Multi-LOD model for describing uncertainty and checking requirements in different design stages

J. Abualdenien & A. Borrmann
Chair of Computational Modeling and Simulation, Technical University of Munich, Germany

ABSTRACT: The design of a building is a collaborative process between multiple disciplines. Using Building Information Modeling (BIM), a model evolves throughout multiple refinement stages to satisfy various design and engineering requirements. Such refinement of geometric and semantic information is described as levels of development (LOD). So far, there is no method to explicitly define an LOD's requirements nor any specification of its uncertainty. Furthermore, despite the insufficient information available in early design stages, a BIM model appears precise and certain. This can lead to false assumptions and model evaluations, for example, in the case of energy efficiency calculations or structural analysis. Hence, this paper presents a multi-LOD meta-model to explicitly describe an LOD's requirements taking into consideration the information uncertainty. This makes it possible to check the consistency of the geometric, semantic, and topologic coherence across the different LODs. The model is implemented as a webserver and user-interface providing a means for managing and checking exchange requirements between disciplines.

1 INTRODUCTION

The *architecture, engineering,* and *construction* (AEC) industry is a collaborative environment which requires an iterative and cooperative exchange of models information (Chiu 2002). For example, developing the structural design demands the architectural design information as an input. In this kind of collaboration, the information quality, such as compliance with regulations and analysis requirements, is essential for exchanging, coordinating and integrating the partial designs at the different stages. The design of a building evolves throughout multiple stages, each characterized by a set of consecutive and calibrated actions, to satisfy the different design and engineering requirements.

Building Information Modeling (BIM) is a promising approach that supports managing and exchanging semantically rich 3D-models between the project disciplines (Eastman et al. 2011). Recently, BIM has been widely adopted in the AEC industry (Young et al. 2009), it improves the process' efficiency and quality by promoting the early exchange of 3D building models. Through the different phases of a construction project, the building model is gradually refined from a rough conceptual design to highly detailed individual components. The sequential refinement of geometric and semantic information is described as *Level of Development* (LOD) (Hooper 2015). LOD is a concept that describes the different stages of the project life-cycle by providing definitions and illustrations of BIM elements at the different stages of their development (BIMForum 2017).

In the early design stages, BIM model information is not yet accurate as it is subject to multiple changes in the subsequent design stages (Knotten et al. 2015). Presently, model-based planning techniques are incapable of managing multiple levels of development including a description of their geometric and semantic information uncertainty. Neither is there a formal definition of a building component's level of development nor is there an explicit description of the fuzziness of information. On the contrary, a BIM model appears precise and certain which can lead to false assumptions and model evaluations, as in case of energy efficiency calculations or structural analysis, which affect the design decisions taken throughout the design stages.

The research project MultiSIM aims to develop methods for evaluating building design variants in early design stages. The variants may have different LODs as well as incomplete and uncertain information. The main approach focuses on providing:

- Consistent management of multiple LODs
- Describing the information uncertainty
- Consistent management of design variants
- Supporting model analysis at the early design stages

As part of this research group, we propose the development of a multi-LOD meta-model, which explicitly describes the LOD requirements of each individual building component type taking into consideration the possible uncertainties.

The multi-LOD meta-model introduces two layers, *data-model level* and *instance level*, which offers high flexibility in defining per-project LOD requirements and facilitates formally checking their validity, such as defining and checking required information to support the *Embodied Energy* calculations at different design stages. This paper discusses the advantages in representing the uncertainties at early design stages and highlights the benefits of systematically managing and checking exchange requirements between disciplines. In order to ensure the model's flexibility and applicability, its realization is based on the existing *Industry Foundation Classes* (IFC). IFC is an ISO standard, which is integrated into a variety of software products (Liebich et al. 2013).

The paper is organized as follows: Section 2 discusses the background and related work of our research. Section 3 provides an overview of the multi-LOD requirements and describes the design concepts, and Section 4 presents the meta-model design. In order to evaluate the multi-LOD model, Section 5 illustrates how it can be used to define and check the requirements of the *Embodied Energy* calculations, and a prototype implementation is discussed in Section 6 in terms of usability and possible integration in the design process. Finally, Section 7 summarizes our progress hitherto and presents an outlook for future work.

2 BACKGROUND & RELATED WORK

2.1 *Level of Development (LOD)*

The concept of LOD is employed to manage the model evolvement through the different stages of the building life-cycle. It organizes the iterative nature of the design process which enhances the quality of the decision taken (Hooper 2015). An LOD describes the BIM elements on a particular stage providing definitions and illustrations (BIMForum 2017) which represents their information quality, i.e. certainty and completeness. The LOD scale increases iteratively from a coarse level of development to a finer one. Consequently, the associated characteristics' quality of the exchanged model elements is increased.

