Aalborg University
Department of Production
Department of Civil Engineering
Aarhus School of Architecture
Department of Building Design

Use of IFC Model Servers

Modelling Collaboration Possibilities in Practice

Kaj A. Jørgensen

Jørn Skauge

Per Christiansson

Kjeld Svidt

Kristian Birch Sørensen

John Mitchell

Final version

May 2008

Preface

This report is a result of a research project, which has been carried out with focus on possible improvements on the way partners in building construction projects collaborate in building modelling by use of model servers based on the international standard Industry Foundation Classes (IFC).

The project was initiated by leading members of IAI in Denmark, who also performed a preliminary study about available model servers. The primary research work has been carried out by researchers at Aalborg University and Aarhus School of Architecture. Several others, however, have also participated and contributed.

The research project has been carried out based on cooperation with partners of a real building construction project – the "Nano Building" Construction Project located at Skjernvej in Aalborg – in the following termed "test project" and "test building". The partners of the test project have been involved in discussions about the research work and have delivered valuable data for the project – primarily the 3D models developed by the architect and the engineers.

It is anticipated that model servers will suit many project/client situations:

- Large projects (for large institutional, design, contracting and FM companies)
- Single building projects (for smaller custom integration e.g. domestic housing)
- Project hosting services (as a model management server, such as design coordination, whole project data management and project management, complex alterations e.g. hospitals, laboratories, etc.)

The primary objective has been to test the use of IFC based model servers to support collaboration in building modelling and end to evaluate the possibilities for practical use. The project includes activities for testing of the functionalities of model servers and for giving feed back to the on-going system development. Further, the project aims at development of a demo, which can show how model servers can support building modelling performed on one shared building model and thereby indicate a possible solution to many current coordination problems.

In the project, the EDM Model Server from EPM Technology in Norway has been used and a close dialog with EPM has been maintained throughout the project. In the preliminary study, also the EuroStep model server was used.

During the project, a considerable delay has occurred because preliminary performed tests showed that the required functionalities of the model server were not ready for use. Thereby, it was impossible to carry out practical use of the model server concurrently with the ongoing test project and to involve the partners as much as originally planned. However, three models from the architect and the engineers of the test project have been used in the testing activities and as the basis of the demo.

This project is partly based on other research projects at Aalborg University and Aarhus School of Architecture. Especially the recently performed project "Building Models and Building Modelling" made recommendations about modelling on shared building models. In addition, important studies have been performed at University of New South Wales in Sydney in Australia.

The authors would like to thank a number of people for valuable input to the project and preparation of this report. First of all, many thanks to Jan Karlshøj from Ramboll, Kristian Agger from Aarhus School of Architecture and Jesper Vaupel from Building Informatics for initiating the project and for reviewing of the draft versions of the report. Also many thanks to Gudbrand Skarseth from EPM in Norway for many discussions about the functionality of model servers in general and the EDM Model Server in particular. In addition, the partners of the test project have contributed with three models and some important discussions. Finally, great thanks to the Danish Ministry of Science, Technology and Innovation for the financial support, which made the project possible.

May 2008

Kaj A. Jørgensen, Aalborg University
Jørn Skauge, Aarhus School of Architecture
Per Christiansson, Aalborg University
Kjeld Svidt, Aalborg University
Kristian Birch Sørensen, Rambøll and Aalborg University
John Mitchell, University of New South Wales

Contents

1. Introduction

Usually, building construction projects are carried out by *multiple collaborating partners*: clients, advisors, contractors, building end-users, etc. There are different collaboration approaches and various amounts of data are exchanged between the partners during the project period including requirements, descriptions, drawings, budgets, etc. [Fox 2007]. In recent years, many partners have started to work with computer based building models, which are also subjects for data exchange. In this report, it is assumed that such models are used and data, which are not represented in building models, are not included in the scope of the report.

Exchange of building model data between multiple partners embrace a number of problems [Bakkmoen 2007] [Fox 2007] [Senate Properties 2007]:

- The partners create and use different building models, which focus on different disciplines or different parts of buildings.
- Different building models are usually created and maintained by use of different modelling tools. This may lead to redundant data and changes of models must often be performed in multiple models.
- Different modelling tools have different building model representations and proprietary file formats. File exchange by use of such file formats requires that all software applications have the necessary interfaces. Otherwise, some work has to be repeated.
- There is always a risk that maintenance operations performed on redundant data lead to inconsistent data.
- Many resources may be lost by performing unnecessary extra work and by working with inconsistent data.

Usually, building construction projects are divided into multiple phases with various engagements from the different partners. This report is only addressing the *design phase* and it is limited to the collaboration between three partners: *architect, building structure engineer and building services engineer*.

1.1 Shared Building Models

Traditionally, each partner in a building project develops several separate models, which often contain redundant data. Even within a single discipline there co-exist several models containing e.g. analysis data, geometrical data for drawings, building part specifications etc. When multiple models are used to describe the same objects from different perspectives, consistency between the models arises as a very important issue. Depending on the capabilities of the CAD tools used and the experience on working with building models in the design teams, it has become practice to seek for possibilities to create shared models, where separate models are inter-linked an containing few redundant objects. It means that e.g. the architect's geometric model consists of links to models delivered by the structural and mechanical engineer supplemented with the objects the architect is responsible for. These inter-linked models are developed through several data exchange iterations.

In contrast to working with multiple models, an alternative approach is to work on a shared building model [Jørgensen 2007], which would obtain data produced in the entire lifetime of the building (see Figure 1) from the time, where the idea of the building is born, through the construction period and the utilisation period to the time of removal of the building and even after that.

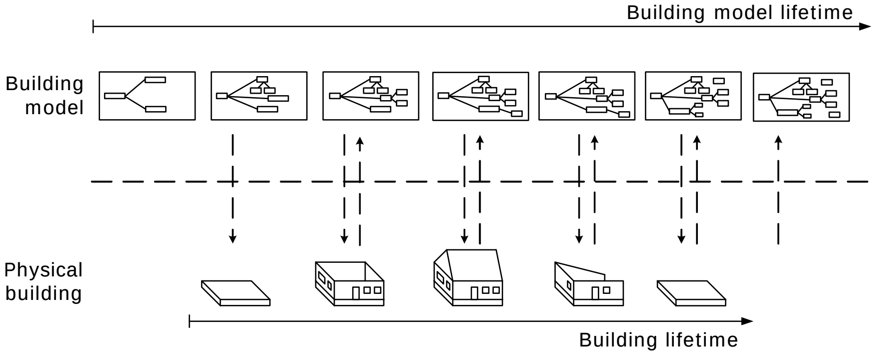


Figure 1 – Building model life time and the life time of the physical building

In this scenario, the model is supplied and extended with data through the design activities, the construction activities, the utilisation activities, etc. [Bakkmoen 2007] [Fox 2007] The contained data is carefully maintained at all time and by all partners so that the model can be reused as much as possible in all life phases. Update operations are performed on the shared model and data extractions will be made from the shared model. Although it is not likely to have a single model implemented with the current conditions in the building industry, it is the primary scope of this report to demonstrate the current practical possibilities for this scenario and its advantages and downsides.

