2023 European Conference on Computing in Construction
40th International CIB W78 Conference
Heraklion, Crete, Greece
July 10-12, 2023

FROM STATIC TO DYNAMIC INFORMATION CONTAINERS

Nidhal Al-Sadoon[1] (✉), Raimar J. Scherer[1] and Karsten Menzel[1]
[1]Institute of Construction Informatics, Technische Universität Dresden, Germany
nidhal.al_sadoon@tu-dresden.de

Abstract

In today's fast-paced construction industry, collaboration is crucial for success. The Information Container for Linked Document Delivery (ICDD) is one of the efforts being made to support collaboration. It is used to store and exchange semantically linked heterogeneous application models. However, it is a static information container, which complicates synchronizing processes in an advanced digital twin environment. In this paper, we propose an approach that enhances information container implementation with dynamic features by automatically updating the linksets whenever a new document version is added to the container. This feature is expected to expand the workability of the ICDD and enhance its usefulness beyond pure data exchange.

Introduction

The construction industry is characterized as a highly interdisciplinary industry in which experts from various domains participate in projects. It is important to consider that construction has specific characteristics regarding production compared to other industries, such as temporality, a bounded location, and a one-off product. It can be compared to a nexus where parties connect in temporary "project networks" to bring together numerous factors that make up a specific project (Vrijhoef and Tong, 2004). Throughout an asset's lifespan in the built environment, many players are involved at various stages, and large sets of application models are generated that need to be shared and exchanged.

To improve these communication processes, many efforts have been made toward structuring the processes of collaboration and data sharing. Building Information Modelling (BIM) stands out as the most notable approach that brings about improved facilities for information management in AEC operation projects. The BIM environment approach is a digital workspace where multiple stakeholders in a construction project can collaborate and share data, however, despite the many improvements BIM environments have brought to the construction industry, they do not entirely succeed in addressing interoperability challenges (Pauwels, 2014). The gap between traditional document-based construction and data-driven construction cannot be adequately closed with BIM's enhanced capabilities, especially in the area of interoperability (Boje, 2020).

The built environment contains more information than just BIM models such as live sensor data, geographic information systems data (GIS), social systems data, environmental data, and facility management data. To meet the requirements of tasks that go beyond traditional BIM, and without preharmonizing them in one common huge data model, a suitable solution is still needed to manage and exchange distributed BIM and non-BIM data from different sources. Context-capturing driven approaches, namely, MultiModel (Scherer and Schapke, 2011) and COINS (Van et al., 2010) were developed to formalize an exchangeable context model without altering the information resources and without coming up with one harmonized big data model. Based on these two approaches, the information container for Data Delivery (ICDD) was standardized in ISO 21597 (ISO 21597-1 2020). The standard provides specifications to facilitate the exchange of heterogeneous original data embedded in a Semantic Web meta-layer that provides metadata and linking capabilities based on two main ontologies; container ontology and linkset ontology. It is possible to link, store, and exchange data from all kinds of sources, including model data, documents, images, and ontology-based data.

Utilizing Semantic Web technologies, information elements within a container can be easily linked with each other and with external data. In this context, an ICDD container can be described as a semantically connected storage folder containing current project information, which can be used just for storing, exchanging, or sharing a particular version, or state of the project data among stakeholders (Werbrouck, 2019).

It is worth mentioning that the standard provides a specification for versioning by adding versioning properties to the Container ontology, namely: *ct:versionID* to indicate the version of the resource and *ct:versionDescription* to describe the version. However, the versioning functionality is applicable only for individuals of the classes *ct:ContainerDescription*, *ct:Document*, and *ct:Linkset*, but, the versioning properties are not applicable to the links themselves. This results in an out-of-date linkset whenever any of its contained documents is versioned. Consequently, to keep the container up-to-date, a new link set needs to be created. However, to create each link in a linkset, the document information and the identifier need to be identified for all link elements, which will typically consume considerable time and effort. To overcome the described limitations, this research proposes an approach that is aimed at providing up-to-date information containers by enabling a fully automatic updating of the linksets to save the time and cost required to create new linksets whenever a related document is versioned. This versioning includes identifying the linksets, where these documents have participated and then automatically updating the links in the linksets. Hence, accessing the most up-to-date data can be insured for all participants in the process.

