

## ADAPTIVE MINIMIZED COMMUNICATION PROTOCOL BASED ON BIM

Ata Zahedi and Frank Petzold  
Technical University of Munich, Munich, Germany

### Abstract

The early conceptual architectural design phases are characterized by a constant interplay of creating variants and assessment of those variants as well as their consistent detailing. Variant comparison plays a significant role in achieving the desired final building. Objectifiable criteria used for the evaluation and comparison of design variants can be used to legitimize decisions and selections as the design process proceeds. Moreover, using these criteria, such as the results of simulations and analysis performed by various domain experts, most likely leads to building designs with better performance. One major challenge in practice today is the management of design information and collaboration between several actors in a building project.

A large portion of the Architecture, Engineering, and Construction (AEC) industry still 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 (Beetz 2009; Borrmann et al. 2018). 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 based on BIM, which aims to introduce a computer-readable, yet adaptive universal method/function which works on schematized information exchange requirements (templates) for different use cases. Furthermore, this concept will be explained and demonstrated using an example scenario.

### Introduction

Decisions made in the early phases of building design significantly affect subsequent design tasks and efforts, such as the end products realization and performance. As we move on from initial conceptual design phases into more and more detail-oriented planning and subsequently precise technical design for construction, the cost and effort to change the overall design or modify previously taken decisions increases dramatically (MacLeamy 2004; Davis 2013). In other words, the architect's ability and freedom to affect the building design decreases intensely.

In the early, yet essential phases, usually only vague

and incomplete information is available in the form of fragments from the client's requirements, building regulations, and environmental necessities. The problem of this procedure is that in the early phases, even though the information is little, nearly all the crucial decisions must be made at this point. The process of designing a building is characterized by a constant interplay of creating variants, their assessment and consistent detailing (Zeiler, Savanovic & Qanjel 2007). During these initial phases, the architect as the designer mainly relies on his personal experience and know-how for detailing, evaluating and comparing design variants. The sheer load of design decisions leaves the architect to decide on a large number of details early on and sometimes without enough knowledge about their consequences (Derelöv 2004).

One way to improve this approach is to integrate, as early as possible, other domain experts into the assessment and evaluation of different aspects of the design. This creates the need for frequent and proper communication as well as the exchange of design information between different disciplines and their respective actors. 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).

The unique nature of the Architectural, Engineering, and Construction (AEC) industry puts a high demand on communication and workflow management and its optimization. The current trend in AEC follows Building Information Modelling (BIM). BIM is a model-based process for the development, integration, and management of all semantical and geometrical information related to construction projects. One significant advantage of BIM is the ease in which exchanging semantically rich 3D-models between different disciplines is possible. This feature in BIM enables new possibilities for the exchange of information in a digital format between different actors, such as architects and consultants. Thus, improving access to computer-aided analysis from the early stages of design. This encourages the early involvement of varying domain experts and thereby enhances the efficiency and overall quality of the design process (Borrmann et al. 2018).

In an attempt to address this gap, the DFG-FOR2363 research group develops methods for evaluating architectural design variants in the early stages of their development by adaptive detailing strategies that allow the detailing, evaluation, and assessment of alternative building design variants, which may be partially incomplete and vague.

In the next section, a brief overview of the state of the art research related to this work is prepared. Following this short literature review, the concept of BIM-based adaptive Minimized Communication Protocol is discussed. Later on, the adaptive universal Feedback function along with its related arguments is described, followed by a demonstrative example. In the end, a brief conclusion along with future steps is addressed.

### State of the art

As the design of a building evolves, many experts from different domains are needed for evaluating various aspects of the design. Moreover, challenges in early design phases are uncertainties (vagueness and incompleteness) both in the model quality (degree of detailing and abstraction) of the design variants and in the results of the combined analysis and simulation procedures, which arise from assumptions taken to perform the analysis. In the concept of adaptive detailing corrections and modifications or so-called detailing of the BIM model can be suggested or provided by the domain-expert or the architect. But without the architect's acceptance, no change will be acknowledged. The corrections and criticism offered by the architect can be assumed as reflexive design, based on experience/know-how, or even based on references and similar models, and could include both objective and subjective aspects of design.

On the other hand, the criticism, corrections and suggested detailing from the domain-experts to the architect is referred to as **Feedback**. The feedback can also be delivered by the expert planners using objective criteria such as simulation/analysis results. The feedback from the domain-expert can include simulation results and design options (Zahedi & Petzold 2018a).

