Lessons Learned from Designing and Using bcfOWL

Oliver Schulz[1,*], Jyrki Oraskari[2] and Jakob Beetz[1]

[1]Department of Design Computation, RWTH Aachen University, Aachen, Germany
[2]Department of Individualized Production, RWTH Aachen University, Aachen, Germany

Abstract
The bcfOWL ontology has been developed as part of the EU Horizon 2020 BIM4Ren project to enable communication between BIM Collaboration Format (BCF) Issues and Linked Building Data (LBD) concepts described on the Semantic Web. This paper evaluates the current approach in bcfOWL based on its use in the BIM4Ren project. The ontology serves as an interlanguage for component-based communication, providing a gateway to the systems of different domains. During its use in the project, new insights into the usability of the ontology were gained. We discuss these findings and provide suggestions to complement the original design principles of bcfOWL, targeting the LBD domain. Our work should guide future research in component-based communication in building-related projects and provide helpful considerations for future ontology designs. We also discuss potential areas of improvement for bcfOWL, including versioning, Ontology Design Patterns and validation. Overall, bcfOWL aims to improve querying capabilities and connectivity with Linked Building Data, making it a valuable tool for building-related projects.

Keywords
BCF, Issue Management, Linked Data, Linked Building Data, Ontology Design

1. Introduction

The bcfOWL ontology [1] was developed as part of the EU Horizon 2020 BIM4Ren project. The BIM4Ren project aimed to harness the potential of BIM to enable energy-efficient renovation of existing buildings across the construction value chain. The development of bcfOWL was explicitly aimed at incorporating the localisation of site images and the spatialisation of tasks created by a project manager and completed by an assigned person in the field. bcfOWL is based on buildingSMART's BIM Collaboration Format (BCF) [2], an open standard for sharing and communicating issues related to building elements. The BCF standard was developed to facilitate the exchange of issue descriptions among various stakeholders in the context of the building information modelling methodology. The standard employs concepts for Topics, Comments, and Viewpoints to describe Issues and is available in two formats: file-based (BCF XML[1]) and server-based (BCF API[2]). While both formats share most concepts, buildingSMART

Proceedings LDAC2023 – 11th Linked Data in Architecture and Construction, June 15–16, 2023, Matera, Italy

*Corresponding author.
schulz@dc.rwth-aachen.de (O. Schulz); oraskari@ip.rwth-aachen.de (J. Oraskari); beetz@dc.rwth-aachen.de (J. Beetz)
0000-0002-4722-4621 (O. Schulz); 0000-0002-4723-3878 (J. Oraskari); 0000-0002-9975-9206 (J. Beetz)

[1]GitHub, BCF XML: https://github.com/buildingSMART/BCF-XML accessed 24.02.2023
[2]GitHub, BCF API: https://github.com/buildingSMART/BCF-API accessed 24.02.2023

has not yet developed a common schema for BCF.

bcfOWL aims to enable access between issue communication-related information and Linked Building Data (LBD) concepts described on the Semantic Web. For instance, bcfOWL can connect BCF issues with Linked Building Data elements and make it searchable using open standards. Moreover, the ontology can be a shared schema to ensure compatibility with and conversion to BCF XML and the BCF API.

In this paper, we want to evaluate the current approach in bcfOWL based on its usage throughout the BIM4Ren project. Original design principles of bcfOWL were introduced in [1] and are still valid. The suggestions in this paper complement these and target the Linked Building Data domain. Therefore, this work should serve as a guide for future research in component-based communication in building-related projects. On the other hand, it shall provide helpful considerations for future ontology designs.

The paper is organised as follows: We begin by introducing the current state of bcfOWL and describe its design and usage throughout the BIM4Ren project. Next, we identify potential areas for improvement that became apparent during our work with bcfOWL and discuss them at the end of each subsection. Finally, we outline a path forward for the ontology and conclude the paper in the last section.

