Törmä, S. and Oraskari, J. and Vu Hoang, N. (2012). Distributed transactional building information management, Proceedings of the First international Workshop on Linked Data in Architecture and Construction (LDAC 2012), Ghent University.

Distributed Transactional Building Information Management

Seppo Törmä, Jyrki Oraskari and **Nam Vu Hoang**
Department of Computer Science and Engineering
Aalto University, School of Science
seppo.torma@aalto.fi, jyrki.oraskari@aalto.fi, nam.vuhoang@aalto.fi

Abstract

Impressive results have been gained in the field of building information modeling – most notably the standardized schema (IFC) and advanced discipline-specific design tools – but the *management* of building information is still in its infancy. Building information is produced in a distributed and parallel fashion by different discipline experts throughout a building project. These partial models are *interrelated* as they describe the same building, although from different perspectives. How can the interdependencies be represented and the contents of the partial models be exchanged, combined, and modified in a systematic manner? We propose an approach based on *Linked Data*: the partial models are converted into Linked Data datasets (utilizing URIs, RDF, and OWL) and published on the Web (allowing browsing and querying with SPARQL). We describe an ongoing research project to determine the feasibility of this approach. The work is based on a IFC-to-RDF converter, and use of RDF store to maintain the resulting dataset. The research problems – link-type modeling, link generation, change discovery, change management, and information scope management – are described.

Introduction

In every significant building project a huge volume of information both about the building and the construction process is produced in a distributed and loosely coupled manner. Different parties produce *partial models* that describe the building from different perspectives, including requirements, architecture, structural design, mechanical systems, fabrication, and construction process.

As the partial models are representations of the same physical end result, they are obviously *interrelated*. Inconsistencies between the partial models will result in potentially costly problems in the construction phase: spatial collisions, omissions, and other incompatible design decisions. The problems can inflict the schedule of the project and have long-lasting effects on the eventual quality of the building.

Building information modeling (BIM) aims to make the partial models more coordinated. An important part of work in the field has centered on the definition of a common data model for building data. The result is a widely used standard IFC (Industry Foundation Classes) currently supported by major BIM authoring tools. Although BIM tools still maintain the models in their own native formats, they have a capability to export their models as IFC files. The tools can thus map their native entities into a common data model. However, this does not meant that the entities in different models — e.g., the wall in an architectural model and a precast element in a structural model — would have any direct relations in that model. When an entity needs to be modified, the information about its links could aid the management of the change across the models. It could be used to provide the modifier an awareness of a dependency, or to generate notifications or change requests to other parties.

The current way to share building information models is typically based on simple exchange of IFC files. While this approach is compatible with the distributed and loosely-coupled nature of building projects, it does not provide a framework for managing the interlinking of models. The relations of models can be interpreted only by human designers, which limits the support that BIM systems could give to management of cross-model dependencies. Model data could be shared also through other mechanisms — centralized repositories, distributed event-based systems, cloud, federated databases, and so on — but none of these directly addresses the problem of cross-model relations either.

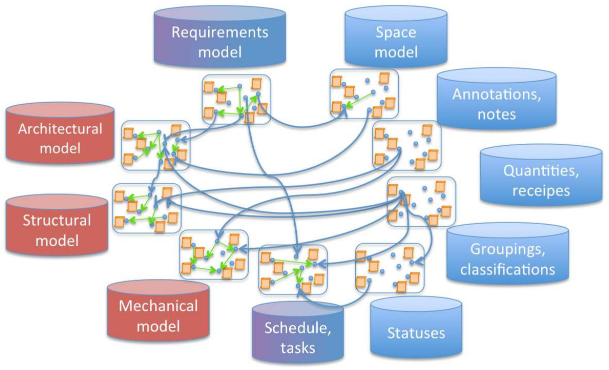

Figure 1: Linked Building Data.

The DRUM project (Distributed Transactional Building Information Management) studies the mechanisms for distributed information management in building information. The focus is on the interrelations of partial models, and the support for cross-model change management.

The starting point of DRUM is to respect the distributed and discipline-centered way of information production and maintenance. That is, the exported IFC models are considered to be shared in a read-only mode and *all changes are always made to native models*, and then exported again to IFC. This is most natural way of working since the maintenance of a model requires discipline-specific expertize and tools. Moreover, changes made to an exported IFC model are difficult to import back to native models, because there is *no proper conversion roundtrip* between native and IFC models. A conversion would loose information and restructure the model, making it very tedious to parse it back into a valid native model.

DRUM studies the problem of model interrelations based on *Linked Data*, a technology that directly addresses the relations of co-existing datasets (Berners-Lee 2006; Bizer, Heath, and Berners-Lee 2009; Heath and Bizer 2011). In addition, by providing a Web-oriented and open linking framework, Linked Data can support the connection of external information sources (annotations, documents, project management tools, etc) to IFC-based models (Fig. 1).