Current State of Information Containers

While standardized ICDD is a comparatively recent achievement, the Multimodel approach underlying it has already been developed, implemented, and verified in the German Mefisto Project (Scherer and Schapke, 2011) to support information analysis and the cross-domain exchange of information in a process-oriented Management Information System for simulation and decision-making. In its infancy, XML-based containers provide a mechanism for describing the models and their content on an abstract level using ontologies. Thus, the MultiModel Container (MMC) version 1 represents a logical envelope for handling distributed, yet inter-related, application models in combination with corresponding link models as a single information resource that is only capable of representing one static information state (Fuchs et al., 2011). Due to the success of the Mefisto project, the MultiModel approach was further extended by buildingSMART introducing MMC version 2, and the standardization of a BIMLV-Container in DIN SPEC 91350 to enable a linked, open-source data exchange under a common application of the existing BIM standards IFC (ISO 16739) and GAEB-XML (Scherer and Katranuschkov, 2019). Between 2012 and 2020 the MultiModel method was further developed in about ten research projects. One of the implementations for the MultiModel approach was in the European Project HESMOS to integrate various information resources thereby extending BIM to an energy-extended MultiModel framework (eeBIM) aiming to provide a new methodology for energy and emissions saving using BIM-based methods of working, and evaluating the developed methodology in performed real-life pilot projects (Katranuschkov, 2014; Scherer et al., 2017).

Another approach was made in the same period in the Netherlands to develop an interdisciplinary container for the exchange of information in the frames of the COINS project. Using linked data approaches, the project aims to standardize a flexible information container for connecting all building data (Alsem and Willems, 2014). Consequently, both approaches formed the basis of ISO 21597.

As soon as the information container approach was standardized, several research studies were conducted to explore its potential to foster the delivery and exchange of project files, including BIM models, drawings, images, and other data models. Based on the information containers ISO standard 21597, Senthilvel et al. (2021) developed a micro-service approach for delivering project files. Hagedorn (2018) developed ICDDToolkitCore, which is a framework that provides functions to open, create, validate, edit, and export containers according to the specification of ISO 21597. Utilizing the ICDD and the Semantic Web technologies, the researcher also proposed the concept of the" toolchain framework" which relies on the idea of a centralized platform for configuring workflows employing construction-specific applications, services, and content within a platform in a compatible and process-oriented manner (Hagedorn et al., 2022). A BIM-based solution for Infrastructure Asset Management System (AMS) was proposed by Hagedorn et al. (2023). Based on stakeholders' requirements across domains, the proposed approach provides asset managers with a strategy for the dynamic use of information containers in the operational phase.

Hamdan et al. (2021) developed an approach in which they linked an IFC model, representing an existing bridge, with ontologies that semantically represent the construction and affecting structural damage as well as other related data, e.g. photos, protocols, etc. Thereby, the models and links were stored in an ICDD. Also, demonstrated in ICDD environment, the Building Concrete Monitoring Ontology (BCOM) is developed by Liu et al. (2021) the ontology allows an IFC- based bridge model to be configured with properties about concrete works so that they can be processed by predefined queries in an asset management application. Using Blockchain technology and the ICDD, Ye and König (2020) present a framework for automated billing by combining the BIM Contract Container (BCC) with Smart Contracts. In addition, interesting research recently proposed a Linked Building Data (LBD) server to link heterogeneous linked building data in a Federated Common Data Environment through the combination of the Solid Initiative for Web decentralization and the ICDD standard (Werbrouck et al., 2022).

To expand on the existing standardized ICDD schema, Al-Sadoon and Scherer (2021) proposed an ontology-based extension that enables the allocation of multiple values for individual elements within IFC files. The approach was developed and validated in the BEST project where the objective was to implement a fire and crowd co-simulation on a building model with changeable spaces (Al-Sadoon et al., 2022). Finally, it is worth mentioning that to retrieve the datasets linked as an RDF in the ICDD, the query language SPARQL is the recommended RDF query language, however, Werbrouck et al. (2019) compare the use of SPARQL with Linked Data querying languages; HyperGraphQL and GraphQL-LD. He concludes that among these three languages, SPARQL is the most expressive. In contrast, GraphQL-based queries are useful for simple queries that involve a known dataset (such as an ICDD container), because they hide SPARQL's verbosity behind elegant and concise syntax.

Finally, based on ICDD standard (ISO 21597), the concept of decentralized digital twins for self-sovereign data sharing and cross-organizational information-based collaboration is researched in iECO (2020) to support a wide variety of work processes and scenarios and to create a common data space for the construction industry.

