Minimized communication protocol based on a multi-LOD meta-model for adaptive detailing of BIM models

Ata Zahedi[1], Jimmy Abualdenien[2], Frank Petzold[1], André Borrmann[2]
[1]Chair of Architectural Informatics, Technical University of Munich, Germany
[2]Chair of Computational Modeling and Simulation, Technical University of Munich, Germany
ata.zahedi@tum.de , jimmy.abualdenien@tum.de (Lead author email address)

Abstract. A continuous generation and evaluation of design variants characterize the conceptual architectural design stages. Variants comparison is a crucial process for making design-detailing decisions. Objectifiable criteria, including results of simulations and analysis, used for evaluation and comparison of design variants can legitimize decisions as the design process proceeds. A major challenge today is the management of design information and collaboration among several actors in a building project. Yet, a large portion of the Architecture, Engineering and Construction (AEC) industry deals with conventional methods to exchange design information. The growing use of building information models is promising but even the most recent developments and practices still rely heavily on human-readable protocols and issue management systems. Considering the potential of schematized computer-readable communications to be analyzed and used for future references and case-based reasoning systems. This paper proposes a novel minimized communication protocol that aims to introduce a computer-readable, yet adaptive universal method which works on schematized information exchange requirements (templates) for different use cases. The concept is demonstrated using an example scenario.

1. Introduction

While the building sector alone consumes around 50% of all the materials obtained from the earth, the architecture, construction, and engineering (AEC) industry still fall greatly behind other advanced industries, such as automotive industry in terms of efficiency and productivity (Hegger *et al.*, 2012). The early conceptual architectural design stages comprise a highly creative process that is characterized by a continues endeavor for the creation of variants, evaluating their performance, and a progressive detailing of their components. Decisions taken in the early stages have a significant influence on the performance and costs of the developed solution. Typically, decisions about the building's envelope, shape and façade are made in a short period of time mostly based on the architect's empirical knowledge, generic values, or rules of thumb (Attia *et al.*, 2012). Since the use of analytical tools requires a mostly completed and very detailed model, the application of simulation tools in the early stages, including the assessment of different design options, is not fully attainable.

Today, the use of digital building models for the design, construction, and operation of buildings is already a hot trend in the building industry. Building Information Modelling (BIM) plays an important role in the construction industry resulting in a significant modernization of the working procedures (Borrmann *et al.*, 2018). However, the use of BIM in the early design stages is difficult to master since this method requires more planning effort and leads to highly detailed designs, which are not necessarily apt for these early stages (Azhar, 2011). Nevertheless, certain potential lies in the direct reuse of data held by digital models for diverse analysis and simulation tasks, e.g. the structural analysis or the energy performance simulation (Borrmann *et al.*, 2018). To address this gap, the research project EarlyBIM (FOR2363)[1], funded by the 'Deutsche Forschungsgemeinschaft' (DFG), is dedicated to the development of strategies and methods for adaptively detailing the partially incomplete and vague design

[1] https://for2363.blogs.ruhr-uni-bochum.de

models in order to assess and compare different design options. This paper proposes an upgrade for our adaptive minimized BIM-based protocol for communication between architects and various specialist planners (domain experts) in the design stages (Zahedi and Petzold, 2018b).

2. State of the Art

2.1 The Design Process in the Early Design Stages

As the building design evolves, many aspects of the design desire different domain-experts to evaluate and comment on them. Among the challenges during the early design phases are the incompleteness of input information and immensity of design decisions regarding numerous variations of design parameters. Moreover, since one way to overcome the gap between data requirements of simulation tools and model quality (degree of detailing and abstraction) in conceptual design is to simplify simulation engines and make assumptions, hence the leading results may appear to be exact but then again are uncertain. Studies show that the AEC professionals become so occupied with managing design information, including manually integrating and coordinating domain-specific design information and representations that, in the end, they manage to create only few design alternatives (Flager *et al.*, 2009).

