Interoperability Definition Framework for Creating Consistent Constraints in BIM Data Exchange

Liu H.[a,b,*], Gao G.[b,c,*], Gu M.[b,c]
[a] Digital Horizon Technology Co., Ltd., China. [b] Tsinghua University, China. [c] Beijing National Research Center for Information Science and Technology (BNRist), China.
liuhan@digital-h.cn; gaoge@tsinghua.edu.cn

Abstract. To improve the interoperability in the data exchange of Building Information Models (BIMs), various standards and methods have been proposed to add constraints to the data exchange process, including data creation, conversion and validation. It is a challenge to ensure the consistency of the constraints when many object types and stakeholders are involved. In this paper, the Interoperability Definition Framework (IDF) method is proposed for integrating various sources of knowledge about data exchange. Automatic checking can be performed on the IDF knowledge graph to ensure the compatibility of the mappings. The knowledge graph also supports converter capability confirmation for user-defined exchange requirements, and consistent ruleset generation for different BIM data formats.

1. Introduction

Building Information Model (BIM) is the digital representation of construction projects and built assets. The BIM data contains abundant geometry and semantic information about a facility, which is supposed to be used as a shared resource of knowledge in the whole life-cycle of a facility. BIM data exchange is the key link between different software tools and different stakeholders in the life-cycle. In the latitude direction, information is shared between heterogeneous software tools and formats for applying various analyses and supporting collaboration among stakeholders. In the longitude direction, the accumulation and transmission of data through the life-cycle phases are supposed to make benefits continuously.

The interoperability of data exchange requires that within a clearly defined boundary, the result of data delivery guarantees the normal function of the applications that receive the data. The methods to ensure interoperability mainly include formal specification of the data exchange requirements, computer-checkable rules to constraint the data formats and contents and qualified software implementations to meet the requirements. Based on the IFC (Industry Foundation Classes) as an open data schema, the IDM-MVD (Information Delivery Manual - Model View Definition) is the current mainstream method for improving interoperability in the engineering industry. Data exchange requirements are well-defined and represented as MVD rulesets, so that software certification and data validation can be performed accordingly.

Recently, the community started to realize that in order to ensure the interoperability of data exchange, the constraints should not only be applied to the data exchange results, but the whole process of data creation, conversion and reception. Sufficient human-readable instructions are also needed for the partitioners to do the right things in each step of data exchange. The IDS (Information Delivery Specification) (buildingSMART, 2020) is a new topic to show this idea of applying consistent constraints for the whole data exchange process. Using IFC as the target format, IDS tries to collect the data constraints and converter settings into an XML ruleset. The human modelers will get the instructions to create the model and to config the exporter. However, there are still challenges in the promotion of IDS-based data exchange applications.

- Data validation on multiple BIM data formats. The MVD technology is widely used for validating the data in IFC format, but there is still a need for a validation method that can be easily transplanted to multiple software-specific formats.
- Software capability confirmation for various exchange requirements. The buildingSMART has started the software certification program (buildingSMART, 2018), but only for some static MVD specifications. It is still a problem to confirm the capability of BIM modeling tools and converters for various user-defined exchange requirements.
- Consistency between the constraints. The creation of IDS usually requires knowledge from modelers, data end-users, IFC experts and software vendors. When many object types and stakeholders are involved, an automated method may be needed to integrate the knowledge and to confirm whether there are conflicts between the constraints.

Aiming at the above challenges, the Interoperability Definition Framework (IDF) method is proposed in this paper for integrating various sources of knowledge about data exchange, which helps provide consistent constraints for the creation, conversion and validation of BIM data. Based on the IDF knowledge graph, automatic checking can be performed to ensure that the knowledge sources are without conflicts. IDS and MVD rulesets can be generated accordingly to provide consistent constraints for both source and target schemas. The capability of BIM converters can be flexibly checked according to user-defined exchange requirements.

2. Related Work

2.1 Specification and Validation of BIM Data Exchange

The IDM-MVD process is a recommended method for specifying the requirements and constraints of BIM data exchange by the buildingSMART organization (See, et al., 2012). Both domain engineers and BIM data engineers are involved in the IDM-MVD process. The IDM document is about the process and participants, use cases, domain concepts, and the information requirements for each domain concept to meet the needs of the use cases. With the help of data engineers, the mvdXML ruleset is composed according to the IDM by binding the domain concepts into IFC data entities. Automatic MVD validation can be performed to on receiving the IFC data to check the conformance of the data against the mvdXML ruleset (Lee, et al., 2015; Weise, et al., 2016; Liu, et al., 2022). It is also shown that MVD is applicable to other open-source data schemas such as CityGML and gbXML (Jeon, et al., 2021). In addition to checking the data results, the buildingSMART software certification program has been deployed to confirm the capability of BIM modeling and converting tools in creating IFC files that conform to the MVD rules. However, the current software certification program is only for some static MVD specifications, such as the IFC4 Design Transfer View (DTV) and the Reference View (RV). It is still a challenge to confirm the capability of software tools according to some user-defined exchange requirements.

