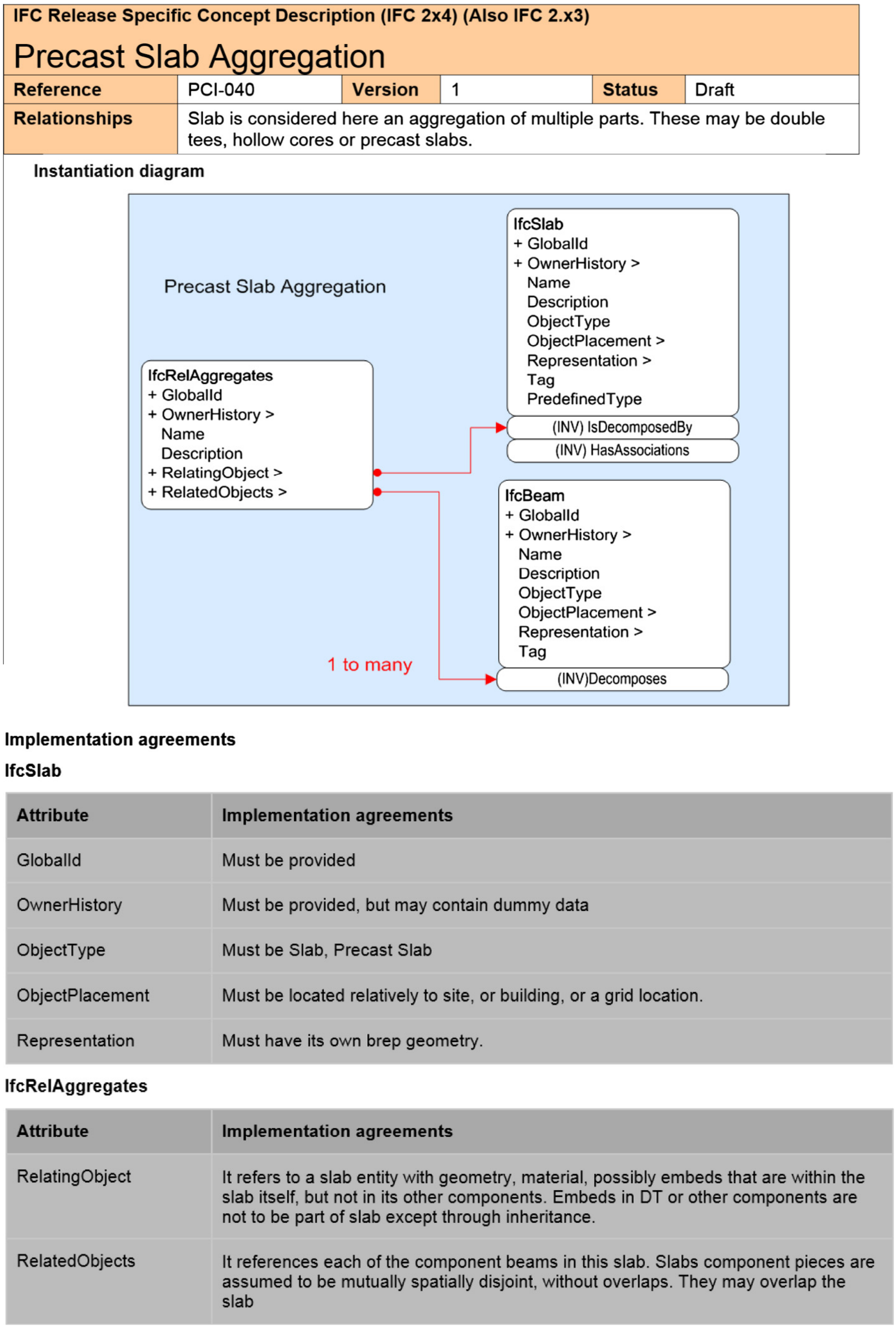


Fig. 1. Process of NBIMS MVDs development.

manageable nor efficient and thus often fails to reflect updates. In particular, the present practice does not allow users to easily categorize information items and systematically determine a data hierarchy. This manual process can cause human errors and omissions in collecting information. Inconsistent definitions of concept documents can result in confusion among software vendors developing their IFC interfaces. In addition, this way of collecting information prevents the sharing of data among MVD developers in various industry domains. Sharing IDM classifications is imperative because relevant industries can use the same terminology and data



IFC Release Specific Concept Description (IFC 2x4) (Also IFC 2.x3)

Precast Slab Aggregation

Reference	PCI-040	**Version**	1	**Status**	Draft
Relationships	Slab is considered here an aggregation of multiple parts. These may be double tees, hollow cores or precast slabs.				

Instantiation diagram

Implementation agreements

IfcSlab

Attribute	Implementation agreements
GlobalId	Must be provided
OwnerHistory	Must be provided, but may contain dummy data
ObjectType	Must be Slab, Precast Slab
ObjectPlacement	Must be located relatively to site, or building, or a grid location.
Representation	Must have its own brep geometry.

IfcRelAggregates

Attribute	Implementation agreements
RelatingObject	It refers to a slab entity with geometry, material, possibly embeds that are within the slab itself, but not in its other components. Embeds in DT or other components are not to be part of slab except through inheritance.
RelatedObjects	It references each of the component beams in this slab. Slabs component pieces are assumed to be mutually spatially disjoint, without overlaps. They may overlap the slab

Fig. 2. A binding document defined for an aggregation of a precast slab and beams.

structure interoperability in data exchange. Equally an issue is the arbitrary inclusion of "nice to have" attributes that makes extra work for front end detailers, without significant downstream benefits.

3.2. Separately defined attributes for each entity

A table-based dataset in the existing IDM does not illustrate how entities and attributes are related, what entities share the same attributes, and what types of relational entities are applied. For example, if several columns need the same representation type such as an extrusion and the same material attributes, such entities and their attributes should be specified in the table iteratively. However, an EXCEL table often precludes users to identifying such shared items. Such capability is required to generate a modularized concept description that should represent all of requirements as a module unit but not depict data redundantly. Since an MVD is composed of such modularized concepts, how a concept is defined greatly affects the development of an MVD. This information for an IDM should be drawn on a knowledge map that can represent the entire relationships and their hierarchy of embedded data. If specifications of data exchange are unorganized, software companies might end up with employing heterogeneous binding processes that map their native objects onto the IFC schema in a different way.

3.3. A situation hard to verify the correctness and redundancy of data

Data in the current IDM practice is often not well managed and organized because of problems in inconsistent data hierarchy and structure. In addition, the current method precludes users from

Table 1
An EXCEL table for PCI MVDs [12].

Information group	Information items	Attribute set	Attributes		P_EM4	A_EM4
Primary CIP or steel structure – assemblies	Slab systems, building cores					
		Geometry	Extruded shapes or solid forms	Required	R	R
				Function	F	F
				Accuracy	P	C
		Material	Material type	Required	R	R
			Quantity	Required	R	O
		Finishes	Geometry	Function	V	F
				Accuracy	P	C
			Surface treatments	Required	R	O
		Assembly relations	Part of building	Required	O	R
		Association relations	Implements structural objects	Required	O	
		Nested relations	Contains components	Required		R
		Connection relations	to precast	Required		O
			to CIP	Required		O
			to Steel	Required		O
		Meta data	Author, version, date	Required	R	R
			Approval status, date	Required	R	R

