
BIMSERVER.ORG – AN OPEN SOURCE IFC MODEL SERVER

Jakob Beetz, Assistant Professor, j.beetz@tue.nl
Eindhoven University of Technology, Eindhoven, The Netherlands

Léon van Berlo, Researcher, leon.vanberlo@tno.nl

Ruben de Laat, Researcher, ruben.delaat@tno.nl

Pim van den Helm, Researcher, pim.vandenhelm@tno.nl
TNO Built Environment and Geosciences, Delft, The Netherlands

ABSTRACT

In this paper the Open Source IFC model server project bimserver.org is introduced. A brief overview of existing IFC toolkits and model servers both as commercial applications as well as research prototypes is given. An implementation-independent meta-model from current IFC STEP EXPRESS schemas is generated and used as a venture point for a database persistency framework. A model server application is introduced that is based on this framework which enables the storage, maintenance and query of IFC based building information models for the collaboration among different stakeholders.

Keywords: IFC, model server, BIM, CAD, collaboration

1. INTRODUCTION

Currently, only very few implementations of model servers exist that enable stakeholders in AEC/FM projects to collaborate on central repositories of building information models (BIM). Most of the few existing commercial implementations are built on the basis of general purpose STEP or PLM/PLCS technologies. Up to now, both the underlying business models as well as the operation of the servers themselves are tailored to the needs of large corporations from other industry domains such as shipbuilding, automotive, and petrochemical. Design and engineering processes as well as the resulting end-products in these domains differ significantly from the building industry. Here, one-of-a-kind artifacts are designed in a collaborative effort by nonrecurring, temporary and multi-disciplinary teams that are mainly composed of members of small and medium enterprises (SMEs). Furthermore, very few of these stakeholders have sufficient data modeling knowledge to the degree needed for an effective use of present day model server products. In order to achieve the long-standing goal of shared product data models throughout the lifecycle of a building, not only open technical, methodological and standardization issues to be solved. The availability of affordable and end-user friendly model server technology is crucial for the advancement of the industry as a whole.

Open, accessible frameworks that focused on the demands of the building industry that have been proposed in the past, such as the IMSvr (IMSvr 2003) and SABLE (Houbaux 2003) projects, which have been reviewed in (Kiviniemi et al 2005) and (Howard & Björk 2007), unfortunately have been discontinued and only a small amount of dedicated research has been done to address the specific problems of shared building information models for collaboration purposes (Weise 2006). However, among the promising open initiatives (Nour et al 2006, OpenIFCtools 2009) that have been started also in more recent years, no common platform for a shared research and development effort has gained enough traction yet to drive a collaborative development of an open source platform forward.

By creating a free software platform (licensed under the GPL v 3) and model server implementation presented in this paper, end-users from the building industry are encouraged to use shared building information models that

have been advocated for such a long time now. As a second goal, the results *-presented here offer a considerably valuable development environment for both fellow academic researchers and commercial software developers.

In the following sections a brief overview of the internal architecture of the platform as a whole and describe the use of an implemented browser based interface from an end user perspective is given. A summary and indications of future work conclude this paper.

2. FRAMEWORK ARCHITECTURE

The bimservers.org development framework has a multi-layered design with a generative, model-driven architecture based on the existing IFC STEP Express models that are transformed into more widely accepted data modeling languages and frameworks to simplify and streamline development with these models. A generic persistency layer using a key-value database approach allows versioning of data models on an object level.