The development of design variants represents a vital aspect of the process of designing a building. Starting with the requirements that the architect receives from the client, followed by external restrictions and regulations, the architect designs possible variants. As the planning process proceeds, more suitable and preferred variants are elaborated in more detail while other drafts are rejected or discarded. Mattern & König (2018) introduce a concept for the model-based management of design variants. The developed idea eliminates the need to use separate models for partially similar and yet different design variants and thus avoids redundant information. So-called option categories are introduced to organize the possibilities

that might evolve when going forward in the design process. Then Graph models are proposed for managing design variants and their interdependencies. However, due to complex interdependencies and restrictions, the authors suggest that the creation of invalid combinations are to be avoided (Mattern & 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.

A major complication for using BIM in early design stages is that despite the insufficient information available in these phases, a BIM model appears precise and specific. Abualdenien & Borrmann (2018) introduce a multi-LOD metamodel for defining information exchange requirements as well as addressing information uncertainty during the early stages of design using BIM models. Their suggested metamodel has two layers of a data-model level and instance level. 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 & Borrmann 2018). These exchange requirements defined using the multi-LOD metamodel of Abualdenien & Borrmann are being used later on in this paper as the aLODx templates. Using this concept depending on the analysis type and the design stage, which corresponds to a level of development, each domain expert will ask for a list of building components plus their proper attributes. Further explanations are given later on in this paper.

Considering the importance of collaboration and internal communication between the parties involved in building design, open BIM Collaboration Format (BCF) was first introduced in 2010 (bcfXML v1) and then further developed in 2014 (bcfXML V2.1). BCF enables BIM-based workflow and communication as an open standard, between different actors and different software vendors (buildingSMART 2019). Using BCF, the project participants create topics (such as issues, proposals, and change requests) that contain various attributes like type, description, comments and many other types of communication information. Each topic is then connected to a model element as well as a viewpoint and possible screen-shots. This eliminates the need to exchange entire bulk digital BIM-models between software applications. Each issue in BCF version 2 could also be equipped with a new feature called BIM-Snippet. BIM-snippets enable the exchange of schematized files. A BIM-Snippet could be another partial IFC file (buildingSMART 2017).

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 and workflow in early stages of design.

In the meanwhile, the authors believe that knowledge, especially in complex projects, is a socially attained phenomena (Zhang & El-Diraby 2011) and acquiring proper information for better decision-making in early design phases is achieved by integrating as many domain-experts as possible in the process. Nevertheless, aiming for computer-readable communications doesn't necessarily eliminate 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.

### **Minimized BIM-based Communication Protocol**

Through the early phase of design, we assume that when the architect requests an analysis from any domain-expert, there could be three different outcomes as following:

- The model is still incredibly immature and not qualified for analysis. So in this case, the response is a simple no, and maybe some general guidelines, most probably in the form of text or predefined categorical responses.
- The analysis could be performed, and the results would be presented to the architect.
- Some details within the model are missing, and these missing details would be reported back to the architect.

Reporting back the missing details to architect confronts him with the decision on which and what details to choose to develop his model further. The problem at this point is that the designer has to decide on so many things which he might not have enough knowledge about them. What could be of great help to the architect at this point is to provide him with some suggestions and enable him to compare them? Who is better to suggest these options to him than the experts and consultants in each domain? Other than being

more informed and knowledgeable compared to the architect in that specific field, they are also more familiar with common practice choices and decisions that are taken by other designers and architects when facing these decisions.

This paper differentiates design variants and options. Variants are the design models developed by the architect as proposed solutions, whereas options represent the feedback and suggestions provided by the different experts. In this concept, Options are partial design models that are being suggested (sent back as feedback) to the architect by the domain-expert as common practice examples to fulfill the shortcomings of his design model for analysis compatibility. These are to be used (accepted or rejected or even reasoned & argued & discussed with the domain-expert) by the architect as a source of inspiration, suggestion or a possible solution provided by a specialist (whom might have better understanding and knowledge in that specific matter or domain, than the architect). Our approach is to establish a common data environment (CDE) between the architect and multiple consultants to communicate and provide the architect with options to choose for further detailing his design variants. This so-called “feedback” that the architect receives from the domain-expert is consist of the following possibilities:

- 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

In this way, the architect would be provided with possible options to overcome design decisions while being informed about the consequences and outcome of those choices. Throughout the design process, the architect is the responsible team leader and can create and modify the design model/variants, and each updated state of design is done only through architect's final acceptance or upon his determination.