The IDS is a recent attempt to provide a more complete specification for the whole BIM data exchange process. The IFC format is the basis of the IDS rules, but the necessary information and instructions to create, convert and validate the data in a non-IFC BIM software system are also included in the IDS ruleset. Different human and computer participants in the data exchange process can get different facets of information from the IDS ruleset. The human modeler will get instructions to perform correct modeling operations, and to configurate the conversion tool for exporting the correct data. Automatic data validation is also expected to be performed both before exporting (in a software-specific format) and after exporting (in IFC).

2.2 Methods for Integrating the Knowledge about BIM Data Exchange

The creation of BIM data exchange rules requires the integration of knowledge about both the engineering domain and information technology. The communication between the domain experts and data engineers may take lots of effort, and there are also tedious tasks for viewing the documents and converting the constraints into computer language. Related studies tried to set up the knowledge model for BIM data exchange specifications, and to support the computer-aided process of creating the runnable rulesets.

The semantic web technology is a promising approach to formally represent the classification of domain concepts and the definition of data schema in order that the knowledge can be shared and reused (Venugopal, et al., 2012; 2015; Lee, et al., 2016). The classification of domain entities and definition of properties are represented as an ontology. The IFC schema and the mappings from domain terminologies to data objects are also represented as ontologies. By reusing the knowledge about domain concepts and mappings, MVD rulesets and documents can be automatically generated according to some formalized representations of exchange requirements. The consistency inside of the ruleset can also be automatically checked to avoid ambiguity and unsatisfied necessary components. In another related study (Son, et al., 2022), the buildingSMART Data Dictionary (bSDD) is used as the definition of domain concepts, and the idmXSD is used as the formal representation of exchange requirements. The information units can be mapped to the IFC data entities with the help of bSDD. When the MVD ruleset is generated, the validity of the ruleset is then automatically checked to ensure that the information units conform to the IFC schema and that the exchange requirements are fully covered.

The above approaches focus on the binding of domain concepts on an open-source data schema, typically the IFC. However, the community starts to realize that the constraints need to be applied to the whole process of data creation, conversion and reception to ensure the interoperability of data exchange, and more software-specific instructions and configurations are needed to support the easy creation of data in a non-IFC modeling tool. Several commercial tools have taken steps in this direction. BIMQ (AEC3, 2016) is an online collaborative tool for specifying the exchange requirements and editing the MVD rules. BIMQ provides a graphical user interface for defining domain concepts, exchange requirements, IFC mappings and constraint rules. Based on the defined MVD rules, BIMQ can also generate configuration files for specific tools like Autodesk Revit and GraphiSoft ArchiCAD, so that the property fields and IFC mappings can be loaded by the tools. The IDS toolkit (ACCA, 2021) initiates the IDS format, and provides user interfaces to define the exchange requirements and constraints. The IDS ruleset is designed to be loaded on an upstream BIM authoring tool for providing human-readable instructions and general constraints on property values, so that the exported IFC data tends to satisfy the exchange requirements.

In comparison, BIMQ tends to make use of the existing configuration files of each software tool, but IDS tends to collect all needed settings and instructions into one standard IDS ruleset, and asks the software vendors to support loading and utilizing the rules. Anyway, compared with the reviewed methods based on the open-source IFC schema, the validation of data compliance and the configuration of data mappings on software-specific formats are still challenging. There is still a need for a methodology to implement software capability confirmation and data validation on various software-specific formats.

3. The Interoperability Definition Framework Knowledge Graph

In this paper, the Interoperability Definition Framework (IDF) method is proposed for integrating the knowledge about data exchange from multiple stakeholders. During the creation

of the IDF knowledge graph, the input knowledge is automatically checked to find the conflicts. The MVD and IDS rulesets can then be generated to provide consistent constraints for the creation, conversion and validation process of BIM data. The knowledge graph can also support the flexible confirmation of software capability for user-defined exchange requirements.

The idea of the IDF method comes from the observation that although the data formats are close-sourced, the software development kits (SDKs) are provided to access the data content. The SDKs are usually in object-oriented programming languages, and by taking the in-memory data structure as a "schema", the specifications on the open-source data formats are likely to be transplanted to describe the constraints for software-specific formats. Based on this idea, the IDF knowledge graph tries to explicitly represent the concept mappings to both the source format and the target format, in order that more generalized software capability confirmation and BIM data validation can be supported.