Linked Building Data

Linked Data is based on the Semantic Web technologies designed for the *representation, publication, and browsing of structural data on the Web*. In this approach identifiers are represented with URIs (Uniform Resources Identifiers), data with RDF (Resource Description Framework), schema with OWL (Web Ontology Language), and queries with SPARQL (SPARQL Protocol and RDF Query Language).

The application of the Linked Data in BIM means that partial models are published by their producers in the Web as RDF datasets. The GUID-identified entities in IFC models are identified through URIs. The properties and relations of entities are represented as RDF statements. The conversion to RDF can be only partial: for instance, it can be useless to convert the geometrical information in a model — usually its largest part — since for efficient geometrical manipulation it needs to be converted to a more suitable geometrical representation, which can be done directly from IFC.

In Fig. 1 there are a set of partial models at the outer skirt, covering both models based on IFC (on the left) and external information sources not represented in IFC (on the right). The important objects from all these models are exposed in Web by giving them a URI. In addition, the essential internal structures of models are exported as RDF statements and essential data is provided as literals in RDF. The linking between these models is provided by linksets in RDF.

The conversion of IFC files into RDF datasets is based on IFC ontology which is produced by converting the IFC schema into OWL (Beetz 2009). Both of these conversiosn are relatively straightforward as the OWL and RDF are mostly more flexible and expressive than EXPRESS (the IFC schema language) and STEP Part21 format (the IFC data representation format). There are some minor issues with the conversion — e.g., the procedural constraints in EXPRESS or global property constraints in RDFS — but they have little practical impact.

A publicly accessible converter for IFC data is available at the Ghent University (Pauwels, De Meyer, and Van Campenhout 2011), and an efficient IFC-to-RDF converter has also been implemented in the DRUM project.

While the partial models need to be converted into RDF datasets, the links can be provided directly as linksets in RDF format, since there are really no other existing representations for them. There can potentially be many independently created linksets between two models. The question of how to make the users of the datasets aware of these linksets. This belong to the area of information scope management discussed more below.

Once an IFC model is converted into an RDF dataset, it can be stored in an RDF store. The contents of the model can be queried with SPARQL queries or browsed with a Web brower.

Research problems

While Linked Data can provide a framework in which to represent the interrelations of models, there are many open questions about how to apply the available representations and technologies. At least the following research problems can be identified:

Link type modeling *What kinds of relations there can be between models?* The first thing to observe is that there can be links at two different levels: between the *models* on the whole and between *entities* within the models. The model-level relations are often defined in advance, already when the models to be created in a project are specified. Once the model-level links are known, they provide the basis for looking for specific types of entity-level links across the models. The model-level links can be classified into:

- *Sequential relations*: A model *is based on* another, such as structural design is based on architectual model.
- *Parallel relations*: Two models *compete* of a shared resource (like common space) or *complement* each other (like designs of two wings of a building).

The types of entity-level links are different between different kinds of entities: between a requirements and a building object, two building objects, or a building objects and an activity. Definition of relevant link types also depends on the cross-model functionalities to be supported. In DRUM a working categorization of links has been developed. There is a need for thourough experimentation and refinement to come up with an *ontology of BIM links*.

Link generation *How to generate the actual links at the entity level?* This is broad problem that will quite probably require a combination of multiple automatic, manual, and semi-automatic methods. Obviously the role of manual methods has to be limited: it is an error prone task and an additional burden to designers.

There are some clear categories of links that can be identified automatically, most notably those based on geometric information of building objects. These kinds of links can potentially be generated by any tool that can import two or more models at the same time, check spatial clashes, and provide detailed information about clashes. We are currently studying the application of Solibri Model Checker and Tekla Structures to this problem.

There are link discovery frameworks for detecting which nodes in two RDF datasets represent same real-world entities. Examples are Silk (Volz et al. 2009) and LIMES (Ngomo and Auer 2011).

Change discovery *How to discover changes between different versions of a same model?* RDF is based on graph data model which means that two graphs need to be compared to identify added, removed, and modified nodes. The task is complicated by two problems: nodes with *no identity*, and with *changing identity*.

The anonymous nodes - referred in RDF as blank nodes - cannot be directly mapped to corresponding nodes across model versions as they only have an identity within one version. In RDF graphs converted from IFC files, approximately 80-95% of nodes are blanks. This complicates the application of existing methods for RDF change detection such as RDFSync (Tummarello et al. 2007). RDFSync is based on the computation of *Minimal Self-contained Graphs (MSGs)* of the model, and on the efficient detection of changes using checksums for MSGs. Unfortunately, the large ratio of blanks results in very large MSGs that connect most of the nodes in the model. An area of active research in DRUM is to come up with efficient strategies to give at least *partially stable identities* for blank nodes.