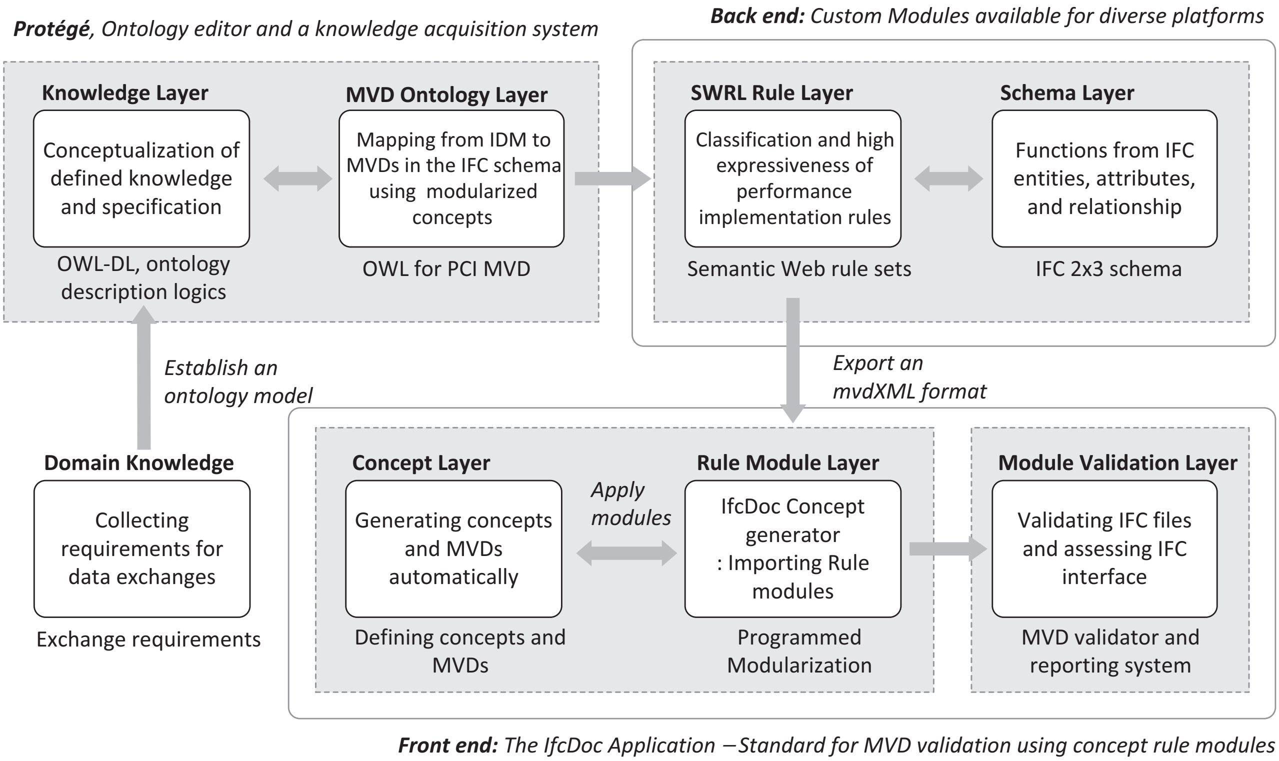


Fig. 3. Architecture of ontological IDM model and MVDs rule definition.

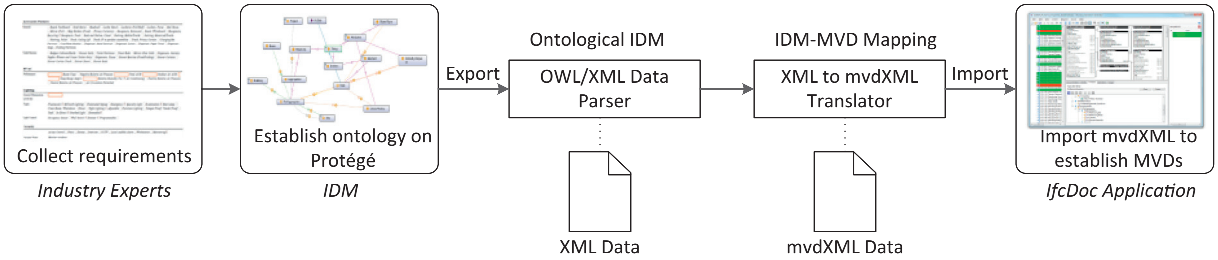


Fig. 4. Data flow for ontological IDM and MVD development processes.

confirming the composition and definition of an MVD pertaining to the contexts defined in an IDM. These extraneous data are identified in the MVD development process, which results in rearranged data and restructured data hierarchy, both of which are needed to guarantee consistency. This error-prone method can impede data translation for an MVD, which is expected to provide explicit

Table 2
Definitions of classes.

Class	Description and instances
Entities	
Project	Activities for design, engineering, and construction leading towards a product
Element	
Beam	A structural object designed to carry loads between points of support
Column	A structural object of slender form that transmits to its base the forces
Slab	A construction component enclosing a space vertically supporting walls and columns
Wall	A construction component subdividing spaces and fulfilling a load
Discrete accessory	Various accessories Included in elements
Building element part	A component as a subordinate part of a building element
Reinforcing element	Bars, wires, strands, meshes, and tendons
Covering	Finishing and treatments of the surfaces
Fastener	Fixing parts to connect or join elements
Spatial element	
Building	Main purpose of construction designed to stand permanently
Building story	Horizontal aggregation of spaces
Site	A defined area of land for construction
Attributes	
Global ID	An unique identifier for software world
Owner history	History and identification of information
Name	A string value for specific name
Object type	A string value defining type of an object
Local placement	Defining the relative placement of a product
Representation	General concept of representing product
Predefined type	Enumerations of defined types
Is decomposed by	Reference to the decomposition relationship
Decomposes	Reference to the decomposition relationship
Has assignments	Reference to the relationship objects
Has associations	Reference to the relationship objects
Relations	
RelAggregates	A general relationship of composition and decomposition
RelAssociates	A relationship referring to external sources
RelAssigns	A generalization of link relationships
Properties	Further defined attribute definitions depending on discipline, regulation, and region

Table 3
Definitions of properties.

Object property	Description and instances
Has global id	A component has only one member of the *Global id* class
Has owner history	A component has only one member of the *Owner history* class
Has name	A component has only one member of the *Name* class
Has object type	A component has only one member of the *Object type* class
Has local placement	A component has only one member of the *Local placement* class
Has representation	A component has only one member of the *Representation* class
Has predefined type	A component has only one member of the *Predefined type* class
Has is decomposed by	A component has only one member of the *Is decomposed type* class
Has decomposes	A component has more than one member of the decomposed type class
Has assignments	A component has only one member of the *Has assignments* class
Has associations	A component has only one member of the *Has associations type* class

binding specifications used by software venders to develop IFC interfaces with their native applications. Unstructured specifications and unordered implementation rules can cause confusion in defining an MVD and executing predefined rules. To address this issue, we need to explicitly define a link between MVD specifications and formal information models.

3.4. Discontinuity of development processes between an IDM and an MVD

Unlike implementable data, manually input data on a table is a set of texts. To translate IDM data into an MVD, MVD developers should manually define each entity and attribute based on table-based data as a reference. In other words, MVD developers should refer to an IDM table as a written document to generate concept descriptions, which is a time-consuming and tedious task. This discontinuous procedure prevents us from confirming the consistency and integrity of MVD information against an IDM.

4. Research methodology

This research suggests an approach to improving definition processes for an IDM and an MVD of each domain industry using more formal and consistent specifications than those used in current practice. In particular, to address the problems in the present MVD development, this paper proposes an approach to an ontology-based IDM and MVD that can formalize domain knowledge and integrate the processes of an IDM and an MVD using ontological principles.

4.1. An ontology and related research

In philosophy, an ontology term is derived as the study of existence as well as formal categories [29]. Ontology can be used to represent formal specifications of abstract views of domains associated with objects, properties, and relationships [16]. Based on fundamental proposition or perspective of the domain industries,

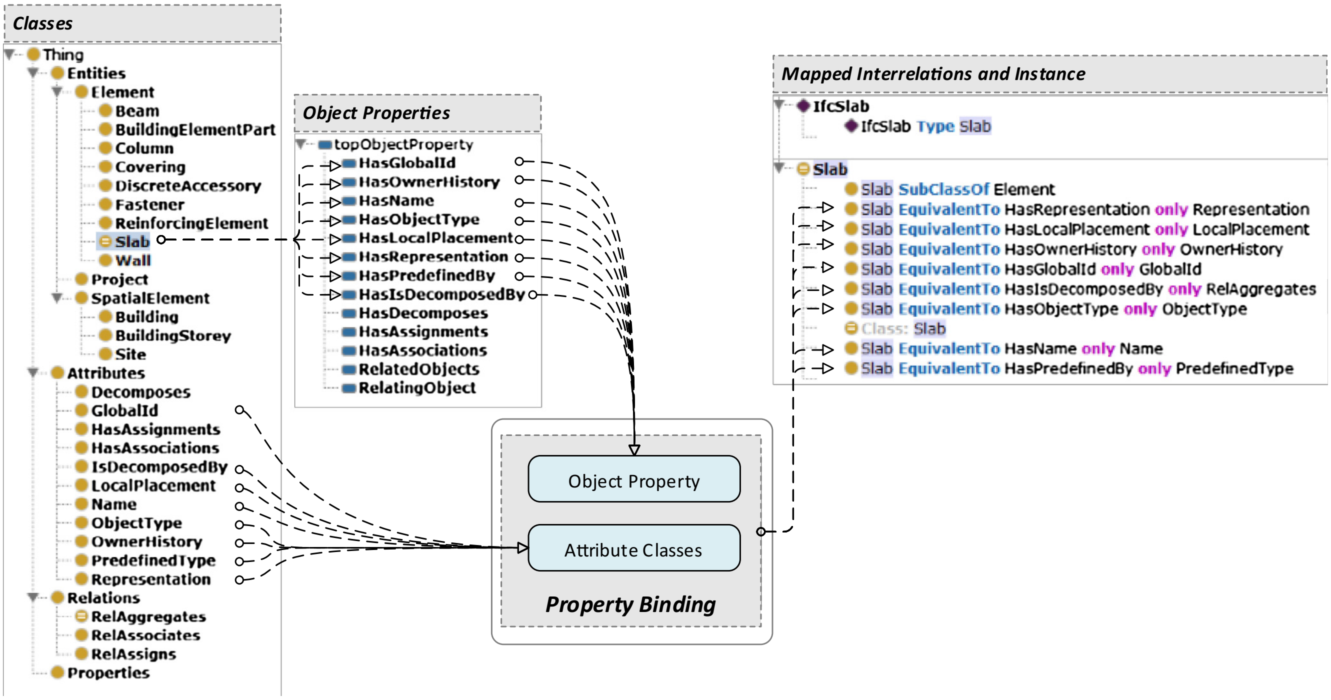


Fig. 5. A *Slab* class and its interrelationships with *Attribute* classes through object properties.