The proposed communication protocol focuses on staying as minimized as possible, which means avoiding to send back and forth any actual digital BIM files (Zahedi & Petzold 2018b). 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 alphanumeric 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

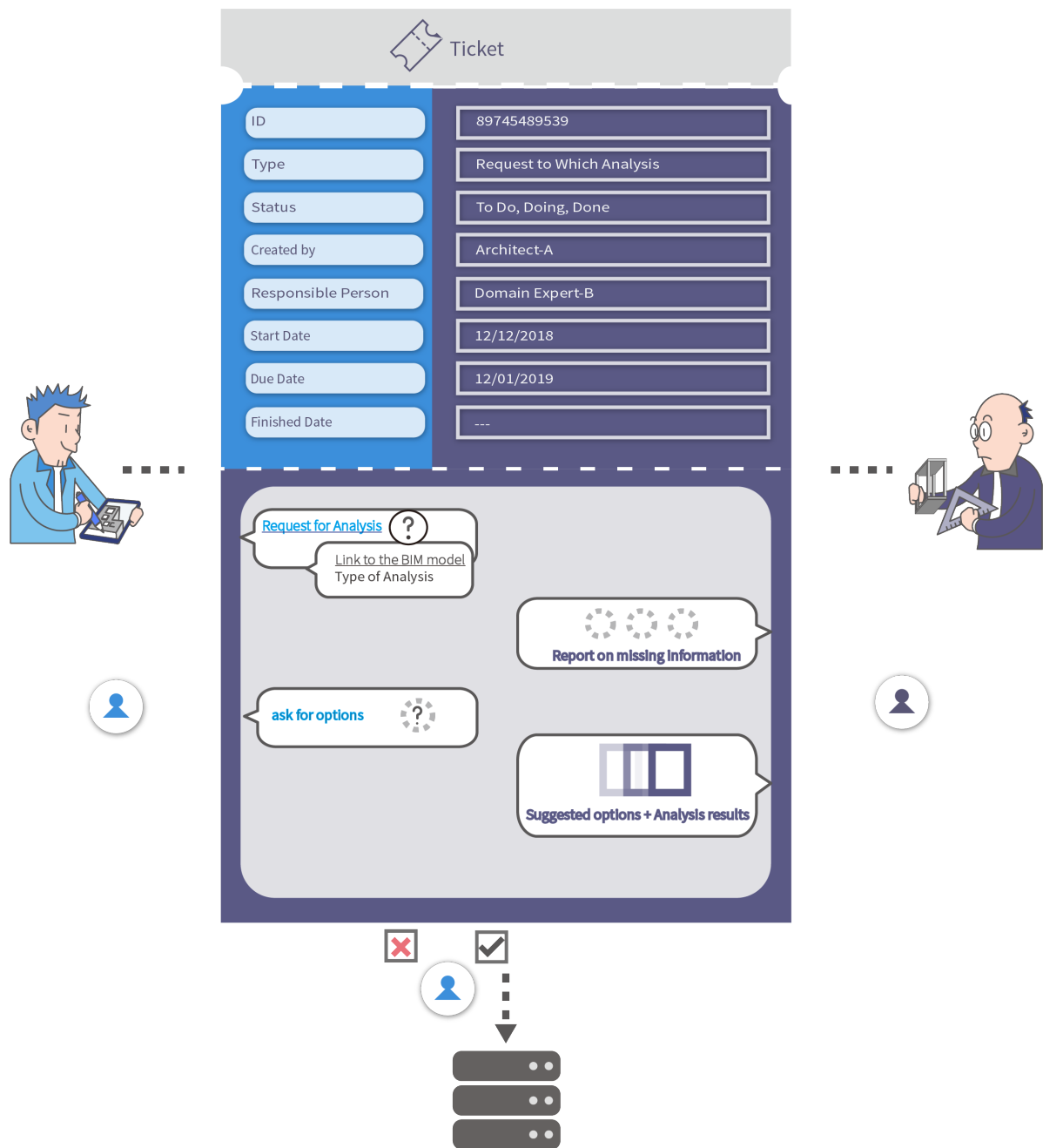


Figure 1: BIM-based Minimized Communication Protocol

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. Multiple 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. Figure 1 demonstrates this combination.