The namespaces in Listing 1 are referred to throughout the paper.

```
prefix bcfOWL:  <http://lbd.arch.rwth-aachen.de/bcfOWL#>
prefix bot:     <https://w3id.org/bot#>
prefix cto:     <https://w3id.org/cto#>
prefix ifcOWL:  <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#>
prefix prov-o:  <http://www.w3.org/ns/prov#>
prefix dot:     <https://w3id.org/cto#>
prefix opm:     <https://w3id.org/opm#>
prefix omg:     <http://w3id.org/omg#>
```

Listing 1: Ontologies with their prefixes used throughout the paper

2. Using bcfOWL in BIM4Ren

The need for an ontology for BCF arose from work on the BIM4Ren project, in which images were spatially located based on plans and models by using BCF [3]. Even though the overall functionality worked with the BCF API, it lacked flexible query abilities. The API had to be extended with new API routes to cope with this drawback. Moreover, in [4], the BCF API was extended with a concept of *Originating Documents* for Viewpoints and *Spatial Representations* for Documents in order to make the camera view locations queryable over different representations of a building. Furthermore, a geometry concept was added to be able to save damage markups within the Viewpoints. However, BCF data had, and still has, very limited possibilities to connect to Linked Data concepts in the LBD domain, such as the Damage Topology Ontology (DOT) [5] and the Building Topology Ontology (BOT) [6].

While these proposed extensions partially addressed the drawbacks, adding to the BCF for each missing condition proved disadvantageous. Therefore, a transfer to an ontology seemed desirable, as it could ensure more flexible query ability and connectivity to other concepts in the LBD domain. Furthermore, the creation of the ontology does not exclude the further

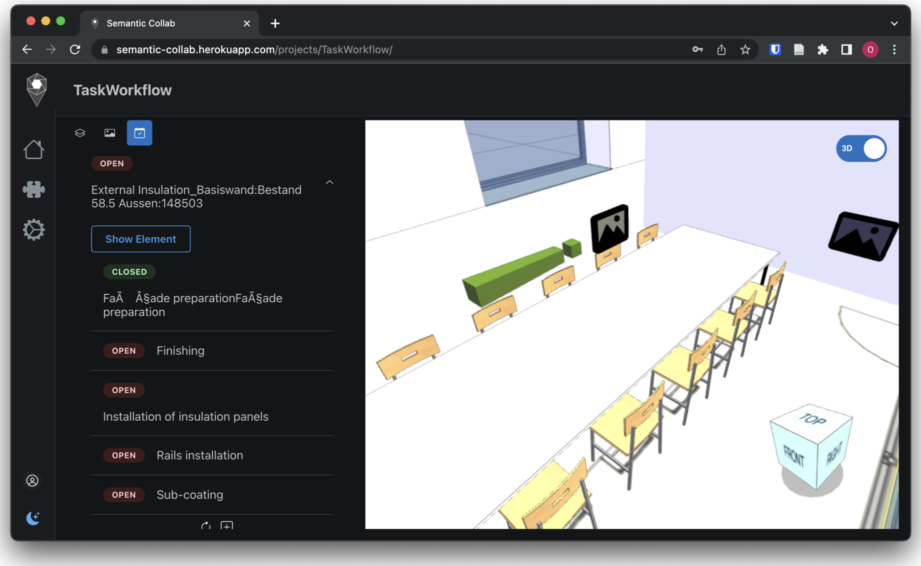

Figure 1: Spatially located images and tasks in the BIM4Ren project. The BCF Viewpoints are located on the 2D floor plan, which is superimposed with the model.

exploitation of the original BCF formats. Still, it merely extends them with a common data format that can be compatible with both the XML and the API versions.

bcfOWL was first presented in [1] as a hand-crafted ontology to express the BCF API and BCF XML formats in the W3C Web Ontology Language (OWL). No new concepts were introduced or changed to ensure consistency between the different BCF representations. The ontology was used throughout the BIM4Ren project for various functions. Initially, it was used to locate images spatially. Then, to express tasks performed in a specific order by a particular person (see Figure 1) on the construction site. To achieve the latter, bcfOWL was integrated with the Construction Task Ontology[3] (CTO), as BCF cannot specify the order of task execution or provide shared task specifications. An example of this integration is shown in the Listing 2.

