Towards Scene Graph Descriptions for Spatial Representations in the Built Environment

Oliver Schulz[1], Jeroen Werbrouck[2], and Jakob Beetz[1]
[1]Chair for Design Computation, Department of Architecture, RWTH Aachen University, Germany
[2]Department of Architecture and Urban Planning, Faculty of Engineering and Architecture, Ghent University
schulz@dc.rwth-aachen.de

Abstract. Although semantic linking for heterogeneous data about the building is already established in container-based environments - such as Common Data Environments, Digital Twins and Digital Archives - the spatial component has been neglected. In the field of computer graphics, however, the principle of the scene graph has already been established for the spatial relationship. Nevertheless, spatial linking can help establish a context between files and make it machine-readable to evaluate this context efficiently. This paper investigates to what extent scene graphs can be used to link heterogeneous files spatially. Based on current workflows and research, objectives for a potential file-level scene graph framework are provided. An example implementation for these objectives is shown in a Linked Data scenario.

1. Introduction

Common Data Environments (CDEs) are used to consolidate all information about a building, help digitise building-related data, and ensure availability for the stakeholders (Preidel et al., 2018). While much research is done in the direction of CDEs, the spatial interlinking of documents (e.g. pictures, plans and models) residing on the CDE is often not explicitly noted or not taken into account. Although current-day Building Information Modelling (BIM) processes – often used in combination with a CDE – come with a coordination process to ensure that the partial models are spatially superimposed (Tulke and Schumann, 2018), these processes can be error-prone since they rely on manual processes. Furthermore, for the scenario of existing buildings, the data is often heterogeneous and not coordinated at all, semantically or spatially. While people may be able to read the different representations of a building and spatially relate them to each other, a generic machine-readable way to connect these is still missing. An example use-case is given in (Schulz and Beetz, 2021), where pictures – taken on-site – are spatially superimposed, first with a digital plan of the building and in the second step with a BIM model. The process builds on an extension of buildingSMARTs BIM Collaboration Format (BCF).

BIMQL (Mazairac and Beetz, 2013) and BimSPARQL (Zhang et al., 2018) provide approaches to use a query language to query the different building elements of a model spatially. Still, they are targeted at the Industry Foundation Classes (IFC) - buildingSMART's standard exchange format for BIM processes and archiving of BIM models - and do not consider different building representations such as plans and images.

In Computer Graphics, scene graphs enjoy great popularity for providing a spatial hierarchy structure for virtual scenes, making them scalable, performant, and accessible for machines (openscenegraph, 2012). Therefore, these scene graphs are commonly used in modelling (and BIM) software, game engines, and scene descriptions of file formats. Relation to the structure of scene graphs can also be observed in the spatial structure of the IFC (Borrmann et al., 2018).

This paper focuses on analysing the spatial relationship between heterogeneous documents in digital asset management. We claim that adopting a file-level scene graph structure can establish a spatial interlinking of these documents, thereby enhancing the overall usability in container-based environments such as CDEs, Digital Twins and Digital archives. Even though spatial information can also be regarded as semantic too, we distinguish through the remainder of this paper between semantic and spatial linking. Furthermore, this work considers spatial as a concept that coordinates can describe. Spatial, as in topological descriptions (under, next to), is not the focus of this paper but can result as a consequence of this work.

The paper is structured as follows. Section 2 describes the current research around scene graphs, the AEC domain and CDEs and Container environments relating spatial connections between heterogeneous files. In Sections 3, we investigate the connection between these topics and define objectives for a file-level scene graph to interlink representations of buildings spatially. In Section 4, we map the objectives to a Linked Data example and provide a prototype as a proof of concept. Finally, we conclude the work in Section 5 and provide an outlook for further research.