The *American Institute of Architects* (AIA) introduced a definition of the term LOD that comprises six levels, starting from LOD 100 reaching LOD 500. Additionally, it adds more flexibility by defining intermediate stages, like LOD 350 which requires the representation of the interfaces between the different building system (BIMForum 2017). Several guidelines have been proposed in an attempt to define the available information at each LOD. Most popularly, *Level of Development Specification* follows the AIA definitions, and *Level of Definition* in the UK (BSI 2017) consists of seven levels and introduces two components: *Levels of model detail* (LOD) representing the graphical content of the models, and *Levels of model information* (LOI) representing the semantic information. Recent approaches propagate the terms *Level of Information* and *Level of Geometry* to clearly distinguish semantic from geometric detailing grades (Hausknecht, Liebich 2017).

In this paper, the abbreviation *LOD* stands for the *Level of Development*, which represents the composition of both *Level of Geometry* (a.k.a. Level of Detail) and *Level of Information* (semantics).

2.2 *Refinement of LODs*

Multiple efforts have been conducted for describing the LODs refinement through the project life-cycle. The main idea is the attempt to represent and formalize the model maturity. Either by explicitly defining relationships or by controlling the amount of added details within an LOD, which makes it possible to check the model's consistency. (Biljecki et al. 2016) argue that five LODs are not enough to capture the building model's development, as the information ambiguity is high. Thus, they restrict the LODs refinement by allowing less specification and modelling freedom using a set of 16 stages. Similarly, (van Berlo, Bomhof 2014) looked into producing a more suitably refined set of LODs for the Dutch's AEC industry, they developed seven LODs after performing multiple geometric tests and analyzing the industrial practices.

From another perspective, (Borrmann et al. 2014) presents a methodology for creating and storing multi-scale geometric models for shield tunnels by explicitly defining the dependencies between the individual levels of detail. For this purpose, a multi-scale product model is developed including a geometric-semantic description of five levels; where the levels 1-3 describe the outer shell in terms of boundary representation of the tunnel volume, boundary surface as well as openings, and the fourth level includes the modeling of the tunnel's interior structure. It is shown how the LOD concept can be integrated into the IFC data model. In order to model the relationship between the different levels and maintain their aggregation, a new relationship class *IsRefinedBy*, a subclass of *Aggregates*, is introduced. The proposed multi-scale model makes use of the parametric modeling techniques to preserve the consistency among the different levels of detail by interpreting and processing the procedural geometry representations. Consequently, the change of a geometric object is propagated by updating all the dependent representations.

In this paper, we adhere the BIMForum's definition, starting from LOD 100 reaching to LOD 500, while making use of its flexibility by introducing intermediate LODs, including LOD 120 and LOD 250, to capture the refinement relationships of the semantic-geometric information.

2.3 *Interoperability*

The design and construction of a building is a collaborative process between multiple disciplines, each expert, such as architect and structural engineer, uses different authoring tool and requires custom specifications to support a particular type of simulations and analysis. With increasing the projects specialization and heterogeneity, the building industry requires a high level of interoperability.

The US national institute of standards and technology confirmed the high annual costs, around $16 billion, resulting from the lack of interoperability between the AEC industry software systems (GCR 2004). Over the last decade, numerous methods of exchanging data in the domain of AEC have been investigated. The aim is to define a common interface for lossless geometric as well as semantic data exchange. Therefore, buildingSMART is promoting the development of the industry standard, *Industry Foundation Classes* (IFC) which was published as ISO standard in 2003 (Liebich 2013). IFC is a free vendor-neutral standard and includes a large set of building information representations, including a variety of different geometry representations and a large set of semantic objects modeled in a strictly object-oriented manner. To allow for dynamic (schema-invariant) extensions and adaptation to local or national requirements, the IFC data model provides the *PropertySet* (PSet) mechanism, which relies on dynamically definable name-value pairs.

Besides exchanging data using IFC, dealing with different kinds of building information, e.g. property sets and definitions, requires a standardized terminology. Thus, the *buildingSmart Data Dictionary* (bsDD) (buildingSMART 2016) was developed as a central repository that stores multilingual definitions of the IFC entities and common schema extensions, for instance, an *IfcWall* entity description and *Pset_WallCommon*. Additionally, bsDD integrates multiple classification systems, including *OMNICLASS* (OmniClass 2012) and *UNICLASS* (Chapman 2013), which are widely adopted for structuring the building information. Each object in the dictionary is identified by a *Globally Unique ID* (GUID) which makes it computer-readable and independent of the object name and language (Bjorkhaug, Bell 2007).

As the IFC data model is too large for authoring tools to handle (Bazjanac 2008), buildingSMART developed the *Model View Definition* (MVD) mechanism as a standard approach for IFC implementation, which reduce the size of models through filtering. An MVD represents a subset of the IFC schema that specifies the requirements and specifications of the exchanged data between the authoring tools (Hietanen, Final 2006). In order to ensure the exchanged data completeness, the required information for each discipline scenario needs to be documented and defined as computer-executable rules (Yang, Eastman 2007). Hence, MVD and its open standard mvdXML (Chipman et al. 2012) can be used to structure the exchange requirements with specific IFC types, entities, attributes (Karlshøj et al. 2012).