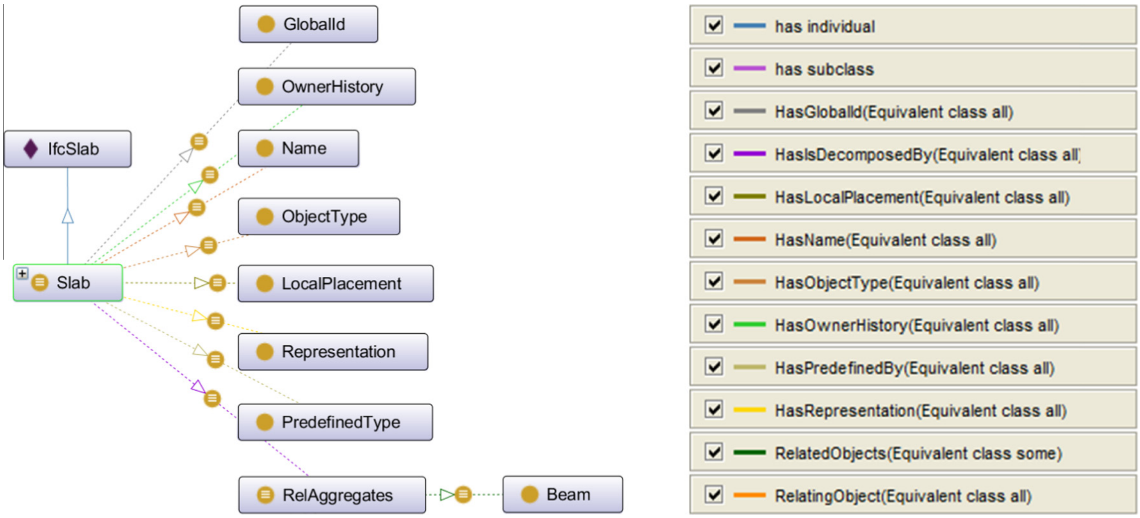


Fig. 6. A *Slab* class and its interrelationships with *Attribute* classes through object properties.

ontology defines certain knowledge classes and hierarchy using semantic relationships. Such capability that clearly represents semantic relationships of domain knowledge can play a pivotal role in developing an IDM and generating a robust link between an IDM and an MVD. In addition, the defined scope of domain information imposes the explicit limitation of knowledge definition using epistemology [14]. Thus, this formal representation of domain knowledge depicts the abstraction of the real world to state explicit specifications using a logical theory of ontology [18]. In addition, the integration of ontology and semantic web technology provides methods to structurally illustrate domain knowledge and to facilitate its reusability [3]. In other words, the informatics systems of an ontology model articulate exchange rules and restrictions for specific industry domains that can resolve the problems of current IDM and MVD development processes.

This research for developing an ontology-based IDM and MVD is an original work, which does not adapt well to the earlier efforts. However, several studies related interoperable data format employed the ontology principles. For providing useful background for this research, a large number of theoretical and practical studies related to conceptual modeling were reviewed. The development of ifcXML representation borrowed the core ontology concepts, high-level IFC kernel entities such as IfcObject, IfcProduct, IfcBuilding, and IfcBuildingElement [20]. In addition, the conceptual modeling was researched for developing an ontology design and its specifications in the mid 20th century [4,13,30]. Classification systems such as OmniClass and Uniclass, provides valuable information for structure configurations. This ontology concept has been adopted in diverse domains that require formal, shareable, and extensible domain knowledge: The philosophical conceptualization of the essence of ontology knowledge describing construction processes was used to represent the knowledge map of constraints, mechanism, and actors in construction infrastructure [14]; a knowledge management framework was proposed

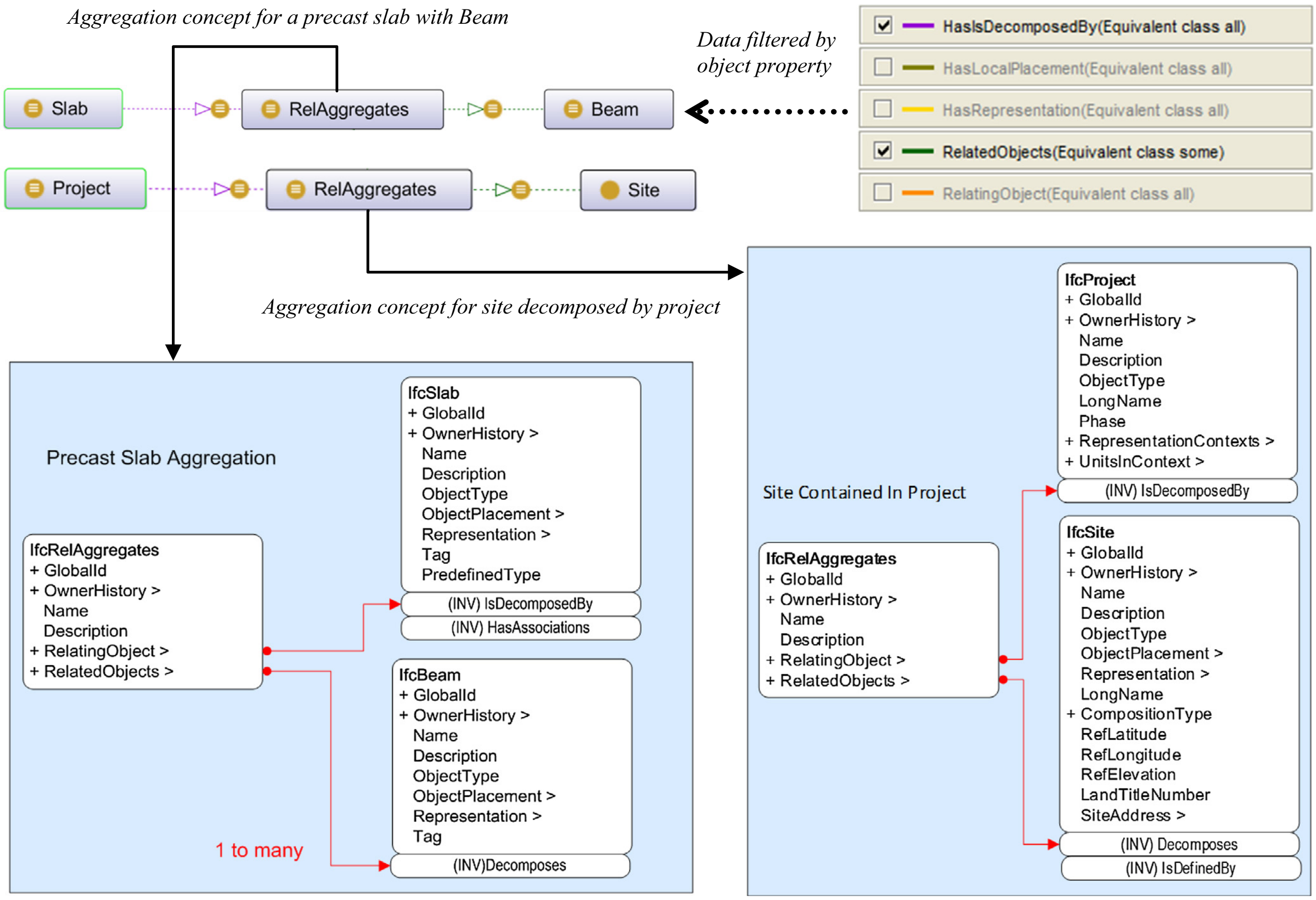
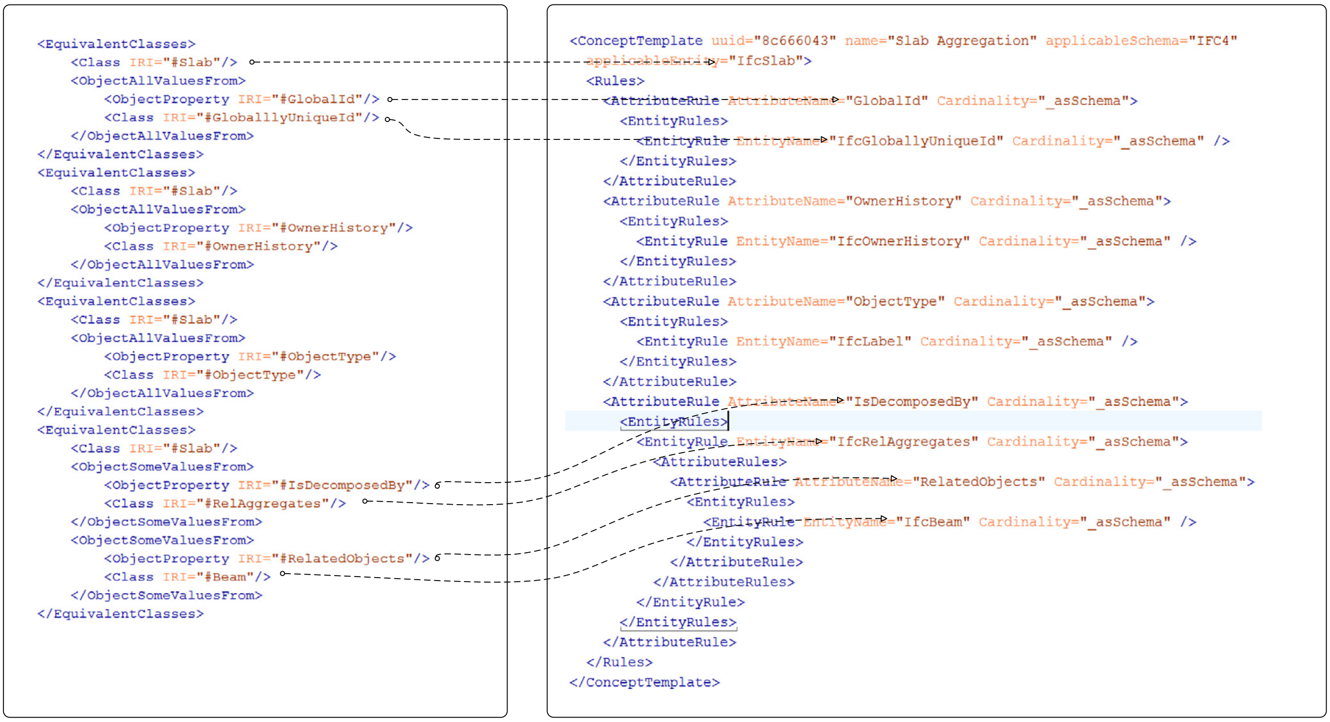