2. Related Work

2.1. Scene Graphs in Computer Graphics

In computer graphics, scene graphs have enjoyed great popularity for a long time. They are especially useful for managing large scenes, i.e. collections of many geometries within a viewport. Scene graphs have the advantage of traversing objects and are performant and accessible for machines (openscenegraph, 2012). Over time, different scene graphs were invented, but their core principles are mostly the same. A scene graph forms a hierarchical tree by means of parent-child relationships. Each node can have exactly one parent but several children. Next to the parent-child relationship, each node comes with a transformation - a combination of location, rotation and scale - in a local coordinate system, while the uppermost node, the root node, defines the world coordinate system. When a parent node changes its transformation, all child nodes are transformed as well but maintain their relative offset, defined by their own transformation, from the parent. Furthermore, geometries - often in the form of triangulated meshes - can be added to the nodes. To handle the performance of large scenes, the geometries can also be provided with different Levels of Detail (LOD), which represent simplified variants of the mesh that are displayed or hidden depending on the distance to the camera. The creation process of LODs ranges from automated processes, where the creator has no influence on the result, to hand-built geometries. These LODs are not to be confused with the LODs often used in BIM. In addition, scene graphs usually have specialised nodes, such as lights, cameras, primitive geometries, and materials that colourise the meshes. (Reiners, 2002)

Scene graphs have a variety of uses, ranging from renderers (web and desktop) and game engines to managing scenes inside files (X3D, USD, glTF), 3D modelling applications and plain 3D viewers.

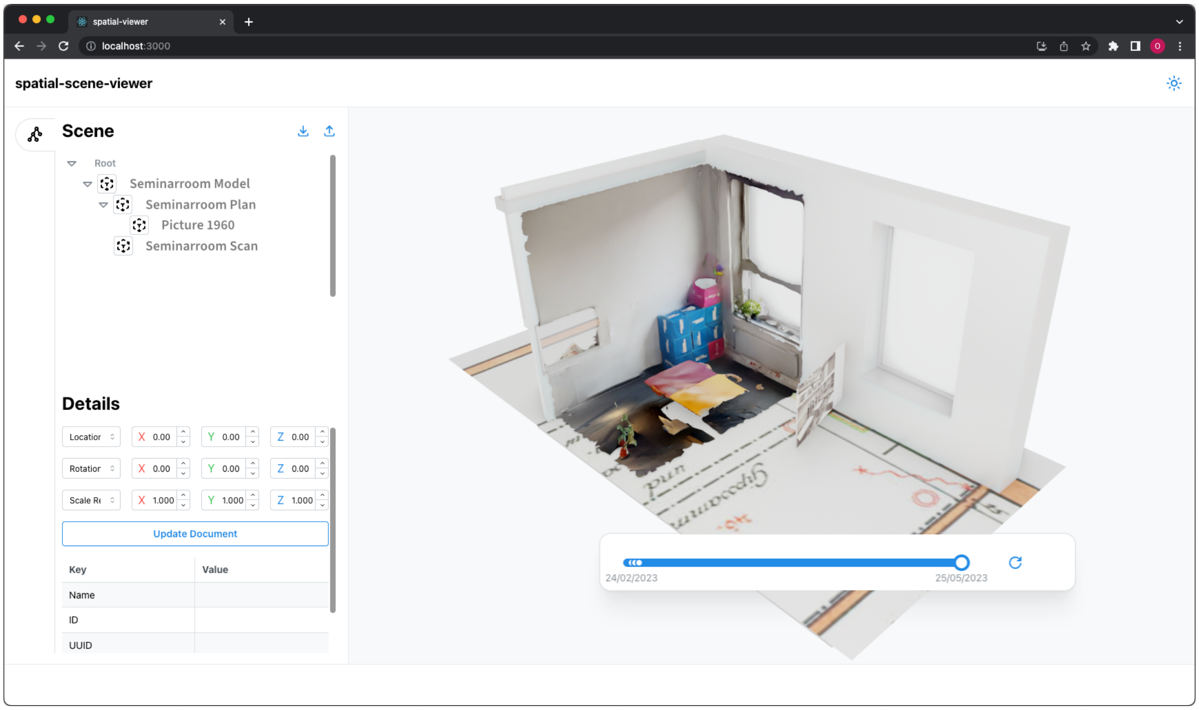

Figure 1: A stylised scene of the Seminar room at the Reiff Museum of the RWTH Aachen University depicting spatially aligned heterogeneous files consisting of a point cloud, an IFC model, a historical plan, and a picture.

