Towards a Decentralised Common Data Environment using Linked Building Data and the Solid Ecosystem

Jeroen Werbrouck[1,*], Pieter Pauwels[2], Ghent University[2], Jakob Beetz[2], Léon van Berlo[3],
[1] Ghent University, RWTH Aachen University
[2] RWTH Aachen University
[3]Netherlands organisation for applied scientific research TNO
*email: jeroen.werbrouck@ugent.be

Abstract

With the emergence of Building Information Modelling (BIM), the construction industry is rapidly catching up with the digital revolution that has boosted productivity in virtually all economic sectors. In current practice, the focus of BIM lies on exchange of documents, often through proprietary formats exchanged using the Industry Foundation Classes (IFC). However, with web technologies such as RDF, OWL and SPARQL, a data- and web-based BIM paradigm becomes within reach. The decentralisation of data and decoupling of information and applications will enhance a more general adoption of Big Open BIM, and is expected to lower the BIM threshold for smaller companies that are active in different phases of the building life cycle. Since one of the promises of the Semantic Web and Linked Data is a highly improved interoperability between different disciplines, it is not necessary to reinvent the wheel for the setup of an infrastructure that supports such a network of decentralised tools and data. In this paper, we evaluate the specifications provided by the Solid project (Inrupt Inc.), a Linked Data-based ecosystem for *Social Linked Data*. Although the exemplary use case of the Solid ecosystem is decentralisation of data and applications for social network purposes, we notice a considerable overlap with recent ambitions and challenges for a web-based AECO industry (Architecture, Engineering, Construction and Operation). This includes standardised data representations, role- or actor-based authorisation and authentication and the need for modular and extensible applications, dedicated to a specific use case. After a brief introduction to Linked Data and its applications in the building industry, we discuss present solutions for building data management (Common Data Environments, multi-models, etc.). In order to translate these approaches towards a Linked Data context with minimal effort and maximal effect, we then review the Solid specifications for use in a construction-oriented web ecosystem. As a proof of concept, we discuss the setup of a web-service for creation and management of Linked Building Data, generated with the Solid-React generator. This application is envisaged as a bridge between the multiple data stores of different project stakeholders and the end user. It acts as an interface to a distributed Common Data Environment that also allows the generation of multi-models.

Keywords: Linked Building Data, Common Data Environments, Decentralisation, Solid

1. Introduction

The building industry is one of the most fragmented industries around the world. At the same time, it is also among the least digitised ones, only leaving 'Agriculture and Hunting' behind (McKinsey & Company, 2015). Although the recent upcoming of Building Information Modelling (BIM) has somewhat closed the gap, multiple challenges still need to be overcome before the sector reaches the full potential offered by digitisation. In other words, the use of integrated, interoperable data, exchanged through connected web services, or the reaching of 'BIM Maturity Level 3' as defined in the notorious BIM wedge (Fig. 1). The main goal of web-based BIM is to provide an answer to data islands that complicate lossless exchange and collaboration between disciplines that focus on the built environment. These disciplines form the AECO industry (Architecture, Engineering, Construction and Operation).