Fig. 7. Interrelationships of a RelAggregates class and concept diagrams of the relations.

for improving the performance of construction project managers [7]; a manufacturing system was defined by engineering ontology models using semantic web [23]; and construction process execution plan was integrated by an ontology-based model to specify the area of constrains [32]. EU e-COGNOS project led by VTT, technical research centre of Finland, provides an IFC-enabled approach with regard to knowledge management activities of European construction firms and ontology-based mechanism of interdependencies [22]. As a practically developed project for building construction, European organizations including the Dutch BAS, the EU CONCUR, and the Norwegian BARBi, developed the Lexicon system. This project addressed the categorization of pragmatic entities and conceptual specifications needed in AEC/FM industries [31]. The implementation framework using ontology patterns was clearly described in the Christopher Alexander's works associated to the ontology specification for mapping patterns [1,2]. The listed efforts explicitly show the benefits and possibilities of the ontology-based knowledge and its management system, which is greatly beneficial to developing specifications and requirements of BIM data exchanges required in an IDM and an MVD. The next section describes how an ontology-based knowledge can improve current IDM and MVD development processes.

4.2. An ontology-based IDM and MVD

While following the NBIM process for MVD development, domain experts and engineers need a formal baseline and a standardized reference of all data definitions, taxonomies, and a hierarchy of harmonized schemas that provide a comprehensive data set of model entities required by software vendors. However, the current heterogeneous binding processes have been caused by different approaches of MVD developers and software vendors. When developing an IDM, they normally account for actual contexts using various modeling abstractions, which lead to numerous representations of exchange requirements. However, a defined MVD should be implemented in strictly formalized patterns such as predefined methods and functions. Hence, practitioners often use an MVD indirectly in their IFC binding process. In terms of a knowledge gap, current practice in transferring an IDM to an MVD is too fragmented to organize exchange requirements of one specific domain. During an IDM process, domain professionals and software vendors have several discussions in order to define the requirements and business rules demanded for the multiple exchange processes of one particular domain. Such requirements are generally written in the paper-based document. The requirements in the IDM document are manually translated by MVD developers according to the IFC schema. The translated contents are written in paper-based or electronic documents. Thus, an IDM and an MVD that involve the same requirements are specified in different types of formats and documents. To ameliorate these challenges, this paper suggests an ontology-based approach to developing an IDM and an MVD that is expected to provide (1) high-level information representations that can be translated from model data to knowledge; (2) a robust knowledge framework that helps capture the appropriate semantics familiar to the software developers; (3) data verification and specifications for the consistency and correctness of IDM information and model views so that MVD developers and software vendors can identify whether a defined MVD includes all IDM requirements; and (4) an integrated process for an IDM and an MVD that can facilitate



```
<EquivalentClasses>
    <Class IRI="#Slab"/>
    <ObjectAllValuesFrom>
        <ObjectProperty IRI="#GlobalId"/>
        <Class IRI="#GloballlyUniqueId"/>
    </ObjectAllValuesFrom>
</EquivalentClasses>
<EquivalentClasses>
    <Class IRI="#Slab"/>
    <ObjectAllValuesFrom>
        <ObjectProperty IRI="#OwnerHistory"/>
        <Class IRI="#OwnerHistory"/>
    </ObjectAllValuesFrom>
</EquivalentClasses>
<EquivalentClasses>
    <Class IRI="#Slab"/>
    <ObjectAllValuesFrom>
        <ObjectProperty IRI="#ObjectType"/>
        <Class IRI="#ObjectType"/>
    </ObjectAllValuesFrom>
</EquivalentClasses>
<EquivalentClasses>
    <Class IRI="#Slab"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#IsDecomposedBy"/>
        <Class IRI="#RelAggregates"/>
    </ObjectSomeValuesFrom>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#RelatedObjects"/>
        <Class IRI="#Beam"/>
    </ObjectSomeValuesFrom>
</EquivalentClasses>
```

Structure of OWL/XML for a slab data

```
<ConceptTemplate uuid="8c666043" name="Slab Aggregation" applicableSchema="IFC4"
  applicableEntity="IfcSlab">
  <Rules>
    <AttributeRule AttributeName="GlobalId" Cardinality="_asSchema">
      <EntityRules>
        <EntityRule EntityName="IfcGloballyUniqueId" Cardinality="_asSchema" />
      </EntityRules>
    </AttributeRule>
    <AttributeRule AttributeName="OwnerHistory" Cardinality="_asSchema">
      <EntityRules>
        <EntityRule EntityName="IfcOwnerHistory" Cardinality="_asSchema" />
      </EntityRules>
    </AttributeRule>
    <AttributeRule AttributeName="ObjectType" Cardinality="_asSchema">
      <EntityRules>
        <EntityRule EntityName="IfcLabel" Cardinality="_asSchema" />
      </EntityRules>
    </AttributeRule>
    <AttributeRule AttributeName="IsDecomposedBy" Cardinality="_asSchema">
      <EntityRules>
        <EntityRule EntityName="IfcRelAggregates" Cardinality="_asSchema">
          <AttributeRules>
            <AttributeRule AttributeName="RelatedObjects" Cardinality="_asSchema">
              <EntityRules>
                <EntityRule EntityName="IfcBeam" Cardinality="_asSchema" />
              </EntityRules>
            </AttributeRule>
          </AttributeRules>
        </EntityRule>
      </EntityRules>
    </AttributeRule>
  </Rules>
</ConceptTemplate>
```

Structure of mvdXML for a slab data

Fig. 8. Translating OWL/XML into mvdXML pertaining to slab aggregation.

MVD development and reduce efforts in manual knowledge translation. One main reason of consolidating two processes is that an ontology-based translation is expected to maintain a relationship between two processes so that stakeholders can review an ontology-based IDM, identify the purposes and sources of the requirements specified in an MVD, and confirm about whether the MVD is well-defined and accurately specified according to an IDM.