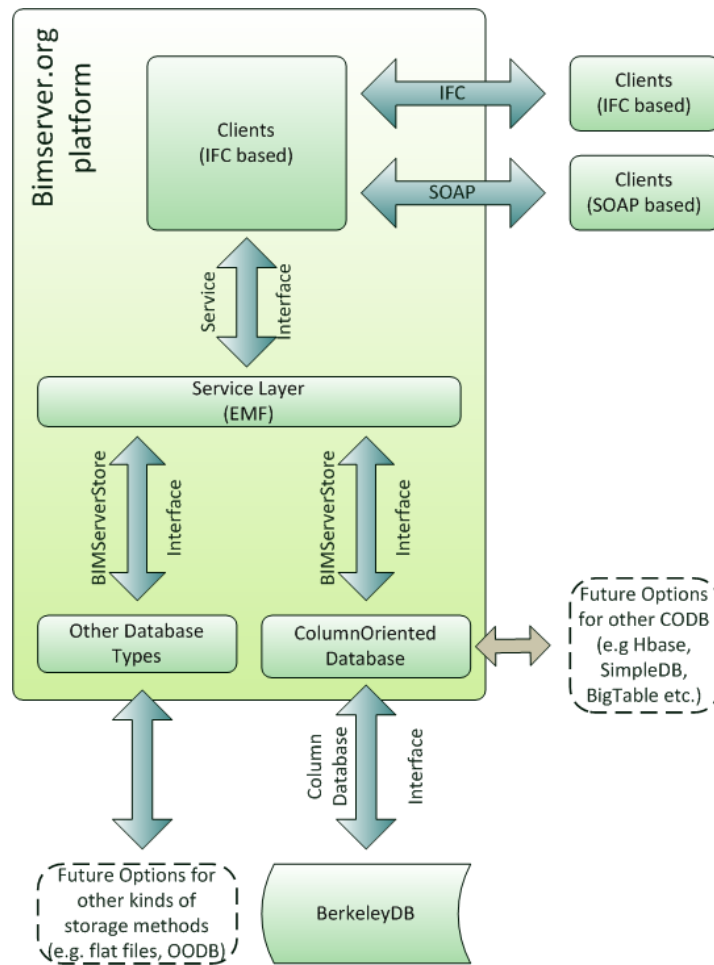


Figure 1: Simplified architecture diagram

At the core of the framework, a generic EXPRESS parser based on the Open Source EXPRESS parser (OSEXpress, 2001) by Lubel and others from NIST is used to create an in-memory meta model of schemas like IFC2x3. With this internal model as a basis, a Meta-Object Facility (MOF) has been created in the Eclipse Modeling Framework (EMF) format, which is a meta model comparable to OMG's XML Metadata Interchange (XMI) (Gerber and Raymond 2003). For most EXPRESS constructs, the mapping and conversion to their EMF counterparts is straightforward: EXPRESS *Entities* are converted to corresponding *EClass* instances, EXPRESS

attributes including the bounds of aggregations such as LIST, BAG, etc are stored as their *EAttribute* counterparts. *Optional* EXPRESS attributes (which are the dominant citizens in IFC schemas) are made *unsettable* in the corresponding ECore model and EXPRESS *enumeration* types are transformed into *EEnum* classifiers in return. However for certain EXPRESS language features, work-arounds had to be introduced since no straight-forward mapping to EMF is possible: EXPRESS *TYPEs* that are derived from primitive types (integer, string, float etc.) are boxed into wrapper classes, since a similar inheritance feature for primitive classes does not exist in EMF or in its main target language Java. While this constitutes a minor inconvenience for developers at the moment, casting operators may be introduced in future versions of the project. In rare occasions inverse relationships cannot be generated in straight forward manners as their *EOpposite* EMF counterparts due to inheritance issues and manually coded work-around had to be introduced into the model. Other work-arounds include the storage of original string values alongside with their converted floating-point representations to prevent rounding and notation issues in round-tripping read/write cycles, a *tristate* wrapper type that to capture the difference between STEP's \$ and *Undefined* and an additional *NULL* value for enumerations that are disallowed in the EMF representation.

Figure 1 shows the various layers that allows for exchangeable components. For example another implementation of the ServiceInterface could provide the BIMserver services based on an in-memory representation of data, or even a network of other services. Another implementation of the BimDatabase interface would allow other types of database (for example relational databases) to be used. The ColumnDatabase interface at last makes it possible to switch to another key-value database like for example the distributed (Project Voldemort 2009).

The generated EMF model is used as the input for a code generator that creates early-binding classes for all entities and types found in the schemas. With this, the bimserver.org initiative also offers a generic early-binding IFC library that hopefully makes the framework a convenient tool for both academic researchers and industrial developers.