So far, the IFC model supports neither the notion of LOD, nor a description of its uncertainty. However, as it is a very widespread and well-established format, we will show how an external meta-model can be used to enrich IFC data by these aspects.

3 MULTI-LOD META-MODEL

3.1 *Requirements analysis*

The efforts and costs required to make changes in a building model in the early stages are relatively lower than in the subsequent stages (Kolltveit, Grønhaug 2004). However, the lack of adequate information impedes taking informed decisions. Hence, it is crucial to maintain the individual component's LOD requirements. Especially in the process of designing a building, the components are associated with diverse levels of development within the same phase, such as load-bearing components can be described with a higher LOD than the interior fittings in the early design stages.

To our knowledge, there is no approach for formally defining and maintaining multiple levels of development throughout the design stages. Neither is there a formal definition of a building component's level of development nor is there an explicit description of the fuzziness of its geometric and semantic information. Therefore, the multi-LOD meta-model is proposed in order to:

– Define component types' LOD requirements
– Model information uncertainty
– Represent a building model on multiple stages
– Describe the relationships between LODs
– Check the consistency between LODs

To manage the requirements of the individual building component types for a specific LOD, a component type is associated with multiple LOD definitions. An LOD definition consists of two separate groups: one for defining the geometric representation and alphanumerical attributes, and another for specifying the semantic alphanumerical attributes. This separation helps to achieve and to maintain the semantic-geometric coherence of the overall model (Stadler, Kolbe 2007; Clementini 2010). Finally, the building model is presented by creating multiple instances from the defined component types.

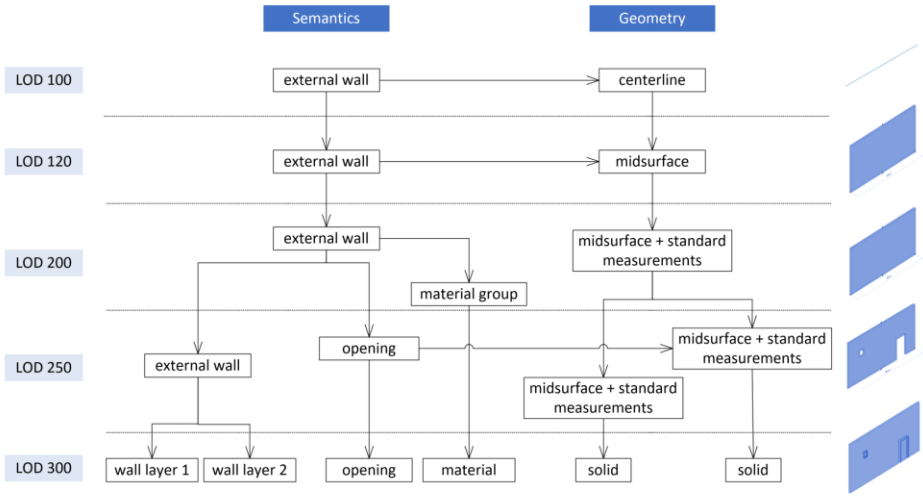

Figure 1. Separation of geometry and semantics on different LODs of an *external wall*

3.2 *Separation of geometry and semantics*

The multi-LOD meta-model aims to maintain a clear separation between the building components' semantic and geometric requirements. In terms of geometry representation of a building component, it is refined along with increasing the level of development. For example, as demonstrated in Figure 1 at LOD 100, an *external wall* is presented as a *centerline*, since in the next LODs additional information is available, such as a *thickness* and *material*, it is possible to render the wall *solid* model in its 3D shape and dimensions. This kind of hierarchical development of a *centerline* towards a *solid* model defines the dependencies between the geometric representations on the different levels of development. Accordingly, the relationships between the semantic requirements are determined, which supports checking the consistency between the multiple LODs.

3.3 *Alphanumerical attributes and fuzziness*

With incrementing the LOD, additional attributes become available, for example, the *construction type* and *material* information can be determined starting from LOD 200. In some cases, it is uncertain whether a specific attribute is available or can be estimated from a specific LOD. Thus, the multi-LOD model provides the ability to specify whether an attribute is mandatory or optional as well as offering a level of precision in specifying the attribute's assigned value in case of uncertainty. The level of precision in assigning the attribute's value is related to its type; it might be achieved by specifying an abstract value, such as a classification, or a fuzziness range. With that said it is possible to model and analyze the known uncertainties of the building model at the early design stages where uncertainty is at its highest.