For communicating the BCF content with the different clients in the BIM4Ren Project, we provided a middleware that allowed us to query the content either by using the BCF API or SPARQL. The content itself was stored on an Apache Jena Fuseki server[4] (see Figure 2).

```
inst:Intervention_Technical_pre_work_inspection
    a       bcfOWL:Topic , cto:Task> ;
    bcfOWL:hasAssignedTo
            inst:Oliver_Schulz ;
    bcfOWL:hasCreationAuthor>
            inst:Oliver_Schulz;
    bcfOWL:hasCreationDate>
            "2022-05-24T00:00:00Z"^^xsd:dateTime ;
```

[3]Construction Task Ontology: https://mathib.github.io/cto-ontology/ accessed 24.02.2023
[4]Apache Jena Fuseki: https://jena.apache.org/documentation/fuseki2/ accessed 24.02.2023)

```
    bcfOWL:hasDescription>
            ”Technical pre-work inspection. The quality
            of the facade needs to be checked, some
            measure be performed, and the accessibility
            needs to be ensures.” ;
    bcfOWL:hasDueDate
            ”2022-05-25T00:00:00Z”^^xsd:dateTime ;
    bcfOWL:hasPriority
           inst:InterventionPriority_three ;
    bcfOWL:hasTitle
            ”Technical pre-work inspection” ;
    bcfOWL:hasTopicStatus
           inst:TopicStatus_Open ;
    bcfOWL:hasTopicType
           inst:InterventionPost_Facade_4 ;
    cto:hasTaskContext
           inst:Intervention_External_Insulation_Basiswand ;
    cto:hasTaskMethod>
           inst:TaskMethod_1> ,inst:TaskMethod_2> .
```

Listing 2: Example from connecting bcfOWL with CTO

In addition, the use of bcfOWL in container-based environments such as the Information Container for Linked Document Delivery (ICDD) [7] and the Linked Data Platform (LDP) [8] - in its manifestation of Solid [9], and ConSolid [10] - was investigated during BIM4Ren. Since bcfOWL was missing the ability to point to containers, the ontology was extended with the concepts of *bcfOWL:TopicsContainer*, *bcfOWL:ViewpointsContainer,* and *bcfOWL:CommentsContainer.* These can be used to point to containers that store BCF content about Topics, Viewpoints, and Comments. It is not necessary that the container content is serialised in RDF, but they could also be used to refer to, for example, JSON or XML content. This extension went hand in hand with testing federated Common Data Environments (CDEs), which was investigated in [11] using the CDE subset of BCF as an example.

3. Potential Areas of Improvements

In this section, we discuss the different potential areas for improvement for bcfOWL regarding future developments in the Linked Building Data domain.

3.1. Versioning

As Issue Management is a process of exchanging information, the details of issues can change frequently. For example, the status of an issue may change from *active* when it is created to *closed* after it has been resolved. Although the BCF API can track these changes using an event system that allows queries to see what has changed, bcfOWL initially focused on the core aspects of BCF and did not consider the event system from the beginning.

Versioning for bcfOWL was first investigated in [12]. Different approaches were considered for describing an issue's different versions in a typical Issue Management process. The core idea underlying the various approaches was to avoid deleting information from the graph and instead

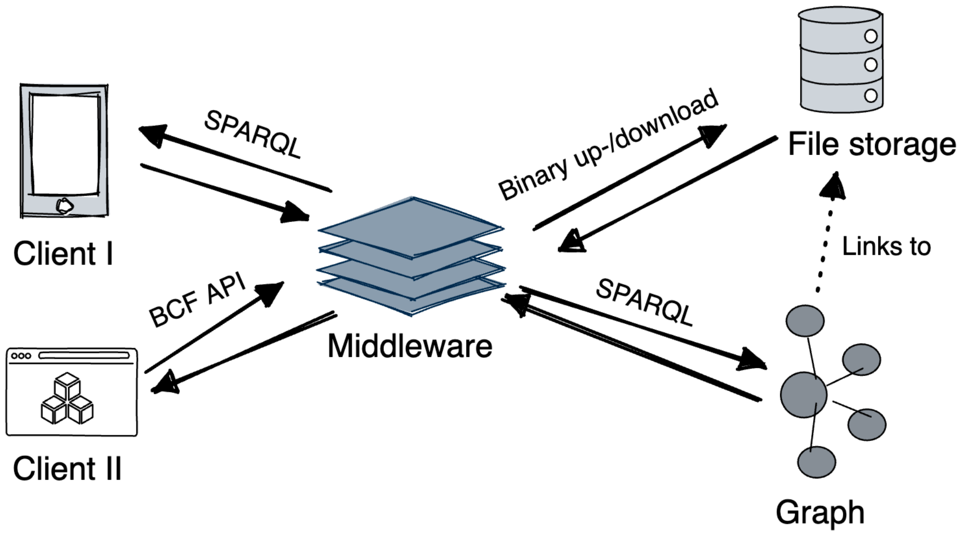

Figure 2: The server infrastructure that was used during the BIM4Ren project.

add new information to it. As a result, the graph retains a comprehensive record of all past events, ensuring a complete history is preserved. The following approaches were considered as most promising to go forward:

1. A *State's* approach, inspired by the LBD ontologies the Ontology for Property Management (OPM) [13] and The Ontology for Managing Geometry (OMG) [14]. These *States* can be used to externalise the information part of a subject to another triple that links to the original subject. Thereby newly created *States* never delete the old *States* and can be filtered by, for example, the parameter time.
2. A *Statement* approach for annotating triples using triples. The approach was first envisioned in [15] and culminated in developing RDF-star [5] which was used for this work.