1.2 IFC and IFC Based Model Servers

In order to provide a common data representation and thereby enable easy exchange of model data, international standardisation work is carried out by the *International Association for Interoperability (IAI)*[1], who has published the neutral data model *Industry Foundation Classes (IFC).*

The IFC work that was commenced in the mid 1990s opens up for more efficient communication between applications containing digital building models. The IAI group was founded to start to work on a more short-term operational communication standard than the since the first half of 1990s ongoing PDES/STEP work (Product Description Exchange Specification/Product Data Exchange using Step - PDES, Standard for the Exchange of Product Model Data - STEP). The IGES (Initial Graphic Exchange Specification) had been around since the early 1980s but did not support model-to-model communication but transfer of graphical primitives. PDES (ISO 10303) was launched as a single international standard to cover all aspects of a neutral format for product data exchange [NBS-DATA 1988] [Wright 2003].

The IFC data model is very comprehensive and is until now the best attempt made to provide support for the idea of collecting all data of a building model in a shared representation. Figure 2 illustrates, how a number of different software applications share

[1] IAI is an alliance of organizations dedicated to bring about a coordinated change for the improvement of productivity and efficiency in the construction and facilities management industry. Members of IAI engage in national-industrial programmes that aim to change the organisation, process and technology of the industry. See www.iai-interoperability.org for more details.

information using the IFC data model. IFC is now approved by ISO as the standard ISO/PAS 16739.

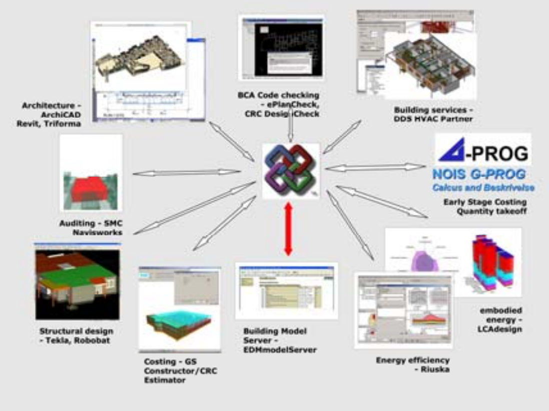

Architecture - ArchiCAD Revit, Triforma
BCA Code checking - ePlanCheck, CRC DesignCheck
Building services - DDS HVAC Partner
-PROG
NOIS G-PROG
Calcus and Beskrivelse
Early Stage Costing Quantity takeoff
Auditing - SMC Navisworks
Structural design - Tekla, Robobat
Costing - GS Constructor/CRC Estimator
Building Model Server - EDMmodelServer
Energy efficiency - Riuska
embodied energy - LCAdesign

Figure 2 – Shared IFC model Environment

IFC concentrate on representation of the core data about building components as model objects and independently of the modelling tools. So, data about how the building is presented is secondary, e.g. surface colours, line weight and line colours.

In addition, the IFC standard includes also specifications of how model data can be represented in data files[2]. Thereby, software vendors can develop interfaces, which can read and write files, where building models are represented by IFC.

Application A
Data File
Application B

Figure 3 – Data exchange based on file

A few software vendors have implemented the complete IFC data model as a *database*, which can then accept and store all data objects that comply with IFC. A computer system with such a database is termed an *IFC model server*. A model server is a database a set of server applications that provide multi-user database management and allow use of the IFC data model as the underlying database structure. Such servers obviously provide means for model upload and download but other functionalities are typically provided. Ideally, applications should be able to exchange data directly with model servers (see Figure 4).

[2] Originally, the ISO 10303 Part 21 file format was developed and specified but lately a file format based on XML has also been created – ISO 10303 Part 28. The Part 21 file format is usually referred to as the IFC file format.

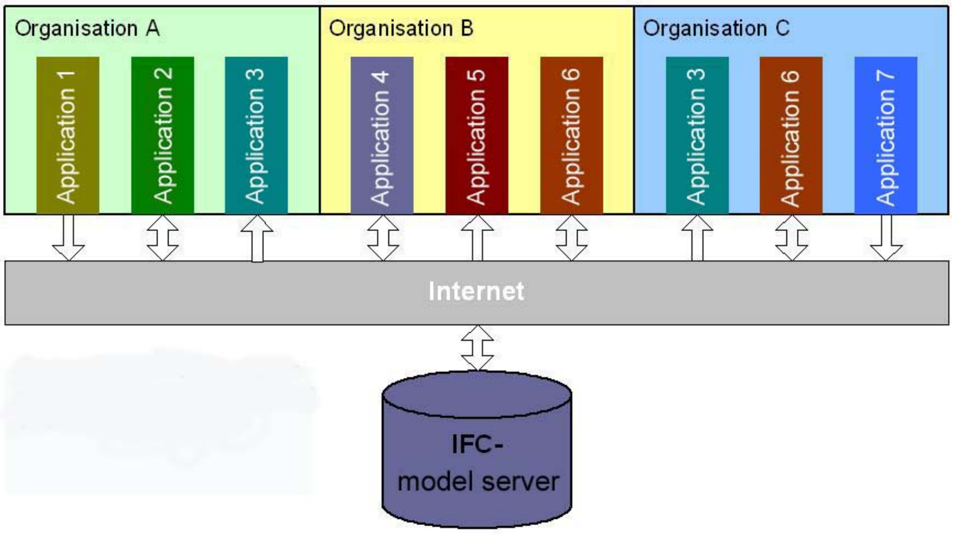


Figure 4 – Data exchange with model server

Hence, the current situation is that a comprehensive international standard for representing building models is available and corresponding model servers can store all kinds of models, which can be represented by IFC. Although some attempts are made [NIBS 2007], there are no widely accepted standard for, what the functionalities of model servers should be and what features they should offer. When it comes to software tools in relationship with IFC, there are serious limitations. Some tools can import IFC formatted files, some can export to IFC formatted files and some can both. However, the way this is done differs quite a lot [Bakkmoen 2007]. Because each tool has its own internal representation, it is sometimes impossible to perform exact transformations and, in such cases, all data are not exported or imported [Erabuild 2008] and [Senate Properties 2007].

2. Modelling Collaboration Scenarios

Different scenarios for collaboration about development of building models in a construction project can be identified. In this chapter, three scenarios are sketched expressing differences in how closely the design partners work together and how the modelling activities are coordinated. One of these is the future collaboration scenario with one shared building model, which is investigated in this report and the others are two commonly used scenarios, where partners develop separate models.