The IDF knowledge graph is composed of 8 interrelated modules, as shown in Figure 1. The namespace "idf:" is used to identify the predicates and objects in the IDF knowledge graph.

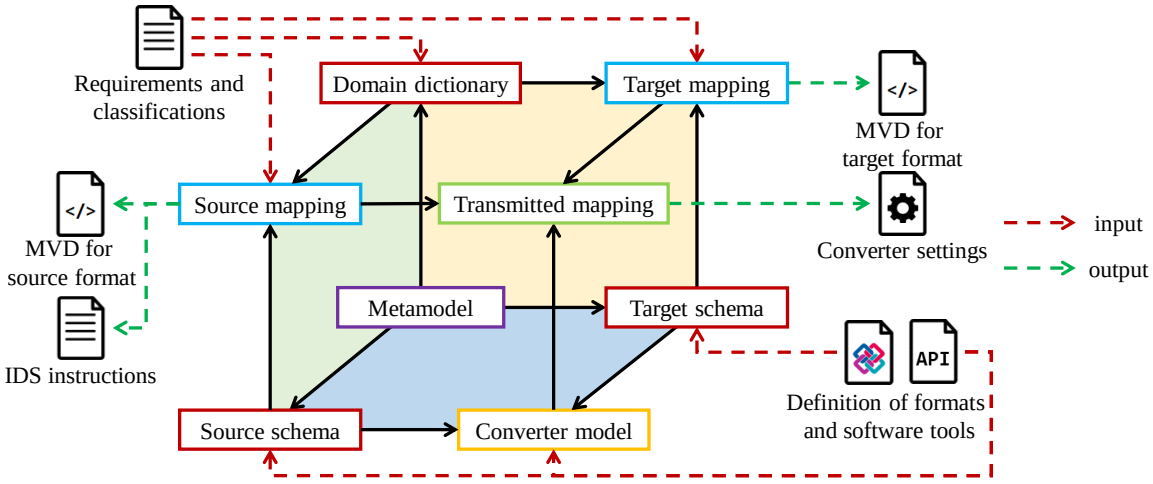

Figure 1. The structure of the IDF knowledge graph, with the input and output documents.

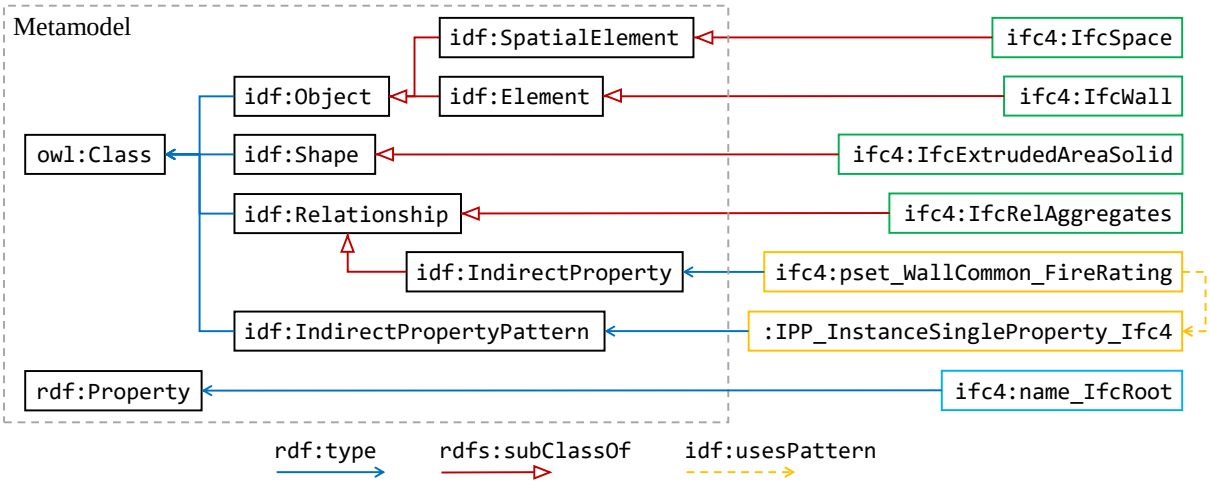

Figure 2. The IDF metamodel and some example attached nodes in IFC schema.

(1) Metamodel. The metamodel defines the basic abstract classes that describe the data of a built asset, which is supposed to be independent of specific data formats. Based on the metamodel, both the domain terminologies and the data classes in different formats can be attached to the metamodel as sub-classes or instances. Many of the IDF abstract classes can be viewed as a simplification of the "upper ontology" classes from ISO 15926-2 (ISO, 2003).