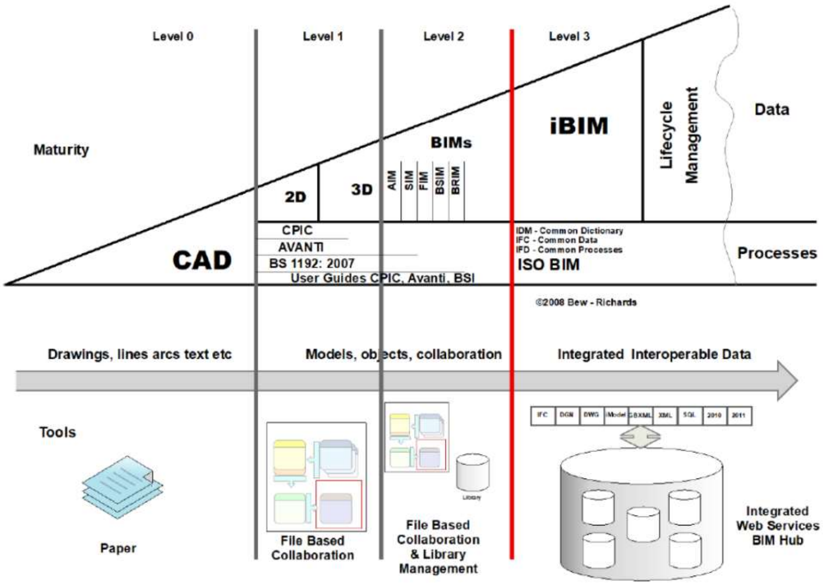


Figure 1: The BIM levels of Maturity as defined by BSI Standards Limited, 2013

Linked Data technologies are considered very promising to reach such high level of interoperability. These technologies rely on the Resource Description Framework (RDF)[1], which is a data model that has been standardised as part of the Semantic Web technology stack (Berners-Lee, Hendler, & Lassila, 2001). Using RDF, individual data concepts can be linked to one another in the form of triples. RDF triples can be understood as very basic sentences, linking a *subject* to an *object* with help of a *predicate* that states the exact relationship between the two. To ensure that each data item is uniquely identifiable over the web, it is characterised by a Uniform Resource Identifier (URI). This strategy, which is the main difference between RDF and other existing data models, allows to semantically enrich information in an open-world way: anything can be said about anything, which opens up possibilities for interdisciplinary collaboration. Furthermore, due to the structured representation of data in the form of a web-wide graph, digital agents are capable of semantically interpreting this data and using it for specific purposes with minimal human intervention. A distinction can be made between the terminology that is used to describe data (e.g. a taxonomy, a *type*) and the actual data individuals that are semantically connected using these structures (e.g. a specific object). The former is referred to as the TBOX (‘Terminological’), the latter as the ABOX (‘Assertion’). Knowledge models for defining conceptual domain schemas are mostly on a TBOX level. Such schemas are called ‘vocabularies’ or ‘ontologies’[2] and give semantic meaning to certain data elements. The combination of multiple vocabularies allows to set up complex models of real problems.

The application of Linked Data concepts for the AECO Industry has been documented multiple times by now (Beetz, Van Leeuwen, & De Vries, 2005; Pauwels & Terkaj, 2016; Pauwels, Zhang, & Lee, 2017; Rasmussen et al., 2017), because it could seriously enhance interoperability between disciplines and collaboration between stakeholders. Linked Data allows to construct vocabularies for each related sub-discipline, connected with one another in a neutral format and distributed over multiple servers. This allows applications that are concerned with different aspects of the built environment to connect and exchange information without information loss. It also allows to link to open data on the web, such as contextual information (e.g. geospatial, governmental, historical or weather data). Furthermore, since vocabularies contain the information to interpret the data in a semantic way, automatic reasoning and rule checking comes within reach (Pauwels et al., 2017). With this in mind, several researchers are currently working on making Linked Data-based BIM more mature. First proposed in (Beetz et al., 2005), the setup of a Linked Data-based version of IFC was officially approved as the ifcOWL ontology (Pauwels & Terkaj, 2016). However, as ifcOWL covers the entire

[1] https://www.w3.org/TR/rdf11-concepts/, W3C

[2] https://www.w3.org/standards/semanticweb/ontology.html, W3C

IFC schema, it is very large and consequentially not really flexible to deal with topics that are not sufficiently covered in the IFC standard, such as existing buildings, Geographic Information Systems (GIS), Facility Management (FM) or circular economy. A new paradigm is therefore proposed by the W3C Linked Building Data Working Group[3], targeting the development of more modular vocabularies that each address a specific building-related topic (Rasmussen et al., 2017). The main advantage of such modular Linked Data vocabularies over traditional, monolithic, domain-specific information models such as the IFC schemas is their modularity, which increases the flexibility to address specific challenges.

While much research and standardisation has been done regarding the Semantic Web, in general as well as for construction, the number of applications that effectively use its potential remains limited: Linked Data technologies have a steep learning curve and there is quite an implementation threshold for developers to actively contribute to an interconnected network of online applications (Verborgh, 2018). In order to stimulate the worldwide developer community, and to separate personal data from the applications that use it, the recent Solid project (Social Linked Data) was founded (Mansour et al., 2016). It aims at a decentralised ecosystem for social web applications, in which users own their data themselves and allow external (micro)applications to use it for a certain (social media) purpose. However, social media are just one use case picked by the Solid team to illustrate its concept. The same set of specifications and tools can be used for other domains as well, especially for disciplines as fragmented and decentral as the construction industry.