Another way to close the gap is to decide on and fill in the missing data. This pretty much means passing through the conceptual design and reaching into more detailed design. Moreover, the architect (other than having a knowhow) is not and should not be an expert in all aspects of design. The alternative according to the adaptive detailing concept is to suggest and provide proper detailing, corrections and modifications from domain-expert to the architect. Nonetheless, without the architect's approval, no change will be recognized. These suggestions and modifications received from the domain-experts are referred to as ***Feedback*** (Zahedi and Petzold, 2018a).

Exploring different paths and developing several design variants is a fundamental recurrent feature in the creative activity of building design. Every building design task/activity starts with some requirements demanded by the client, followed by some city restrictions and regulations. As the planning activity moves forwards, more suitable and preferred variants are explained in more detail while other drafts are rejected or discarded. Proper management of design variants and avoiding data redundancies has been a topic of many researches. Mattern & König (2018) introduced so called option categories to organize the possibilities that evolve in design process. Furthermore, the authors suggested graph models to manage interdependencies within different design variants. Nevertheless, due to complex situations and restrictions, the authors suggest that the creation of invalid combinations are to be avoided (Mattern and König, 2018). This paper aims to utilize the concept as mentioned above later on in its backend system during the temporal development between options and design variants.

2.2 BIM Collaboration Format (BCF)

There exists strong evidence in the literature that knowledge, especially in complex projects, is a socially attained phenomena. In other words, obtaining appropriate information for better decision-making in early design stages could be achieved by integrating as many domain-experts as early as possible in the design process (Kvan, 2000). Turk (Turk, 2000) argued that based on speech act theory (Searle *et al.*, 1980) the primary goal of communication is requesting and fulfilling commitments (between participants) rather than just exchanging information. In other words, when the structural engineer advises to the architect that the supporting columns are too far apart, besides the information exchange he is trying to commit the architect to fix it.

Taking into account the significance of internal communications in building design, the buildingSMART organization (formerly known as International Alliance for Interoperability - IAI) introduced an open standard called BIM Collaboration Format (BCF) to enable BIM-based workflow and communication between different software tools. Through BCF, actors create topics (e.g. issues, proposals, change requests) that comprise various types of communication information (e.g. type, description, comments, screenshots and viewpoints inside model) addressed in BIM data models. This way only topics and not the entire bulky BIM models will be exchanged between different BIM authoring tools. In 2014, BCF-V2 was equipped with a RESTful API and the possibility to exchange machine-readable topics, using so called BIM-Snippets. Example of a BIM-Snippet could be a partial IFC file (buildingsmart-tech.org, 2019)

There's no doubt that BCF is a well-established and -supported communication protocol that is vendor-neutral and opensource. Almost all commercial BIM software have implemented BCF in their systems too. However, BCF XML is dominantly used for grasping human-readable data regarding issue management. Even though BCF v2.1 is capable of encompassing so-called BIM-Snippets to encapsulate schematized arbitrary data, BCF is still mostly used to address human-readable issue management in AEC and examples of implementing BIM-Snippets are not yet commonly introduced. The authors of this paper, while not arguing the reputation and usefulness of BCF, merely intend to present another approach for BIM-based communication workflow and structured dialogue in early stages of design. A machine-readable structured dialogue between project participant could help the AEC industry with its fragmented nature to learn from various projects and communications. Nevertheless, aiming for computer-readable communications does not necessarily eliminates the need for social interactions. Still, in our protocol, there is room for comments and freestyle text and explanations. However, since the AEC industry is fragmented with many small and medium-size companies, and in most cases, collaborations between companies are limited to the duration of one project, we believe that being able to learn from the partnerships and communications of various building projects might be of great help to improve this less advanced industry.

3. Multi-LOD Meta-Model and Adaptive Requirements Template

A major challenge for using BIM-authoring tools in the early design stages is that despite the insufficient information available in these stages, a BIM model appears precise and certain, which can lead to false assumptions and model evaluation. Additionally, although the Level of Development (LOD) (BIMForum, 2019), is a well-known concept that describes the quality and quantity of the information contained within a BIM model, the currently available specifications are informal, textual descriptions and graphical illustrations.