The architect will create the request for a specific type

of analysis and sent to the responsible domain expert. This request contains the desired nature and scope of analysis. In the next step, the domain expert will check the content of the sent BIM model and create a feedback report for the architect. This feedback report contains the missing building components and attributes within the BIM model that are essential for the analysis to run. In response to this report, the architect not knowing precisely what he should choose to fill in for the missing details, asks the domain expert for suggestions or in the context of our research group so-called Options. Proceeding, the domain expert will provide the architect with some common practice

suggestions for filling in the missing details in the design model along with the proper analysis results, using those suggested details. This way the architect can be informed about the consequences resulting from each of those options and choose them wisely. In the next step, a universal method/function called "feedback" will be defined to implement the exchange of missing components and attributes along with proposed values for them as options.

### Feedback Function

This adaptive feedback function in its method signature is as follows:

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

This universal method, based on its use case, will receive different arguments. These arguments are explained one by one as follows:

*actionType*: The first argument is called *actionType*. It represents the use case of the feedback function. Possible *actionTypes* are as follows:

*missingObject*: This action type deals with general use case scenarios, where some building components are missing overall, e.g., reporting back to the architect that all the openings (windows) are missing.

*missingObjectProperty*: This action type refers to an attribute needed for a building component that is empty and needs to be fulfilled by the architect.

*createNewObject*: This action type refers to a newly created building component (probably as part of an option) by the domain-expert (consultant), which is subject to the acceptance or disapproval of the architect. This volatile, newly created component nevertheless will have a GUID that is unique and will be used to refer to this object (explained later on).

*deleteObject*: This action type is used when the consultant suggests an unnecessary building component to be deleted by the architect.

*updateObjectProperty*: This action type is used to indicate a value to the architect, provided by the domain-expert, to fill in a missing property.

*optionGroupID*: The next argument in the feedback function is not mandatory and is used to group multiple suggestions. The proposed details could either be individually suggested, or they could be grouped. This could be achieved with an optional argument inside the feedback function. This argument provides the possibility to arrange some modifications together in a package called option so that the architect can choose to either accept the whole package, with all the included changes, or none of it. These packaged suggestions could not be and should not be handpicked to preserve the consistency and effectiveness of the suggested option.

*GUID*: This argument refers to the Globally Unique ID of a particular building component inside a given BIM model on the BIM server. In case the *actionType* in the feedback function is *missingObject*, then this argument (*GUID*) would be the unique id of the building component that is to contain 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 lacking property. And finally, when the *actionType* is *deleteObject* then this argument is the GUID of the building component which is recommended for deletion by domain-expert. The use of this argument would be apparent upon explaining the demonstrative examples.

*aLODx*, *objectID*, *propertyID*: In each design stage and depending on the level of development, the individual domain experts request detailed requirements of exchanging building information models. Based on the explicit exchange requirements that are essential for each analysis type (examples in our research group are the Life Cycle Assessment (LCA) & Structural Analysis), a specific scheme will be defined, which contains all the essential components (spatial and semantical building components) with their corresponding crucial attributes and LOD within the BIM data model. This, in turn, formulates a certain aLODx for a specific analysis regarding a particular building project type. The following template, called aLOD1-TP4, illustrated in Figure 2, shows an example of the scheme (aLODx) for information required from TP4 (Life Cycle Assessment-LCA) in our research group.