2.1 Characterisation of Scenarios

The primary differences between the three scenarios are described regarding a set of fundamental characteristics for the work performed on shared model data:

Redundancy	Redundant model data are data, which occur more than once and represent the same fact. If redundant data need to be updated, it is important to update all occurrences.
Integrity and consistency	Integrity refers to accuracy or correctness of data, i.e. that model data represent real world facts correctly. Consistency refers to logical coherence, i.e. that there are no contradictions in the data. One way of getting inconsistent data is when redundant data area being updated and not all occurrences are updated. However, data can be inconsistent in many other ways and it must be emphasised that integrity and consistency are not identical. Data can be consistent even though they are not correct. In order to secure and preserve integrity and consistency of model data, it is necessary to establish rules and procedures for users to obey. It is also necessary to perform periodic checking and comparison. Inaccurate and inconsistent data can cause misinformation.
Accessibility	Model data can be accessed differently by different users or clients. In the following, users are separated into three groups: 1) the model creators, 2) model coordinators and 3) model clients. Users can have direct access or can be granted access through more privileged users.
Model responsibility	Certain responsibilities are normally always associated to the model data. So, it must be clearly identified in the user organisation, who is responsible for the various tasks performed on model data.
Model ownership	Ownership refers to who owns the model in the design phase.
Performance	Performance refers in this context to the use of resources for creation and maintenance of model data.

2.2 Separate Models

In this scenario, each partner creates one ore more separate models, which most often relate to different disciplines. Coordinators and model clients get views of the models, typically drawings, from the model designers. The models will often be partly redundant and some objects will thus be represented in multiple models and changes to these objects have to be repeated in each relevant model.

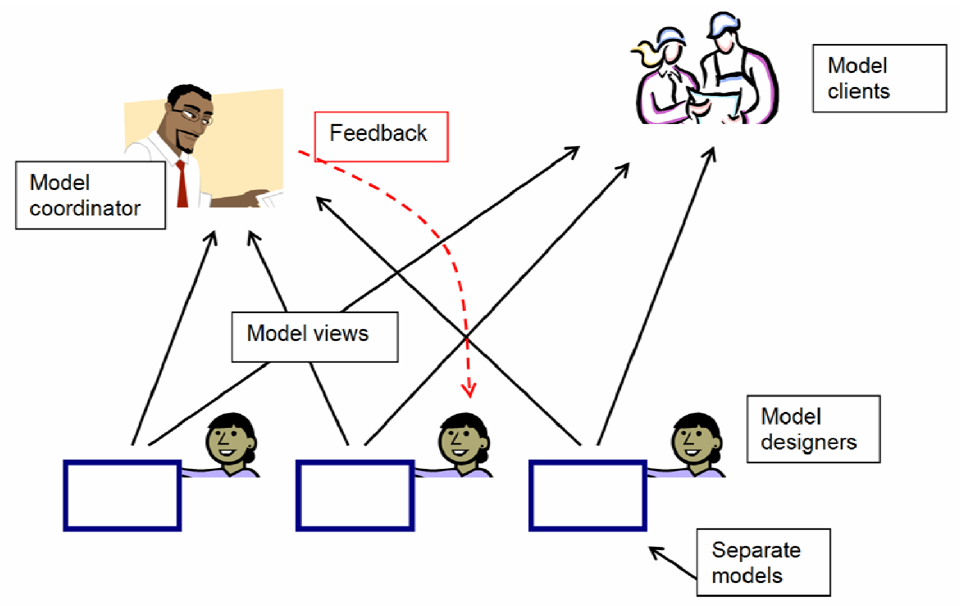


Figure 5 – Collaboration scenario with separate building models

2.3 Separate Models with Aggregated Model

This scenario is identical to the previous scenario except that an aggregated model is created from time to time.[3] The primary purpose of this model is the support for design coordination. The design coordination can benefit from views of the aggregated model directly and many model client views can be produced from the aggregated model.

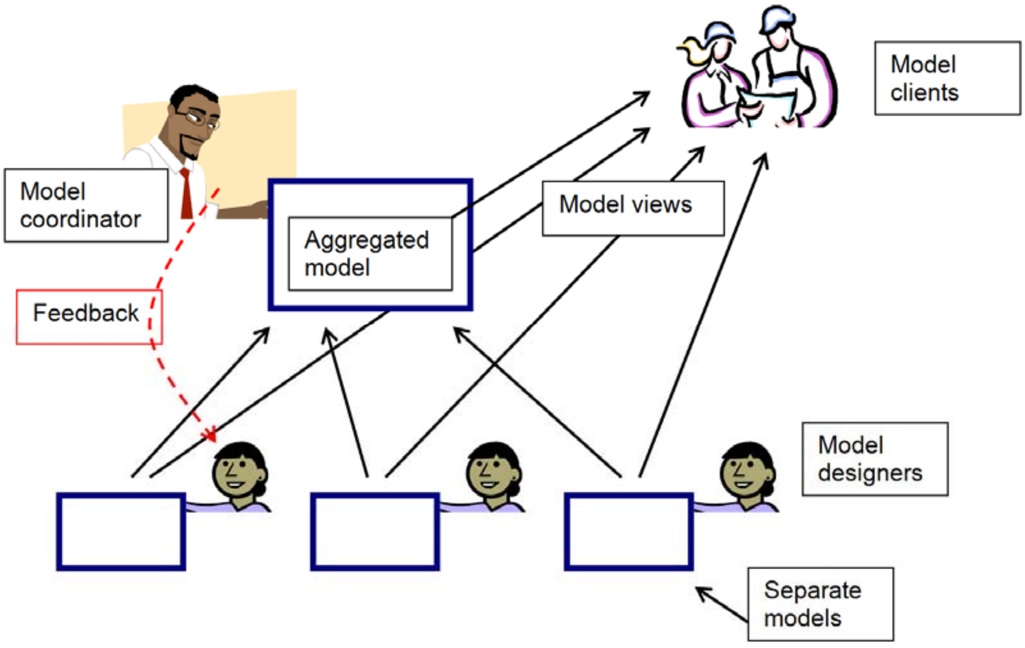


Figure 6 – Collaboration scenario with separate building models and an aggregated model

[3] A number of tools are available for model aggregation, e.g. Solibri Model Checker and Navis Works.

2.4 One Shared Model

In this scenario, all partners collaborate on the design and maintenance of one shared model [Jørgensen 2007] [Plume 2007]. Coordinators as well as model clients may be granted access to produce their own views.