Multiple research projects are focusing on the development of knowledge models for describing the built environment in a Semantic Web context. In order to implement these knowledge models and move towards BIM Maturity Level 3, usable applications need to be developed. Modular web applications for the building industry, able to communicate with each other and exchange disparate data with use of HTTP and JSON, were recently re-introduced as 'BIM bots'[4], often built using the BIM bot compliant BIMserver framework (Beetz, van Berlo, de Laat, & van den Helm, 2010), a well-known implementation of Model Servers that is based on IFC. Going one step further, the idea of BIM bots could be extended with Linked Data, broadening their range towards topics that are not typically covered by IFC. As the Semantic Web technology stack offers the possibility to connect various domains, there is no need to start a specific Linked Data ecosystem for construction. With this in mind, this paper discusses the use of the specifications provided by the Solid project. As we consider a way for collaborative project data management the first step for a network of Linked Building Data tools, a prototypical management service is implemented within the Solid ecosystem as proof of concept, using the Solid-React generator, which hides the complexity of authentication, authorisation, data storage etc.

2. Collaboration strategies for building projects

Different strategies exist by now to reach a more streamlined collaboration within building projects. A Common Data Environment (CDE) is a virtual storage location for collecting and managing documentation of building projects, mostly offered as a cloud service. Because all project information is managed in this common environment, the chance of misunderstanding and information loss is strongly reduced. Access to certain documents can be defined by the project's Information Delivery Manual (IDM), based on the international BuildingSMART standard for bundling and structuring Information Requirements (IR). Examples of commercial CDEs are, for instance, Autodesk's BIM 360 and Trimble Connect. CDEs can be optimised to work with certain data formats, either proprietary or open. Standards such as IFC allow information to be exchanged between applications from different software vendors, which, in most cases, use proprietary data formats natively. The BIMserver platform (bimserver.org) implements IFC in a model server and allows developers to build web services or plugins that can be connected to the shared IFC database and exchange data via a common API (e.g. BIMSie (van Berlo, 2015)). Recent developments show that the main CDE suppliers are open to develop a standardised open API for CDEs. This initiative just recently started as a follow up from the German DIN SPEC (91391-1/2) initiative that defines the features and expectations from a CDE.

[3] https://w3.org/community/lbd/, Linked Building Data Community Group

[4] http://bimbots.org/, Van Berlo, L.

An approach related to CDEs, but more focused on the container structure in which project information can be stored, are the so-called multi-models. The most recent development in this domain is the Information Container for Data Drop (ICDD), which is in the final stage of standardisation as ISO 21597. It is the successor of the Dutch COINS standard and follows a tradition of ontology-based multi-model management (Gürtler, Baumgärtel, & Scherer, 2015; Schapke, Katranuschkov, & Scherer, 2010; Törmä, 2013). In its own words, ICDD has been developed '*in response to the need of the construction industry to handle multiple documents as one information delivery or data drop'*. Its internal structure is based on two ontologies: a Container ontology and a Linkset ontology. As the former defines the classes and properties used for description of metadata about the container, the latter provides the definitions for the semantic links between documents. The standard allows to refer to identifiers on a sub-document level, such as individual IFC GUIDs or pixel zones within an image. An ICDD container can be considered a semantically connected 'dump' folder with the available project information, and can, for example, be used when project data needs to be transferred from one project stakeholder to another. A container in *.iccd format (using a ZIP compression) has a fixed structure, containing subfolders with the main ontologies, the 'payload documents' (e.g. imagery and IFC files) and the 'payload triples' for the relationships between those documents. The main folder contains an *index.rdf* file that contains the documents of the project and some metadata. The *content.rdf* file that is stored in the 'payload triples' folder can also refer to external documents or URIs. An overview of the ICDD structure is given in Fig. 2, depicting the folder structure and a subset of a linkset in Turtle format.