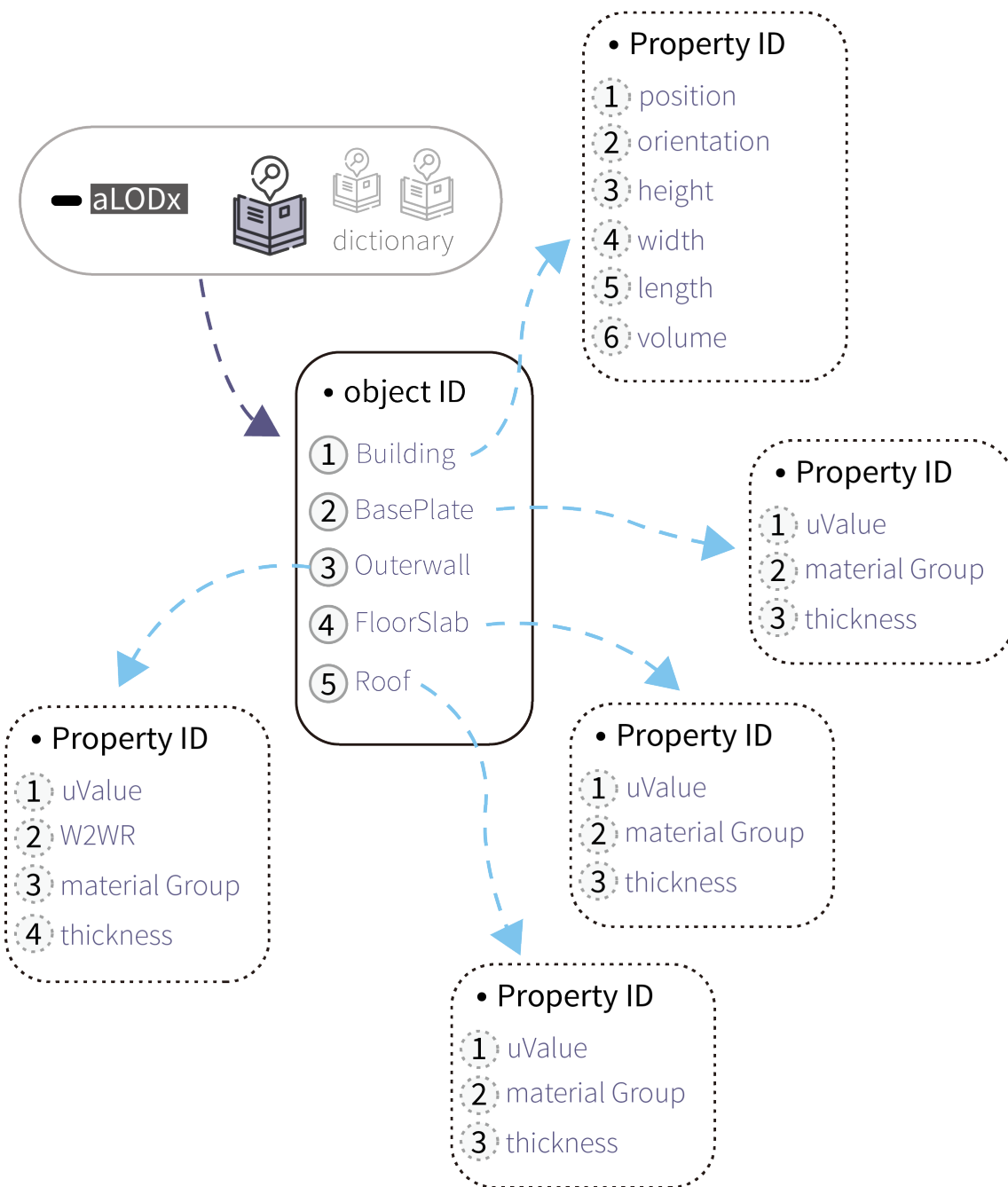


Figure 2: aLOD1-TP4

To define these requirements, this paper uses the multi-LOD metamodel introduced by (Abualdenien & Borrmann 2018). aLODx acts as a dictionary, which contains the required information/details by different domain experts for their simulation engines to work. Inside our concept, the aLODx serves as an adaptive lookup table (using ObejectID & PropertyID) that both the architect and the domain expert will refer to when they are communicating with each other. This way, they can both be sure about what type of building component (object) and what sort of attribute (property) they are talking about.

*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.

### Demonstrative Example

Figure 3 shows the conceptual framework of this minimized communication protocol using an example scenario for LCA. The scene starts when, at some point during the early design stages, the architect requests for an LCA analysis. By doing so, a ticket will be