Attributes	LOD 120		LOD 200		LOD 300	
	existing	fuzziness	existing	fuzziness	existing	fuzziness
Dimensions	✓	±10 cm	✓	-	✓	-
Wall thickness			✓	±10 cm	✓	±5 cm
Opening position			✓	±10 cm	✓	±5 cm
Opening percentage			✓	±5 cm	✓	-
Material			✓	material group (wood, concrete,..)	✓	material

Figure 2. Example of assigning geometric-semantic attributes and fuzziness of an *external wall*

Figure 2 provides geometric and semantic attributes of an *External Wall* component type for LODs 120 to 300. The *surface* dimensions exist starting from LOD 120 with a permissible fuzziness range of ±10 cm, while no fuzziness is permitted afterward. Additionally, the information describing *wall thickness* and *opening position* are available starting from LOD 200 with ±10 cm of fuzziness and then reduced to ±5 cm on LOD 300. Considering a different type of fuzziness, the information about *material* can be available from LOD 200, where at this level; it is defined by specifying the *material group*, such as *Ceramic*, whereas afterward on LOD 300 the exact material value, like *Brick*, should be assigned.

4 META-MODEL DESIGN

The multi-LOD meta-model design provides means for defining a project-specific data-model, incorporating formal LOD definitions for individual component types. It introduces two layers: *data-model level* defines the component types as well as their geometric and semantic requirements for each LOD. The *instance level* represents the building model by instantiating multiple instances of the component types defined on the data-model level.

The meta-model design complies with the object-oriented modelling principles, which offers high flexibility and extensibility. It allows for a dynamic definition of any component types as well as their attributes for the different LODs. This provides the flexibility required when dealing with different con-

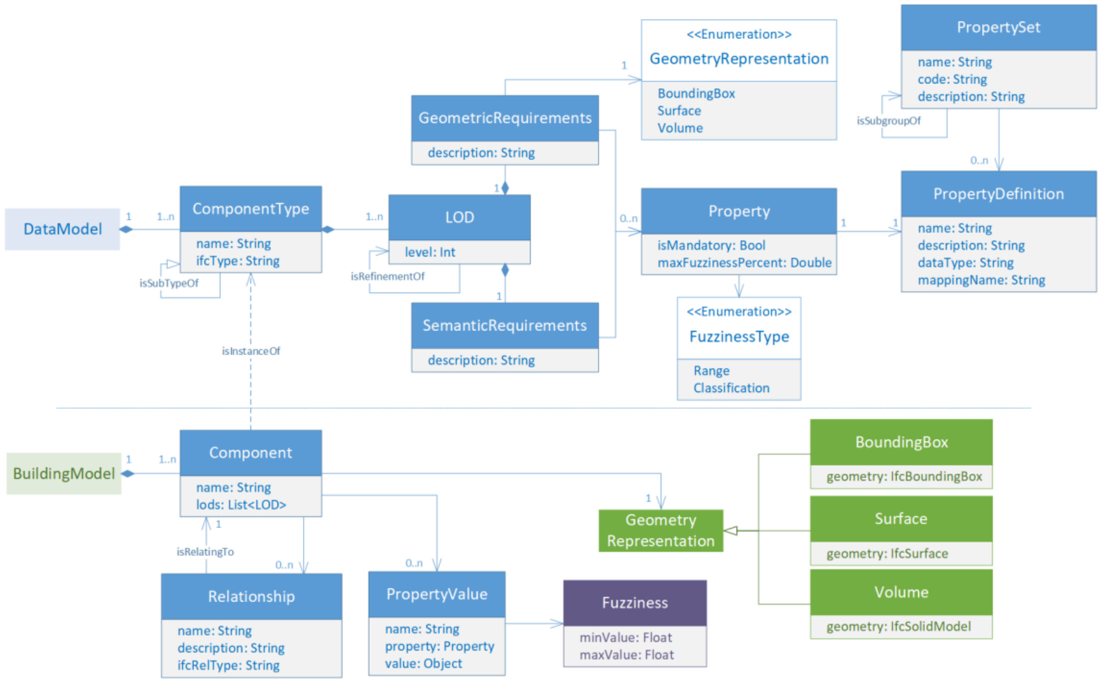

Figure 3. Multi-LOD meta-model (UML diagram)

struction types, different domains, and different analysis tools. At the same time, the meta-model provides a consistent way to query information about LOD definitions on both the *data-model level* as well as the *instance level*. Thereby, as illustrated in Figure 3, a component type definition is represented as a separate class, where it is linked to an IFC type, *IfcWall* as an example, and associated with a list of LOD definitions. The component types are mapped to instances of the IFC data model, on the one hand, to make use of the geometry representations defined there and on the other hand to experiment with real-world data produced by IFC-capable BIM authoring tools.

An LOD definition is produced out of two objects, geometric and semantic requirements. Both requirements are explicitly described in the form of properties. The details of each property are determined in addition to the permissible fuzziness and geometry representation. The properties are managed by means of grouping, the *PropertySet* class. A *PropertySet* includes multiple *PropertyDefinition* instances defining property details but excluding its fuzziness. At the same time, the fuzziness type and maximum percentage as well as whether the property is mandatory are specified when assigning a *PropertyDefinition* to an LOD property. This brings multiple advantages, including decoupling the property definition from the LOD requirements, and flexibility of using the same property definition in multiple LODs along with different fuzziness.