To define the different component types' LOD requirements in a formal manner, Abualdenien & Borrmann (Abualdenien and Borrmann, 2019) have introduced a multi-LOD meta-model. As illustrated in Figure 1, the meta-model consists of two layers: (1) the *data-model level* to define the component types' LOD requirements, and (2) and the *instance level* to represent the actual building components as well as their relationships. The data-model level facilitates the definition of various component types, such as *wall*, *window*, *and slab*. Each component type is associated with multiple LOD definitions and linked to an Industry Foundation Classes (IFC) (buildingSMART, 2019) class, which makes it possible to use the rich geometry representations and test with real-world building models. An LOD definition includes a geometry representation, *BoundingBox* as an example, as well as geometric and semantic properties, which are assigned to a vagueness type and maximum percentage. The explicit specification of vagueness to the LOD requirements facilitates a model's analysis and supports informed decision-making in the early design stages. The instance level, on the other hand, represents the building components and maps their properties as well as geometry representation to the requirements defined at the data-model level.

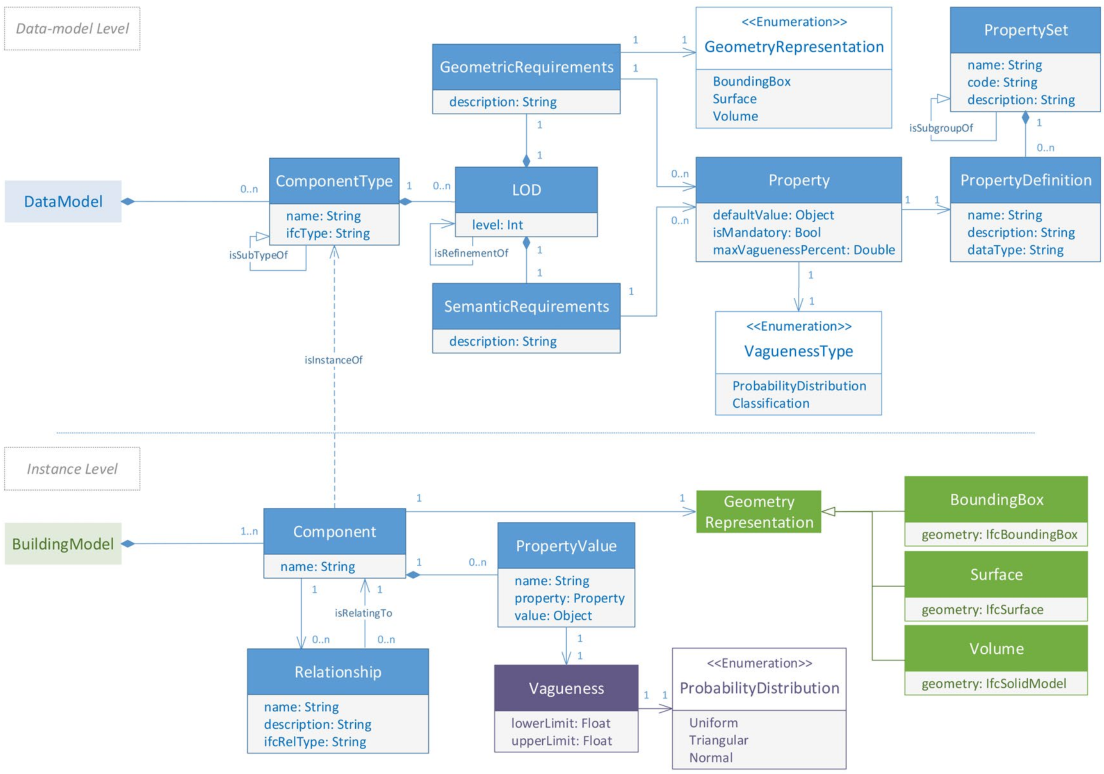

Figure 1: Multi-LOD meta-model (Abualdenien and Borrmann, 2019)

Besides communicating the information vagueness to the project's participants, a major benefit of defining the requirements at the multi-LOD meta-model is to formalize the vague input used by the different analysis tools. Accordingly, the domain experts collaborate to alleviate the impact of the information vagueness. As demonstrated in Figure 2 and 3, the concept is implemented as a web-server with a user interface (UI) providing the possibility for managing and checking exchange requirements between different domains (Abualdenien and Borrmann, 2019). These exchange requirements defined on the multi-LOD meta-model are used as the basis for the *aLODx* templates that are used in communication protocol proposed in this paper (please refer to Section 4). Accordingly, depending on the analysis type and the design stage, which contains diverse levels of development for the different component types, every domain-expert specifies a list of building components in addition to a set of properties.