As of now, a decision on what versioning system should be used is not yet made. The *State's* approach is very much in line with current approaches in the LBD Community Group and is straightforward to implement with current Linked Data technologies. *Statements* with RDF-star showed a performance comparable to the *State* approach but still have open questions, for example, on how to express it in an ontology and how to use it for reasoning. In contrast, the statements approach could be used to only slightly modify the current structure of bcfOWL and add annotations to the triples. The States approach, on the other hand, would mean a significant restructuring of bcfOWL.

While this is not necessary for all processes, versioning should be considered from the beginning when creating new ontologies. For describing version content, PROV-O [16] is considered a starting point.

[5]W3C, RDF-star and SPARQL-star https://w3c.github.io/rdf-star/cg-spec/editors_draft.html accessed 27.02.2023

3.1.1. Signing

We considered the possibility of a user signing the changes on the graph when creating a new problem description or adding descriptions of how the problem was handled. The idea is to use a public key (PK) infrastructure so that only the user can create the signature, but anyone can check who has signed the part of the graph. The signatures can serve two purposes: to indicate ownership of the content, but more importantly, to provide non-repudiation of the claims, e.g. in a situation where someone declares the issue fixed. While such a safeguard is not necessary for all ontologies, it is a very important factor, especially in the construction industry - where many projects end up in court. Data protection should be ensured, particularly in areas where processes, responsibilities, tasks and the completion of tasks are recorded.

Carroll, Sayers et al., and Hogan have proposed algorithms to create a canonical serialisation of an RDF graph in [17], [18] and [19], respectively. The algorithms allow for checksum calculation for sub-graphs and, thus, when encrypted with the user's private key, can be used to sign the content changes.

3.2. Validation

As seen in Figure 2, the client in the BIM4Ren workflow is able to communicate changes and additions in the BCF context via the middleware with the BCF API and SPARQL. While we identified the static structure of the API as a disadvantage in specific scenarios, its rigidness can also be an advantage, for example, when POST and PUT operations are performed. The precise structure of the API prevents incorrectly formed BCF content from being sent to the server. In these cases, the API gives direct feedback to the client and makes it aware of this via error messages. In the case of SPARQL Updates, however, this is not necessarily the case. Although a request is also returned as incorrect if it does not comply with the SPARQL syntax, it is not validated for completeness or content. While OWL allows us to reason, it is not necessarily helpful in validating the data content because of its open-world assumption [20]. The Shapes Constraint Language (SHACL) [21] could be a helpful addition to BCF on the semantic web by validating it with SHACL shapes before sending the content to the graph. An example shape for bcfOWL is provided in Listing 3.

```
bcfOWL:TopicShape
    a sh:NodeShape ;
    sh:targetClass bcfOWL:Topic ;
    sh:property [   sh:path bcfOWL:hasTitle ;
                    sh:datatype xsd:string ;
                    sh:minCount 1 ;
                ] ;
                [...],
    sh:property [
                    sh:path bcf:description ;
                    sh:datatype xsd:string ;
                ] .
```

Listing 3: Example of how a BCF Shape could look like

However, a potential conflict may arise if versioning via RDF-star is used as described in Section 3.1, since SHACL cannot yet work with RDF-star. However, there are first discussions within the SHACL community to incorporate RDF-star [6] [7]. A potential workaround to this problem in the meantime could be to enable validation via regular SPARQL queries - as described in [20] - but this would only be a temporary fallback until the SHACL community resolves this matter.

While the open-world assumption is integral to the Linked Data philosophy, it is not always helpful when it comes to tightly defined schemas. This can be remedied by using SHACL to validate the data for conformance using the shapes even before the content is stored in the graph. It should be noted that OWL and SHACL are not mutually exclusive and can be used complementary to each other. Therefore, if an ontology maps to a fixed schema, the corresponding SHACL shapes should be created in addition to the description in OWL. However, it should be further investigated whether checking incoming issues that first have to be checked by means of SHACL has a negative effect on the performance and responsiveness of the server communication.