Figure 2 shows the main classes in the IDF metamodel (in the dashed box) and several attached classes and instances from the IFC schema.

The "indirect property pattern" is introduced in the IDF method for representing a "complex path" to link a root entity node with a required property value, which can be defined based on a specific data structure, or just be defined as virtual connections between domain terminologies. For example, an "instance property" in the "Revit API" schema can be represented as an indirect property pattern shown in Figure 3. The data nodes and attributes are organized according to the Revit API definitions. Some important nodes can be assigned with a name (such as "PSetName"), which can be used in generating mvdXML rules.

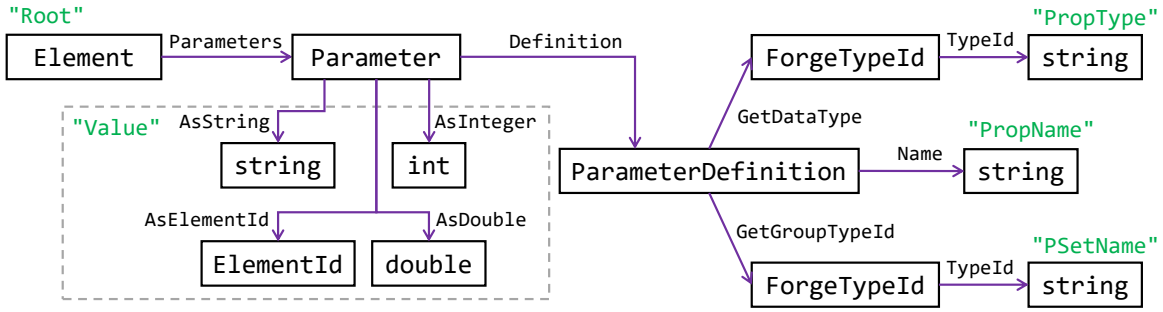

Figure 3. An example indirect property pattern "instance property" for Revit API schema.

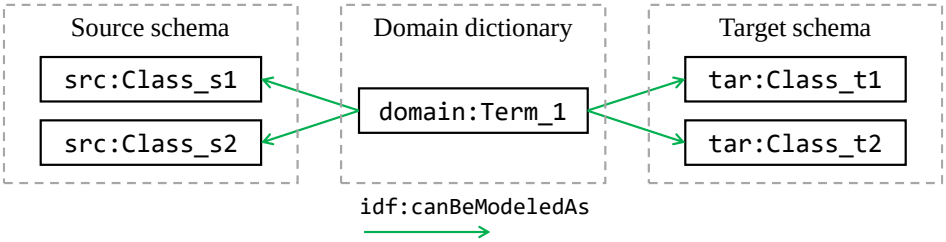

Figure 4. Some example source mappings and target mappings.

(2) Domain dictionary. The domain dictionary is a collection of natural language terms that are involved in the exchange requirements. The terms can be grouped into objects, relationships, properties and so on, and then be attached to the metamodel. The entries of a domain dictionary may be from an existing classification system (such as bSDD and OmniClass), which introduces the hierarchy between the classes, and constraints the usage of properties and relationships.

(3) Source schema & target schema. To define the exchange requirements between different BIM formats, each format is needed to be represented as a schema model and attached to the metamodel. The ifcOWL can be directly used as a schema model, and the built-in property sets and quantity take-offs can be appended to the IFC schema model as indirect properties. For a non-IFC format, a similar schema model can be defined for representing the classification of data types, and the data structure to access the direct and indirect properties.

(4) Source mapping & target mapping. A mapping model is a set of "idf:canBeModeledAs" triples, each of which points from one node in the domain model to several nodes in a schema model. The mappings can be defined between two classes, between two properties, or between two indirect property patterns. Figure 4 shows an example of source mappings and target mappings of a domain term to different data schemas. In real-world cases, the knowledge about the mappings may come from different stakeholders. The source mapping may come from the modelers, indicating the routines to model the domain concepts in the source format. The target mapping may come from the model receivers, indicating the allowed data types in the target format according to the exchange requirement.

(5) Converter model. The converter model is established between the source schema and target schema, defining the capability of the converter. In the converter model, the ability of the converter to transform a source data item (a class or a property) to a target data item is defined as an "idf:Conversion" node. One source data item may be allowed to convert into several optional targets, and each conversion is "enabled" by one or more "idf:ConverterOption" nodes. Some of the options are exclusive (such as a group of radio buttons in the converter UI), and a valid configuration of the converter can be defined as a set of non-exclusive option nodes. An example converter model is shown in Figure 5.