Upon the upload of a new revision by an end-user, a parser for IFC instance models, encoded in a Part 21 STEP Physical File Formats (SPFF) notation instantiates instances of the generated early-binding classes. Other parsers, like the part 28 ifcXML notation parser are work in progress. For the main purpose of database-persisted, versioned models, a persistency meta model has been developed that stores revisions of IFC models on a per-object basis: For each entity instance in the model, a unique key is generated composed of an object id, the id of the project and the revision number itself. This enables the generation of version deltas referred to as Change Sets in the bimserver.org platform: For a model $M=\{A,B\}$ of two entities A and B , where A contains a reference attribute to B , two records in the database are created along with their attribute values (either primitive types or object references). If a change of B is uploaded, a new record B' will be created while the original B is kept in the DB resulting in $M'=\{A,B,B'\}$. Even if a changed object A' refers to a new object C in a later revision of the model $M''=\{A,A',B,B',C\}$, the initial model state M can still be reconstructed by choosing the appropriate earlier revision ids. Currently, the creation of these Change Sets is left to the client-side application. In future versions of the framework this is replaced by automated server-side processes that generate version deltas autonomously. The absence of unique, non-ambiguous object identifiers that is inherent to the IFC model schema, in which only a fraction of instances are equipped with a GUID, and the file-oriented nature of STEP models in general, makes such automated delta-generations a non-trivial task. Candidate approaches for such detection mechanisms have been suggested by (Weise 2006) and (Arhau & Lombardo, 2006), however a clean solution of this generally recognized, yet still unsolved problem domain might require a fundamental change of underlying data structures and modeling methods altogether. Apart from these graph-topology comparison approaches, we are also looking into spatial-heuristic comparisons of model versions based on the geometric representation of building components. In such an approach, spatial comparison operators could be used to match seemingly disparate components in two model versions (old and new) which cannot be aligned by their GUIDs based on their level of geometric incidence. The framework provides a branching system based on the notion of sub projects. By adding IFC object data to a project that is a sub project of another, the data automatically extents both projects. A main project joins the database objects that are common to all subprojects.

To retrieve complete revisions or partial models that have been checked into the server, a number of query mechanisms have been implemented. Single objects can be extracted by their unique object identifiers, IfcRoot-

descendants can be retrieved by their IFC GUID, all instances of a class can be retrieved (e.g. all doors) and for more sophisticated queries Java query code can be sent to the server where it is compiled and executed at runtime. As an illustration of such query code, consider the code snippet in fig. 2, which also illustrates the use of the early binding model API and the *wrappedValue* solution. Here, a model is queried for all doors on the first storey that exceed a height of two meters.

```

public class Query implements QueryInterface {

    private IfcDatabase<Long> model;
    private PrintWriter out;

    @Override
    public void query(IfcDatabase<Long> model, PrintWriter out) {
        List<IfcBuildingStorey> stories = model.getAll(IfcBuildingStorey.class);
        Map<Float, IfcBuildingStorey> orderedStories = new TreeMap<Float, IfcBuildingStorey>();
        for (IfcBuildingStorey storey : stories) {
            orderedStories.put(storey.getElevation().getWrappedValue(), storey);
        }
        if (orderedStories.size() > 1) {
            IfcBuildingStorey firstFloor = stories.get(1);
            for (IfcRelContainedInSpatialStructure rel : firstFloor.getContainsElements()) {
                for (IfcProduct product : rel.getRelatedElements()) {
                    if (product instanceof IfcDoor) {
                        IfcDoor ifcDoor = (IfcDoor)product;
                        if (ifcDoor.getOverallHeight().getWrappedValue() > 2) {
                            out.println(ifcDoor.getName().getWrappedValue()+
                                " " + ifcDoor.getOverallHeight().getWrappedValue());
                        }
                    }
                }
            }
        }
    }
}

```

Figure 2: example custom query code

XML configuration files that are editable by end users allow a fine-grained control over classes, attributes and relationships that should either be left out of query results or included. In future, this mechanism can be used to allow the extraction of sub models according to predefined Model View Definitions (MVD) in the context of the Information Delivery Manual (IDM) initiative or project specific filter lenses.

Based on TNO's IfcEngine library (TNO IFC Engine 2002), a number of geometry operations have been integrated into the framework using a Java Native Interface JNA that run on Windows, Apple (unix) and Linux platforms. The current main applications of this functionality are the generation of export formats (CityGML including GeoBIM extension, Collada, KML/KMZ) based on complete model revisions or as partial models based on the queries described earlier, an O3D viewer that is integrated into the web client and clash detection functionality that is reported in (Helm et al 2010).