3.3. Generic Framework

bcfOWL is a translation of a specific Issue Management format to an ontology. Thereby, its concepts - although theoretically applicable in many different scenarios - are always tightly connected to BCF. This approach is comparable to translating the IFC EXPRESS Schema to ifcOWL [22]. The different components of BCF, the Topics, Viewpoints and Comments, are generally not unique to BCF, but on the opposite, very generic concepts that could be used for scenarios outside Issue Management. As an example:

The concept of the BCF Viewpoint is used throughout the BIM4Ren project to spatially locate images taken on-site in the digital building model, as seen in [3].

BCF is the only standardised and widely supported format that is currently allowing to achieve this in a software-independent manner. But in order to use this concept, the Viewpoint is always accompanied by the BCF concepts of Topics and Comments. Furthermore, the Viewpoint lacks vital information to be usable on its own. For example, it does not come with an author or a date. This information has to be saved to the Topic. However, this only partially solves the problem since a Topic can have multiple Viewpoints simultaneously. Therefore, from an ontological perspective, using the Viewpoint from BCF to describe camera views of real pictures on-site without intending to remain in the BCF context makes little sense.

Although bcfOWL and BCF use the general concepts, they do not describe them sufficiently to be used meaningfully in other scenarios. Therefore, more generic frameworks are needed to describe these concepts that can then be used to create BCF information.

A solution to this problem could be to base bcfOWL on the principles of Ontology Design Patterns (ODP) [23]. These principles require keeping ontologies small and modular since large ontologies tailored to specific schemas are usually challenging to reuse for other purposes. Suitable examples in the LBD community built on these principles are the Building Topology

[6] GitHub, Modifications to SHACL to incorporate RDF-star: https://github.com/w3c/rdf-star/issues/122 accessed: 24.02.2023

[7] GitHub, RDF-star and SPARQL-star support for SHACL: https://github.com/w3c/shacl/issues/15 accessed: 24.02.2023

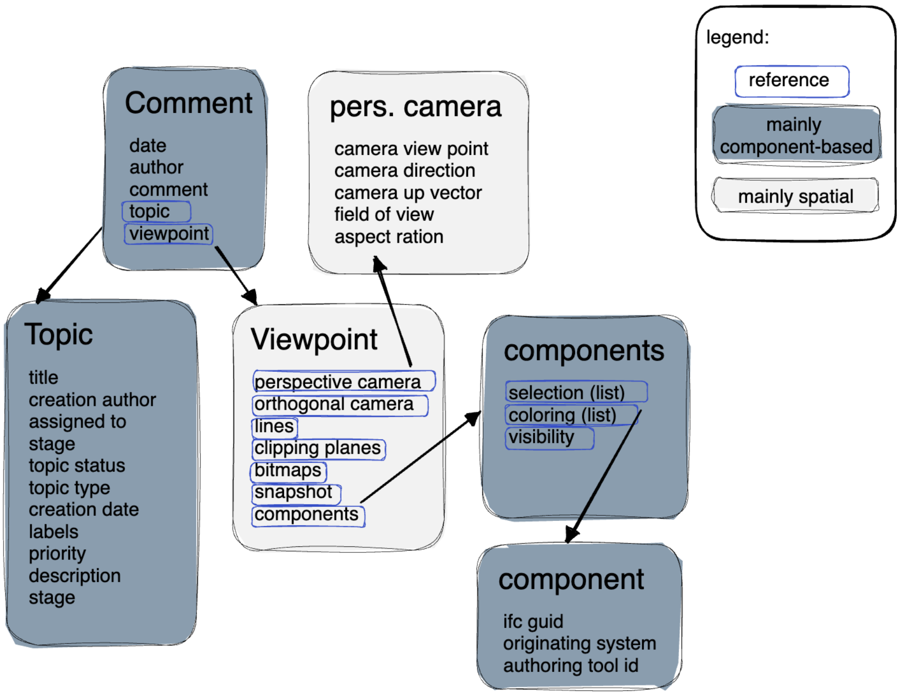

Figure 3: The main concepts of BCF are the Topic, the Comment and the Viewpoint with its many sub-components. The information in this these concepts can be classified into two categories: component-based communication and spatial

Ontology (BOT) [6] for describing the core components of a building, like zones, spaces and elements and the Damage Topology Ontology (DOT) [5] for describing damages to buildings.