(6) Transmitted mapping. The transmitted mapping is an automatically generated mapping model between the source schema and the target schema. Each entry in the transmitted mapping model is an "idf:TransMappingItem" node, which "transmits" the link between one source data item to one target data item through one or more terms in the domain dictionary, and is "implemented" by one or more conversions in the converter model. An example entry in the transmitted mapping model is shown in Figure 6.

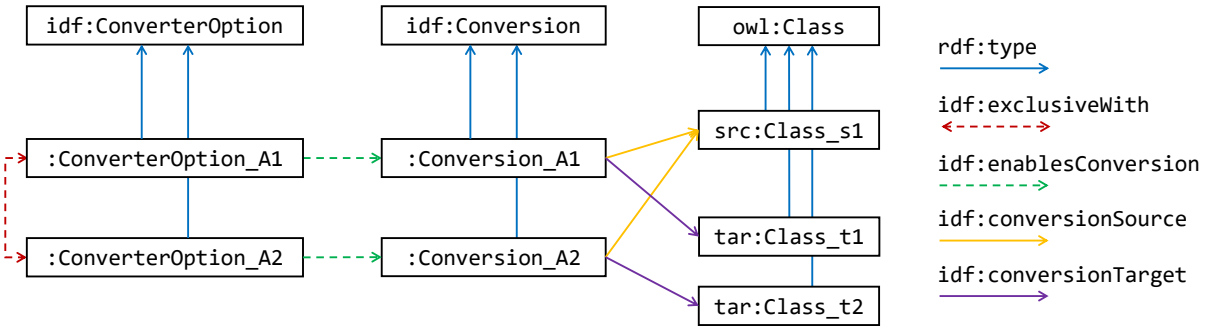

Figure 5. An example converter model.

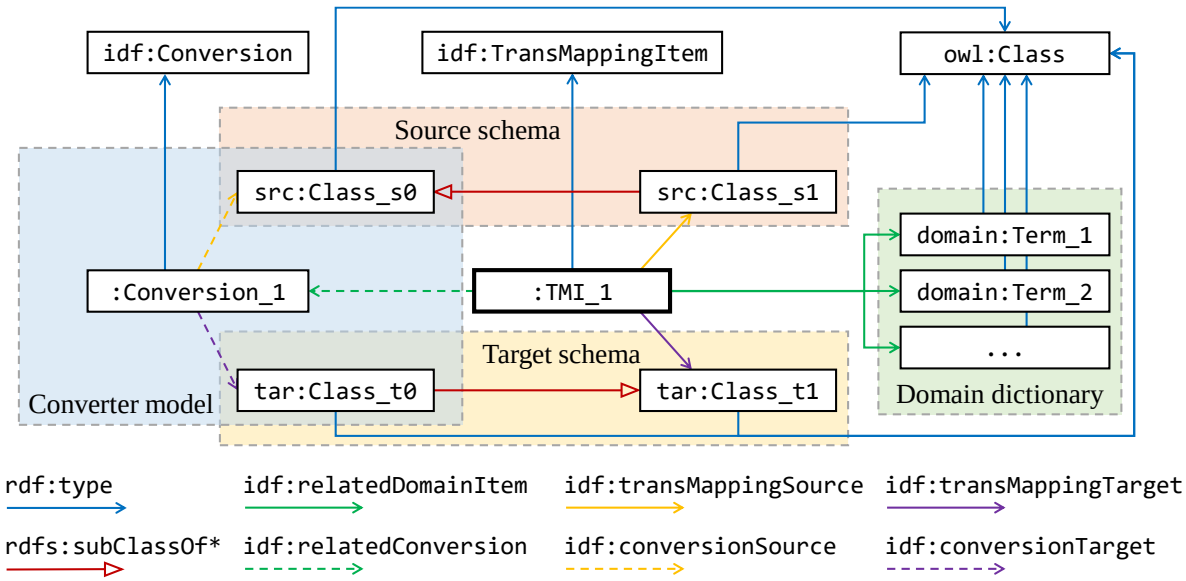

Figure 6. An example entry in the transmitted mapping model.

In summary, when the modules in the IDF knowledge graph are arranged to be in a cubic shape (as shown in Figure 1), the "lower layer" modules are about the intrinsic definitions of data formats and software tools, and the "upper layer" modules are about the domain concepts, modeling routines and exchange requirements. The knowledge sources required to generate different modules are usually provided by different domain engineers or software vendors, and

in the process to build up the IDF knowledge graph, the compatibility between the definitions can be checked.