Dynamic Information Containers Approach

The current state of information containers implemented based on ISO 21597 could be referred to as a "static information container". A typical ICDD structure is created based on the container and linkset ontologies. Documents are added, and then optionally different link types can be used to create links between these documents and the elements they contain. In this way, the container fulfills the basic purpose specified by ISO 21597, namely delivering, storing, and archiving documents. On the other hand, it is undeniable that a construction project is dynamic by nature. Along the life cycle of a facility in the built environment, information models are subject to be modified. The building model, for example, evolves not only during the design phase but also during the operational phase when the building is maintained and renovated. Therefore, the conversion of the ICDD status from static to dynamic is important for maintaining a valid information container. However, it is worth noting that the ISO 21597 standard enables versioning functionality by adding versioning properties to ontologies and restricting them to specific domains and ranges (ISO 21597-1 2020). To indicate the version of a resource, the property *ct:versionID* shall be used, while the corresponding property *ct:versionDescription* can be used for a detailed description. As an additional feature, the property *ct:priorVersion* can be used to track the version history of resources. Therefore, the versioning functionalities specified in ISO 21597 are limited to version identification and version history tracking.

In light of the versioning restriction specified in the standard, this research presents an approach developed to extend the versioning functionalities. It is important to mention that implementing this approach requires one small change to the ICDD structure, which is changing the cardinality of the document in the *LinkElement* class from "exactly 1 document" to "minimum 1 document" to allow for the versioned document to be assigned to the related linkset. The extended functionalities will be performed in

```
Procedure : Step 1_ Automatic naming for document versions
INPUT: New Document.
Get last version of the document in the container by filename (without version ID)
IF the document exists in the container THEN
    // assign the property ct:priorVersion
    ct:priorVersion ← last version property
    ct:versionID ← last document version ID + 1
ELSE
    // if the document does not exist, create a new version
    ct:versionID ← 1
END IF
filename concatenated with '-Ver' and versionID
Add the new document to the container with version ID, assign the property ct:priorVersion.
```

```
Procedure : Step 2_Automatic link updating
INPUT: List of the Document Versions (DV List)
// Set the N to Last version ID of the document in the list.
N ← Last version ID
// Initialize an empty list to store updated link elements.
Updated linkElement list ← ∅
FOR i ← N - 1 TO 1 DO
    // Get all link elements from Linkset where document ID = i
    linkElementset ← get all linkElement from Linkset
      where DV List (i) ∈ 'has document'  AND linkElement ∉ Updated linkElement list
      FOR each linkElemen IN linkElementset DO
        ADD linkElemen TO Updated linkElement list
        Get Identifier of the linkElement
        // Check if the identifier exists in the last document version (N).
        IF identifier IS Exist In DV List (N) THEN
            ADD DV List (N) to the linkElement as 'has document'
        END IF
    END FOR
END FOR
```

Figure 1: Step 1 and 2 of the approach

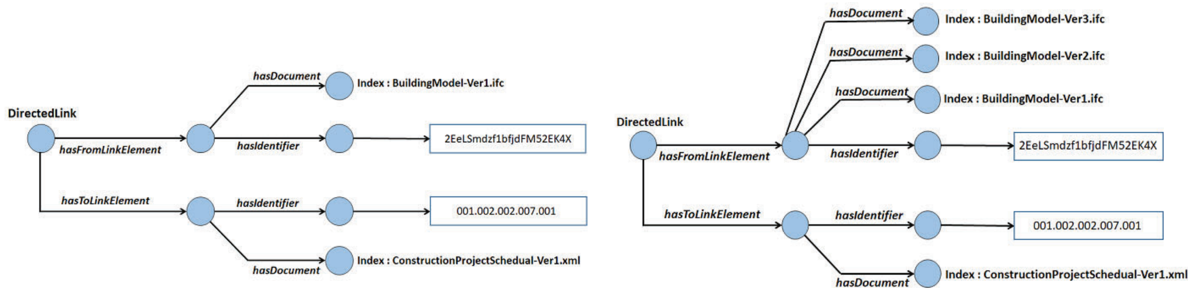

Figure 2: An instance of link element (left) based on ISO 21597 (right) based on the proposed approach

two main steps: automatic name extending for the documents when added to the container and automatic updating of the related linkset, as shown in Figure 1. This could be described as follows:

When creating an ICDD, first the container is created, then the document is added and finally, the links are created and saved in linkset. The first step of implementing the approach starts with adding the documents. Once a document is uploaded into the container, it is checked whether it already exists or not (currently, in the prototype the comparison is made using the original file name). If the file not exists, it becomes the first version, and a version number is automatically added to its name based on the equation in (1). For this purpose, the property *ct:versionID* is used, which is specified in the ICDD standard schema:

$$File\ name\ in\ the\ ICDD = Original\ file\ name + -Ver(versionID) \quad (1)$$

However, if the file is already existing in the container, the property *ct:priorVersion* is used to get the last version of it, then, the subsequent version ID is added to the file name, and the file is added to the Payload Documents folder. The second main step begins when the added file is not the first version. This is done first by searching for the link elements in the linksets that have the name of the newly added file as an instance of "has document", then defining the identifiers of the related linked elements, and finally searching for the value of each identifier. If the value exists, the new version of the file is added as an additional instance of "has Document" to the corresponding link element. This step is repeated until all link elements from the first step are checked. In this way, the linksets are automatically updated whenever a new version of a document is added to the container as illustrated in Figure 2. The following points are worth mentioning:

- The approach applies to both linking types: shallow linking at the document level as well as deep linking among elements within documents (linking identifiers across different documents).
- Additionally, it is applicable for all links types specified in ISO 21597-2.
- The automatic update of the link elements depends critically on the element's ID, which is a key aspect for successful implementation.
- Sensitivity to the element's ID implies that when a building element is deleted from a building model and then add it again to the next version of the same model, it will be considered a new element and consequently, it will not be updated in the related linkset.

As this approach is still in its early stages of development, it is anticipated that more advantages and limitations of its functionality will become apparent.

Case Study

To verify the approach, an exemplary case study is created containing two files: a one-story building model with architectural design as well as the related work progress schedule file. Additionally, for the implementation of the ICDD, the Multi-Model Engine (MME) presented by Al-Sadoon et al. (2022) is enhanced by the proposed approach. Consequently, the MME provides the following functions:

- Creating the container, adding documents, and creating links based on ISO 21597.
- Adding dynamic values based on the ICDD extended schema (Al-Sadoon et al., 2022).
- Automatic document versioning and linksets updating based on the proposed approach.

So, using the MME, the container is created then the two files BuildingModel.ifc and ConstructionSchedule.mpp are uploaded to the container as internal documents. Here the first step of the approach is implemented, and both file names are extended to include the version number (-ver1) as shown on the left side of Figure 3. A random wall element is selected from the building model and linked to