Although many ontologies are created to bring an existing schema into the Semantic Web, it should be checked beforehand whether they can be built on top of existing higher-level ontologies. If these do not exist, these ontologies could be created first, potentially keeping the ODP principles in mind.

The core components of BCF and bcfOWL can be divided into two main concepts: component-based communication information and spatial information. Even though these components cannot always be clearly assigned to only one of the concepts, we will show the most important overlaps in the following two subsections.

3.3.1. Component-Based Communication Information

The concepts of Topics and Comments in BCF mainly contain semantic information for carrying messages and providing a context for these messages. An example structure for BCF can be seen in Figure 3. Most of the properties in these concepts - like the title, status, assignment etc. - can be used for communication in general. For example, the ISO 10303 [24] describes an approval process that contains substantial overlaps with the BCF information. The VDI/bs 2552 Part 11.2 [25] describes another component-based process with significant overlaps with the BCF standard. These examples show that many component-based communication processes are founded on a common information root, but no connection exists between them. Furthermore, is the link in BCF to the targeted building elements in the communication relatively weak and placed in a sub-concept of the Viewpoints. This prevents the potential of BCF from being adequately exploited. By linking to the respective building elements, it should be possible to find out which problems selected building elements have or have had in the past. This would be an essential function in the digital planning of a building and its elements over its entire life cycle - for example, in the domain of facility management.

However, this link in BCF cannot be easily queried from the server without querying or downloading the entire database [3]. bcfOWL can circumvent this problem by querying via SPARQL, but this again has the disadvantage that this can only happen in a BCF context. If information were noted for building elements in a non-BCF context, this would lead to a theoretical misuse of bcfOWL. A higher-level framework for component-based communication could bypass this problem and maintain compatibility with different processes.

Many ontologies, processes and schemas represent different subsets of component-based communication. Still, a higher-level schema is missing so far, which would allow for establishing compatibility and synergy between them. The higher-level ontology can then be used to describe the content of, e.g. BCF without clearly labelling it as bcfOWL. In fact, it could be argued that bcfOWL itself does not need to be specifically defined but that certain SHACL shapes, which check for BCF compliance and use the higher-level ontology, define whether the incoming data is BCF compatible or not.

3.3.2. Spatial Information

BCF provides users with a standardised and software-independent way to locate camera positions in a digital building model. While primarily targeted towards BIM, it can also be used in CAD or non-BIM models. While this is a significant achievement, it again comes with the disadvantage that this is always connected to the BCF context. Therefore, the core parts of the Viewpoint responsible for spatially locating an issue could be described in an upper-level ontology for locating and aligning geometry and documents like plans, models and images or, in summary: spatial representations of the building. These spatial representations are supposed to be structured in such a way that they can be placed in a spatial context to each other. This would allow heterogeneous data to be spatially linked to each other, which is a missing feature, especially for existing buildings, where often no coordination process of the digital data has taken place. Nevertheless, this process is also relevant for new buildings to be planned, too, since the coordination process is not always carried out in a satisfactory manner, and the possibility of spatial linking means that these errors can be rectified more quickly. An initial approach to the concept of spatial representations in BCF can be seen in [4], where BCF has been extended to include the functionality to locate damage in a digital building which could potentially be used to locate DOT damages.

In addition to the semantic component of communication, a further investigation of the spatial context becomes evident. Although we can already locate and link photos and geometries to each other, many open questions remain about how spatial representations should be linked to each other in a spatiotemporal context. A framework that allows us to link heterogeneous data spatially in a machine-readable way is not yet clearly available for the AEC domain, neither by legacy means nor in the Linked Data context. Although this problem is usually circumvented by methods such as coordination processes, this does not always prevent errors. Without a spatial structure, correcting these errors can be very time-consuming. A spatial framework based on ODPs could address this issue as well.

4. Conclusion

The bcfOWL ontology, developed as part of the EU Horizon 2020 BIM4REN project, provides a promising approach to facilitating the communication of issue-related data in the context of BIM. By providing a means of expressing issue data on the Semantic Web, bcfOWL enhances the query ability and connectivity in the context of Linked Building Data, serving as an interlanguage for communication between different systems and domains. However, the use of bcfOWL in the BIM4Ren project has provided new insights into its usability, which this paper discusses.