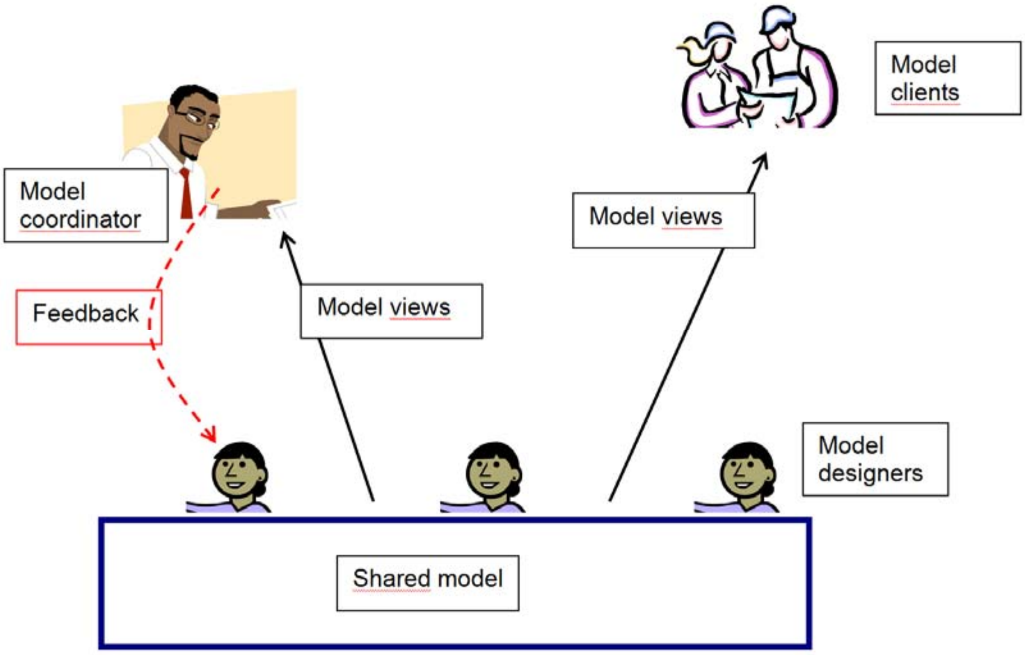


Figure 7 – Collaboration scenario with one shared building model

2.5 Comparison of Scenarios

The three scenarios are compared by use of the dimensions identified in section 2.1.

Scenarios	*1.* *Separate models*	*2.* *Separate models with aggregated model*	*3.* *One shared model*
Redundancy	Redundancy exists. When update operations are performed, some tasks must be repeated in other models too.	Redundancy exists. When update operations are performed, some tasks must be repeated in other models too.	Redundancy can be avoided and update operations are not repeated.
Integrity and consistency	Rules and procedures are developed in each separate organisation. Cross checking is difficult to perform.	Rules and procedures are developed in each separate organisation but cross checking can be performed on the aggregated model and supported by software.	Rules and procedures must be developed together. Checking is performed on the shared model.
Accessibility	The coordinators and clients access typically the model through extracted views.	The coordinators access most often only the aggregated model. Clients access the model through extracted views.	Users of all groups can be granted access on group or individual level and can produce their own views at any time.
Model responsibility	Each organisation is responsible for its own model.	Each organisation is responsible for its own model.	The total organisation is responsible for the model but each organisation can be granted responsibility for relevant parts of the model.
Model ownership	Each partner has ownership over their own models.	Each partner has ownership over their own models.	The ownership of building model is shared.
Performance	Resources are wasted, when the same data must be created or maintained in multiple models.	Resources are wasted, when the same data must be created or maintained in multiple models.	Resources are not wasted on repeated operations.

2.6 Project Webs and Model Servers

Collaboration among building process stakeholders is today often supported by Project Webs, Internet based Electronic Document Management systems.[4]

The Project Web is a shared *file* container for a project stored on a server, which can be accessed through the internet. Files of any kind (text documents, spreadsheets, CAD models etc.) are organized in a hierarchy of directories. Similarly, the model server is a shared *object* container for building model objects.

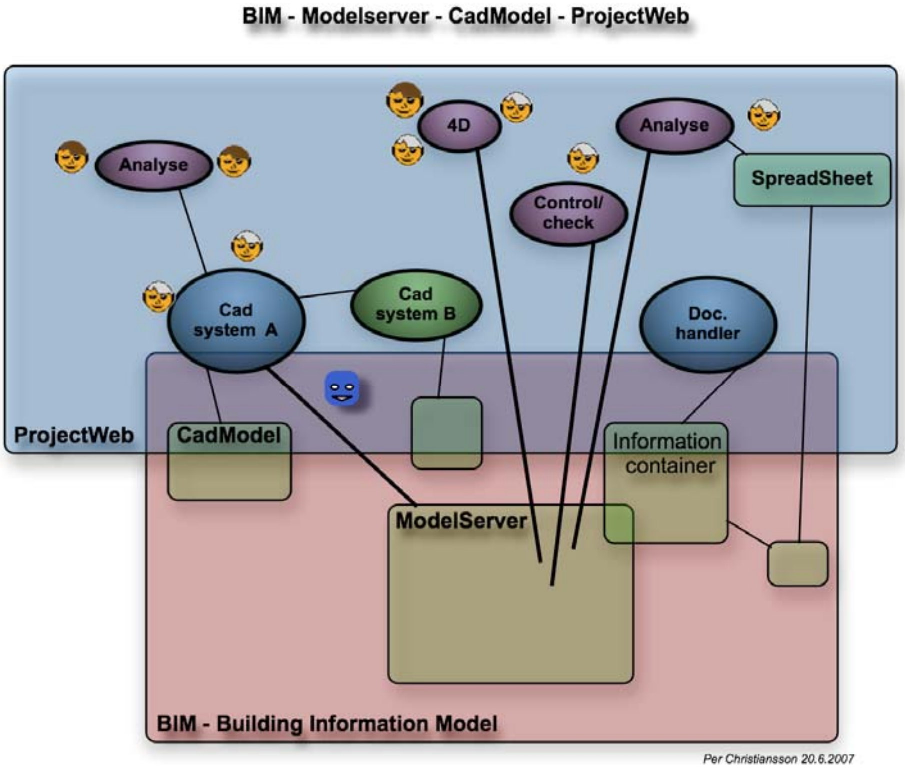


Figure 8 Information/model containers are accessed from analyses/simulation/control programs. Organization, collaboration, work methods, responsibilities are also influenced by and influencing ICT tools properties.

3. Model Servers

As already stated, model servers are special database systems, by which multiple users can share building models. Users can be granted access rights to a model server and can then, as a basic functionality, upload models to a server and download models from a server. In addition, model servers potentially provide high level functions of coordination and project life cycle management and operational data management [Plume 2007] [InPro 2008].

[4] Today this is compulsory for public building projects (over 3 million DKK) in Denmark.