The choice of the key-value database BerkeleyDB engine as the backbone of the persistency framework (after less successful experiments with traditional object-relational mapping frameworks) led to a robust, well-performing and transactional framework that allows the handling of large models such as the HITOS reference model (Lê et al 2006, Howard & Björk 2007) with its more than a million entity instances.

Many of the functions provided are also exposed as SOAP interfaces through a WSDL file, which allows developers the integration of remote server interaction into their own applications. The ability to refer to distinct revisions of objects in a RESTful manner provides a powerful mechanism for distributed models.

3. SERVER AND WEB CLIENT FROM AN END-USER PERSPECTIVE

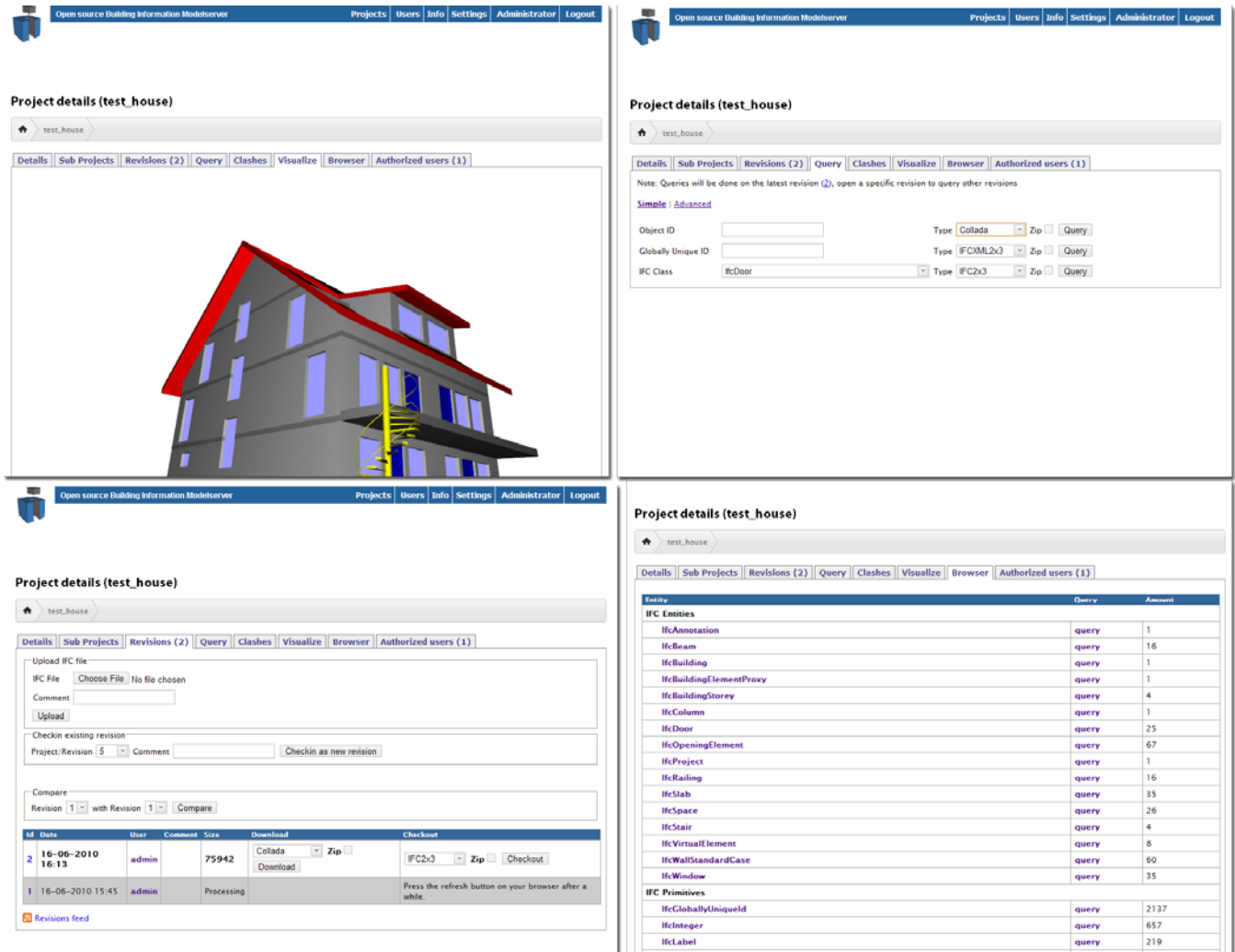


Figure 3: screenshots of different modules of the model server exposed through the web browser client: a) visualization module (upper left), b) query module (upper right), c) revision overview (lower left) and d) object browsers (lower right)