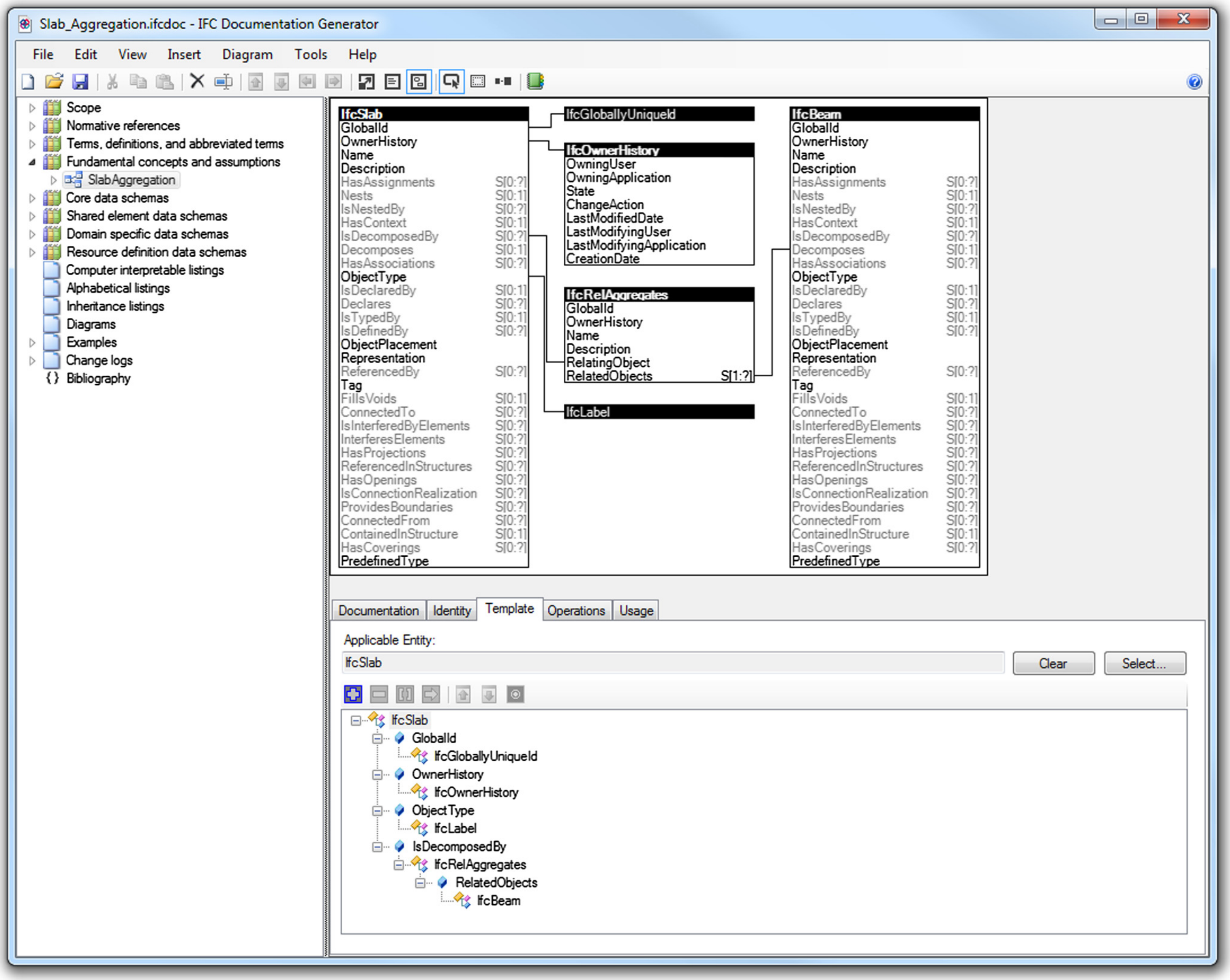
4.3. Overview of framework

Fig. 3 represents the architecture for defining an ontological engineering model for an IDM and an MVD. The ontological engineering model specifies a conceptualized definition and a formal classification for representing specific IDM requirements and the explicit scope of an MVD. The process of defining an ontological data model is executed by Ontology Web Language (OWL)-Description Logic (DL) implementable on Protégé, a Java-based open source for ontology editors that generates queries and rules in a formal framework. However, the ontology model executed on Protégé can slightly differ from ontological principles in terms of theory and practice [28]. Even though the features of these two paradigms differ with regard to the semantics of constructs that evaluate consistency of an ontological model, their structures representing classes and properties are almost the same. In this framework, Semantic Web Rule Language (SWRL) and OWL help define a schema of rules and queries and a knowledge classification of a predefined ontological model. Based on the ontology for the formal specifications of an MVD, SWRL allows us to define a hierarchical knowledge map and dictionary. In a back-end process in Fig. 3, SWRL rules can be shared and extended to the native binding of various types of software products. In a front-end process, SWRL rule sets that are programmed modularization, also can be implemented on the IfcDoc application, which is a MVD documentation tool developed in .NET C# language.

Fig. 4 represents the simplified data flow of an ontology-based IDM and MVD processes in Fig. 3. The collected data exchange requirements by domain professionals are ultimately transferred to the IfcDoc application. In addition, as shown in Fig. 4, the ontology model written in OWL/XML is parsed based on the predefined specifications for establishing a modularized concept. Since an IDM written by ontological principles includes explicit correlations among entities, attributes, and properties, a knowledge module can be extracted consistently and explicitly from an ontology model. Basically, the module, a concept description, is designed as a format of a template that can be reused for various domains. Thus, based on such templates, relevant entities can be assigned to them on the IfcDoc tool to generate an associated MVD. The parsed OWL/XML model is translated to mvdXML so that the defined IDM can be imported into the IfcDoc tool. mvdXML aims to efficiently share predefined model views and reuse existing requirements and their rules for developing MVDs of other disciplines [8,21]. mvdXML containing the definitions of data exchanges based on the IFC schema can be exported from and imported to the IfcDoc tool. A translator developed as a Protégé add-in is executed based on an embedded mapping table, which specifies a correspondence for binding an input set of data, an IDM, into an output set of values, an MVD using the IFC schema.

5. Case study

An ontology formalizes IDM and MVD knowledge and helps generate an explicit link between concept rules and formal information models that can reduce effort and time in developing an MVD. As a case study, the precast concrete IDM was established based on ontological principles capable of representing the formal specifications of MVD knowledge. This case study addressed the semantics of an MVD for the precast concrete domain. It is one of the most advanced domains that consists of not only a fully advanced MVD but also complex performance implementation affecting concrete and steel domains. Based on precast concrete



Applicable Entity: IfcSlab

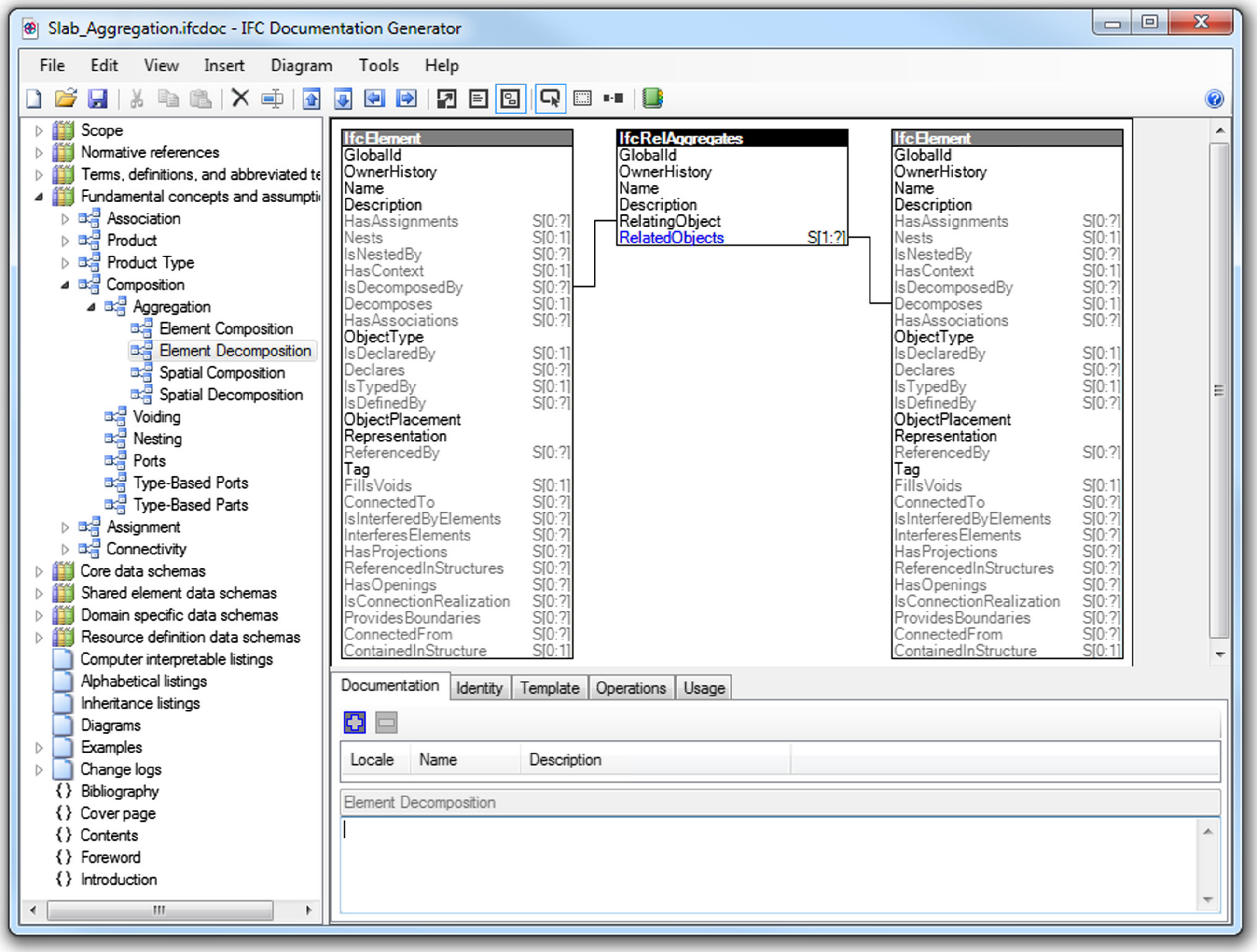
Fig. 9. The structural relation of entities and attributes of slab aggregation defined on the IfcDoc tool.