In some cases, multiple components fall under the same category, such as *Heating, Ventilation, and Air Conditioning* (HVAC) systems, and share several properties. Hence, the *ComponentType* class supports defining sub-types of a specific component type through inheritance. This means a sub-type inherits the parent component type's requirements in addition to specifying additional specific requirements.

After defining the component types' requirements, the building model is represented by multiple instances of the available types. An instance is assigned to a geometry representation, which complies with IFC, such as *IfcSurface*, and its properties are filled with values. In terms of fuzziness, its range is automatically transferred from the maximum fuzziness percentage defined at the component type level, for example, 4% and an attribute value of 250 cm is translated into a range of ±10 cm. Moreover, at the instance level, it is possible to increase the limitation of the range values, such as to be between -5 cm and +7 cm. Finally, the connections between the individual components within the same LOD, including aggregation and association, are presented through the *Relationship* class. With that said, the meta-model allows checking if the instance of a given type on a particular LOD complies with the requirements defined in terms of semantic fuzziness and geometric representation.

5 USE CASE: EMBODIED ENERGY ANALYSIS

Life Cycle Assessment (LCA) is one of the most established and well-developed methods for assessing the potential environmental impacts and resource consumption throughout a product's life-cycle (Ness et al. 2007). As one of its applications, LCA is used to calculate the *Embodied Energy* which is represented as the sum of non-renewable energy consumption during the life cycle (Merkblatt 2010). Performing the LCA calculation involves multiple geometric and semantic information of the building model, including the building location, dimensions, number of storeys and window-to-wall ratio. Additionally, custom energy-related attributes are required for each

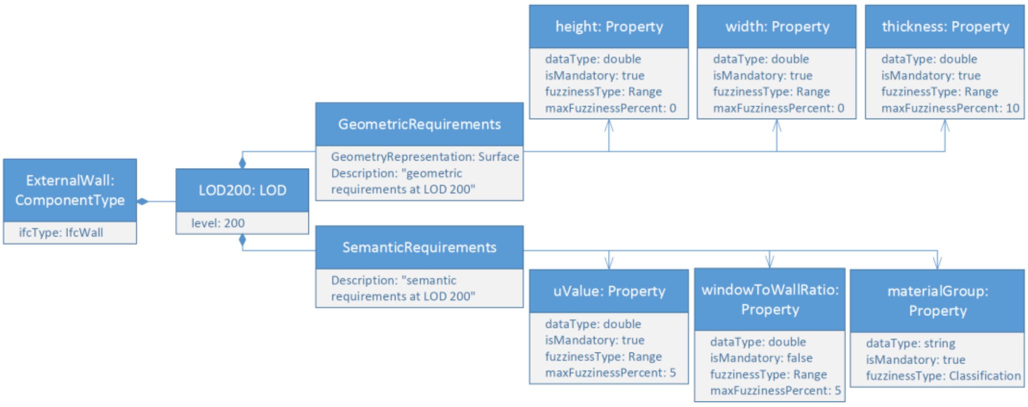

Figure 4. Data-model level example of an *external wall* at LOD 200 supporting embodied energy calculation (UML diagram)

component and need to be transferred when exchanging the model, such as the *Thermal transmittance* (U-value).

In order to include this information in the model, it has to be provided in a correct way in the BIM authoring tool. Here, the multi-LOD meta-model comes into play where it allows defining the data-model of the individual component types. For instance, Table 1 lists the component types and their required attributes for the LCA calculation on LODs 120, 200 and 300.

Table 1. Required components and attributes for LCA calculation in different LODs.

LOD	Available Components	Required Attributes
120	building	dimensions, location
200	floor, roof, wall	thickness, material, U-value
300	windows	material, U-Value

LOD 120 is limited to the building model's generic information. Whereas, at LOD 200 and 300, information about the windows and walls including the *thickness* and *material* become available. Therefore, a *U-value* of each component type can be provided. Having these attributes specified at the data-model level guarantees their association at the instance level and provides a way to check the model's validity. Based on the data listed in Table 1, Figure 4 illustrates a data-model level of an external wall at LOD 200. The geometric requirements include the *height, width,* and *thickness*, where the thickness attribute permits a fuzziness range of 10%. For the semantic requirements, the *U-value* and the material group are required. As the *window-to-wall* ratio is expected to be precise at LOD 300, it is not required and allows a fuzziness range of 5%.

6 IMPLEMENTATION

To evaluate the proposed multi-LOD model for practical use, the data-model level is implemented as a webserver and *User Interface* (UI). The UI provides a user-friendly way for defining disciplines, levels of development, property sets and component types. Figure 5 provides an overview of the system design.