4. Adaptive Minimized Communication Protocol Based on BIM

In a general sense, any request for analysis during early design stages could yield to three possible outcomes as following:

1. The design model is extremely immature and therefore not capable of analysis.

2. Some geometrical details or semantical information which are expected to be present in the model are missing. In this case, the analysis might be performed with simplification of simulation tools (or making assumptions regarding the missing input data) or the domain-expert reports the missing information to the architect as this information is required to carry on with the analysis.
3. The analysis could be performed, and the results will be presented to the architect.

Making design decisions requires knowledge and experience in various domains; hence, the architect confronts multiple challenges while overcoming the lack of information by making estimations. Therefore, the early domain-experts involvement facilitates evaluating the consequences of the design decisions as well as provides the required assistance by providing design suggestions/options.

This paper differentiates design *variants* and *options*. The term variants are given to the design models developed by the architect, whereas options represent the feedback and suggestions provided by the different experts. The options are to be used (accepted or rejected) by the architect as a source of inspiration, suggestion, or a possible solution provided by a domain-expert. It is important to note that throughout the design process, the architect is the team leader and the only responsible for creating and modifying the design model/variants. At the same time, the architect is the team member who is liable for updating the design state and approving the final decisions.

To support communication among the project participants in a means of structured dialogue, the authors propose establishing a Common Data Environment (CDE). Besides formalizing the communication, the proposed approach makes it possible to share bulk product and model data, as well as facilitating proposing design options to the architect for further detailing the available design variants. The proposed design options, so-called "feedback", which are provided by the domain-expert, consist of the following responses:

- report on the missing details in the design model
- possible options that could serve to fulfill the shortcomings in the design model
- the results of analysis or simulations when each of those options is selected

The proposed communication protocol focuses on staying as minimized and lite as possible, which means avoiding to send back and forth any actual digital BIM files. Even partial models (as options) will rest on the CDE, and the protocol will only inhale their links and globally unified IDs (*GUID*). In the case of updating/suggesting attributes and properties for already existing building components in the digital design model, these alphanumerical values will be included in the messages. But in the case of creating/suggesting new building components/objects, they will be only referred/linked to inside the messages.

The proposed communication system is consisting of two parts. One part would be an issue tracking system or so-called ticketing system. Via this part, just like any other ticketing system, requests and responses will be managed, and their progress will be controlled. Priorities can be set for each ticket, and their responsive person can also be traced back. Various tags can be assigned to tickets which makes the coordination and communication more seamless and transparently traceable. The other part would serve and inhale the essence of the feedback provided by various consultants and domain experts. In the next section, the method/function called "feedback" will be defined.

4.1 Feedback Function

This universal method based on its use case will receive different arguments. This adaptive feedback function along with its' arguments are explained one by one as follows:

Feedback (actionType, optionGroupID, GUID, aLODx, objectID, propertyID, value, rating)

***actionType*:** The first argument is called *actionType*. It represents the use case of the feedback function. Possible *actionTypes* are *missingObject, missingObjectProperty, createNewObject, deleteObject, updateObjectProperty*. *missingObject* is to report for some building components that are missing overall (e.g., all the openings are missing). *missingObjectProperty* refers to an attribute required for a building component that is unfilled. *createNewObject* refers to a newly created building component (probably as part of an option) by the domain-expert which will have a GUID that is unique and will be used to refer later on in the protocol. *deleteObject* refers to an unnecessary building component to be deleted by the architect. *updateObjectProperty* is used to suggest a value to fill in missing property.

***optionGroupID*:** This argument is used to group multiple suggestions. The proposed details could either be individually suggested, or they could be part of a package all sharing the same *optionGroupID*. To preserve the consistency and effectiveness of the suggested option, packaged suggestions should be treated (accepted or rejected) as a whole and not individually.