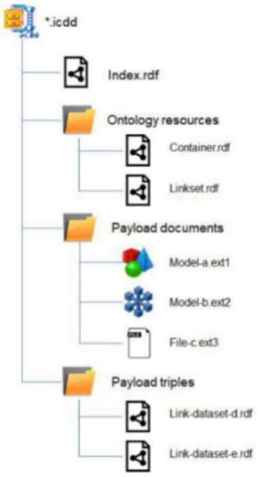


Figure 2: Structure of an ICDD container (source: ISO 21597-1)

In this section, a brief introduction was given to CDEs and the ICDD standard for multi-model containers. While CDEs focus on a cloud environment for enhanced collaboration, Linked Data-based multi-models allow to establish specific links between project documents and sub-document information. A Linked Data-based CDE would need to combine the possibilities offered by CDEs, while allowing to establish links between RDF data and documents, on a fine-grained data level. This way, project sources such as imagery, point clouds, geometry etc. can be linked with RDF graphs about topology, products and properties. In the next section, we review the specifications of the Solid ecosystem for use in such a Linked Data-based CDE.

3. Solid specifications

The initial use case of the Solid ecosystem is social media. Decoupling of applications and data lets users stay the owner of their own data and allow the applications that suit their needs the most to access that data. From a more general perspective, it embodies a movement that aims to realise the web 'as originally envisaged' by Tim Berners Lee (Berners-Lee, Dimitroyannis, Mallinckrodt, & McKay, 1994; Berners-Lee et al., 2001): personal data storage, standard communication between apps and the use of a 'universal' data format in the form of RDF[5]. The decoupling of apps and data is as relevant for the building sector as it is for social data. The BIMserver realises this within context of IFC; the Solid framework could be a candidate for doing this in a Linked Data context. In the following section, we review the main specifications for the Solid framework[6], and suggest how they could be implemented for projects dealing with the built environment.

3.1 Identification and authentication

In Solid, identification of actors happens through WebIDs[7]. In Section 1, it was stated that in Linked Data, every 'resource' is identified through a unique identifier over the web, a URI. Applied to WebIDs, this means that any actor can have their own URI, that can relate to other resources (e.g. personal data or other people's WebIDs) and be linked to by other resources as well. This generates a global 'social graph' with fundamental relationships between a person, his data and other people's data. Within Solid, the WebID is used as a URL (Uniform Resource Locator), which is a specific type of URI that, apart from identifying the resource, also allows to access it. An example of a WebID could be *https://jwerbrouck.solid.community/profile/card#me.* It identifies both the resource that should be included in triples that link to this person, and the web address that allows to access his personal data card. This card is part of the 'datapod' of the actor: the server (personal or hosted by a providers) that contains an inbox, a public folder and a private folder. According to the specifications, a 2019 update will include a section where access rights for third-party applications can be managed: read, write, append and control. At the time of writing, this is only managed by logging in to the application with the personal webID. This system of datapods that allow specific apps to access the data is the main difference with current social networks, that require users to upload their data. The WebID also serves as the basis for authentication of actors, instead of usernames, by making use of the WebID-TLS protocol[8] and cryptographic certificates generated by the browser.

3.2 Authorisation

To determine whether a certain actor (person or application) has access to particular data, an RDF-based authorisation mechanism called 'Web Access Control' (WAC) (Hollenbach, Presbrey, & Berners-Lee, 2009) is used. Actors can be identified by their WebIDs, which are stored in *.acl graphs that state which webIDs should be allowed to read or update data. The specification also allows to refer to groups of users instead of specific ones. An *.acl file can be linked to an entire folder (such as the basic folders for public and private data), but can also be mapped in a more fine-grained way to subfolders or individual files or RDF graphs. Currently, the specification does not allow for mapping on resource level, which could be beneficial for Linked Data-based building projects (Oraskari & Törmä, 2016).

3.3 Content Representation

In general, datapods make a difference between Linked Data resources (e.g. in the form of Turtle, JSON-LD, etc.) and non-Linked Data resources (e.g. imagery, PDF files, etc.). All the resources are grouped into subfolders of either the private or the public data folder. As indicated in section 3.2, access rights can be regulated per graph or document, based on the .acl specification.