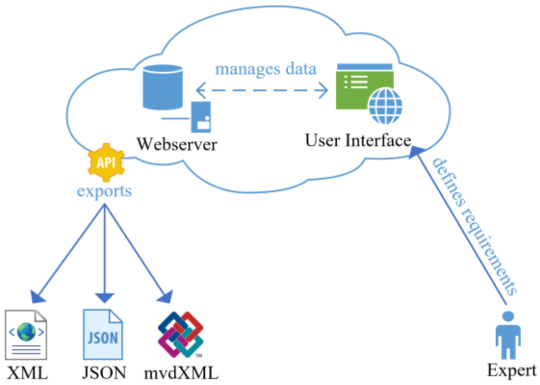

Figure 5. Data-model level system design

The main idea is that every discipline is capable of defining its own property sets, and then assigning particular properties to a specific component type's LOD. The property sets management screen is demonstrated in Figure 7. A property set can have sub-sets in order to minimize the properties redundancy. Additionally, a property is assignable to multiple disciplines. Finally, the properties are associated to an LOD at the component types' screen. Figure 6 shows the *WallType* component details screen. The *General* tab is for defining the component name, IfcType and description. Whereas the second tab *Requirements* facilitates associating every LOD with properties including a specification of their fuzziness. The properties are grouped based on their *Property Set* name, following the naming scheme *Pset_**, for

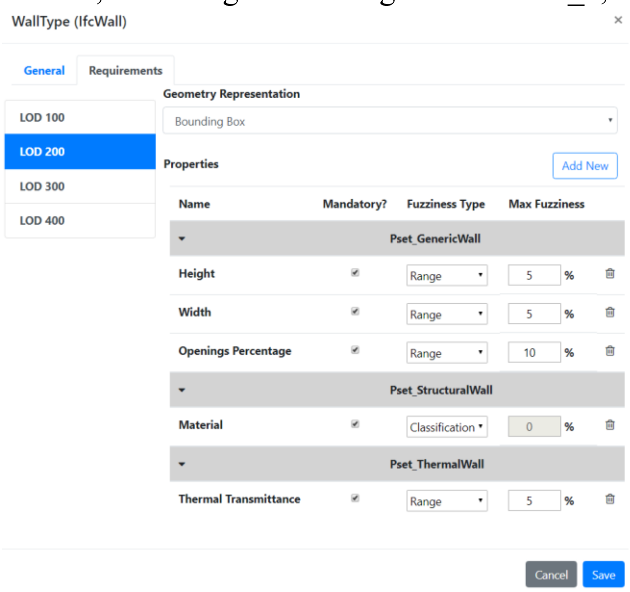

Figure 6. Component details screen of *WallType* (UI prototype)

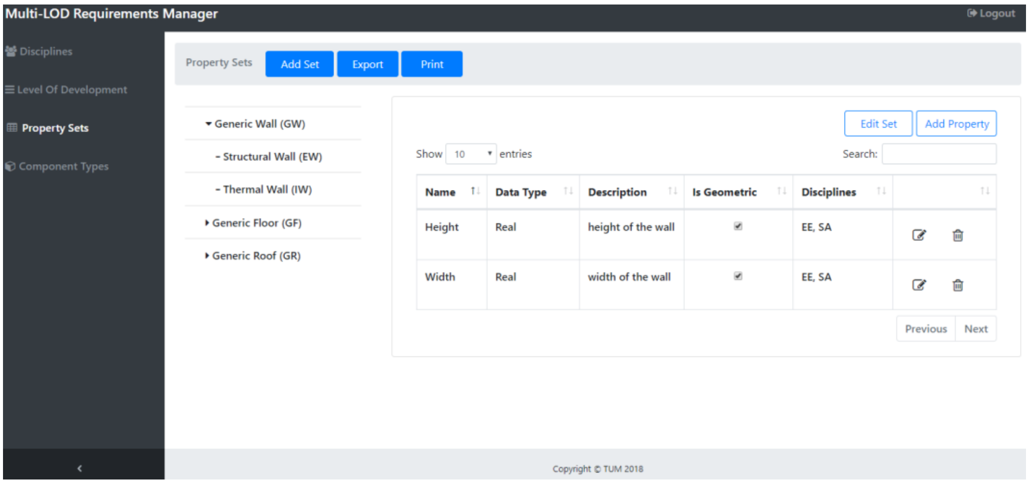

Figure 7. Property Sets management screen (UI prototype)

instance, *Pset_ThermalWall*. For improving the usability and increase the data integrity, the bsDD's *Application Programming Interface* (API) is employed. It assists the process by listing the commonly known IFC elements, properties, and classifications to the user. Consequently, this mapping to the bsDD's *GUID* provides additional context and meaning to each value, which improves interoperability between different disciplines and assists the model analysis.