***GUID*:** This argument refers to the Globally Unique ID of a particular building component inside a given BIM model on the CDE. In case the *actionType* in the feedback function is *missingObject*, then this argument (*GUID*) would be the unique id of the building component that contains the missing objects. Like with the example of all the openings being absent, then this GUID could be the Building's GUID. But when the *actionType* is *createNewObject* then this argument will be the GUID of the temporarily newly created object that is suggested to the architect by the domain-expert. With the *missingObjectProperty* and *updateObjectProperty* as *actionType*, then this argument is the *GUID* of the building component that accordingly has a missing property, or a new value is suggested for its absent property. Finally, when the *actionType* is *deleteObject* then this argument is the GUID of the building component, which is recommended for deletion by domain-expert.

***aLODx*, *objectID*, *propertyID*:** In each design stage, the individual demand detailed requirements when exchanging building models. The presence of these requirements is crucial to perform model analysis, such as Life Cycle Assessment (LCA) and Structural Analysis. To support this kind of information exchange, the defined components and their LOD requirement at the multi-LOD meta-model (explained in Section 3), which are intended for a particular analysis, are called aLODx. This way, the requirements for a specific analysis are formulized. Inside the proposed concept, an aLODx serves as an adaptive lookup table (using ObejctID & PropertyID) that both the architect and the domain expert will refer to when they are communicating with each other. Accordingly, they can both be sure about what type of building a component (object) and what sort of attribute (property) they are talking about.

The requirements demonstrated in Figure 2 represent an example of the scheme (aLODx) of the properties required from the structural analysis specialists in the authors' research group. Using this screen, the domain-experts are capable of defining the property sets that are expected to be present in the exchanged model to perform their analysis. For each of the defined the properties, it is possible to specify the property's details as well as assigning a corresponding set of disciplines, including structural analysis (SA), operational energy (OE), and embodied energy (EE).

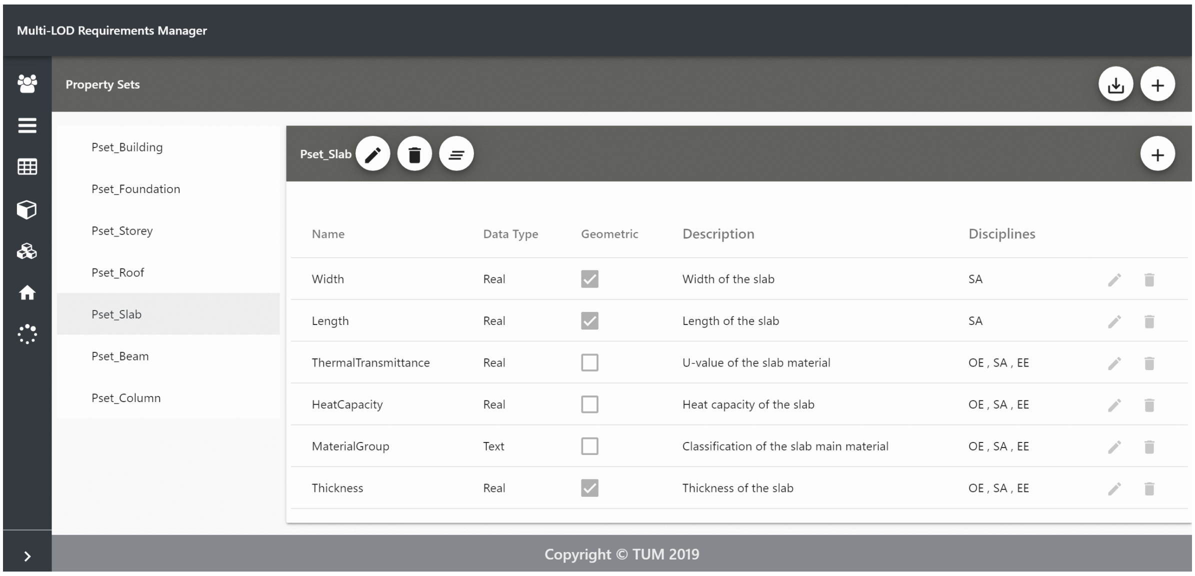

Figure 2: Prototype: property sets screen (Abualdenien and Borrmann, 2019)

After defining the exchange requirements, the corresponding component types are defined and mapped to their properties. Figure 3 illustrates the *FloorSlab* component type's details screen. The *General* tab defines the component name, IfcType, and description. Whereas the second tab *Requirements* facilitates the association of properties at every LOD, including a specification of the vagueness type and percentage.