[5] https://inrupt.com/, Inrupt

[6] https://github.com/solid/solid-spec, Inrupt

[7] https://www.w3.org/2005/Incubator/webid/spec/identity/, W3C

[8] https://www.w3.org/2005/Incubator/webid/spec/tls/, W3C

3.4 Solid SDK

As indicated in Section 1, the number of applications that use Linked Data remains limited. Since a decentralised and distributed ecosystem such as Solid relies on third party developers for application development, it needs to lower this development threshold. With this in mind, a dedicated Software Developer Kit (SDK) for programming in React and Angular is provided. This includes a generator that preconfigures the solid specifications (e.g. for security settings), so developers can focus on the functionality they want to implement, instead of implementing the specifications themselves. It also includes the Comunica framework (Taelman, Van Herwegen, Vander Sande, & Verborgh, 2018) for querying Linked Data on the web, with SPARQL (SPARQL Protocol and RDF Query Language)[9] or its less verbose and more accessible variants GraphQL-LD (Taelman, Vander Sande, & Verborgh, 2018) and LDflex (Verborgh, 2018). The use of these more accessible query languages is also expected to improve the development rate of Linked Data applications.

3.5 Solid for construction

The above mentioned specifications of Solid could easily be implemented for an AECO-oriented CDE, possibly extended towards an ecosystem of Linked Data BIM bots. Future research should determine the optimal configuration, although for this first paper we propose a preliminary approach, which will also be used in the proof-of-concept application of Section 4.

In this workflow, the project manager creates a private project subfolder in his or her datapod. All project stakeholders have their own datapod as well, and get the URI of the main project folder of the manager. This does not mean that they have access to the contained files, as this can be managed individually. This project folder should contain at least two graphs in the private data section. The first one defines the minimal project information; the 'skeleton' of the project in the form of its topology. We suggest to use the modular Building Topology Ontology (BOT) (Rasmussen et al., 2017) as recommended by the W3C Linked Building Data Community Group. BOT allows to identify different zones within a building (site, building, storey, space) and define the relationships between them. The second graph contains the WebIDs of different stakeholders and their function(s) in the project. This stakeholders graph will be used to define access rights based on the role in the project. In case a handover of the project data happens, or roles are switched within one project, access rights can therefore immediately update.

Project stakeholders then create a project folder in their own pod as well, and upload the information that is their responsibility, while linking it to the general project URI from the manager. They can request access to project information from other stakeholders, to be able to link to this information as well. Granting access to other project information might be done by the manager (who has access to the entire project database by default) or by the responsible stakeholder. This way, an interlinked, virtual environment starts to exist, which links the data and documents from the project stakeholders into a web-based graph; a Linked Data-based Common Data Environment. Ideally, only binary information (imagery, point clouds) is stored as documents, while other information such as topology and product data is stored in an RDF graph. However, the described workflow also supports a more traditional document-based approach: a project management app that implements the ICDD ontologies could easily generate an ICDD dump file of the project, mirroring the distributed online model.

Apart from this rough workflow, the Solid SDK and implementation of, for example, GraphQL-LD to ease querying, could significantly stimulate development of Linked Building Data applications. A general workflow for aligning such applications needs to be made in the future, so they can automatically communicate with one another and exchange information that is required for a certain use case. This workflow could be based on the IDM and MVD specifications and include a method to describe use cases in a modular way, where tools can be chained together, automatically validating their inputs and outputs, for example with use of SHACL (Shapes Constraint Language)[10]. Such tools can

[9] https://www.w3.org/TR/rdf-sparql-query/, W3C

[10] https://www.w3.org/TR/shacl/, W3C

then use the RDF data in the distributed project pods for advanced reasoning, simulation, model enrichment etc. At the moment, however, the first step is the setup of a prototype application for basic management of building data.

4. Proof of concept: ConSolid

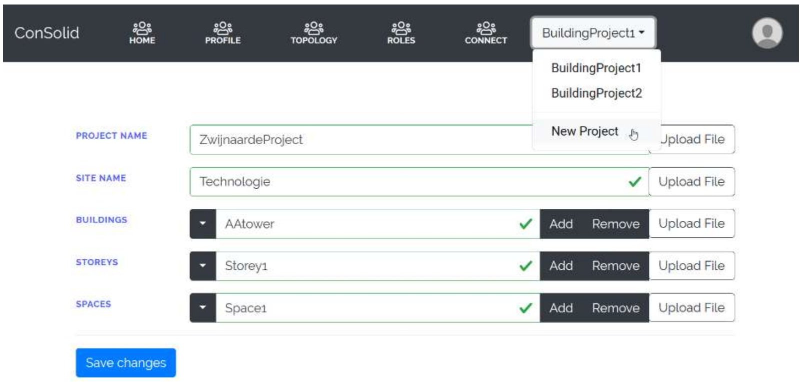
In this section, the workflow discussed in Section 3.5 will serve as a guideline for the development of a rudimentary application for building data management, which is entitled ‘ConSolid’ (Construction Solid)[11]. In the ideal case, the ConSolid app aids in the following building project management topics:

- Configuration of the building project and its topology
- Semantic enrichment of a building project with geometry, product data and properties
- Configuration of the stakeholders and their role within the project
- Uploading documents to building project folders
- Linking document-based information to each other and to RDF graphs
- Validating the distributed model with predefined validation shapes (e.g. SHACL)
- Querying building project information
- Generating ICDD project containers

The test application was generated with the Solid React generator[12]. As the development is work in progress, not all functionality has been implemented at the time of writing: initial work has focused on implementing project and topology creation, role assignment and uploading, linking and querying of project information by the project manager. Semantic enrichment of the project with product data, properties and geometric information has not been implemented yet. Strategies to integrate SHACL validation in the ConSolid tool are discussed in (Werbrouck, Pauwels, Senthilvel, & Beetz, 2019).

4.1 Basic functionality

The generated application includes login functionality with one’s WebID. After logging in, the WebID is stored, which allows the application to perform actions with the data stored in the user’s pod. In the current implementation, we distinguish 3 main tabs, apart from the general ‘Home’ and ‘Profile’ tab: ‘Topology’, ‘Roles’ and ‘Connect’, respectively for managing the basic information of the building, stakeholders and their roles and the connection of distributed project data.



PROJECT NAME: ZwijnaardeProject

SITE NAME: Technologie

BUILDINGS: AAtower

STOREYS: Storey1

SPACES: Space1

Figure 3: Loading a building project from a user’s datapod in the ‘Topology’ tab

[11] The project code is available at https://github.ugent.be/jmauwerb/solid-cde

[12] https://solid.inrupt.com/docs/writing-solid-apps-with-react

4.2 Project Topology

If the user is the owner of any project, it can be selected in the navigation bar and loaded from the distributed pods of the stakeholders (Fig. 3). The basic building topology is then displayed in the ‘Topology’ tab, where it can be updated. Documents may be uploaded to the pod and linked to particular zones in the building, as an alternative to the more advanced linking in the ‘Connect’ tab. If no projects have been created yet, a new project can be created from scratch, stating the basic topology and setting the creator of the pod as the project manager. As mentioned earlier, the ontology that is used for topology description is BOT, although other ontologies such as ifcOWL could be used as well.

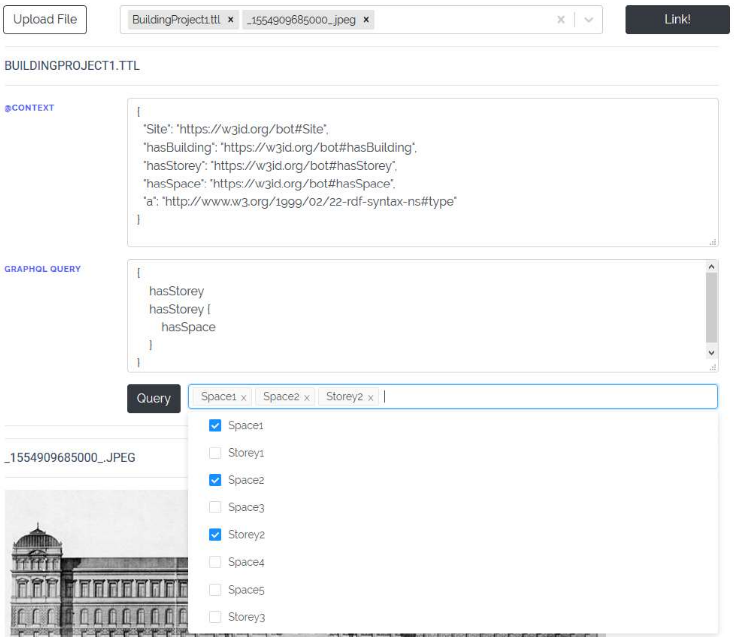
4.3 Role management

The ‘Roles’ tab allows the project manager to assign certain roles to certain WebIDs. These roles are stored in a graph in the main project pod (using FOAF) and can be used for managing access rights. A schema may be developed with some standard access rights that are mapped to certain roles, to automatically set the rights that occur the most. Role groups can be configured from this data, which means a more flexible approach than person-based access rights.