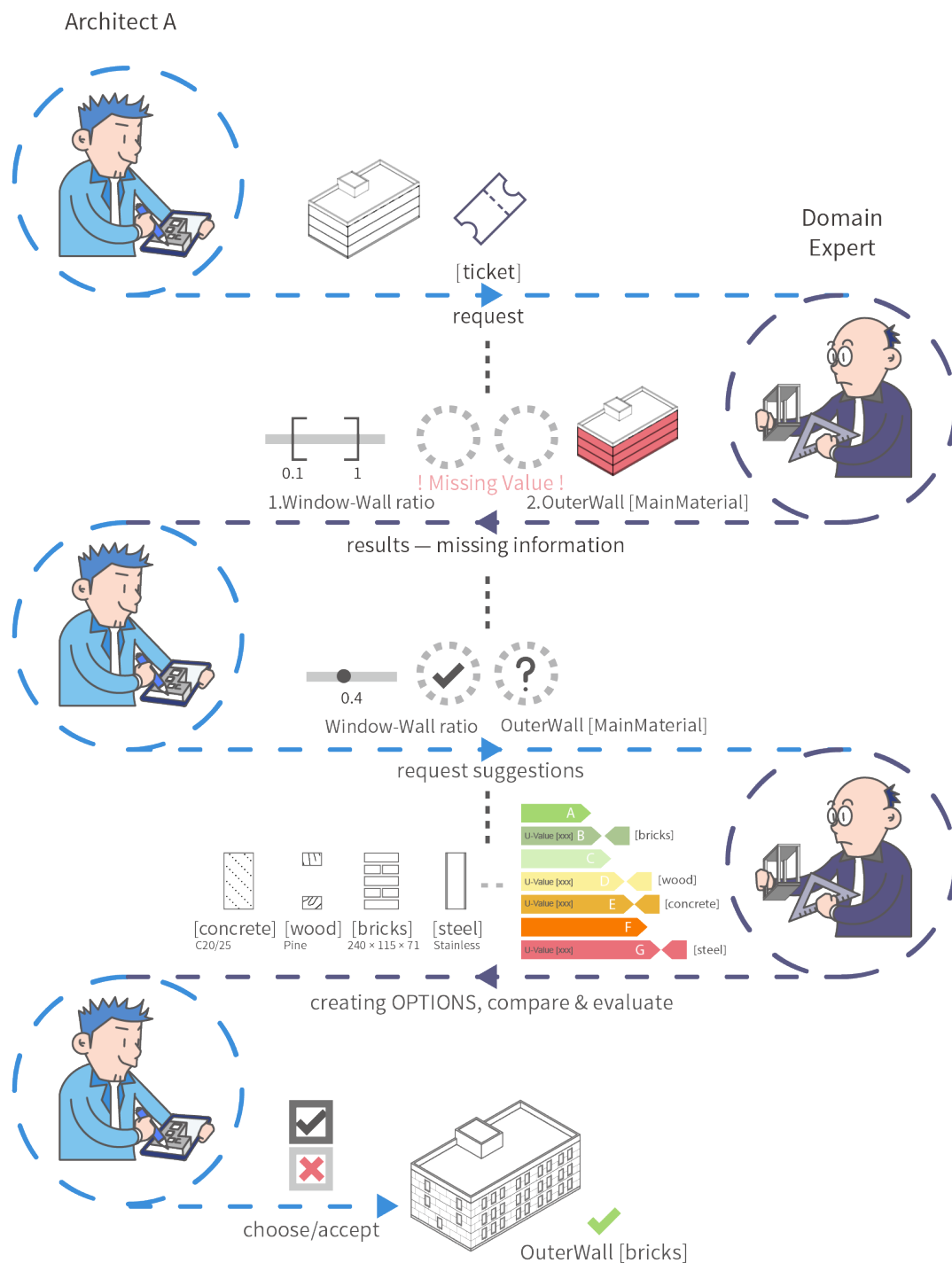


Figure 3: Example Scenario for LCA

issued and a lite message mainly in the form a link, which clarifies the scope of the requested analysis, will be sent to the LCA specialist.

But on the other hand, not being sure of his choice about the outer-walls material, he asks for suggestions from the domain expert, together with the proper evaluation of the suggested options. The specialist creates four possible options for outer-walls material, namely steels, bricks, wood, and concrete. Then he

evaluates them and sends back the results along with their comparison to the architect. This way, the architect can decide intuitively for the better option according to the evaluation results and the system will automatically add the details (outer-walls material) to his model.