Key aspects of building model servers are:

- Database sharing via data net
- User's rights / security
- Concurrent usage
- Discipline views (partial models/views), and ad hoc queries
- Version control
- Transaction processing
- Model merge
- Speed/performance/integrity/consistency
- Audit (user's roles, decisions, and issue tracking)
- Data protection (mirroring/backup)
- Model update history
- Storage

Potential benefits to be gained from a building server environment are:

- Flexible multi-disciplinary management
- Closer mapping between discipline/application data
- Large project scalability
- Wider & sufficient building structure/services engineering support for real projects
- Full life cycle support
- Ownership & security system
- Any selection of data
- Management mechanisms for long transactions and auditing

3.1 Other tools with similar functionalities

Many existing CAD tools offer different possibilities to share building models. If for instance a model consists of a set of linked files, it is often possible for multiple users to work concurrently on separate files – each file one user at a time. Some CAD tools based on one database for the whole model, such as ArchiCAD, Tekla Structures and AutoDesk Revit, support simultaneous multi user facilities. In the following, a short description of ArchiCAD's Teamwork facility is described as a representative.

ArchiCAD Teamwork is based on the native ArchiCAD model database. Teamwork allows members of a project team to work on separate parts of the same project, relatively autonomously, and merge their work into a shared model. In this way teams of designers can share an entire building model and its documentation.

Teamwork is controlled by a number of team members. On top is the coordinator of the team which is the administrator who coordinates the team. Next, is the team leader who is regarded the project responsible. Next in the hierarchy are a number of teammates, who can be granted access rights to work on a part of the shared model. To enable this, the teammate is prompted to reserve a 'Workspace'. Finally there are viewers, who are able to make read-only access to the shared model. When a team member has reserved a workspace, it is locked by the team member and cannot be changed by other team members.

Examples of typical workspaces are:

- A combination of stories
- Layers
- Animations
- Sections

- Elevations
- Interior Elevations
- Detail Drawings
- Worksheets
- Layouts

Workspaces can even be smaller parts of the building. Especially, a workspace can include ArchiCAD Hot Linked Modules. Such a module can be stored outside the shared model and can be referenced dynamically and multiple times from the shared model. In the shared model, hot linked modules will be shown by hollow square marks on their hotspots in different colours.

At any time the team members can send changes to the shared model and receive changes that others have made. Further, team members can add notes to elements in each other's workspace. During the work, no changes on the shared model can be seen by other team members until the changes are uploaded again. After upload the workspace can be unlocked. The practical use of these advanced locking facilities and system controlled user right management depend on local working culture, project organisation and interpersonal trust.

3.2 EDM Model Server and Comparable IFC Model Servers

The EDM Model Server is based on the Express Data Manager technology (EDM), which EPM[5] has developed based on the data modelling language Express (ISO 10303-11). The Express Data Manager EPM Technology is a database product, where the database structure is defined by Express. In connection with this, EPM uses an extension to the Express language Express-X with programming and query language features.

Eurostep Model Server (EMS) from Eurostep in Finland[6] is based solely on the Java programming language. It is, therefore, available on any server platform supporting Java. EMS can also be installed on multiple databases: MySQL, SQL Server and ORACLE. Eurostep has developed an XML-based Product Model Query Language (PMQL). Further, a standard browser based client application is provided. EMS also provides a conversion utility from the model geometry to Virtual Reality Models (VRML) for 3D visualisation.

A third IFC Model Server was developed by VTT Building and Transport and SECOM Co., Ltd[7], in 2002. The IFC Model Server also provides model server functionality over the Internet. This is offered in a technological independent way by using XML and Simple Object Access Protocol (SOAP) for communication between the model server and client software. Thereby, the model server functionality can be utilized independent on the programming language used to create the model server clients.

Along with the internationally available model servers, the SABLE server is also important to mention. The SABLE (Simple Access to the Building Lifecycle Exchange) project[8] has the objective to make it easier to communicate between client applications and multiple model servers. If each model server has an interface to the SABLE server, client applications need only to use one standard interface to enable connection to all model servers. As indicated,

[5] EPM Technology: http://www.epmtech.jotne.com/

[6] EuroStep: http://www.eurostep.com/

[7] SECOM: http://www.secom.co.jp/english/

[8] SABLE project: http://www.blis-project.org/~sable/

SABLE is a middleware server and not a model server it self. Eurostep has been the primary supporter of SABLE and EMS is the first SABLE compliant model server with a SABLE interface. The development of SABLE is currently ajourned.

Oracle[9] has also introduced a collaboration system for the construction industry Collaborative Building Information Management (CBIM). This system enables communication between partners through a range of visualisation and analysis modules. The system can import IFC models but IFC export is not supported. Furthermore, it is not possible to perform merge operations.

4. EDM Model Server

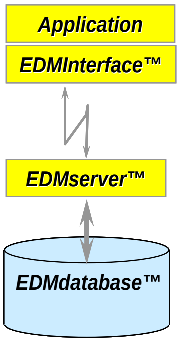


Figure 9 – Express Data Manager architecture

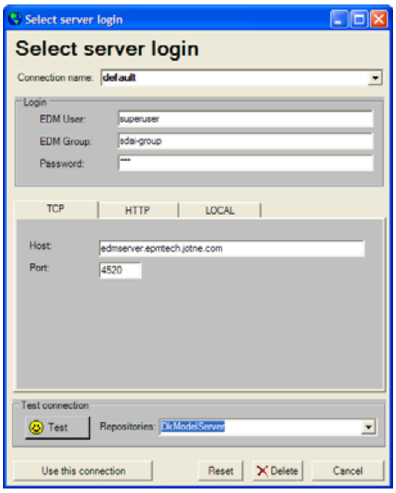
The primary components of the EDM Model Server are the EDMdatabase and the software, which handles it, the EDMserver. Each end-user client application can be based on the EDMinterface, which contains common services for communication with the EDMserver. As this communication can be based on both TCP and HTTP, client applications can connect locally or via the internet. Potentially, this means that e.g. CAD tools can connect directly to the EDMserver through the EDMinterface.

A special client application has been developed – the EDMModelServerManager (MSM) – for remote management of the server. In this project, the MSM application is the only application, which has been used. Therefore, the functionality of EDM Model Server is only seen through MSM.

Access to the model server is controlled by usernames and passwords. Some users may for instance be constrained regarding update operations. By this feature, it is also possible to delay the time when models are published to certain users.

Part of the login procedure is to select the connection mode, TCP or HTTP and selection of repository. Only one repository can be selected in MSM but multiple instances of MSM can be started in parallel and, in each instance, different repositories can be selected.

[9] Oracle: http://www.oracle.com/



Select server login

Connection name: default

Login

EDM User: superuser

EDM Group: sdai-group

Password: ***

Host: edmserver.epmtech.jotne.com

Port: 4520

Test connection

Repositories: DkModelServer

Figure 10 – Model server login

New empty repositories can be created with MSM and, within a repository, different primary objects like project objects, site objects, building objects, etc. can be created and manipulated directly with MSM. Creation of new repositories with content is most often performed by uploading a model from a file. With MSM, models can be uploaded and downloaded as files. Multiple versions of a model can be stored in the same repository at the model server. Instead of replacing the existing model, the model server can upload a model with a new version number added to the name.