The multi-LOD webserver stores the component types' requirements into a relational database and exports them as XML and JSON formats using *REpresentational State Transfer* (REST) API as shown in Figure 8. To facilitate using these requirements as exchange requirements and validate their existence, the webserver exports them into the common formats supported by the BIM authoring tools, such as *PropertySets* file provided by Autodesk Revit, and translates the requirements into mvdXML rules. Hereby, it is possible to use the requirements for external services, such as a Revit plugin, for automatically generating and ensuring the exchanged building models attributes completeness.

```xml
<ComponentTypes>
▼<ComponentType id="1" name="WallType" ifcType="IfcWall">
  ▼<LOD id="2" componentId="1" level="200">
    ▼<PropertySets>
      ▼<PropertySet id="1" name="Pset_GenericWall">
          <Property id="1" name="maxHeight" description="Maximum height"
          dataType="Real" isMandatory="true" fuzzinessType="1"
          fuzzinessTypeName="Range" maxFuzzinessPercent="5.0"
          propertyDefinitionId="1" isGeneric="true"/>
          <Property id="2" name="maxWidth" description="Maximum Width"
          dataType="Real" isMandatory="true" fuzzinessType="1"
          fuzzinessTypeName="Range" maxFuzzinessPercent="5.0"
          propertyDefinitionId="2" isGeneric="true"/>
          <Property id="3" name="maxLength" description="Maximum Length"
          dataType="Real" isMandatory="true" fuzzinessType="1"
          fuzzinessTypeName="Range" maxFuzzinessPercent="5.0"
          propertyDefinitionId="3" isGeneric="true"/>
          <Property id="5" name="openingPercentage" description="Windows
          Openings percentage" dataType="Real" isMandatory="true"
          fuzzinessType="1" fuzzinessTypeName="Range"
          maxFuzzinessPercent="10.0" propertyDefinitionId="5"
          isGeneric="false"/>
        </PropertySet>
      ▼<PropertySet id="4" name="Pset_StructuralWall">
          <Property id="6" name="materialGroup" description="The main material
          classification" dataType="Text" isMandatory="true" fuzzinessType="2"
          fuzzinessTypeName="Classification" maxFuzzinessPercent="0.0"
          propertyDefinitionId="6" isGeneric="false"/>
        </PropertySet>
      ▼<PropertySet id="2" name="Pset_ThermalWall">
          <Property id="4" name="thickness" description="Wall thickness"
          dataType="Real" isMandatory="true" fuzzinessType="1"
          fuzzinessTypeName="Range" maxFuzzinessPercent="10.0"
          propertyDefinitionId="4" isGeneric="true"/>
          <Property id="7" name="thermalTransmittance" description="Thermal
          Transmittance (u-value)" dataType="Real" isMandatory="true"
          fuzzinessType="1" fuzzinessTypeName="Range" maxFuzzinessPercent="5.0"
          propertyDefinitionId="7" isGeneric="false"/>
        </PropertySet>
      </PropertySets>
    </LOD>
    <Parent/>
  </ComponentType>
</ComponentTypes>
```

Figure 8. Example of a *WallType* LOD200 XML format

7 CONCLUSION AND FUTURE WORK

The multi-LOD meta-model offers a high-level interface that provides a consistent way for defining and querying LODs in term of their semantic and geometric requirements. As the LOD requirements take into account the permissible fuzziness, the known uncertainties are explicitly modelled, which delivers great advantages in assessing and verifying the model consistency in the early design stages. The meta-model introduces two layers, *data-model level* and *instance level*. This offers a high degree of flexibility in defining per-project LOD requirements and facilitates formal checking of their validity, such as requiring specific information for *Embodied Energy* calculations, *Building Performance* simulations, or *Structural* analysis.

As part of evaluating the model, the *data-model level* is implemented in a webserver and a corresponding user-interface. The system provides a means for managing the exchange requirements between the project disciplines for every LOD. In this way, the requirements' consistency, correctness, and completeness are maintained. Additionally, the system exports the exchange requirements into JSON, XML, and an automatically generated mvdXML rules to encourage their integration in the modeling process. As a next step, further research is necessary to develop a methodology for describing the detailed refinement relationships of the building elements and checking their consistency.

8 ACKNOWLEDGEMENTS

We gratefully acknowledge the support of German Research Foundation (DFG) for funding the project under grant FOR 2363.

9 REFERENCES

Bazjanac, Vladimir (2008): IFC BIM-based methodology for semi-automated building energy performance simulation. Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley, CA (US).

Biljecki, Filip; Ledoux, Hugo; Stoter, Jantien (2016): An improved LOD specification for 3D building models. In *Computers, Environment and Urban Systems* 59, pp. 25–37.

BIMForum (2017): 2017 Level of Development Specification Guide. Available online at http://bimforum.org/lod/, checked on 1/11/2018.

Bjorkhaug, L.; Bell, H. (2007): Ifd: IFD in a Nutshell. In *SINTEF Building and Infrastructure*.