It was discovered during the process that the ontology is insufficient for the functionality needed for use- cases in other component-based communication processes. While these processes share many similarities with the BCF concepts, they still differ in ways that prevent proper deployment. To address these shortages, this paper proposes decomposing the BCF concepts into parts to form BCF from several generic multi-purpose ontologies based on the Ontology Design Patterns, enabling other similar processes. Furthermore, with future developments, aspects such as versioning and validation should be strengthened in order to ensure the best possible applicability.

The consideration made in this paper should complement the original design of bcfOWL. While the suggestions made in [1] should ensure a sound basis when creating an ontology, this paper focuses on potential improvements and considerations that should help to make the ontologies concepts more reusable.

It is imperative to mention that a basic generalisation of our observation is impossible. However, it remains to be said that the translation of a schema from legacy formats to Linked Data is a frequently chosen approach, in which our presented points should at least be checked for applicability. This should benefit the overall goal of making ontologies more reusable, thus avoiding too much duplication of content between different ontologies.

Therefore, future work will further pursue component-based communication and spatial contextualisation. The goal is to establish higher-level ontologies that can be used to represent general functionalities and processes within buildings and CDEs and from which, in turn, the core components of, e.g. BCF can be derived. A comprehensive investigation is required to estimate how serving legacy formats, such as BCF, using the Semantic Web technology stack impacts the performance and robustness of client-server communication. The prototype servers, which were created within the framework of BIM4Ren to link BCF communication with Linked Data, cannot be sufficiently compared with servers from commercial providers. Nevertheless, by conducting a thorough evaluation, we can determine the optimal level of feature integration that maximizes both user value and server performance.

Acknowledgments

This research has been funded by the European Unions Horizon 2020 project BIM4Ren.

References

[1] O. Schulz, J. Oraskari, J. Beetz, bcfOWL: A BIM collaboration ontology, in: Proceedings of the 9th Linked Data in Architecture and Construction Workshop, Luxembourg, 2021, pp. 1–12. https://linkedbuildingdata.net/ldac2021/files/papers/CIB_W78_2021_paper_122.pdf.

[2] buildingSMART, BIM Collaboration Format (BCF), https://technical.buildingsmart.org/standards/bcf/, no date.

[3] O. Schulz, J. Beetz, Image-documentation of existing buildings using a server-based BIM Collaboration Format workflow., in: EG-ICE 2021 Workshop on Intelligent Computing in Engineering, Berlin, 2021, pp. 108–117. doi:10.14279/depositonce-12021.

[4] O. Schulz, Spatialization of damage to buildings using a BIM collaboration Format approach, in: 32. Forum Bauinformatik 2021, Darmstadt, Germany, 2019, pp. 341–348. doi:10.26083/tuprints-00021521.

[5] A.-H. Hamdan, M. Bonduel, R. Scherer, An ontological model for the representation of damage to constructions, in: Proceedings of the 7th Linked Data in Architecture and Construction Workshop - LDAC2019, Lisbon, Portugal, 2019, pp. 64–77. https://ceur-ws.org/Vol-2389/05paper.pdf.

[6] M. H. Rasmussen, M. Lefrançois, G. F. Schneider, P. Pauwels, BOT: The building topology ontology of the W3C linked building data group, Semantic Web 12 (2019) 143–161. doi:10.3233/SW-200385, https://research.tue.nl/en/publications/bot-the-building-topology-ontology-of-the-w3c-linked-building-dat.

[7] ISO, ISO 21597: Information container for data drop - Exchange specification, https://www.iso.org/standard/74389.html, 2020.

[8] S. Speicher, J. Arwe, A. Malhotra, Linked Data Platform 1.0, https://www.w3.org/TR/ldp/, 2015.

[9] A. V. Sambra, E. Mansour, S. Hawke, M. Zereba, N. Greco, A. Ghanem, D. Zagidulin, A. Aboulnaga, T. Berners-Lee, Solid: A Platform for Decentralized Social Applications Based on Linked Data, Technical Report, 2016. http://emansour.com/research/lusail/solid_protocols.pdf.

[10] J. Werbrouck, P. Pauwels, J. Beetz, E. Mannens, ConSolid: A Federated Ecosystem for Heterogeneous Multi-Stakeholder Projects, http://www.semantic-web-journal.net/content/consolid-federated-ecosystem-heterogeneous-multi-stakeholder-projects, 2022.