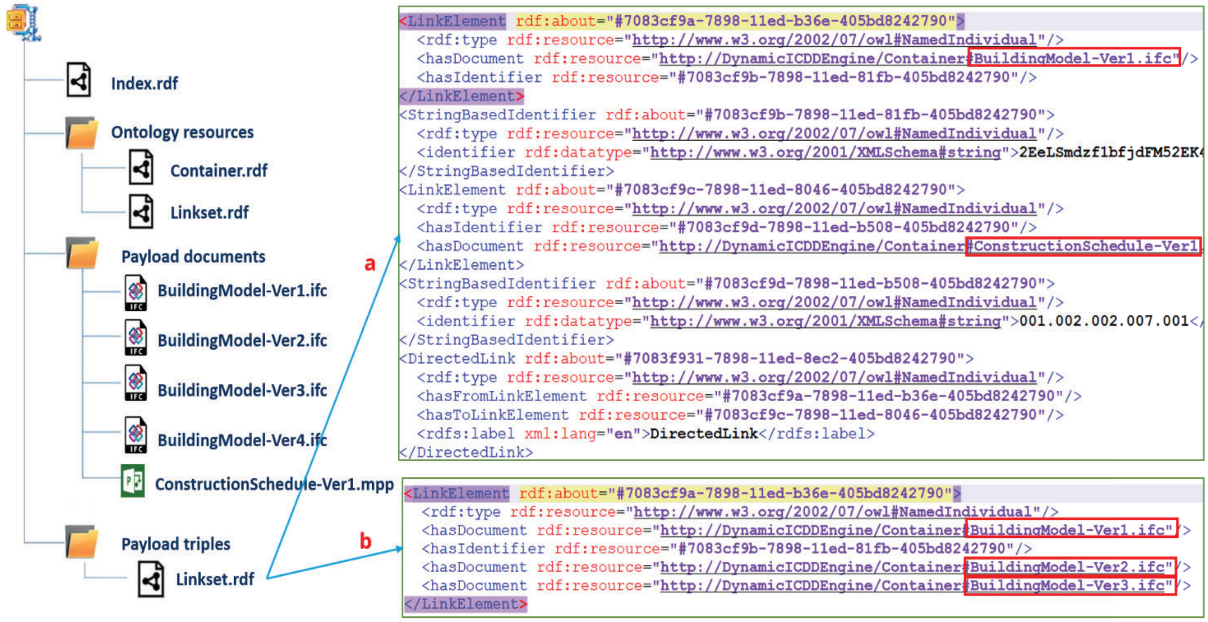

Figure 3: Information Container for the case study

its related activity in the work progress schedule file using *ls:DirectedLink* type. Hence, an information container is created that contains two internal documents in the Payload documents folder. In addition, a deep link is created between elements from each document that is saved as an RDF in the Payload Triples folder as illustrated in the top-right (a) of Figure 3. As it stands, the created information container complies with ISO 21597, but when the building model is revised, for instance, a new version of the Linksets needs to be created from scratch which is time-consuming. This shortcoming is addressed by the proposed functionality of automatic versioning and Linkset updating. These functionalities could be applied also to the other file in the same link; the schedule data model, as in the real scenario when the design model is changed the related schedule and cost models should be revised.

However, for the demonstration purpose, only one file is selected in the implementation. So, when the design is changed a new version of the building model is issued and uploaded to the container. Existence checking is performed for this file and the last version number is defined, then the new model has been given an extended file name; "BuildingModel.ifc-ver2" and is added to the Payload documents folder. Then, the function of link automatic updating took place in three steps; the previous version of the file was searched in the Linksets, the element's ID is defined and the newest version of the file name is added as an additional instance of "hasDocument" to the related LinkElement as shown in the bottom-left (b) of Figure 3. The same steps took place when the third version of the building model is issued. In the second and the third versions of the building model two different design changes took place that did not include the building element "wall" in the experimented link. However, for verification purposes, this wall was deleted in the fourth version of the building model. In such. When it is uploaded to the container, it has been given a new extended name (-ver4) and added to the Payload document folder because it differs from the previous version of the building model but when it comes to the link element updating, the wall ID was not found therefore, the newly file name was not added to the LinkElement.

Conclusion and future work

Even though information containers are developed to serve as a repository, maintaining their validity is of utmost importance. Having up-to-date containers, however, cannot be secured solely by using ISO 21597. Hence, we proposed an approach in this paper to extend the information containers' usability from static to dynamic containers by adding the functionalities of automatic document versioning and automatic linkset updating. To verify the approach, an exemplary case study comprising two files linked on a subdocument level using *DirectedLink* type. The future work will involve validating the approach through the development of Smart Advanced Service functionalities for the support of planners in the context of construction planning in environmental assessment (iECO, 2020). By having an ICDD that is up-to-date, we can have not only valid storage containers but also a dynamic information container that could be used along the building life cycle, separately on their own or integrated with cloud-based services, to keep track and provides control over changes in the linked documents. As such, the dynamic ICDD aimed at facilitating efficient collaboration among players in a construction project and providing more valuable services for overarching interdisciplinary decision-making.

Acknowledgments

This research was made possible with the funding support of the project iECO (intelligent Empowerment of Construction Industry). This support is herewith gratefully acknowledged.

References

Al-Sadoon, N. and Scherer, R.J., (2021). October. IFC semantic extension for dynamic fire safety evacuation simulation. In *Proc. of the Conference CIB W78* (pp. 11-15).

Al-Sadoon N, Katranuschkov P, Scherer RJ, (2022). Extending ICDD Implementation to a Dynamic Multimodel Framework. In *Proc. of the Conference ECPPM* (doi.org/10.1201/9781003354222-15).

Al-Sadoon N, Menzel, K., Scherer RJ. (2022). Multimodel Framework for Digital Twin Empowerment. In *Proc. of the Conference ICCCBE 2022* (submitting).

Alsem, I.D. and Willems, P.H., (2014). The management of information over the life-cycle of a construction project using open-standard BIM.