In agreement with our minimized communication protocol, the feedback function will be used in two stages. The first one is when the feedback function is

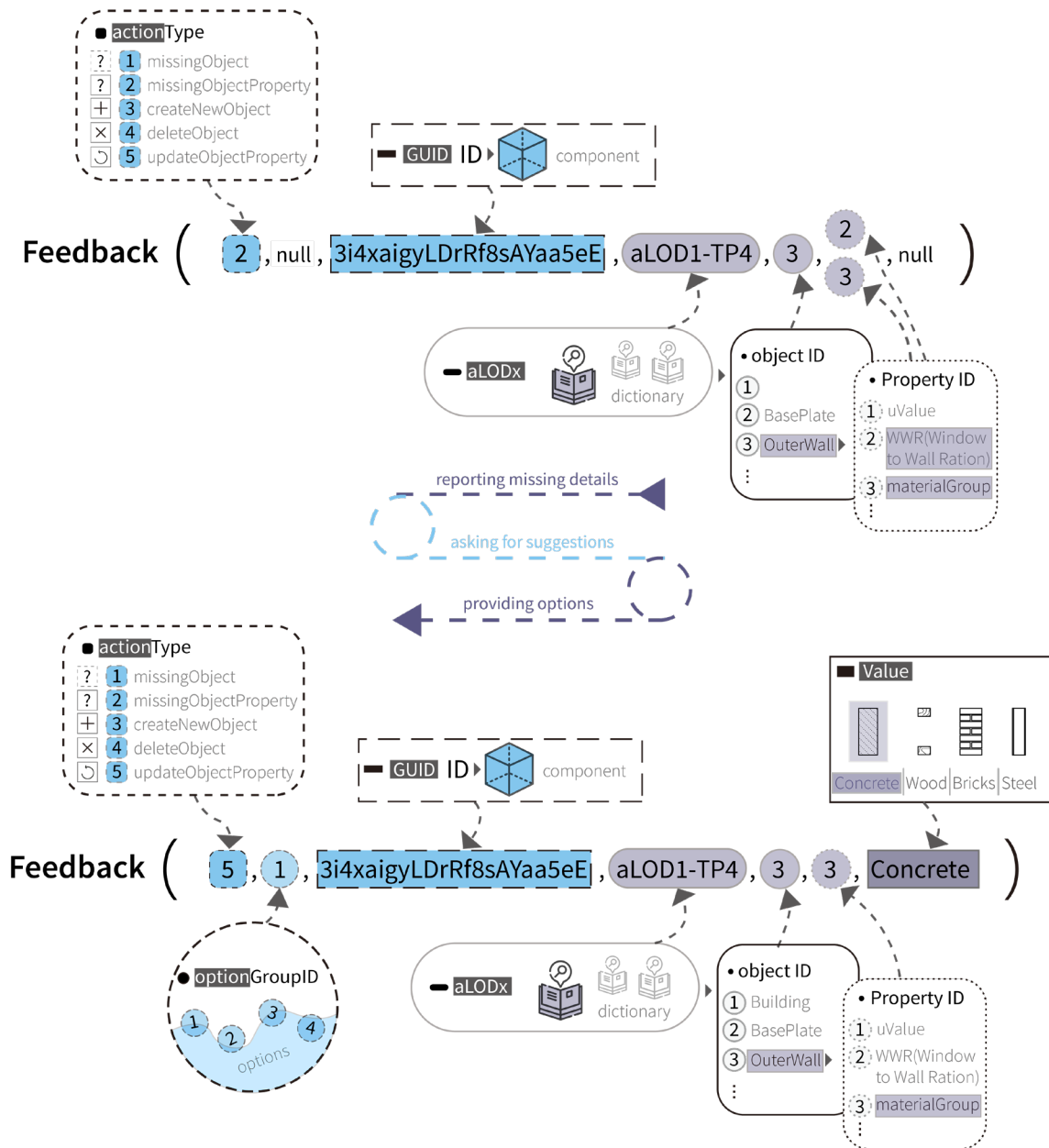


Figure 4: The use of Feedback function based on the example scenario for LCA

used for reporting the missing details (that are essential for the requested analysis) in the design model. In this example, using the *missingObjectProperty* as the *actionType*, the feedback function mentions two individual missing properties in the building (the argument *GUID* refers to the whole building). By pointing to the *aLOD1-TP4*, interpreting *objectID* 3 leads to the *OuterWall* and *propertyID* 2 and 3, point out respectively *Window to Wall Ratio (WWR)* and *Material Group* as missing properties. The upper part of Figure 4 illustrates this feature using the feedback function.