2.2. AEC Domain

In the AEC domain, a variety of heterogeneous data is used, such as plans, images, point clouds and 3D and BIM models (Figure 1). All of these represent a specific space, which is either represented abstractly (plans and models) or exactly (point clouds and images). The representation always depends on the time of creation and can be outdated in the case of the exact representation. In the case of the abstracted version, there is the additional circumstance that these can also be versions that never got beyond the planning stage and are, therefore, never realised in the physical space. What all these spatial representations have in common, however, is that they represent the same physical space, but this is often not linked in a machine-readable way. A typical workflow in architectural offices is, for example, to load plans into the architecture application when renovating existing buildings and to use these as a basis for a (BIM) model. Although this provides a spatial link, it is often deleted from the model at a later point in time or not exported. As a result, the spatial link between some of the elements is established, but this is often lost in the course of the procedure. A possible reference to the plans could be exported within the IFC. A spatial composition in IFC is clearly defined and works by linking local coordinated systems. The principle is called “local placement” (Borrmann et al., 2018) in the IFC context and is similar to the principle of a scene graph in its structure. Furthermore, it is possible to reference external files via the “IfcExternalReference”.

In addition to the IFC structure, the Building Topology Ontology (BOT) (Rasmussen et al., 2019) describes the spatial composition of buildings in the context of Semantic Web-based building de-

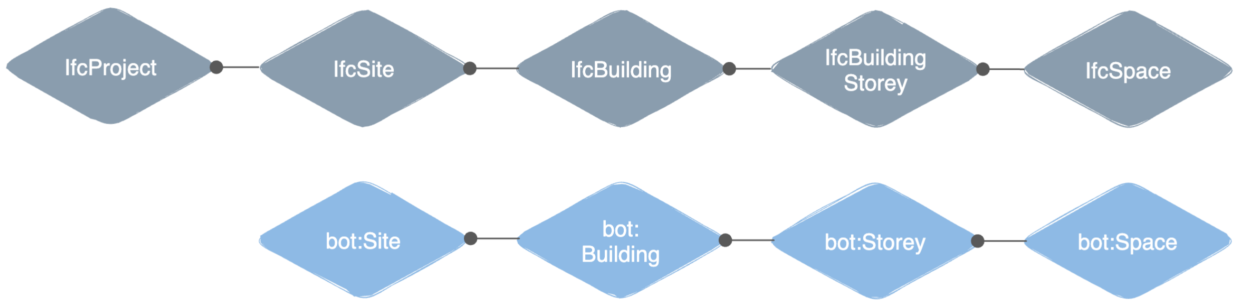

Figure 2: Comparison of the spatial composition of IFC and BOT. Both describe a basic topology of a building or a collection of buildings. The spatial hierarchy of both could be described as scene graphs.

scriptions. The building is divided into zones arranged hierarchically and extends from the site to the storey and the individual spaces. The spatial composition between IFC and BOT shows considerable similarities, and thus BOT is also similar in structure to the scene graphs (Figure 2).

2.3. Common Data Environments

Common Data Environments serve as a single access point in the AEC domain to project-related information. They are defined in ISO 19650 (ISO, 2018) and DIN SPEC 91391 (DIN, 2019). Furthermore, the DIN SPEC 91391 describes that all content is stored in information containers.

While much research has been done towards CDEs in general (Preidel et al. 2021, Oraskari et al. 2022) and the semantic linking of information containers and elements, the spatial component of CDEs tends to be neglected. Partially, this is due to the fact that for BIM methods, coordination processes are often agreed upon to ensure a spatial superposition of the different models from the beginning. But since individuals implement these processes, they are naturally prone to errors. Detecting and solving them later in an ongoing process can be very time-consuming. Moreover, these procedures and today's planning methods did not yet exist in many cases to plan and document existing buildings. Therefore, a computer-readable spatial connection of the different heterogeneous files is not given. Although 2D plans are regularly used as a basis for BIM designs for an initial model - especially in architecture offices - this connection often only exists within the proprietary software applications.

The BIM Collaboration Format (BCF), a feature included in many CDE solutions, is one exception in the spatial linking inside CDEs. It can be used to describe spatially located and component-based issues inside models and can therefore serve as a spatial link between them. The drawback of this spatial link via BCF is that it is susceptible to non-coordinated models. When a model has not undergone the coordination process, the issues will be located at the wrong position in the coordinate system of this file. We identified this issue before and proposed a potential solution in (Schulz and Beetz, 2021) and (Schulz, 2021). Nevertheless, BCF is not intended to be the spatial anchor for all CDE-related files.