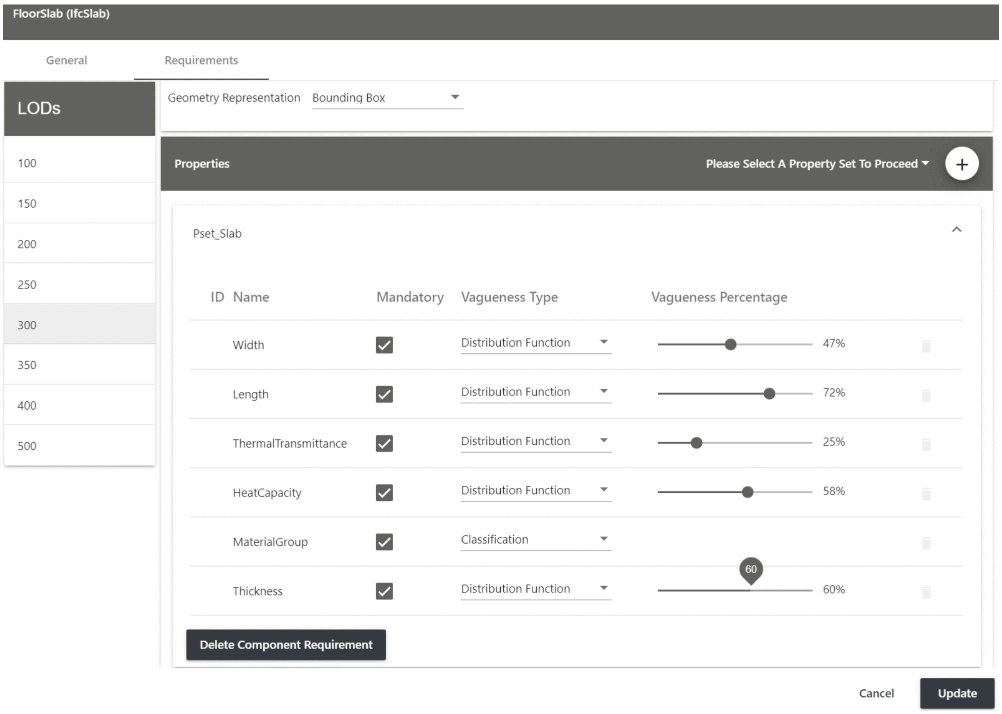

Figure 3: Prototype: component type details screen (Abualdenien and Borrmann, 2019)

value: This argument, if present depending on the actionType, will inhale the suggested values in the form of different options for the architect. Through the demonstrative example in the next section, the use of this argument will be more clarified.

rating: This new argument represents the normalized evaluation or rating/grade of the suggested option according to its performance with respect to the (based on the result of the) requested analysis. For example, this rating could be a number between 1 to 5, while 5 being the best.