Objects of a model can be selected via the spatial structure hierarchy (project-site-building-storey). This hierarchy can be manipulated the ordinary way with expand and collapse and, when objects are selected, various data about the objects can be presented in separate reports.

A special functionality of EPM Model Server is the check-out and check-in operations. Total models or partial models can be checked out for external update and later checked in again. At check-out, a special locking mechanism marks the checked out objects in the model server. Other users may still read these objects but only the user, which performed the check-out, or the user's super user are allowed to make changes. Normally, the checked out model or partial model is modified by a modelling tool and then re-entered by check-in. During check-in, a merge operation is carried out. During that, re-entered objects will replace the existing objects, new objects will be added and missing objects will be removed from the model server. A successful check-in will release all locks, created at check-out. The merge operation relies on correct handling of the Global Unique Identifier (GUID) for each model object. To manage the locking mechanism, a special check-out data set is created on the server at check-out and a special model server property set is added to the model and written to the check-out file. Thereby, the check-out, model update and check-in round trip can only work correctly, if the modelling tool can maintain the custom property set. Currently, ArchiCad is the only modelling tool, which can take proper care of the custom property set.

See Appendix A for further details about the functionalities of the EPM Model Server and the Model Server Manager.

5. Demo of Model Server Use

5.1 Models from the Test Project

The basis of the performed studies and the developed demos is the models created by the partners of the test project. Three separate models were available: an architectural model, a structural model and an HVAC model. The models were created with three different modelling tools: Architectural Desktop from Autodesk, Tekla Structures from Tekla and MagiCad from Progman.

All three tools were insufficient regarding export to IFC files and import form IFC files. Especially import is absolutely insufficient. All three models were examined from generated IFC files representing the models in IFC version 2x2.

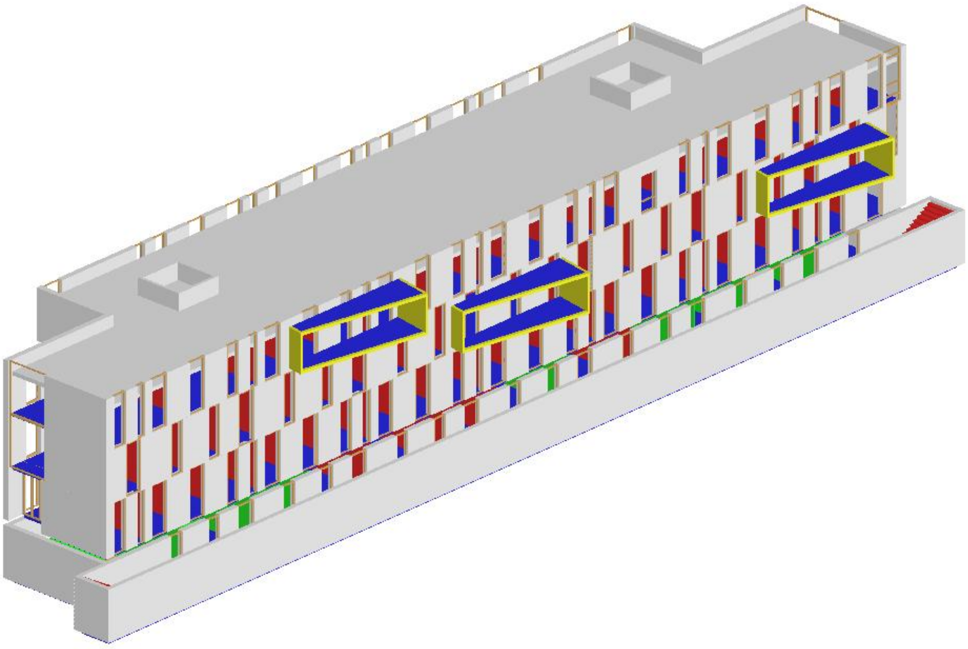


Figure 11 – The architect model

Both the structural model and the HVAC model were created based on the geometry from the architectural model. For the structures, it is done by importing the architectural model as the background for repeated modelling in Tekla. Unfortunately, this is currently the only way Tekla can import IFC files. MagiCad is a tool working on top of the AutoCAD environment like ADT. Therefore the architectural models were used as references for the building installation models in the native dwg-format. Consequently, the three models are separate models, where some objects are actually duplicated.

The result of the modelling performed by the structure engineer is a separate model with detailed walls, floors, steel structures etc.

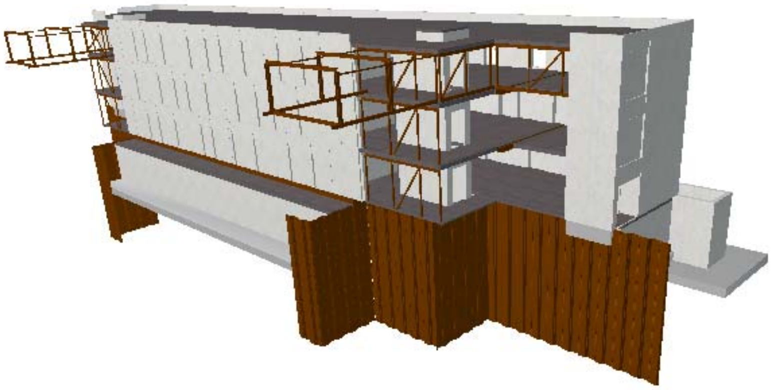


Figure 12 – The structural engineer model

By examining the architect model and the structural engineer model, it is observed that walls, for instance, are modelled in two different ways. In the architect model, each wall is a single object with internal specification of material layers in accordance with how it must be transformed to IFC representation. In Tekla on the contrary, each wall was modelled as two objects, one for each of the two concrete material layers. Tekla also supports the use of components were inner and outer wall can be modelled in one operation and be treated as connected objects, but this facility was not used in the project. The insulation material layer was not modelled in the structural model. Such a representation of walls will not be exported correctly to IFC. Another minor observation is that the origin of the system of co-ordinates is placed differently in the two models. This is of course a consequence of lacking coordination while creating the two separate models.

The HVAC model made by the building services engineer contains radiators and pipes for heating and water distribution. When exported to IFC file, walls, floor slabs and other construction objects were excluded. So, the intersections between pipes and the structural objects were not taken care of.