4.4 Data connection and container generation

In the ‘Connect’ tab, the user can upload files to their project pod. A stakeholder that is logged in may upload files to her pod. Project files to which the user has access rights can be linked to one another, as documents as well as on sub-document level. As indicated in Section 3.3, these project documents can be Linked Data as well as non-Linked Data.

In order to allow optimal linking of documents and triples, different strategies may be used for different file formats. Since the ICDD specification does not state how sub-document identifiers for certain document formats are to be defined, there is no agreed upon system to refer to IFC GUIDS, pixel zones etc. As this is a separate issue that is not considered part of this work, this prototype is currently limited to linking and displaying RDF graphs and imagery. In future versions, 3D model viewers, text editors, etc. could be implemented as well to support a broader variety of documents.



```
{
 "Site": "https://w3id.org/bot#Site",
 "hasBuilding": "https://w3id.org/bot#hasBuilding",
 "hasStorey": "https://w3id.org/bot#hasStorey",
 "hasSpace": "https://w3id.org/bot#hasSpace",
 "a": "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
}
```

```
{
  hasStorey
  hasStorey {
    hasSpace
  }
}
```

☑ Space1
□ Storey1
☑ Space2
□ Space3
☑ Storey2
□ Space4
□ Space5
□ Storey3

Figure 4: uploading and linking documents in the ‘Connect’ tab. Linked Data files may be queried with non-specialist languages such as GraphQL-LD or LDflex.

To simplify the querying of the Linked Data files, a test engine for GraphQL-LD has been implemented (Fig. 4). This means that an end-user does not need to know the details of Linked Data to query the available information. The query results can then be individually selected to link to other RDF resources, documents or sub-document identifiers. As Solid pods also follow a container-like structure, the implementation of functionality for generating an ICDD container should be quite straightforward.

4.5 Testing

Full testing of the tool's functionality has not happened at the moment of writing, although the different modules (project creation, topology, querying) have been tested separately and iteratively during the development process. It lies within the ambitions to simulate a more complete project when the above-mentioned modules are more tightly integrated with one another.

5. Conclusion and Future Work

5.1 Conclusion

In this paper, we suggested the use of Linked Data for semantically connected Common Data Environments. Because of the universality of RDF, such CDE could play a central role in a network of modular applications that each envisage a certain activity in the Building Life Cycle. The scope of such applications is therefore not limited to 'typical' construction activities, but could also include adjacent domains such as historical or geographical data, Facility Management, circular economy etc. Modular domain models, as proposed by the LBD Community Group, enable a more flexible approach to projects that work with disparate data. It was shown that the idea behind the Solid ecosystem is similar to the concept of BIM bots, but then in a Linked Data environment. A brief overview of the basic specifications offered by this ecosystem for social linked data was given, and the potential for managing Building Projects was illustrated. To finish, a prototype application 'ConSolid' was discussed, using the SDK that preconfigures the specifications for the application. This service is still in a very early development phase and thus highly experimental. Nevertheless, some basic functionality shows that a decentralised management of building information is achievable with the available technology stack.

5.2 Future Work

As this work is only an introduction, future work might focus on multiple aspects. One aspect is further develop the ConSolid application towards a more mature tool that can be used in practical situations. Apart from the basic implementation discussed in Section 4, vocabularies for describing product information and properties need to be implemented to be able to create realistic projects. This is a prerequisite for testing the overall capabilities of the use case in a later development stage. Another focus should lie on how multiple tools can access the building project data and communicate with one another. For example, establishing a link with (existing) modelling packages will make the creation of a basic Linked Building Data model more intuitive. Due to the decentralisation principle, other web applications that are authorised to fetch the project data may further enrich or analyse the project, e.g. by performing building performance simulations. When many tools are each dedicated to one small task, they could be used as atomic building blocks in a tool chain that addresses larger use cases. Semantic Web technologies such as SHACL allow validation of the information that is exchanged in such chain. Having a way to configure and connect multiple, modular tools should also allow building disciplines that lie beyond construction to collaborate and use the same data. However, such scenarios require a more thorough interlinking of information on data level: rather than being stored in files, the available information needs to be represented in RDF format as much as possible.