4. Application Case of the IDF Method

In this section, taking the data exchange task between Autodesk Revit and IFC as an example, an application case of the IDF method is proposed to show the steps to create the IDF knowledge graph (section 4.1), to perform converter capability checking (section 4.2) and to generate MVD and IDS rulesets (section 4.3). The case is initiated from a project to establish a city-level BIM handover standard for planning approval, which involves a fine-grained classification system for domain objects and the required properties. The BIMs are required to be exported to IFC format for compliance checking against an MVD ruleset on submission, and several modeling instruction handbooks are also needed for some mainstream BIM software tools. In this project, the domain concepts, data mappings and software-specific instructions are provided by different organizations, so it is of great value to check the correctness of the mappings and configurations to ensure that the exchange requirements can be satisfied.

4.1 Creating the IDF knowledge graph

The IDF knowledge graph is created in the following steps.

(1) Obtaining the schema models for BIM formats. In this case, the involved BIM formats are IFC and RVT. The IFC classes and attributes can be obtained from the EXPRESS specification file. The IFC built-in properties can be obtained from the XML specifications in the IFC documentation. The RVT data can be accessed through the Revit SDK in C# language. In our implementation, the RVT schema model is generated using the reflection mechanism in C#, so that the inheritance between the classes and the implementation of interfaces can be automatically extracted from the SDK to form the class definitions. The fields, properties and zero-parameter methods are also extracted to form the property definitions in the RVT schema.

(2) Obtaining conversions and options of the converter. Autodesk has open-sourced the RVT-to-IFC exporter, which implements the conversions for entities and properties (the conversions for geometries and relationships are not yet considered in this paper). The entity conversion is configured by a mapping file, which defines the exporting rules from RVT built-in classes to IFC classes. In our implementation, one mapping file is considered as a converter option. Each converter option enables several entity conversions, and different mapping files are exclusive to each other. The IFC built-in properties are exported from the "property calculators" in the exporter. By regarding each calculator is as a RVT built-in property, the conversions for built-in properties are established. For user-defined properties, a conversion is assigned to link two indirect property patterns.

(3) Checking the mappings from the domain dictionary to each schema model. The domain dictionary model has defined 5516 classes and 2458 properties in total. The authors of the domain classification system also provided the mappings to IFC classes. The mappings to RVT classes are provided by engineers from another organization. The RVT mappings and IFC mappings are represented as the source mappings and target mappings, respectively. On creating each mapping model, the following checking is performed to ensure that the mapping of a subclass does not conflict with its parent classes. Let c, p be two domain concept classes, and **D**, **Q** be two sets of data classes from the same schema, i.e.

$$\mathbf{D} = \{d \mid c \ \text{idf:canBeModeledAs} \ d\}; \ \ \mathbf{Q} = \{q \mid p \ \text{idf:canBeModeledAs} \ q\}. \qquad (1)$$

The conflict checking is to find whether the mapping targets of the parent covers all mapping targets of the children, which can be represented with the following expression:

$$(c\ \text{rdfs:subClassOf}+\ \ p) \wedge (\mathbf{Q} \neq \emptyset)\ \rightarrow \forall d\ \exists q(d\ \text{rdfs:subClassOf}*\ q) \qquad (2)$$

The checking task found 61 conflicts among 4565 IFC mappings, and 662 conflicts among 3197 RVT mappings. The authors are advised to modify the mappings to resolve the conflicts, or allow the program to automatically expand the parent mappings to cover the child mappings. Finally, in the auto-expansion task, 40 and 510 mappings are added to IFC and RVT mapping models, respectively.

(4) Automatic generation of transmitted mapping. The transmitted mapping is automatically generated from other modules in the IDF knowledge graph. The SPARQL query shown in Figure 7 is performed, and the query result records are grouped by the (?SOURCE, ?TARGET) pairs. Each unique pair corresponds to a new "idf:TransMappingItem" node. The corresponding "?DOMAIN" nodes and "?CONV" nodes are linked to the transmitted mapping item. The task found 1419 query results, and then 76 transmitted mapping items are generated accordingly.

```
SELECT ?DOMAIN ?SOURCE ?TARGET ?CONV WHERE {
   GRAPH [source mapping] {
      ?DOMAIN idf:canBeModeledAs ?SOURCE.
   }
   GRAPH [target mapping] {
      ?DOMAIN idf:canBeModeledAs ?TARGET.
   }
   { ?CONV idf:conversionSource ?SOURCE. } UNION
      { ?CONV idf:conversionSource ?S_CONV. ?SOURCE rdfs:subClassOf+ ?S_CONV. }
   { ?CONV idf:conversionTarget ?TARGET. } UNION
      { ?CONV idf:conversionTarget ?T_CONV. ?T_CONV rdfs:subClassOf+ ?TARGET. }
}
```

Figure 7. The SPARQL query for generating the transmitted mapping.