2.4. Containers for heterogeneous AEC documents

Grouping resources that share one or more common traits is relevant in the AEC industry and beyond. Within AEC standards, ISO 19650 defines information containers as a "*named persistent set of information, retrievable from within a file, system or application storage hierarchy*" (ISO, 2018). A term commonly used in the AEC industry to denote such collections of heterogeneous project information is multi-models (Scherer and Schapke 2011, Gürtler et al. 2015).

ISO 21597's Information Container for Linked Document Delivery (ICDD) (ISO, 2020) is an industry standard that realises this vision. It was invented to handle the handover of collections of files, using a compressed ZIP folder (.icdd) as a dump of all heterogeneous project datasets. Apart from semantically describing its content in a metadata document (`index.rdf`), ICDD specifies an approach to link sub-document identifiers between heterogeneous datasets (e.g. URIs, GUIDs, image zones etc.). These sub-document links are recorded in the `links.rdf` document. Nevertheless, it is mainly used for semantically linking these files and not spatially.

In an interdisciplinary (Web) context, the Data Catalog (DCAT) vocabulary (Albertoni et al., 2023) focuses on providing semantic interoperability between data catalogues published on the Web, similar to the above-mentioned notion of containers. Using URLs, DCAT provides a scalable hierarchy to describe and cross-reference catalogues, metadata records (datasets) of specific files and distinct representations (distributions) of those datasets, which may differ in, for example, file format, temporal or spatial resolution, language. The spatial resolution of a DCAT distribution is mainly oriented towards the tiling of 2D Geographic Information Systems (GIS) applications and less towards the 3D spatial linking of datasets which is the topic of this paper. DCAT allows for catalogues to nest other catalogues apart from datasets, so it is possible to create a hierarchical tree of sub-containers and documents.

3. Scene Graphs for the AEC Domain

While some file formats can reference external files, scene graphs mostly focus on file- and application-internal scene compositions. Next to describing the different transformations of nodes inside these files, they can describe the geometry, lights, cameras etc. Especially in file size, geometry descriptions are one of the largest elements within scene graphs. They mostly consist of lists of vertices and normals or describe different operations such as extrusions and intersections. Moreover, features like lights, materials, and geometries have no added value for describing the spatial composition of different files and not their internal geometries. Therefore, a potential solution using scene graphs can be reduced to core concepts focusing on spatial composition.

In the AEC domain, this missing functionality of spatial alignment is usually addressed by a coordination process to ensure that the origins of the different files align. However, no information is provided to describe the alignment. The stakeholders need to rely on the spatial validity, and issues in the spatial superimposition may be found at a later point in time when it is a time-consuming process to fix them. In addition, it is possible to reference other files in some applications and file formats, such as IFC. However, since the files are not always accessible in different applications and the applications usually only have these functions internally and do not provide access via an API or export, this is not a viable approach for CDEs and container environments. Therefore, the information on how files are spatially connected to each other needs to be accessible on a meta file-level that is machine-readable rather than in-file.

As described in Section 2.1, scene graphs come with extra functionality, not regarding the spatial composition. In fact, when looking at file sizes, the major part of a file describing a scene is occupied with geometry. Nevertheless, some of these features are not necessary when spatially aligning files since information such as geometry, materials and lights are already included in the referenced files. Therefore, file-level scene graphs should be rather lightweight compared to the structure of common scene graphs. Based on the scene graph structure and the requirements of the AEC industry, we considered the following core objectives for file-level scene graphs:

Objective 1 Spatial Composition. Nodes act as *spatial actors* that are part of the parent-child tree and form the spatial composition of a scene. Spatial actors can have multiple children. Documents can refer to these spatial actors in order to be spatially located in the tree hierarchy.

Objective 2 Transformations. The *transformations* act as the spatial points in a Cartesian coordinate system and come with a location, rotation, and scale, either described as vectors or matrices. They are responsible for providing spatial information to individual spatial actors and are intended to enable spatial querying.

Furthermore, the following secondary objectives were identified:

Objective 3 LODs. For file-level scene graphs, *LODs* should be able to be used not only for different degrees of geometry complexity but also to describe the overall extent of a file with a bounding box. Such a bounding box could, for example, be used as a LOD that enables advanced spatial queries. For example, to assess whether an image is located inside a model.

Objective 4: Regions. Since a file can show several spatial representations - for example, a PDF page can have several plans - it is necessary to define regions within the files, which can then be put into a spatial context. Thus it is possible that a file appears several times in a spatial hierarchy.