With this in mind, we hope this work can contribute to the advancements of the BIM community towards an open BIM practice that is characterised by integrated web services and interoperable data.

Acknowledgements

The authors would like to acknowledge the support by both the Special Research Fund (BOF) of Ghent University and the BIM4REN project, which has received funding from the European Union's Horizon2020 Research and Innovation Program under grant agreement No 820773.

References

Beetz, J., van Berlo, L., de Laat, R., & van den Helm, P. (2010). BIMserver. org–An open source IFC model server. *Proceedings of the 27th CIB W78 Conference on Information Technology in Construction*, 8.

Beetz, J., Van Leeuwen, J. P., & De Vries, B. (2005). An Ontology Web Language Notation of the Industry Foundation Classes. *Proceedings of the 22nd CIB W78 Conference on Information Technology in Construction*, *2006*, 670. Technical University of Dresden.

Berners-Lee, T., Dimitroyannis, D., Mallinckrodt, A. J., & McKay, S. (1994). World Wide Web. *Computers in Physics*, *8*(3), 298.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, *284*(5), 28.

Gürtler, M., Baumgärtel, K., & Scherer, R. J. (2015). Towards a workflow-driven multi-model BIM Collaboration Platform. *Working Conference on Virtual Enterprises*, 235. Springer.

Hollenbach, J., Presbrey, J., & Berners-Lee, T. (2009). Using RDF Metadata to enable Access Control on the Social Semantic Web. *Proceedings of the Workshop on Collaborative Construction, Management and Linking of Structured Knowledge (CK2009)*, *514*, 167.

Mansour, E., Sambra, A. V., Hawke, S., Zereba, M., Capadisli, S., Ghanem, A., … Berners-Lee, T. (2016). A Demonstration of the Solid Platform for Social Web Applications. *Proceedings of the 25th International Conference Companion on World Wide Web*, 223. International World Wide Web Conferences Steering Committee.

McKinsey & Company. (2015). *Imagining Construction's Digital Future*. Retrieved from https://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/imagining-constructions-digital-future

Oraskari, J., & Törmä, S. (2016). Access Control for Web of Building Data: Challenges and Directions. *EWork and EBusiness in Architecture, Engineering and Construction*, 45.

Pauwels, P., & Terkaj, W. (2016). EXPRESS to OWL for Construction Industry: Towards a recommendable and usable ifcOWL Ontology. *Automation in Construction*, *63*, 100.

Pauwels, P., Zhang, S., & Lee, Y.-C. (2017). Semantic Web Technologies in AEC Industry: A Literature Overview. *Automation in Construction*, *73*, 145.

Rasmussen, M., Pauwels, P., Lefrançois, M., Schneider, G. F., Hviid, C., & Karlshøj, J. (2017). Recent Changes in the Building Topology Ontology. *LDAC2017-5th Linked Data in Architecture and Construction Workshop*.

Schapke, S. E., Katranuschkov, P., & Scherer, R. J. (2010). Towards ontology-based management of distributed multi-model project spaces. *Proceedings of the 27th CIB W78 Conference on Information Technology in Construction*.

Taelman, R., Van Herwegen, J., Vander Sande, M., & Verborgh, R. (2018). Comunica: a modular SPARQL Query Engine for the Web. *International Semantic Web Conference*, 239. Springer.

Taelman, R., Vander Sande, M., & Verborgh, R. (2018). GraphQLLD: Linked Data Querying with GraphQL. *ISWC2018, the 17th International Semantic Web Conference*.

Törmä, S. (2013). Semantic Linking of Building Information Models. *2013 IEEE Seventh International Conference on Semantic Computing*, 412. IEEE.

van Berlo, L. (2015). BIM Service Interface Exchange (BIMSie). Retrieved from https://www.nibs.org/page/bsa_bimsie

Verborgh, R. (2018). Designing a Linked Data Developer Experience. Retrieved 25 April 2019, from https://ruben.verborgh.org/blog/2018/12/28/designing-a-linked-data-developer-experience/

Werbrouck, J., Pauwels, P., Senthilvel, M., & Beetz, J. (2019). *A checking approach for distributed building data*. Presented at the 31st Forum Bauinformatik, Berlin, Germany.