[11] J. Oraskari, O. Schulz, J. Werbrouck, J. Beetz, Enabling Federated Interoperable Issue Management in a Building and Construction Sector, in: Proceedings of the 29th EG-ICE International Workshop on Intelligent Computing in Engineering, EG-ICE, 2022, pp. 92–101. doi:10.7146/aul.455.c200, https://ebooks.au.dk/aul/catalog/view/455/312/1848-2.

[12] J. Oraskari, O. Schulz, J. Beetz, Towards describing version history of BCF data in the Semantic Web, in: Proceedings of the 10th Linked Data in Architecture and Construc-

tion Workshop Co-Located with 19th European Semantic Web Conference (ESWC 2022), Hersonissos, Greece, 2022, pp. 87–98. https://ceur-ws.org/Vol-3213/paper08.pdf.
[13] M. H. Rasmussen, M. Lefrancois, M. Bonduel, C. A. Hviid, J. Karlshø, OPM: An ontology for describing properties that evolve over time, in: Proceedings of the 6th Linked Data in Architecture and Construction Workshop, London, United Kingdom„ 2018, p. 11. https://ceur-ws.org/Vol-2159/03paper.pdf.
[14] A. Wagner, M. Bonduel, P. Pauwels, R. Uwe, Relating geometry descriptions to its derivatives on the web, in: Proceedings of the 2019 European Conference for Computing in Construction, European Council on Computing in Construction (EC3), 2019, pp. 304–313. doi:10.35490/ec3.2019.146, http://hdl.handle.net/1854/LU-8633667.
[15] O. Hartig, B. Thompson, Foundations of an Alternative Approach to Reification in RDF, http://arxiv.org/abs/1406.3399, 2014. doi:10.48550/arXiv.1406.3399. arXiv:1406.3399.
[16] P. Missier, K. Belhajjame, J. Cheney, The W3C PROV family of specifications for modelling provenance metadata, in: Proceedings of the 16th International Conference on Extending Database Technology, EDBT '13, Association for Computing Machinery, New York, NY, USA, 2013, pp. 773–776. doi:10.1145/2452376.2452478, https://doi.org/10.1145/2452376.2452478.
[17] J. J. Carroll, Signing RDF Graphs, in: D. Fensel, K. Sycara, J. Mylopoulos (Eds.), The Semantic Web - ISWC 2003, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2003, pp. 369–384. doi:10.1007/978-3-540-39718-2_24.
[18] C. Sayers, A. Karp, Computing the Digest of an RDF Graph, Technical Report, Mobile and Media Systems Laboratory HP Laboratories Palo Alto, 2004. https://www.semanticscholar.org/paper/Computing-the-digest-of-an-RDF-graph-Sayers-Karp/e49756a0bf7bcf6ce4b033c4f5261b283d0be394.
[19] A. Hogan, Canonical Forms for Isomorphic and Equivalent RDF Graphs: Algorithms for Leaning and Labelling Blank Nodes, ACM Transactions on the Web 11 (2017) 1–62. doi:10.1145/3068333.
[20] H. Knublauch, SHACL and OWL Compared, https://spinrdf.org/shacl-and-owl.html, 2017.
[21] H. Knublauch, D. Kontokostas, Shapes Constraint Language (SHACL), Technical Report, World Wide Web Consortium (W3C), 2017. https://www.w3.org/TR/shacl/.
[22] J. Beetz, J. van Leeuwen, d. Vries, IfcOWL: A case of transforming EXPRESS schemas into ontologies, AI EDAM 23 (2009) 89–101. doi:10.1017/S0890060409000122.
[23] A. Gangemi, V. Presutti, Ontology Design Patterns, in: Handbook on Ontologies, 2009, pp. 221–243. doi:10.1007/978-3-540-92673-3_10.
[24] ISO, ISO 10303-1012:2018 - Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 1012: Application Module: Approval, Technical Report, International Organization for Standardization, 2018. https://www.iso.org/standard/76144.html.
[25] VDI, VDI/bS 2552 Part 11.2: Building Information Modeling - Exchange Requirements - Slots and Openings, Technical Report, Verein Deutscher Ingenierue (VDI), buildingSMART, 2022. https://www.vdi.de/richtlinien/details/vdibs-2552-blatt-112-building-information-modeling-informationsaustauschanforderungen-schlitz-und-durchbruchsplanung.