After downloading the platform either as a stand-alone JAR executable or as a WAR archive for deployment on servlet containers like Apache Tomcat, interested end-users can start the server with a single mouse click, since the default configuration needs little to no adaption to run on common Windows, Apple (unix) or Linux in both 32bit or 64bit environments. Next to the server that also contains a built-in Jetty web server, a web browser based client is currently the main interface for end users. Here, users can be set up that will be assigned to projects. After creating a new project and adding users that are allowed to check revisions in, an initial upload is made.

Different modules now allow a number of operations directly in the web browser:

An O3D-based visualization allows the visual control of uploaded models (Fig.3a). This module can also be used for the visualization of query results. Models can be queried using different approaches described in section two (Fig. 3b). A revision interface lets users download revisions and browse their history (Fig. 3c) and an object browser provides statistical overviews of objects, primitives and relations within a given revision of the model

(Fig. 3d). Furthermore, clash detection results of arbitrary revisions can be displayed and a sub projects module allows the creation of version branches for the management of design alternatives.

4. USER EXPERIENCES

Numerous users have experimented with the software since its initial release. A growing amount of informal user feedback is collected through the mailing lists and during hands-on workshops. The spectrum of reported user experiences is very broad. Almost all users agree that the archiving function, and the certainty that the latest information is in one place, make the BIMserver software added value for any engineering consortium.

The most appreciated features of the BIMserver software are the authentication and merging concept on top of the subprojects. Most experts agree that the merging as it is implemented in the BIMserver so far (June 2010) needs improvement. The software 'just' places all IFC objects found in the subprojects into one IFC file. Using the same objects in subprojects will result in double objects in the main project. For most users however this doesn't seem to be a problem. They use the merging capabilities to evaluate the geometry consistence of the main model. The authentication system provides users with write access to their own subproject, but also grants them read access to download and evaluate (other parts of) the model.

A frequent request is the finer control of changes introduced by different users in the collaboration process the. Currently, the BIMserver software supports this with a 'early warning system' that warns users to avoid model inconsistency between different users. This feature is seen as a highly important feature by almost all users, but no users have really tried it in practice. The same goes for the RSS feed for model updates; the GUID based change finder and the model summary: all users recognize the importance and are enthusiastic about the feature, but usage in practice is minimal.

Relatively insignificant features like the ability to upload/download IFC models in a ZIP file are highly appreciated because of the reduction of upload/download time on slow internet connections. The Google Earth network link is popular with architects. Architects send the URL of the latest model in KMZ to their clients at the start of a project. Clients can check the status of the design at any given moment. Also the export features for IFC to Collada, CityGML and ifcXML are reasons for users to install and use the software.

Advanced users are looking forward to the configurable clash detection. In this new feature users can filter objects based on subproject or object type to check collision between them. Early adopters agree that this could be the most important reason for new users to use the software.

A prospective connection between IFC objects in the BIMserver to Document Management Systems (DMS) and client side software (for example by using the SOAP interface) is regarded as one of the most promising future additions to the BIMserver software. The current web client user interface receives negative feedback. It is perceived as being too technical. Professional user interface design could mean a higher user rating.

5. SUMMARY AND OUTLOOK

In this paper the IFC-based Open Source building information model server bimserver.org has been introduced. Being both a software development platform as well as a working IFC server available to both SME and large enterprise end-users at no cost and with minimal technical knowledge required, the project aims to gain traction in both research and development communities. The platform provides granular revision control on an object-level for IFC model populations and can be adapted to work with other EXPRESS schemas in the future. It is based on an early-binding API provided by layered code generators on accepted (meta) modeling languages that allows comfortable interactions with IFC instances programmatically and provides flexible interfaces for versioned BIM models both by a SOAP API as well as by a web client interface prototype for end users. It features a number of modules such as visualizations, clash control and flexible query and filter mechanisms and is extendable. A small but vivid multi-national developer community and a few hundred subscribers to mailinglists have provided valuable feedback. The platform has been installed in several companies for test purposes and frequent user meetings in the Netherlands have been very positive and encouraging for the project.