domain knowledge, this study established the precast concrete ontology, which specifies IDM requirements based on the scope of MVD rules and provides implementable modules on the IfcDoc application. This precast concrete ontology model designed based on 96 concept descriptions of the precast concrete MVD produces the formal classification of an MVD, which consists of syntactic constraints pertaining to the IFC schema defined by the EXPRESS language and semantic rules with regard to the specifications of BIM data exchanges. In addition, this case study demonstrates that the proposed ontological approach can address the problems of the current method for MVD development as described in Section 3. To evaluate the improvement of the suggested MVD development process with a precast concrete ontology model, a predefined precast concrete MVD and its concept descriptions were compared with ones established by new approach.

Ontology that is machine readable taxonomy of classes, subclasses, and relationship specifies hierarchical description of information pertaining to performance implementation and business rules [26]. Ontology also uses a set of knowledge concepts within a domain and the relational semantics among concepts, which allow users to employ reasoning and searching features. As shown in Table 2, the required classes and instances of a precast concrete domain in the case study of an ontology model consist of the four information categories, entities, attributes, relations, and properties. This data structure follows the one of concept templates of the IFC 4 repository, so that its ontological IDM can be readily translated into mvdXML using the IFC schema. The four types of classes are interrelated by a set of object properties in Table 3. For instance, the relationship of a Slab object and an ObjectType attribute is defined by an ObjectType property: *Slab HasObjectType only ObjectType*. The ontology-based IDM defines precise terms and explicit scope of data exchanges for depicting their requirements. There are several benefits of using an ontology model for MVD knowledge modeling: (1) an ontology model can be easily managed and shared among domains, (2) a knowledge map helps produce data modules using explicit relationships, (3) a robust knowledge structure developed by an ontology allows consistent validation and reasoning, and (4) the processes of IDM and MVD development can be integrated using implementable data and semantic relationships defined by the means of classes and properties of an ontology [32]. This section illustrates the benefits of this proposed ontological approach, representing detailed examples of a case study.

5.1. Organized and arranged information using ontological principles

An IDM can be well-organized and categorized by the ontological principles using ontology modeling applications such as the Protégé. Hence, the ontology-based knowledge set is easily manageable and editable using explicit data hierarchy, which facilitates categorizing items and determining data structure. In particular, users can avoid



Element Decomposition

Fig. 10. Detailed concept templates originated from an ontology model.

inputting redundant data and designing relationships with wrong items. Fig. 5 represents a *Slab* class and its interrelationships with attribute classes through object properties. Since classes are interrelated with object properties, users can establish a consistent data set with certain relational restrictions. For example, a *Slab* class can involve a relationship with a *ObjectType* class through a *HasObjectType* property. In addition, this electrically defined ontology template improves knowledge searching and promotes sharing data with industry experts in accordance with their specific purposes such as development of their MVDs. Thus, the formal classification structure of a precast concrete MVD that is extensible and shareable knowledge allows users to focus on necessary exchange constraints and ultimately results in saving significant time and effort in developing an MVD. Reusing defined knowledge and distributing them also support integration of industry knowledge, distinct terminology, and a hierarchical structure.

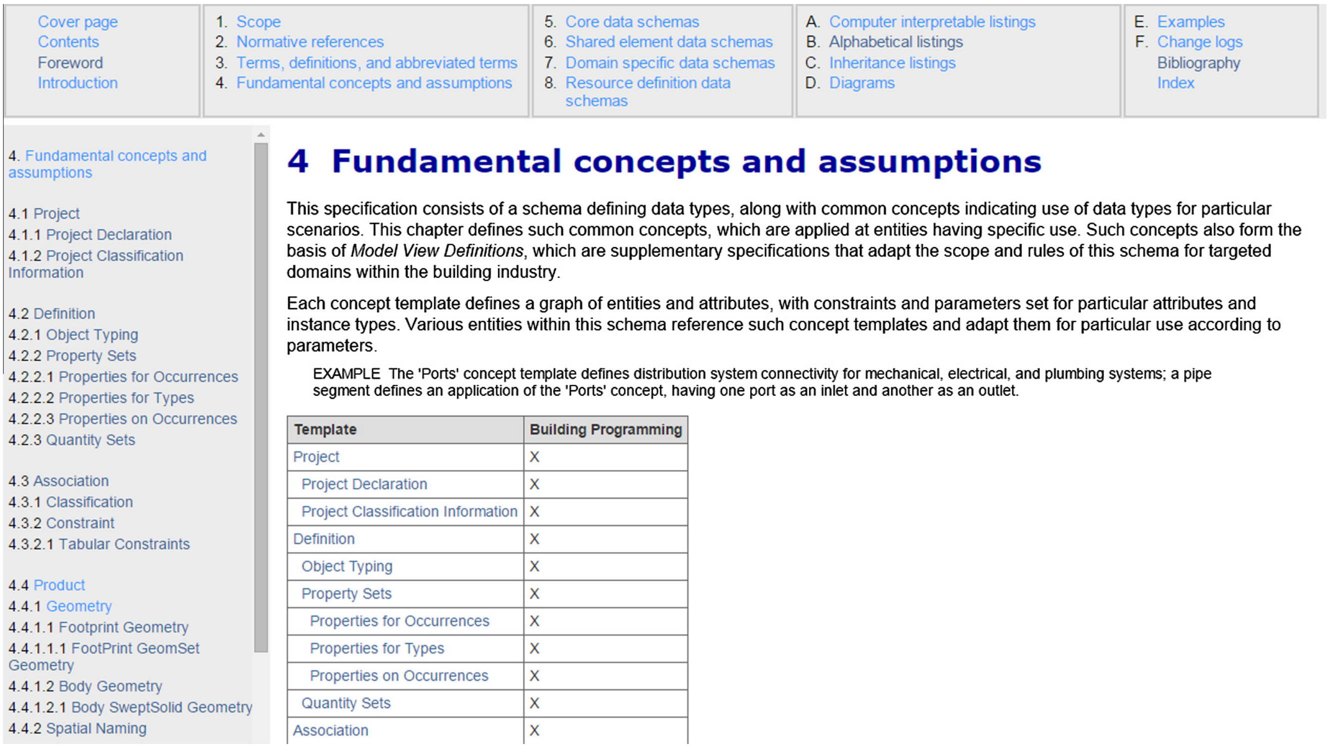
5.2. Knowledge map representing explicit links of items

To grasp the entire structure of defined data, advanced representations such as a knowledge map [9] and clustering [24] are imperative. Such diverse visual diagrams about the knowledge map that efficiently illustrate interrelationships of IDM ontology can be executed on the Protégé, which helps users identify the correlations shared by entities and define a module for a concept description for MVD development. One of network diagram applications is the OntoGraf, which helps demonstrate the node-based diagram of OWL ontologies in the Protégé tool. Fig. 6 represents the graph pertaining to a *Slab* class and its neighborhood. The diagram that corresponds to the OWL description depicts relationships explored through incremental expansion of the graph.

Fig. 7 illustrates the interrelationships of a *RelAggregates* class with its neighborhood classes. Since the relationships can be managed by a filter function and expanded by a restriction to certain relationship types, which help reduce graph complexity, MVD developers can identify a concept module efficiently. As shown in Fig. 7, data filtered by *HasIsDecomposedBy* and *RelatedObjects* properties can be designed as concept descriptions, Precast Slab Aggregation and Site Contained in Project. In other words, explicit relationships depicted on a node-based diagram would help users not only to understand domain knowledge but also to retrieve required data, which allow MVD developers to recognize where a concept description comes from and to identify how accurately an MVD is defined. Thus, a developed MVD ontology enables more effective inquiry for developing concept modules of an MVD that can be specifications for establishing IDM data at the initial phase of MVD development process.