5. Case Study: Structural Analysis in the Early Design Stages

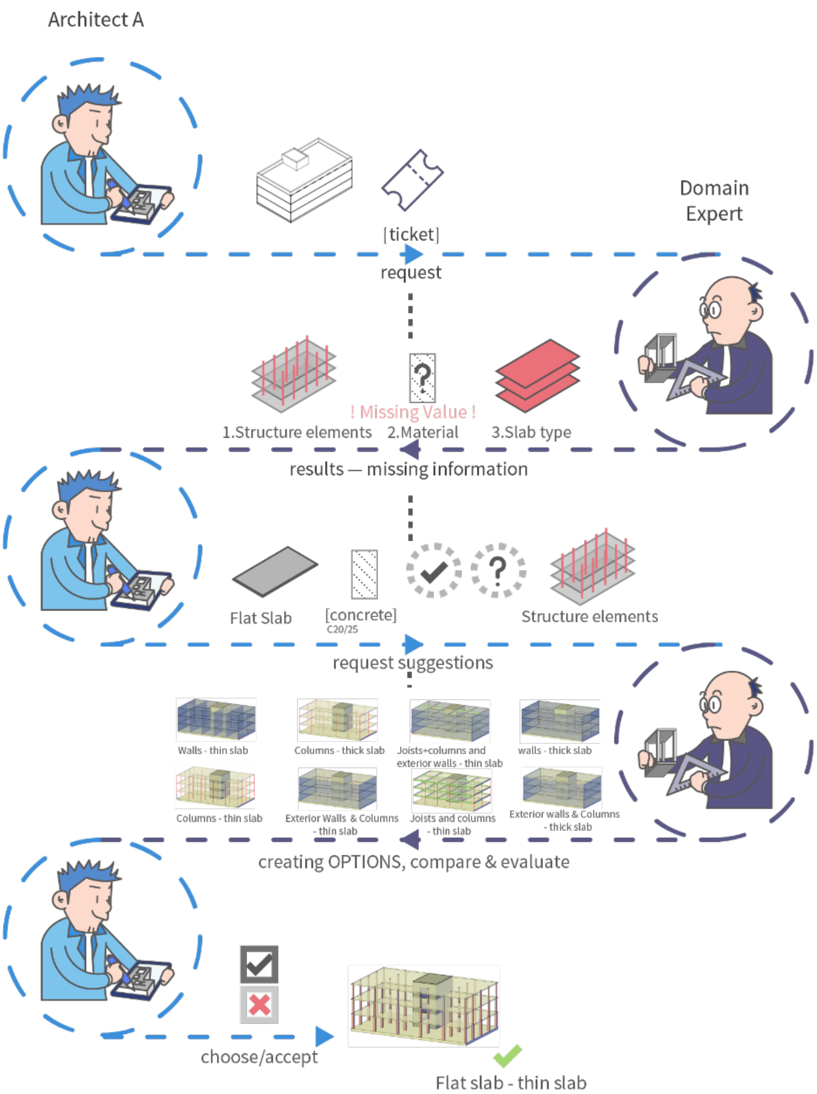

Figure 4: Demonstrative example using Structural analysis

Figure 4 demonstrates an example of the communication in case of structural analysis. Upon requesting for analysis, a ticket will be issued which clarifies the scope of analysis. The domain-expert, as the recipient of this ticket (following the link to the CDE) checks the model against the requirements and sends the architect a feedback regarding the missing details. In this simplified example slab type, main material group and structural elements are missing. The architect decides for flat slab and concrete but asks the structural engineer in return to give him options regarding the formation and possible types of structural elements. The structural engineer will provide three option, one with the mainly interior walls, one with exterior walls as load-bearing with columns and bars (joists), and the other with mainly columns and bars.

Furthermore, the structural engineer also provides the options with ratings according to their structural performance. At the same time, the architect may have also other criteria in mind for making design decisions, such as spatial layout flexibility or preferred window-to-wall ratio & etc. This makes the point again clear that he is the only responsible team leader to make design decisions.

With regard to our minimized communication protocol, the feedback function will be used in two stages and in multiple forms. At first it is used to report the missing details, namely in this example *missingObjectProperty* and *missingObject* as *actionType* are used in multiple instances. In the next round as the structural expert proposes options, *createNewObject* is used and various components (objects) are being created (temporarily) as part of a packaged option. All new objects within an option will share the same *optionGroupID*, but each will be referred to by their unique ID (*GUID*). Figure 5 shows the demonstration of the feedback function in the case of creating new columns.