Boje, C., Guerriero, A., Kubicki, S. and Rezgui, Y., (2020). Towards a semantic Construction Digital Twin: Directions for future research. *Automation in Construction*, *114*, p.103179.

Fuchs, S., Kadolsky, M. and Scherer, R.J., (2011), June. Formal description of a generic multi-model. In *2011 IEEE 20th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* (pp. 205-210). IEEE.

Hagedorn, P., (2018). Implementation of a validation framework for the information container for data drop. In *Tagungsband 30. Forum Bauinformatik* (pp. 147-155).

Hagedorn P, Block M, Zentgraf S, Sigalov K, König M. (2022). Toolchains for Interoperable BIM Workflows in a Web-Based Integration Platform. *Applied Sciences. 12*(12):5959.

Hagedorn, P., Liu, L., König, M., Hajdin, R., Blumenfeld, T., Stöckner, M., Billmaier, M., Grossauer, K. and Gavin, K., (2023). BIM-Enabled Infrastructure Asset Management Using Information Containers and Semantic Web. *Journal of Computing in Civil Engineering*, *37*(1), p.04022041.

Hamdan, A.H., Taraben, J., Helmrich, M., Mansperger, T., Morgenthal, G. and Scherer, R.J., 2021. A semantic modeling approach for the automated detection and interpretation of structural damage. *Automation in Construction*, *128*, p.103739.

IECO Consortium. 2022. https://ieco-gaiax.de/. Web Resource (last accessed: April 2023).

ISO 21597-1 (2020). Information Container for linked document delivery – Exchange Specification – Part 1: Container. International Organization for Standardization, Geneva, Switzerland, ICS 35.240.67 IT applications in building and construction industry, 41 p.

Katranuschkov, P., Scherer, R.J., Weise, M. and Liebich, T., (2014). Extending BIM for energy simulation and design tasks. In *Computing in Civil and Building Engineering* (pp. 625-632).

Liu, L., Hagedorn, P. and König, M., (2021). An ontology integrating as-built information for infrastructure asset management using BIM and semantic web. In Proceedings of the European Conference on Computing in Construction (pp. 99–106).

Pauwels, P., (2014). Supporting decision-making in the building life-cycle using linked building data. *Buildings*, *4*(3), pp.549-579.

Scherer, R.J. and Katranuschkov, P., (2019), July. Context capturing of multi-information resources for the data exchange in collaborative project environments. In Proceedings of the European Conference on Computing in Construction.

Scherer, R.J., Katranuschkov, P. and Baumgärtel, K., (2017). Open eeBIM platform for energy-efficient building design. In *eWork and eBusiness in Architecture, Engineering and Construction* (pp. 387-395).

Scherer, R.J. and Schapke, S.E., (2011). A distributed multi-model-based management information system for simulation and decision-making on construction projects. Advanced Engineering Informatics, 25(4), pp.582-599.

Senthilvel, M., Oraskari, J. and Beetz, J., (2021), August. Implementing Information Container for linked Document Delivery (ICDD) as a micro-service. In *EG-ICE 2021 Workshop on Intelligent Computing in Engineering* (p. 66).

Van Nederveen, S., Beheshti, R. and Willems, P., (2010). Building information modelling in the Netherlands: a status report. In *W078-Special Track 18th CIB World Building Congress* (p. 28).

Vrijhoef, R. and Tong, M.K., (2004). Understanding constructions as a complex and dynamic system: an adaptive network approach. In *12 the annual conference on lean construction.* (pp. 61-73).

Werbrouck, J., Senthilvel, M., Beetz, J. and Pauwels, P., (2019). Querying heterogeneous linked building datasets with context-expanded graphql queries. In *7th*

Linked Data in Architecture and Construction Workshop (Vol. 2389, pp. 21-34).

Werbrouck, J., Pauwels, P., Beetz, J. and van Berlo, L. (2019). Towards a decentralised common data environment using linked building data and the solid ecosystem. In *36th CIB W78 2019 Conference* (pp. 113-123).

Werbrouck, J., Pauwels, P., Beetz, J. and Mannens, E., 2022. Lbdserver-a federated ecosystem for heterogeneous linked building data. *Semantic Web Journal* (submitting).

Ye, X. and König, M., 2020. Framework for automated billing in the construction industry using BIM and smart contracts. In *Proceedings of the 18th International Conference on Computing in Civil and Building Engineering: ICCCBE 2020* (pp. 824-838).