Objective 5 History of Transformations. In order to keep track of changes in the spatial composition, the scene graph should provide a history of changes made to the transformations. While this Objective is not necessary for all processes (e.g. for archiving purposes), it is an added value in processes that undergo active changes, where it is important to follow up on who changed what and when.

4. Using Scene Graphs in Linked Data

Even though the objectives of this paper have been introduced as technology-agnostic, we illustrate the validity of the approach with a Linked Data-based (LD) proof of concept. This choice is motivated by the domain-independent, graph- and Web-based nature of LD technologies (Bizer et al. 2009), alignment with ongoing research on their usage for the AECO industry, and their identification as one of the few technologies that support the full set of FAIR (Findable, Accessible, Interoperable, Reusable) principles (Mons et al., 2017).

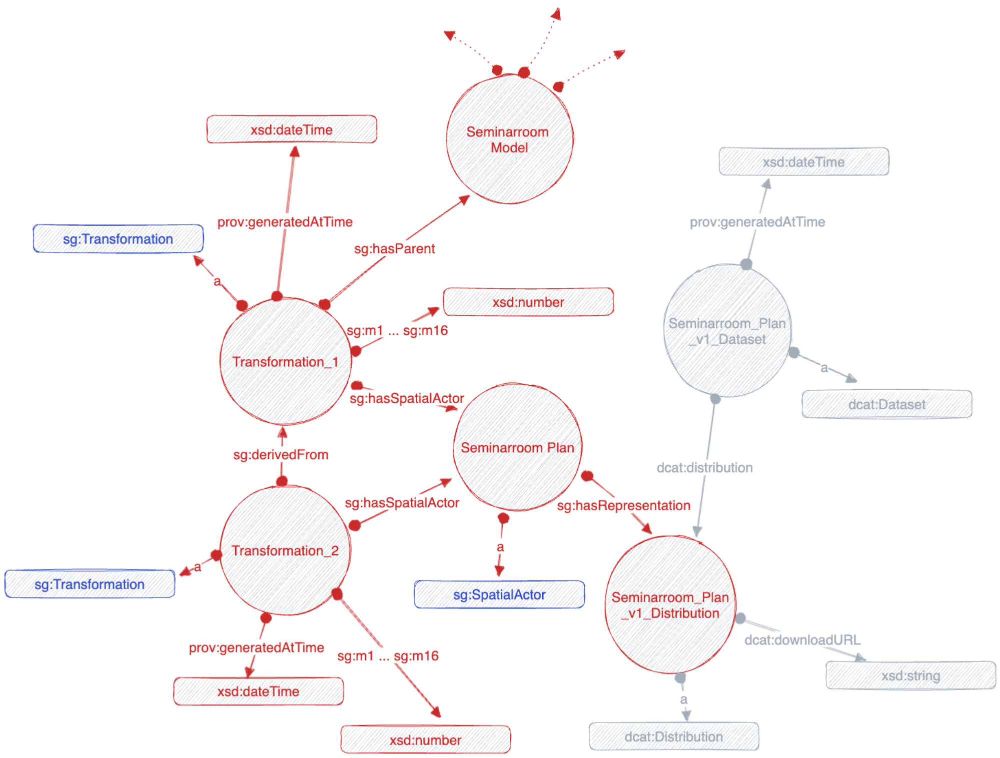

Figure 3: Example graph of the implementation of a scene graph in Linked Data.

The scene graph concept in our examples is described with the prefix *sg:*, but it does not represent an elaborated ontology and, thus, has not been published. The prefixes used throughout this section are mentioned in Listing 1.

Listing 1: Prefixes used throughout this paper.

```
PREFIX sg:      <http://example.org/scenegraph#>
PREFIX rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd:     <http://www.w3.org/2001/XMLSchema#>
PREFIX prov:    <http://www.w3.org/ns/prov#>
PREFIX dcat:    <http://www.w3.org/ns/dcat#>
```

The LD-based proof of concept covers the core Objectives 1 and 2 and the secondary Objective 5. Objectives 3 and 4 will be addressed in future work. The proof of concept is threefold and focuses on *dcat:Distribution* to describe the heterogeneous files, *sg:Transformation* to point at a spatial point (Objective 2) at a specific time (Objective 5) and *sg:SpatialActor* as a link to the file and, therefore, as a gateway between the transformations and the file distributions. The transformations are linked to the spatial actors via *sg:hasSpatialActor*. Figure 3 represents a sample of the graph that could describe the scene in Figure 1.