A number of improvements and extensions will be added to the platform in the near future. Some of these developments are the implementation of more intelligent model merge strategies, the extension of query

possibilities and the addition of further export formats such as gbXML. In the long run, an integration with other initiatives such as IDM, MVD, and ongoing work in the application of Semantic Web technologies (Böhms et al 2008), (Beetz 2009) would be as desirable as the more tight integration of a native geometry kernel. All this depends on future funding and community contributions, making the BIMserver.org initiative a valuable addition to the tool chain that drives open standards in model-based collaboration in the building industry forward.

REFERENCES

- Arthaud, G., & Lombardo, J. (2006) "Automatic Semantic Comparison of STEP Product Models". *Innovations in Design & Decision Support Systems in Architecture and Urban Planning*, 447–463.
- Beetz, J. (2009). "Facilitating distributed collaboration in the AEC/FM sector using Semantic Web Technologies" PhD Thesis, Eindhoven University of Technology. ISBN 978-90-6814-618-9
- Böhms, H. M., Bonsma, P., Bourdeau, M., & Josefiak, F. (2008). "Semantic Product Modelling with SWOP's PMO". In *Proceedings of the international conference on EWork and EBusiness in Architecture, Engineering and Construction: ECPPM 2008*, p. 95.
- Gerber, A., & Raymond, K. (2003). "MOF to EMF: there and back again". In *Proceedings of the 2003 OOPSLA workshop on eclipse technology eXchange* 60-64.
- Helm, P. van den, Böhms, M. & Berlo, L. van (2010) "IFC-based clash detection for the open-source BIMserver". To appear in the *Proceedings of the International Conference on Computing in Civil and Building Engineering ICCCBE 2010*.
- Houbaux, P., (2003) "SABLE (2005) Simple Access to Building Lifecycle Exchange project by Eurostep".. URL: <http://www.blis-project.org/~sable/>, last accessed 01.06.2010
- Howard, R., & Björk, B. C. (2007). "Building Information Models—Experts' view on BIM/IFC developments". In *Proceedings of the 24th CIB-W78 Conference*, Maribor.
- Jørgensen, K., Skauge J., Christiansson, P., Svidt K., Sørensen K.B., Mitchell, J. (2008) "Use of IFC Model Servers. Modelling Collaboration Possibilities in Practice". *Technical report by Aalborg University and Aarhus School of Architecture*. http://it.civil.aau.dk/it/reports/2008_Ifc_model_server.pdf
- Kiviniemi, A., Fischer, M., & Bazjanac, V. (2005). "Integration of multiple product models: Ifc model servers as a potential solution". In *Proceedings of the 22nd CIB-W78 Conference on Information Technology in Construction*.
- Lê, M. A. T., Mohus, F., Kvarsvik, O. K., & Lie, M. (2006). "The HITOS project-A full scale IFC test". In *eWork and eBusiness in architecture, engineering and construction: proceedings of the 6th European Conference on Product and Process Modelling*, 13-15 September 2006, Valencia, Spain, p. 191
- Nour, M., Firmenich, B., Richter, T., & Koch, C. (2006). "A versioned IFC database for multidisciplinary synchronous cooperation". In *Proceedings of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering, Montreal, Canada*. pp. 636-645
- OpenIFCTools (2009) "Open IFC Tools", <http://www.openifctools.org/>. Last accessed 01.06.2010
- OSExpress,(2001), "Open Source EXPRESS Parser", <http://osexpress.sourceforge.net>, last accessed 01.06.2010
- Project Voldemort (2009) "Project Voldemort: A distributed database". <http://project-voldemort.com/>, last accessed 01.06.2010.
- TNO IFC engine, (2002), "TNO IFC Engine series", <http://www.ifcbrowser.com/>. Last accessed 01.06.2010
- Weise M. (2006). "Ein Ansatz zur Abbildung von Änderungen in der model-basierten Objektplanung. (An approach for the representation of changes in model-based design)" *Doctoral Thesis, Institute of construction Informatics, Report 4 (Ed. R. J. Scherer)*, Technische Universität Dresden, Germany, ISBN 3-86005-557-7