5.3. Reasoning for defined knowledge

An ontological model helps search and retrieve data without in-depth knowledge and a specific inquiry on a domain area. Since ontology has the uniqueness pertaining to class names, an ontological inference process can automate the process of reasoning and



Cover page
Contents
Foreword
Introduction

1. Scope
2. Normative references
3. Terms, definitions, and abbreviated terms
4. Fundamental concepts and assumptions

5. Core data schemas
6. Shared element data schemas
7. Domain specific data schemas
8. Resource definition data schemas

A. Computer interpretable listings
B. Alphabetical listings
C. Inheritance listings
D. Diagrams

E. Examples
F. Change logs
Bibliography
Index

4. Fundamental concepts and assumptions

4.1 Project
4.1.1 Project Declaration
4.1.2 Project Classification Information

4.2 Definition
4.2.1 Object Typing
4.2.2 Property Sets
4.2.2.1 Properties for Occurrences
4.2.2.2 Properties for Types
4.2.2.3 Properties on Occurrences
4.2.3 Quantity Sets

4.3 Association
4.3.1 Classification
4.3.2 Constraint
4.3.2.1 Tabular Constraints

4.4 Product
4.4.1 Geometry
4.4.1.1 Footprint Geometry
4.4.1.1.1 FootPrint GeomSet Geometry
4.4.1.2 Body Geometry
4.4.1.2.1 Body SweptSolid Geometry
4.4.2 Spatial Naming

4 Fundamental concepts and assumptions

This specification consists of a schema defining data types, along with common concepts indicating use of data types for particular scenarios. This chapter defines such common concepts, which are applied at entities having specific use. Such concepts also form the basis of *Model View Definitions*, which are supplementary specifications that adapt the scope and rules of this schema for targeted domains within the building industry.

Each concept template defines a graph of entities and attributes, with constraints and parameters set for particular attributes and instance types. Various entities within this schema reference such concept templates and adapt them for particular use according to parameters.

EXAMPLE The 'Ports' concept template defines distribution system connectivity for mechanical, electrical, and plumbing systems; a pipe segment defines an application of the 'Ports' concept, having one port as an inlet and another as an outlet.

Template	Building Programming
Project	X
Project Declaration	X
Project Classification Information	X
Definition	X
Object Typing	X
Property Sets	X
Properties for Occurrences	X
Properties for Types	X
Properties on Occurrences	X
Quantity Sets	X
Association	X

Fig. 11. Model view definition document automatically generated from the IfcDoc tool.

validating data in an ontology model. For example, an MVD developer for a steel domain can search for a dataset of a reinforcing element such as a reinforcing bar using an *Element* class and *Object Property* that provides an accurate data hierarchy and required attributes and relationships employed by a precast concrete MVD. As another example, when an MVD developer needs to design a concept module for *Relaggregation*, predefined modules as shown Fig. 7 can be reused. In addition, users can identify and access accurate data of a precast concrete domain, because the precast concrete ontology is designed based on a comprehensive precast concrete handbook. Such data help MVD developers confirm the previous use of a relevant IDM and share a consistent approach for defining the same IDM data. Such features can be executed on ontology applications such as the Protégé tool, which provides users with embedded libraries that can evaluate the correctness of an information category and links for entities. Furthermore, SWRL allows users to define rules for their specific goals. Through such semantic reasoning feature, specifically defined rule sets in SWRL can be implemented to verify relationships and accuracy of ontology knowledge. Such capability improves the consistency of IDM information, allows users to define specific requirements for an IDM, and facilitates the process of MVD development. Protégé-OWL provides a set of rule libraries that supports modifying and implementing SWRL rules. Implementation libraries involve core SWRL, SQWRL, ABOX, TBOX, mathematical expressions, XML, extensions, and RFD built-ins libraries. For example, *greaterThanOrEqual*, the mathematical built-in of the core SWRL library, provides a feature that indicates that a slab with more than one rebar is a PrecastSlab. The automated interpretation of an MVD is expected to help industry domain experts and software vendors determine if IFC instance models comply with the syntactic and semantic rule sets defined in MVD requirements.

5.4. Integrating processes for IDM and MVD development

An ontology model exported in an OWL/XML file format can be parsed for generating concept modules and translated as an mvdXML file, which can be imported into the IfcDoc tool. The IfcDoc application is open to the public, and thus, an ifcdoc file or an mvdXML file that users defined can be easily shared to discuss and communicate the well-formedness of concept specifications and MVDs. In addition, the IfcDoc application supports reusing the predefined concepts and rules using the import feature. Since the IfcDoc tool can automatically generate MVD documentation, an ontology-based IDM is efficiently translated in the IFC schema in accordance with a specific MVD. The translator from OWL/XML to mvdXML contains a mapping table that describes XML representation matches between OWL and mvdXML syntaxes using the IFC schema. Fig. 8 shows the translation from OWL/XML into mvdXML pertaining to a slab aggregation. *GlobalId* in an ObjectProperty instance and *GloballyUniquedId* in a Class instance are mapped to *GlobalId* in an AttributeRule instance and *IfcGloballyUniqueId* in an EntityRule. In terms of relationships of a slab, *IsDecomposedBy* in an ObjectProperty instance and *RelAggregates* in a Class instance are mapped to *IsDecomposedBy* in an AttributeRule instance and *RelAggregates* in an EntityRules. In addition, to designate related objects, *RelatedObjects* in an ObjectProperty instance and *Beam* in a Class instance are mapped to *RelatedObjects* in an AttributeRule instance and *IfcBeam* in an EntityRules. In such translating processes, a mapping table defines that a *GloballyUniqueId* class should be generated as *IfcGloballyUniqueId* and a *Beam* class should be generated as *IfcBeam*. Since ontology definition follows the data structure of concept templates in the IFC4 repository, it allows MVD developers to use ontology as a baseline to prepare their MVD specifications. Unlike the previous approach for manually developing an IDM and an MVD, such translation method in the proposed approach transforms written information into machine readable data without having to modify them. Moreover, the integrated development processes for an IDM and an MVD facilitates sharing concept descriptions and rule sets among the same domain experts through the IfcDoc application.

Fig. 9 represents the structural relationships of entities and attributes with regard to a slab aggregation defined by an mvdXML file on the IfcDoc application. The explicit attribute such as

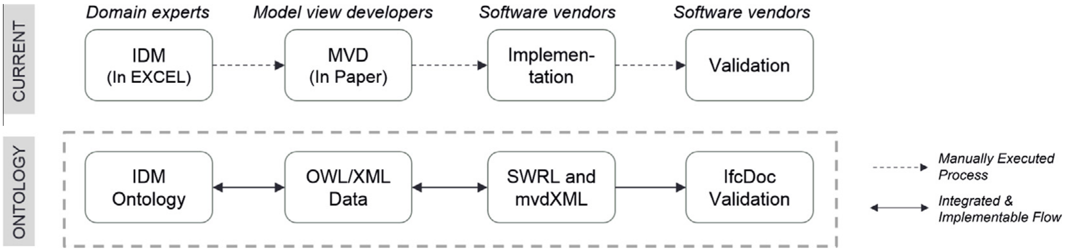


Fig. 12. Comparison between the current and the ontology-based approached for MVD development.

GlobalId, *OwnerHistory*, and *ObjectType* is black-colored and the inverse one such as *HasAssignments*, *IsDecomposedBy*, and *Decomposes* is gray-colored in the diagram. The black line represents the semantic link: for example, *GlobalId* must have the type of *IfcGloballyUniqueId* and *ObjectType* must have *IfcLabel* value. The diagram depicts that the IfcSlab entity of a P21 file must involve attributes for *GlobalId*, *OwnerHistory*, *ObjectType*, and *IsDecomposedBy* relationships. In addition, attributes with no relationship such as *Name*, *Description*, and *Tag* describe that the IfcSlab optionally contains values and types for the attributes. These relational skeleton and semantic requirements are defined in the IDM ontology and translated from OWL/XML.

5.5. Model view definition document

Using the proposed method, the authors generated an engineering ontology that can classify and identify precast concrete domain information. The ontology was translated into mvdXML. Fig. 10 shows the imported concept templates using the mvdXML file. Since these concept templates are translated from the definitions of domain knowledge ontology data, MVD developers and software vendors easily identify their IDM information when they need to update the specifications of an MVD. In particular, this mvdXML data allows domain experts to maintain the consistency of exchange requirements knowledge. In addition, this approach enables MVD developers to reduce time and efforts to reorganize and re-categorize exchange requirements of an IDM for generating concept modules. Defined concept descriptions on the IfcDoc tool can be documented automatically in the HTML format. Fig. 11 represents the MVD document in the HTML format, which follows the same format of the IFC 4 schema. Thus, this approach efficiently integrates IDM and MVD processes using an ontology model and the IfcDoc tool through the mvdXML format.