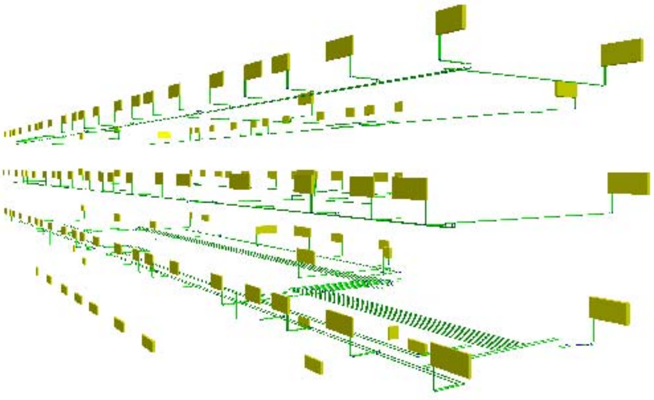


Figure 13 – The installation engineer model

5.2 Model Server Collaboration Demo

In contrast to the approach followed by the partners of the building construction project, this demo will show the possibilities for a scenario, where the shared building model is used by all partners and thereby gradually extended and updated by all partners. The basis is the architect model and the engineers make extensions and updates on this model.

All remote model operations are performed with ArchiCad because – as stated above – this is currently the only modelling tool, which can handle partial models (check-out, check-in and preservation of the special model server property set) and which can handle both IFC import and IFC export properly.

The demo is carried out in three steps. First, the base model is created and uploaded to the model server. Then, the model is modified to include the result of the structural engineering work and, afterwards, some selected HVAC objects are added to the model. Finally, some minor isolated changes are made on the model.

5.2.1 The Base Building Model is Created and Uploaded

The base building model is the model, which is considered the foundation for the collaboration between multiple partners of the project. Hence, this model is developed to a certain point by only one partner, usually the architect. During this development, the model server can be used to publish different initial versions of the building model.

As described, the demo model was prepared in ADT and further with Archicad. As the demo focuses on a minor part of the building model, some specific objects were selected and named (see Figure 14). The model was then exported from ArchiCad and uploaded to the model server – model name 'NanoBuilding'.

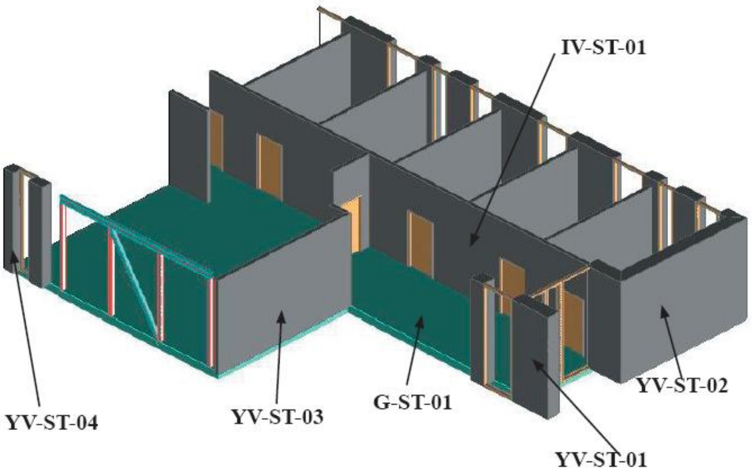
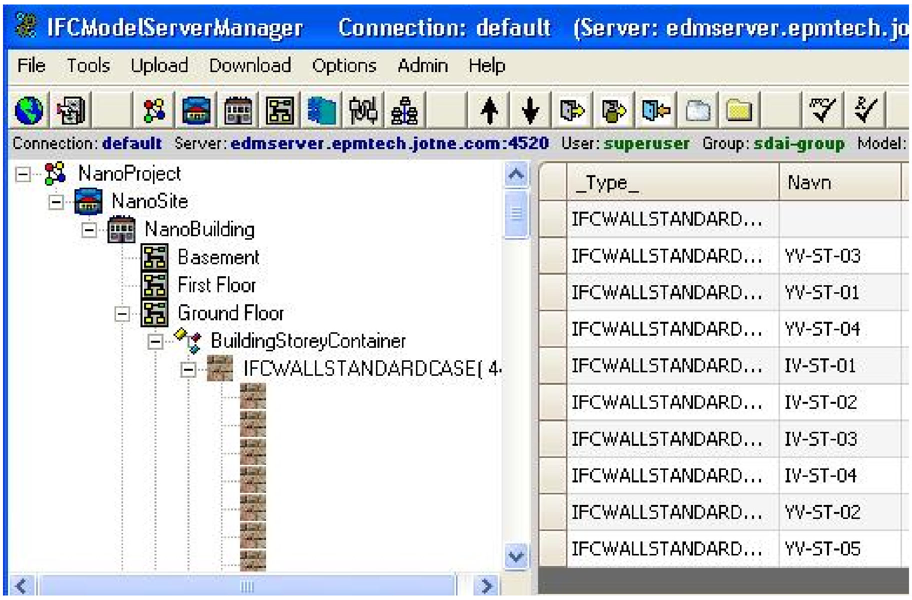


Figure 14 – Objects named on the ground floor (ST)

After upload to the model server, the IFC containment structure and objects can be viewed in MSM.


IFCModelServerManager Connection: default (Server: edmserver.epmtech.jo
File
Tools
Upload
Download
Options
Admin
Help

Connection: default Server: edmserver.epmtech.jotne.com:4520 User: superuser Group: sdai-group Model:

- NanoProject
  - NanoSite
    - NanoBuilding
      - Basement
      - First Floor
      - Ground Floor
        - BuildingStoreyContainer
          - IFCWALLSTANDARDCASE(4

Type	Navn
IFCWALLSTANDARD...	
IFCWALLSTANDARD...	YV-ST-03
IFCWALLSTANDARD...	YV-ST-01
IFCWALLSTANDARD...	YV-ST-04
IFCWALLSTANDARD...	IV-ST-01
IFCWALLSTANDARD...	IV-ST-02
IFCWALLSTANDARD...	IV-ST-03
IFCWALLSTANDARD...	IV-ST-04
IFCWALLSTANDARD...	YV-ST-02
IFCWALLSTANDARD...	YV-ST-05

Figure 15 – IFC containment structure (left) and selected wall objects (right) shown in MSM

5.2.2 Construction Details are added to the Model

In the demo, it is chosen to illustrate this step of the collaboration scenario by making changes of a few walls and the ground floor slab, i.e. the thickness of these objects is changed and the material layers are defined. This work is performed by downloading the complete model form the model server, importing it in ArchiCad, where the changes are carried out. The resulting model is then exported from ArchiCad and uploaded to the server. The model is stored in the same repository as a new version and named 'NanoBuilding_v1'.