Borrmann, André; Flurl, Matthias; Jubierre, Javier Ramos; Mundani, Ralf-Peter; Rank, Ernst (2014): Synchronous collaborative tunnel design based on consistency-preserving multi-scale models. In *Advanced Engineering Informatics* 28 (4), pp. 499–517.

BSI (2017): PAS 1192-2:2013 Specification for information management for the capital/delivery phase of construction projects using building information modelling. Available online at http://shop.bsigroup.com/Navigate-by/PAS/PAS-1192-22013/, checked on 1/11/2018.

buildingSMART (2016): Data Dictionary. Available online at https://www.buildingsmart.org/standards/standards-tools-services/data-dictionary/, checked on Jan 2018.

Chapman, I. (2013): An introduction to Uniclass2. Available online at https://www.thenbs.com/knowledge/an-introduction-to-uniclass-2, checked on 3/22/2018.

Chipman, T.; Liebich, T.; Weise, M. (2012): mvdXML: Specification of a standardized format to define and exchange Model View Definitions with Exchange Requirements and Validation Rules: BuildingSmart.

Chiu, Mao-Lin (2002): An organizational view of design communication in design collaboration. In *Design studies* 23 (2), pp. 187–210.

Clementini, Eliseo (2010): Ontological impedance in 3D semantic data modeling. In Eliseo Clementini (Ed.): Ontological impedance in 3D semantic data modeling. 5th 3d Geoinfo Conference (Ed. by TH Kolbe, G. König & C. Nagel), ISPRS Conference (Vol. 38, p. 4)., vol. 38, p. 4.

Eastman, Charles M.; Eastman, Chuck; Teicholz, Paul; Sacks, Rafael (2011): BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors: John Wiley & Sons.

GCR, NIST (2004): Cost analysis of inadequate interoperability in the US capital facilities industry. In *National Institute of Standards and Technology (NIST)*.

Hausknecht, Kerstin; Liebich, Thomas (2017): BIM-Kompendium: Building Information Modeling als neue Planungsmethode: Fraunhofer IRB Verlag.

Hietanen, Jiri; Final, Status (2006): IFC model view definition format. In *International Alliance for Interoperability*, pp. 1–29.

Hooper, Martin (2015): Automated model progression scheduling using level of development. In *Construction Innovation* 15 (4), pp. 428–448.

Karlshøj, Jan; See, R.; Davis, D. (2012): An Integrated Process for Delivering IFC Based Data Exchange. Available online at https://bips.dk/files/integrated_idm-mvd_processformats_14_0.pdf, checked on 4/15/2018.

Knotten, Vegard; Svalestuen, Fredrik; Hansen, Geir K.; Lædre, Ola (2015): Design management in the building process-a review of current literature. In *Procedia Economics and Finance* 21, pp. 120–127.

Kolltveit, Bjørn Johs; Grønhaug, Kjell (2004): The importance of the early phase: the case of construction and building projects. In *International Journal of Project Management* 22 (7), pp. 545–551.

Liebich, T.; Adachi, Y.; Forester, J.; Hyvarinen, J.; Richter, S.; Chipman, T. et al. (2013): Industry Foundation Classes IFC4 official release. In *Online: http://www. buildingsmart-tech. org/ifc/IFC4/final/html*.

Liebich, Thomas (2013): IFC4—The new buildingSMART standard. In Thomas Liebich (Ed.): IC Meeting. IFC4—The new buildingSMART standard. bSI Publications Helsinki, Finland.

Merkblatt, S. I.A. (2010): 2032 (2010): Graue Energie von Gebäuden. In *Zürich, Schweizerischer Ingenieur-und Architektenverein*.

Ness, Barry; Urbel-Piirsalu, Evelin; Anderberg, Stefan; Olsson, Lennart (2007): Categorising tools for sustainability assessment. In *Ecological Economics* 60 (3), pp. 498–508. DOI: 10.1016/j.ecolecon.2006.07.023.

OmniClass, A. (2012): A strategy for Classifying the Built Environment: USA. Available online at http://www.omniclass.org/, checked on 1/11/2018.

Stadler, Alexandra; Kolbe, Thomas H. (Eds.) (2007): Spatio-semantic coherence in the integration of 3D city models. Proceedings of the 5th International Symposium on Spatial Data Quality, Enschede.

van Berlo, LAHM; Bomhof, F. (2014): Creating the Dutch national BIM levels of development. In Van Berlo, LAHM and Bomhof, F (Ed.): Creating the Dutch national BIM levels of development, pp. 129–136.

Yang, Donghoon; Eastman, Charles M. (2007): A Rule-Based Subset Generation Method for Product Data Models. In *Computer-Aided Civil and Infrastructure Engineering* 22 (2), pp. 133–148.

Young, N. W.; Jones, Stephen A.; Bernstein, Harvey M.; Gudgel, John (2009): The business value of BIM-getting building information modeling to the bottom line. In *Bedford, MA: McGraw-Hill Construction* 51.