4.2 Converter Capability Checking for User-defined Exchange Requirements

Based on the IDF knowledge graph, the user may provide a set of domain concepts (with mappings already defined) as user-defined exchange requirements, and a set of non-exclusive converter options, then the capability of the converter can be automatically checked. By executing the SPARQL query shown in Figure 8, it can be confirmed whether all concerned domain concepts are covered by the enabled conversions. In the experiment, the input option set contains an IFC type mapping file. The concerned domain concepts are chosen as 1915 classes in the mechanical, electrical and plumbing disciplines. Finally, 481 classes are found to be satisfied. While for other classes, it is recommended to either modify the IFC mappings and RVT mappings, or update the converter functions and configurations. Then the IDF creation and checking can be performed again to get refreshed results.

```
SELECT ?DOMAIN ?TMI ?CONV WHERE {
   VALUES (?OPT) { [the input converter options] }
   ?OPT idf:enablesConversion ?CONV.
   ?TMI idf:relatedConversion ?CONV.
   ?TMI idf:relatedDomainItem ?DOMAIN.
}
```

Figure 8. The SPARQL query for converter capability checking.

4.3 Generating MVD and IDS Rulesets for the Source Schema

An mvdXML ruleset defines several tree-shaped subgraph templates on a data schema to link the root entities with the required attribute nodes. Based on the IDF knowledge graph, the

requirements and constraints can be mapped into two mvdXML rulesets for the source schema and the target schema, respectively. The entity types and attributes defined in a schema model can be used to generate the subgraph templates, and the indirect property patterns can also be transformed into subgraph templates. Similarly, the IDS ruleset is also generated with more human-readable instructions on how to model the domain concepts and properties in the source software, and which options are applied to the exporter. The generated mvdXML ruleset and IDS ruleset for the Revit schema are shown in Figure 9.

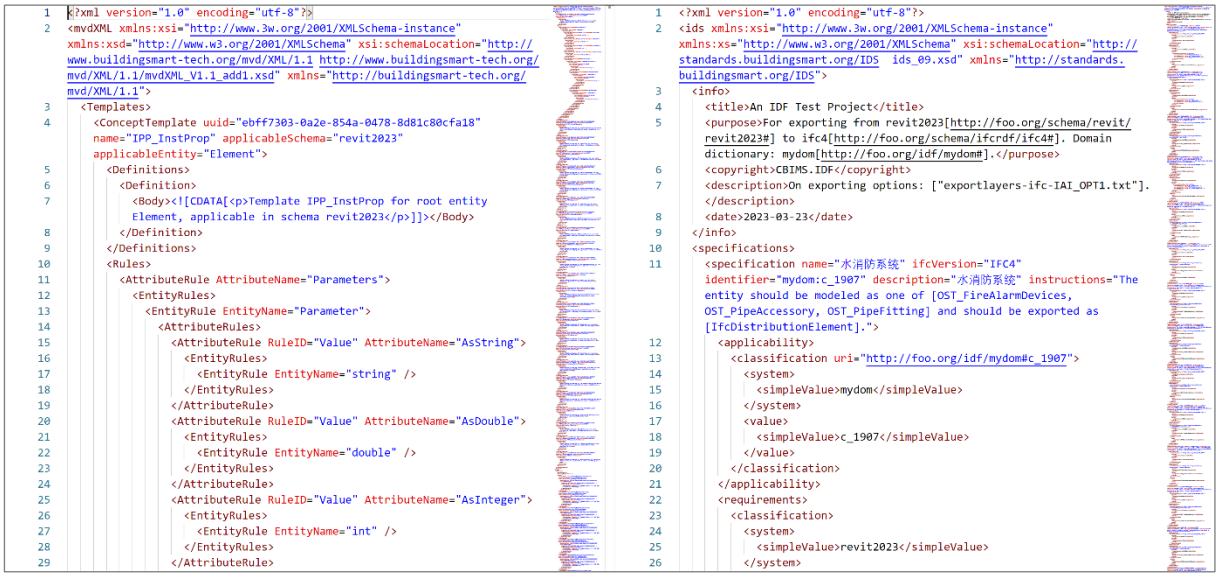

Figure 9. Generated mvdXML ruleset (left) and IDS ruleset (right) for the Revit schema.

The program to create the IDF knowledge graph, run the checking tasks, and generate the rulesets is implemented based on the dotNetRDF toolkit in C#. The experiment is performed on a PC with a 3.19GHz CPU. The whole process finished in 5.06 seconds, which shows that the efficiency of the method fits real-world tasks.