5.6. Summary of a suggested approach and its limitations

The suggested approach using a case study consists of four big steps: (1) establishing IDM ontology on the Protégé, (2) validating a dataset using semantic reasoning, (3) parsing OWL for generating concept modules and translating OWL/XML into mvdXML, and (4) importing mvdXML into the IfcDoc tool to generate MVD documentation. Using the case study of a precast concrete domain, this paper represents diverse benefits of the proposed approach to developing ontology-based IDM and integrated MVD processes. This approach will help formalize domain knowledge for an IDM and define consistent concept modules for an MVD. In addition, ontological principles enable MVD developers to manage classes, attributes, and properties within a single knowledge map that helps efficient and consistent data update and management. Furthermore, since the current IDM approach does not support hierarchical data and follow consistent terms or dictionaries, this ontology is expected to play a pivotal role for sharing a standardized terminology among domain industries, which minimize ambiguity and allows precise mapping between IDM to the concepts to MVD. Hence, explicit data modules and manageable relationships among entities of an ontological IDM will reduce effort and time in defining an MVD and in developing IFC interfaces of BIM authoring tools. In particular, the parser and the translator from OWL/XML and into mvdXML will allow users to generate MVD documentation automatically based on defined ontology that ultimately helps integrate a process of an IDM and an MVD.

With regard to limitations, this study focuses on ontology of a precast concrete domain that involves domain specific knowledge and data exchange requirements. Since an IDM involves various types of requirements for data exchanges, ontology for all aspects of other domains should be established and researched for identifying strengths and weaknesses of the proposed approach. In addition, the rule transition framework that exports rules defined in SWRL as mvdXML and imports them into IfcDoc should be further improved in order to address diverse rule types in SWRL. As a next step, the evaluation of an ontology-based MVD will be implemented by comparing two MVDs shown in Fig. 10: An ontology-based precast concrete MVD and a developed MVD using an existing method. The actual candidate criteria to compare two set of MVDs will be to: (1) calculate the number of implementable and missed concepts and rules, (2) identify the percentage of reusability of existing concept descriptions, (3) evaluate times and efforts in defining concepts and MVDs, and (4) verify the efficiency of documenting concepts and the consistency of domain knowledge (see Fig. 12).

6. Conclusions

Design professionals are generally aware that the trend of building design is toward BIM [15]. However, lack of software interoperability and functionality are rated as one of the greatest obstacles in the BIM application [6]. As a result, domain experts require an MVD as indispensable specifications for their distinct business regulations and data exchange requirements. Based on the aforesaid critical success factors, an ontological IDM and MVD development is a critical need for professionals because no common and robust definition for building semantics has been agreed upon, which results in vague and separate IDM data and a lack of consistency. Hence, an ontology-based approach to developing an IDM and an MVD is imperative to collect required data, shared predefined requirements, and implement the dataset for guaranteeing consistent data structure and integrating MVD development processes. This proposed ontology-based framework would help accurately recognize domain knowledge and appropriate requirements for developing reusable concept modules. Moreover, this integrated approach would contribute to accelerating the use of BIM and facilitating the process of current design and construction work for the AEC industries, guaranteeing interoperable data exchanges. This opportunity will offer the values of greater regulatory predictability and consistency that reduce human errors in a design process and improve design quality. The ultimate goal of the ontology-based IDM and MVD is not only to put in place an automated knowledge framework, but also to offer the impetus to gear up the AEC industry toward greater interoperability through the secured deployment of the IFC-based BIM applications. Hence, this research related to the automated process of a design would bolster the adoption of BIM and its accessories starting during the predesign phase, collaborating with the practitioners at government organizations and diverse industries.

References

[1] C. Alexander, The Timeless Way of Building, vol. 1, Oxford University Press, 1979.

[2] C. Alexander, S. Ishikawa, M. Silverstein, A Pattern Language: Towns Buildings, Construction, Center for Environmental Structure Series, 1977.

[3] C.J. Anumba, R.R. Issa, J. Pan, I. Mutis, Ontology-based information and knowledge management in construction, Constr. Innov.: Inform., Process, Manage. 8 (3) (2008) 218–239.

[4] B.C. Björk, Requirements and Information Structures for Building Product Data Models, VTT Technical Research Centre of Finland, 1995.

[5] buildingSMARTalliance, National Building Information Modeling Standard, 2007.

[6] buildingSMARTInternational, IFC Coordination View V2.0, 2015. <http://www.buildingsmart-tech.org/specifications/ifc-view-definition/coordination-view-v2.0/summary> (retrieved 09.22.15).

[7] V. Charlesraj, S.N. Kalidindi, An ontology-based knowledge management framework for performance improvement of construction project managers, in: Paper Presented at the 23rd International Symposium on Automation and Robotics in Construction, 2006.

[8] T. Chipman, T. Liebich, M. Weise, mvdXML: specification of a standardized format to define and exchange model view definitions with exchange requirements and validation rules, BuildingSMART Int. (2012).

[9] I. Digital Alchemy, IFC BIM Validation Service, 2005. <www.DigitalAlchemyPro.com> (retrieved December 21, 2014).

[10] C. Eastman, Y.-S. Jeong, R. Sacks, I. Kaner, Exchange model and exchange object concepts for implementation of national BIM standards, J. Comput. Civ. Eng. 24 (1) (2009) 25–34.

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

[12] C. Eastman, R. Sacks, I. Panushev, S. Aram, E. Yagmur, Information Delivery Manual for Precast Concrete, PCI-Charles Pankow Foundation, 2009. <http://dcom.arch.gatech.edu/pcibim/documents/IDM_for_Precast.pdf> (retrieved May 5, 2015).

[13] A. Ekholm, S. Fridqvist, A conceptual framework for classification of construction works, Electron. J. Inform. Technol. Constr. (itcon) 1 (1996) 25–50.

[14] T. El-Diraby, H. Osman, A domain ontology for construction concepts in urban infrastructure products, Autom. Constr. 20 (8) (2011) 1120–1132.

[15] M. Fischer, J. Kunz, The scope and role of information technology in construction, in: Paper Presented at the Proceedings-Japan Society of Civil Engineers, 2004.

[16] T.R. Gruber, A translation approach to portable ontology specifications, Knowl. Acquisit. 5 (2) (1993) 199–220.

[17] J. Hietanen, S. Final, IFC model view definition format, Int. Alliance Interoperab. (2006).

[18] P. Katranuschkov, A. Gehre, R.J. Scherer, An Ontology Framework to Access IFC Model Data, ITcon (2003) 413–437.

[19] Y.-C. Lee, C.M. Eastman, W. Solihin, R. See, Modularized rule-based validation of a BIM model pertaining to model views, Autom. Constr. 63 (2016) 1–11.

[20] T. Liebich, ifcXML – XML Schema Language Binding of EXPRESS for IFC, IAI Publ. MSG-01-001, 2001.

[21] T. Liebich, A. Geiger, P. Katranuschkov, K. Linhard, R. Steinmann, M. Weise, mvdXML Schema. Version 0.5, 2011. <http://www.buildingsmart-tech.org/downloads/accompanying-documents/formats/mvdxml-documentation/mvdXML_V1-0.pdf> (retrieved September 20, 2014).

[22] C. Lima, T. El Diraby, B. Fies, A. Zarli, E. Ferneley, The e-COGNOS project: current status and future directions of an ontology-enabled IT solution infrastructure supporting knowledge management in construction, in: Paper Presented at the Winds of Change: Integration and Innovation of Construction, Construction Research Congress, 2003.

[23] H.K. Lin, J.A. Harding, A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration, Comput. Ind. 58 (5) (2007) 428–437.

[24] J.W. Newkirk, NUnit, 2005. <NUnit.org> (retrieved February 2, 2015).

[25] R. Sacks, I. Kaner, C. Eastman, Y.-S. Jeong, The Rosewood experiment—building information modeling and interoperability for architectural precast facades, Autom. Constr. 19 (2010) 419–432.

[26] M. Venugopal, C. Eastman, R. Sacks, J. Teizer, Improving the robustness of model exchanges using product modeling concepts for IFC schema, in: Paper Presented at the Proc. of the ASCE International Workshop on Computing in Civil Engineering, Miami, US, 2011.

[27] M. Venugopal, C.M. Eastman, R. Sacks, J. Teizer, Semantics of model views for information exchanges using the industry foundation class schema, Adv. Eng. Inform. 26 (2) (2012) 411–428.

[28] H.H. Wang, N. Noy, A. Rector, M. Musen, T. Redmond, D. Rubin, M. Horridge, Frames and OWL side by side, in: Paper Presented at the Presentation Abstracts, 2006.

[29] G. Witmer, Dictionary of Philosophy of Mind – Ontology, 2004. <http://www.artsci.wustl.edu/~philos/MindDict/ontology.html> (retrieved May 13, 2015).

[30] J. Wix, Computerised exchange of information in construction, Build. Constr. Core Model (1996).

[31] K. Woestenenk, The LexiCon: structuring semantics, in: Paper Presented at the Proceedings of CIB W78 Conference on Distributing Knowledge in Building, 2002.

[32] B. Zhong, L. Ding, H. Luo, Y. Zhou, Y. Hu, H. Hu, Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking, Autom. Constr. 28 (2012) 58–70.