The second time the feedback function is used for suggesting four different material groups proposed for

outer-walls to the architect. This time the *updateObjectProperty* is used as the *actionType*, and each *value* for the material group is represented as an option via the *optionGroupID* argument. The lower part of Figure 4 demonstrates the usage of the feedback function for suggesting the options as mentioned earlier to the architect.

### Feature work

Through this paper, an example was explained using the minimized BIM-based communication protocol for LCA analysis. Next steps include the use of feedback function and the minimized BIM-based communication protocol for structural analysis and the proper example options suggested by domain-experts

in this field. Right now, a relational database is designed and developed in MySQL to store the history of communications based on this minimized BIM-based protocol. Furthermore, developing a plugin for Revit is in progress, which enables the communication to happen inside Revit based on this minimized protocol and be stored in the established database.

## Conclusions

This paper introduced a concept for a BIM-based minimized communication protocol to enhance the collaboration and transparency of the communication during the critical early phases of design. This BIM-based communication protocol is built on schematized information requirements that are adaptively defined by domain experts depending on analysis type, building project type and a suitable level of development of the digital design model. The goal is to store the history of these communications and use this accumulated data (since this data is structured and is machine-readable) for supporting decision making (e.g., via case-based reasoning) in future use cases.

Utilizing this lite 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 afterwards for further use cases.

## Acknowledgments

The authors would like to gratefully acknowledge the financial support received from the German Research Foundation (DFG) within the framework of the Collaborative Research Group FOR 2363.

## Reference List

- ABUALDENIEN, J., and BORRMANN, A., 2018. Multi-LOD model for describing uncertainty and checking requirements in different design stages. *Proceedings of the 11th European Conference on Product and Process Modelling (ECPPM 2018)*, Copenhagen, Denmark
- BEETZ, J., 2009. Facilitating distributed collaboration in the AEC/FM sector using Semantic Web Technologies. *Eindhoven: Technische Universiteit Eindhoven*
- BORRMANN, A. et al., 2018. Building Information Modeling: Why? What? How? *Building Information Modeling*: Springer, pp. 1–24
- BUILDINGSMART, 2017. *BIM Collaboration Format v2.1 Technical Documentation*. buildingSMART. Available at: <[https://github.com/BuildingSMART/BCF-XML/tree/release\\_2\\_1/Documentation](https://github.com/BuildingSMART/BCF-XML/tree/release_2_1/Documentation)> [Accessed 18 December 2017]
- BUILDINGSMART, 2019. *Open BIM Collaboration Format (BCF) intro*. buildingSMART. Available at: <<http://www.buildingsmart-tech.org/specifications/bcf-releases>> [Accessed 3 April 2019]
- DAVIS, D., 2013. *Modelled on software engineering: Flexible parametric models in the practice of architecture*. RMIT University
- DERELÖV, M., 2004. *On Conceptual Design Reliability*. Linköpings University. Thesis
- FLAGER, F. et al., 2009. Multidisciplinary process integration and design optimization of a classroom building. *Journal of Information Technology in Construction*, 14(14), 595–612
- MACLEAMY, P., 2004. Collaboration, integrated information and the project lifecycle in building design, construction and operation. *WP-1202, The Construction Users Roundtable*
- MATTERN, H., and KÖNIG, M., 2018. BIM-based modeling and management of design options at early planning phases. *Advanced Engineering Informatics*, 38, 316–329
- ZAHEDI, A., and PETZOLD, F., 2018a. Utilization of Simulation Tools in Early Design Phases Through Adaptive Detailing Strategies. *Proceedings of the CAADRIA 2018. The 23rd Conference on Computer-Aided Architectural Design Research in Asia*. Beijing, China, pp. 11–20
- ZAHEDI, A., and PETZOLD, F., 2018b. Seamless integration of simulation and analysis in early design phases. *Proceedings of the IALCCE 2018. The Sixth International Symposium on Life-Cycle Civil Engineering*. Ghent, Belgium, pp. 2007–2015
- ZEILER, W., SAVANOVIC, P., and QUANJEL, E., 2007. Design decision support for the conceptual phase of the design process
- ZHANG, J., and EL-DIRABY, T. E., 2011. Social semantic approach to support communication in AEC. *Journal of Computing in Civil Engineering*, 26(1), 90–104