Later, with the generated rulesets, the data validation tool for software-specific mvdXML rules is also implemented based on the reflection mechanism of C#. The tool walks the data structure through the Revit API classes and properties according to the “AttributeRule”s in mvdXML.

5. Conclusion

With the application case of the IDF method on real-world domain concepts and mappings, it is seen that when the knowledge required for data exchange is from multiple stakeholders including software vendors, modelers and the authority, the mappings are easy to be with conflicts. The IDF knowledge graph is helpful in finding the conflicts and providing feedback on modifying the mappings to better meet the exchange requirements. The integrated knowledge supports generating MVD and IDS rulesets for different software tools, as well as confirming the capability of converters for user-defined exchange requirements, which helps to apply consistent constraints for the whole data exchange process.

The future interests to improve the IDF method will be on the following topics to improve the adaptability of converters on various user-defined exchange requirements. First, other than simple type mappings, a domain concept is usually mapped to a subset of a data class with additional “applicability rules”. Such rules are to be included in the knowledge graph, and the applicability rules for a target format should be automatically converted to the instructions for modeling in the source format. Second, rather than type-level configurations, the BIM

converters may also support instance-level overrides of data type mappings. More detailed instructions are needed in the IDS rulesets for such instance-level configurations. Third, the converter capability checking with an input set of options is implemented in this paper. With more fine-grained definitions of options, there might be an algorithm to traverse all possible combinations of options to find proper converter settings.

Acknowledgment

This work was supported by the Science and Technology Research and Development Program of China State Construction Engineering Corporation “Research and Application of Key Technologies for Building Materials Quality Traceability Based on Independent and Controllable Blockchain” (CSCEC-2022-Z-5).

References

buildingSMART (2020). Information Delivery Specifications (IDS). https://www.buildingsmart.org/standards/bsi-standards/information-delivery-specifications-ids/, accessed February 2023

buildingSMART (2018). b-Cert Documentation. https://www.b-cert.org/Documentation/, accessed February 2023.

See, R., Karlshoej, J. and Davis, D. (2012). An integrated process for delivering IFC based data exchange. https://standards.buildingsmart.org/documents/IDM/IDM_guide-IntegratedProcess-2012_09.pdf, accessed January 2022.

Lee, Y.-C., Eastman, C. M. and Lee, J.-K. (2015). Validations for ensuring the interoperability of data exchange of a building information model. Automation in Construction. Elsevier, 58, pp. 176-195.

Weise, M., Liebich, T., Nisbet, N., Benghi, C. (2016). IFC model checking based on mvdXML 1.1, In: eWork and eBusiness in Architecture, Engineering and Construction: ECPPM, 2016 Limassol, Cyprus.

Liu, H., Gao, G., Zhang, H., Liu, Y.-S., Song, Y. and Gu, M. (2022). MVDLite: a fast validation algorithm for Model View Definition rules. In: Proceedings of the International Workshop on Intelligent Computing in Engineering (EG-ICE), pp. 12-22.

Jeon, K. et al. (2021). A relational framework for smart information delivery manual (IDM) specifications. Advanced Engineering Informatics. Elsevier, 49, p. 101319.

Venugopal, M., Eastman, C. M. and Sacks, R. (2012). Configurable model exchanges for the precast/pre-stressed concrete industry using semantic exchange modules (SEM). In: International Conference on Computing in Civil Engineering. pp. 269-276.

Venugopal, M., Eastman, C. M. and Teizer, J. (2015). An ontology-based analysis of the industry foundation class schema for building information model exchanges. Advanced Engineering Informatics. Elsevier, 29(4), pp. 940-957.

Lee, Y.-C., Eastman, C. M. and Solihin, W. (2016). An ontology-based approach for developing data exchange requirements and model views of building information modeling. Advanced Engineering Informatics. Elsevier, 30(3), pp. 354-367.

Son, S., Lee, G., Jung, J., Kim, J. and Jeon, K. (2022). Automated generation of a model view definition from an information delivery manual using idmXSD and buildingSMART data dictionary. Advanced Engineering Informatics. Elsevier, 54, p. 101731.

AEC3 (2016). BIMQ guide. https://www.bimq.de/en/bimq-guide/, accessed February 2023.

ACCA (2021). What is IDS (Information Delivery Specification) and what is it used for. https://biblus.accasoftware.com/en/what-is-ids-information-delivery-specification-and-what-is-it-used-for/, accessed February 2023.

ISO, 2003. ISO 15926-2 Industrial automation systems and integration -Integration of life-cycle data for process plants including oil and gas production facilities - Part 2: Data model. Geneva: ISO.