The tree structure (Objective 1) derives from linking an *sg:Transformation* to an *sg:SpatialActor* with the predicate *sg:hasParent*. It was refrained from linking a *sg:SpatialActor* directly via a parent-child relationship to another *sg:SpatialActor* since this could cause confusion in case of an edit with a new parent-child relationship. By the *sg:Transformation* linking to its parent node, new and subsequent transformations concerning the same spatial actor can be provided with a new and independent sg:hasParent predicate. This ensures the flexibility to change the Scene Graph structure afterwards. The new transformations are then linked to their predecessor transformations via *sg:derivedFrom*, ensuring easier query ability. An example SPARQL query (W3C SPARQL Working Group 2013) resulting in the newest transformation for a sg:SpatialActor is provided in Listing 2.

The transformations themselves are described with a 4x4 matrix - in Figure 3 represented by *sg:m1...sg:m16* - providing information for location, rotation and scale and a *prov:generatedAtTime* value. In order to define the final position of a transformation in a Cartesian coordinate system, the 4x4 matrices must be multiplied in the order of the parent-child relationships, beginning with the utmost parent.

Listing 2: Query for getting the newest *sg:Transformation* for the *ex:SeminarroomPlan*

```
SELECT DISTINCT ?transform
WHERE
{
    ?transform a sg:Transformation ;
                 sg:hasSpatialActor ex:SeminarroomPlan ;
                 prov:generatedAtTime ?created .
}
ORDER BY DESC(?created)
LIMIT 1
```

A prototype implementing these basic features is published on GitHub[1] and depicted in Figure 1. It provides a simple viewer to view the spatial documents and their alignments over different time periods. Queries used by the viewer and sample data are also provided in the repository.

5. Conclusion and Future Work

This work has investigated spatial contextualisation at the file level and identified the need for the AEC domain, as this aspect - unlike semantic linking - has been rather neglected so far. A potential approach was identified as a generic scene graph structure providing a spatial hierarchy at the file level. Core- and secondary Objectives for a lightweight scene graph were established and mapped to a Linked Data syntax. The first tests for this approach were conducted in a prototype published on GitHub. The research is intended to fill the missing spatial component in research and practice between containerised approaches such as CDEs, Digital Twins and Digital Archives.

In future research, we will address the spatial query ability of the scene graph structure. While the current approach with simple transformation matrices allows querying for single points and vicinity to points, it is not including any queries for, e.g. locations inside specific spaces. A potential approach could be to use the LODs of scene graphs to describe the extent of a file and query for points inside these extents. In addition, Objective 4, the implementation of regions, and Objective 5, the use of LODs, have not yet been incorporated into the prototype to date. Further research is required to determine how regions can be partitioned within files and incorporated into

[1] Spatial Viewer Github: https://github.com/Design-Computation-RWTH/spatial-viewer-eg-ice-2023 (Accessed 22.05.2023)

the scene graph. Furthermore, it will be investigated to what extent new versions of files can be included in the scene graphs. Although the differences between updates from version to version are mostly manageable, these can cause changes to the extent, which in the worst case, could render the current transformation of the scene graph invalid. Finally, it shall be examined to what extent the scene graphs can be incorporated in a larger spatial context, e.g., in connection with GIS.

Acknowledgement

This research is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project number 501812634 and the Strategic Basic Research grant (grant no. 1S99020N) of the Research Foundation Flanders (FWO).

References

Albertoni, R., Browning, D., Cox, S., Gonzales Beltran, A., Perego, A. and Winstanley, P. 2023, Data Catalog Vocabulary DCAT - Version 3, W3C Recommondation, W3C. https://www.w3.org/TR/vocab-dcat-3.

Bizer, C., Heath, T. and Berners-Lee, T. 2009, 'Linked Data: The Story so Far', *International Journal on Semantic Web and Information Systems* **5**, 1–22. http://dx.doi.org/10.4018/jswis.2009081901.

Borrmann, A., Beetz, J., Koch, C., Liebich, T. and Muhic, S. 2018, Industry Foundation Classes: A Standardized Data Model for the Vendor-Neutral Exchange of Digital Building Models, *in* A. Borrmann, M. König, C. Koch and J. Beetz, eds, 'Building Information Modeling', Springer International Publishing, Cham, pp. 81–126. http://link.springer.com/10.1007/978-3-319-92862-3_5.