Changing identities have different reasons. Firstly, not all BIM tools can maintain the identities of all entities across model versions, since some exported entities have no corresponding objects in their native models. Secondly, it is typical practice for a designer who wants to change an existing entity in a model to first delete the old entity and then design a new entity in its place. This is common when more detail is added into existing entities, such as reinforcements to concrete. For economy of work, a detailed design can then be copied and pasted into the place of many other similar entities. However, maintaining the identities of nodes in essential to prevent the cross-model links to break unnecessarily. Methods to overcome this problem has been studied in DSNotify system (Popitsch and Haslhofer 2010), which attempts to match recently removed entities with newly created ones.

Change management *How to coordinate changes that affect multiple models?* Assuming that IFC models - such as an architectural model and a structural model - has been converted into RDF datasets and that cross-model links has been defined between entities that represent same real-world entities, there are many levels at which change management can be supported:

1. *Preventive*: Awareness of the connection can be provided to prevent superfluous changes.
2. *Reactive*: Change notifications to the affected parties can be automatically generated.
3. *Proactive*: Change requests to potentially affected parties can be created when a change is needed.
4. *Transactional*: Using distributed versioning the parties can commit their tentative model versions when designs have reach a consistent state.

When the architect is planning a change in her model - for instance, the position of a wall - the tool can provide her with an awareness of the structures that have been designed inside the wall. Once she makes a change, a notificaiton to dependent models can be sent. Or if she desires, she can send a change request while doing her own changes. Finally, she and the other designers can work on the change collaboratively. They can create new tentative versions of their models to create new designs and once everyone has finished the work, commit the changes together.

Information scope management *How to represent and manage information about what parties, datasets, and linksets belong to a project? How to implement that in a distributed manner?* At the model-level this is a question of model metadata that can be represented using — and perhaps extending — dataset description ontologies such as VoID (Alexander and Hausenblas 2009). At the entity-level the problem can be addressed with protocols that support registration of links between models. Each link is a new linkset can registered with the datasets whose entities it refers to. When the dataset is afterwards accessed, the registered links will be available for traversal.

Some of the above-mentioned problems — especially the problem of changing identities — should eventually be solved in BIM tools but before that happens, their effects are encountered in the information management solutions. If the change in the identity of an entity cannot be detected, the links associated with the entity will be broken. A large number of broken links can severely decrease the usefulness of linksets. In addition, the change cannot be discovered in a proper manner if identities cannot be equated. For now, to study Linked Building Models, these problems need to be solved outside BIM tools.

Summary

DRUM studies the application of Linked Data technologies to management of building information. Linked Data provides a framework that focuses on the central question in exchange, combination, and modification of co-evolving partial models of a building: the interlinking of the models. However, the previous applications of Linked Data are quite different from building information management, and many questions of how to use the technologies need to be solved.

Linked Building Data is an emerging area of research. Previously the IFC data model has been translated to an OWL ontology. There are translators that can convert IFC models into RDF datasets, one implemented in DRUM project. There are initial working categorization of BIM link types but a more principled ontology needs to be worked out. The development of efficient methods for change discovery, models versioning, and link generation are currently under work. Information scope management and collaborative change management methods are future research topics in DRUM.

References

Alexander, K., and Hausenblas, M. 2009. Describing linked datasets-on the design and usage of void, the'vocabulary of interlinked datasets. In *In Linked Data on the Web Workshop (LDOW 09), in conjunction with 18th International World Wide Web Conference (WWW 09*. Citeseer.

Beetz, J. 2009. *Facilitating distributed collaboration in the AEC/FM sector using Semantic Web Technologies*. Ph.D. Dissertation, PhD Thesis, Eindhoven University of Technology. ISBN 978-90-6814-618-9.

Berners-Lee, T. 2006. Linked data-design issues. *W3C* 2009(09/20).

Bizer, C.; Heath, T.; and Berners-Lee, T. 2009. Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)* 5(3):1–22.

Heath, T., and Bizer, C. 2011. Linked data: Evolving the web into a global data space. *Synthesis Lectures on the Semantic Web: Theory and Technology* 1(1):1–136.

Ngomo, A., and Auer, S. 2011. Limes-a time-efficient approach for large-scale link discovery on the web of data. In *Proceedings of IJCAI*.

Pauwels, P.; De Meyer, R.; and Van Campenhout, J. 2011. Interoperability for the design and construction industry through semantic web technology. *Semantic Multimedia* 143–158.

Popitsch, N., and Haslhofer, B. 2010. Dsnotify: handling broken links in the web of data. In *Proceedings of the 19th international conference on World wide web*, 761–770. ACM.

Tummarello, G.; Morbidoni, C.; Bachmann-Gmür, R.; and Erling, O. 2007. Rdfsync: efficient remote synchronization of rdf models. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, 537–551. Springer-Verlag.

Volz, J.; Bizer, C.; Gaedke, M.; and Kobilarov, G. 2009. Silk–a link discovery framework for the web of data. In *Proceedings of the 2nd Linked Data on the Web Workshop*. Citeseer.