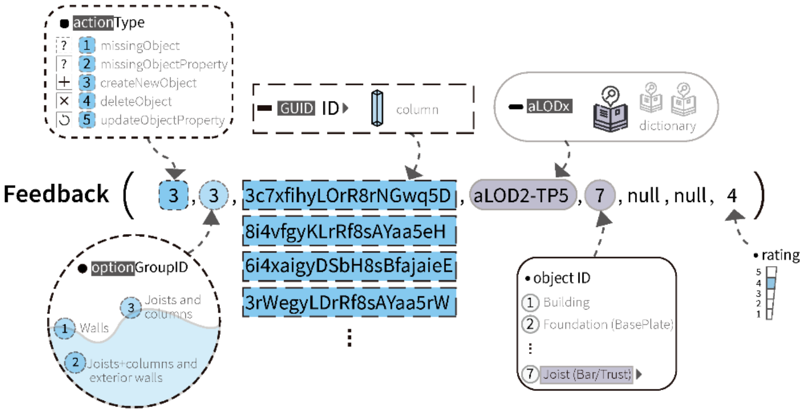

Figure 5: Feedback function explained

6. Conclusions

This paper introduced a BIM-based minimized communication protocol that is based on a meta-model approach for defining exchange requirements for the different component types at multiple LODs. The protocol enhances the collaboration and transparency of communication during the important early phases of design. This BIM-based communication protocol is built on schematized information requirements that are adaptively defined by domain-experts for the different analysis types and building project types using the multi-LOD meta-model. The protocol's aim is to store the history of these communications and use this accumulated data for supporting decision making in future use cases. By means of this minimized protocol, computer-readable communications take place, which can be filtered and analyzed for future use cases. Using this protocol, all communications, variant evaluations, and decision-making will be documented and traceable afterward for further use cases.

7. Acknowledgments

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

8. Reference list

Abualdenien, J. and Borrmann, A. (2019) ‘A meta-model approach for formal specification and consistent management of multi-LOD building models’, *Advanced Engineering Informatics*, pp. 20–42.

Attia, S. *et al.* (2012) ‘Simulation-based decision support tool for early stages of zero-energy building design’, *Energy and Buildings*, 49, pp. 2–15.

Azhar, S. (2011) ‘Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry’, *Leadership and management in engineering*, 11(3), pp. 241–252.

BIMForum (2019) *LOD | BIMForum: Level of Development Specification*. Available at: https://bimforum.org/lod/ (Accessed: 3 April 2019).

Borrmann, A. *et al.* (2018) ‘Building Information Modeling: Why? What? How?’ in *Building Information Modeling:* Springer, pp. 1–24.

buildingSMART (2019) *IFC Introduction - buildingSMART,* 3 April. Available at: https://www.buildingsmart.org/about/what-is-openbim/ifc-introduction/ (Accessed: 3 April 2019).

buildingSMART (2019) *Open BIM Collaboration Format (BCF) intro*. Available at: http://www.buildingsmart-tech.org/specifications/bcf-releases (Accessed: 3 April 2019).

Flager, F. *et al.* (2009) ‘Multidisciplinary process integration and design optimization of a classroom building’, *Journal of Information Technology in Construction*, 14(14), pp. 595–612.

Hegger, M. *et al.* (2012) *Energy manual: sustainable architecture*: Walter de Gruyter.

Kvan, T. (2000) ‘Collaborative design: what is it?’ *Automation in Construction*, 9(4), pp. 409–415.

Mattern, H. and König, M. (2018) ‘BIM-based modeling and management of design options at early planning phases’, *Advanced Engineering Informatics*, 38, pp. 316–329.

Searle, J.R. *et al.* (1980) *Speech act theory and pragmatics*. (10): Springer.

Turk, Z. (2000) ‘Communication workflow approach to CIC’, in *Computing in Civil and Building Engineering (2000)*, pp. 1094–1101.

Zahedi, A. and Petzold, F. (2018a) ‘Seamless integration of simulation and analysis in early design phases’, in *Proceedings of the 6th IALCCE Conference*, pp. 2007–2015.

Zahedi, A. and Petzold, F. (2018b) ‘Utilization of Simulation Tools in Early Design Phases Through Adaptive Detailing Strategies’, in *Proceedings of the 23rd CAADRIA, Beijing, China, 17-19 May 2018*, pp. 11–20.