DIN 2019, 'DIN SPEC 91391-1:2019-04 Common Data Environments (CDE) for BIM projects - Function sets and open data exchange between platforms of different vendors - Part 1: Components and function sets of a CDE; with digital attachment', https://dx.doi.org/10.31030/3044838.

Gürtler, M., Baumgärtel, K. and Scherer, R. J. 2015, Towards a Workflow-Driven Multi-model BIM Collaboration Platform, *in* L. M. Camarinha-Matos, F. Bénaben and W. Picard, eds, 'Risks and Resilience of Collaborative Networks', IFIP Advances in Information and Communication Technology, Springer International Publishing, Cham, pp. 235–242. http://dx.doi.org/10.1007/978-3-319-24141-8_21.

ISO 2018, 'ISO 19650-1:2018 Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) - Information management using building information modelling - Part 1: Concepts and principles', https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/80/68078.html.

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

Mazairac, W. and Beetz, J. 2013, 'BIMQL – An open query language for building information models', *Advanced Engineering Informatics* **27**(4), 444–456. https://www.sciencedirect.com/science/article/pii/S1474034613000657.

Mons, B., Neylon, C., Velterop, J., Dumontier, M., Da Silva Santos, L. O. B. and Wilkinson, M. D. 2017, 'Cloudy, increasingly FAIR; revisiting the FAIR Data guiding principles for the European Open Science Cloud', *ISU* **37**(1), 49–56. https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/ISU-170824.

openscenegraph 2012, ‘What is a Scene Graph?’, https://www.openscenegraph.com/index.php/documentation/knowledge-base/36-what-is-a-scene-graph.

Oraskari, J., Schulz, O., Werbrouck, J. and Beetz, J. 2022, 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, pp. 92–101. https://ebooks.au.dk/aul/catalog/view/455/312/1848-2.

Preidel, C., Borrmann, A., Exner, H. and König, M. 2021, Common Data Environment, *in* A. Borrmann, M. König, C. Koch and J. Beetz, eds, ‘Building Information Modeling: Technologische Grundlagen und industrielle Praxis’, VDI-Buch, Springer Fachmedien, Wiesbaden, pp. 335–351. https://doi.org/10.1007/978-3-658-33361-4_16.

Preidel, C., Borrmann, A., Mattern, H., König, M. and Schapke, S.-E. 2018, Common Data Environment, *in* A. Borrmann, M. König, C. Koch and J. Beetz, eds, ‘Building Information Modeling: Technology Foundations and Industry Practice’, Springer International Publishing, Cham, pp. 279–291. https://doi.org/10.1007/978-3-319-92862-3_15.

Rasmussen, M. H., Lefrançois, M., Schneider, G. F. and Pauwels, P. 2019, ‘BOT: The building topology ontology of the W3C linked building data group’, *Semantic Web* **12**(1), 143–161. https://research.tue.nl/en/publications/bot-the-building-topology-ontology-of-the-w3c-linked-building-dat.

Reiners, D. 2002, Scene Graph Rendering. https://www.lsi.usp.br/~mkzuffo/PSI5787/opensg/scenegraph-rendering.ieeevr2002.pdf.

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), 582–599. https://www.sciencedirect.com/science/article/pii/S1474034611000644.

Schulz, O. 2021, Spatialization of damage to buildings using a BIM collaboration Format approach, *in* ‘32. Forum Bauinformatik 2021’, Darmstadt, Germany, pp. 341–348. https://doi.org/10.26083/tuprints-00021521.

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

Tulke, J. and Schumann, R. 2018, BIM-Manager, *in* A. Borrmann, M. König, C. Koch and J. Beetz, eds, ‘Building Information Modeling: Technology Foundations and Industry Practice’, Springer International Publishing, Cham, pp. 293–302. http://link.springer.com/10.1007/978-3-319-92862-3.

W3C SPARQL Working Group 2013, SPARQL 1.1 Overview, Technical report, World Wide Web Consortium (W3C). https://www.w3.org/TR/sparql11-overview/.

Zhang, C., Beetz, J. and de Vries, B. 2018, ‘BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data’, *Semantic Web* **9**(6), 829–855. http://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/SW-180297.