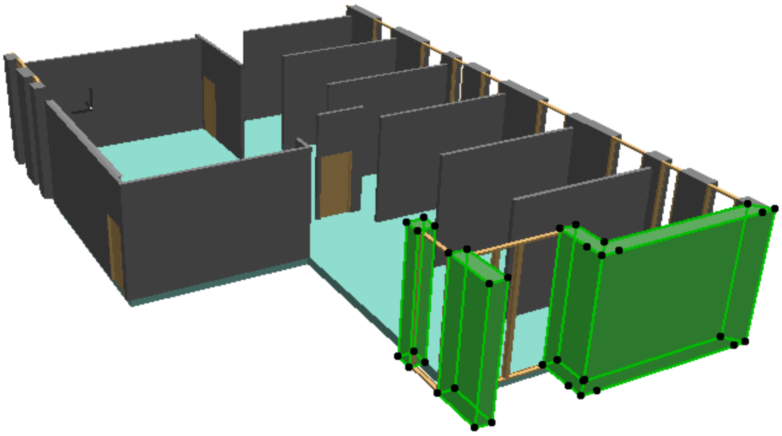
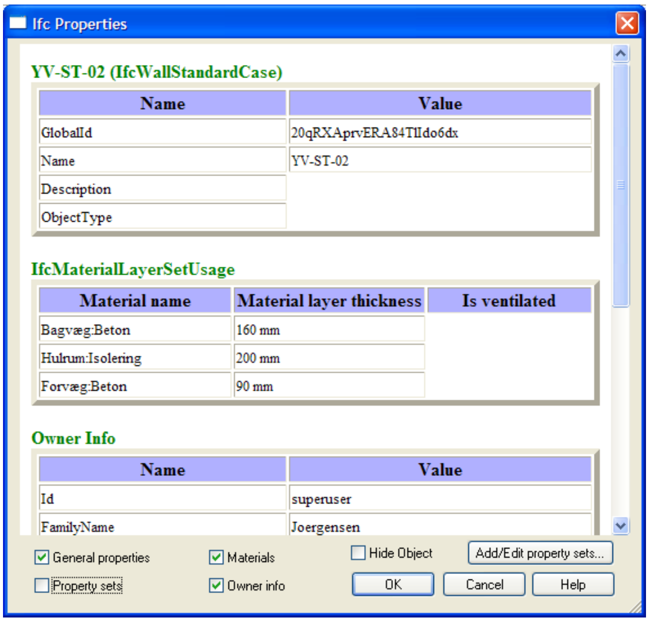


Figure 16 – The walls which has been changed regarding thickness and material layers

The DDS viewer can show the data about the material layers.



YV-ST-02 (IfcWallStandardCase)

Name	Value
GlobalId	20qRXAprvERA84TlIdo6dx
Name	YV-ST-02
Description	
ObjectType	

IfcMaterialLayerSetUsage

Material name	Material layer thickness	Is ventilated
Bagvæg:Beton	160 mm	
Hulrum:Isolering	200 mm	
Forvæg:Beton	90 mm	

Owner Info

Name	Value
Id	superuser
FamilyName	Joergensen

☑ General properties ☑ Materials □ Hide Object

□ Property sets ☑ Owner info

Figure 17 – *Data about material layers as shown in the DDS viewer*

Another result of the structural engineer is also illustrated in the demo, i.e. some columns and beams are added to the model. This work is also performed in ArchiCad, uploaded to the model server and stored as a new version – 'NanoBuilding_v2'.

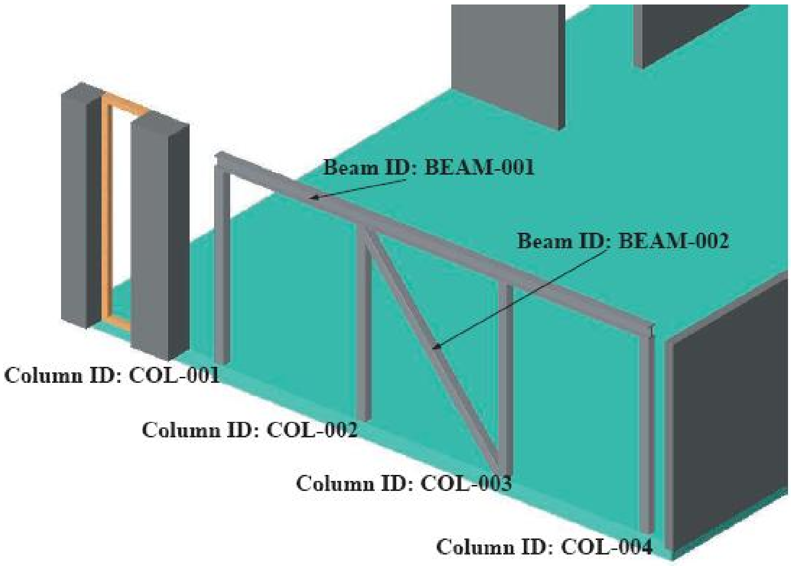


Figure 18 – Column and beam objects added to the building model

5.2.3 HVAC Objects are added to the Model

Further in the demo, it is chosen to illustrate the collaboration with the installation engineer by adding a few selected HVAC objects to the model. The objects are simply copied from the separate HVAC model by means of ArchiCad, uploaded to the model server and named 'NanoBuilding_v3'.

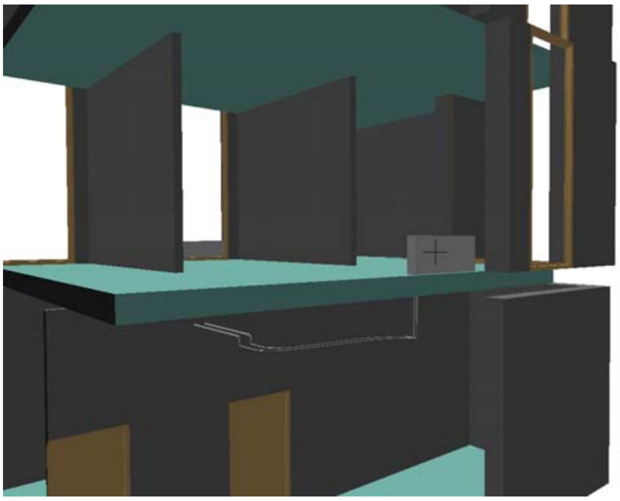


Figure 19 – HVAC objects added to the building model

It must be stated that holes for the pipes are not created in the demo. This is, however, an important issue related to the modelling as proposed in this scenario.

5.2.4 Isolated Changes are made on the Model

In order to show the most important form of collaboration, a special part of the demo is created. When major work has been performed on the model as described in the previous steps, the collaboration changes so that necessary changes of the model are performed by different partners and in random order. In this phase, it is necessary to secure that multiple users can not update the same objects at the same time and thereby risk loss of work. Therefore, such changes require locking of the objects, which needs to be changed. If major changes have to be performed on objects across the model, the entire model can be reserved (locked) but most often it will only be necessary to reserve only smaller parts of the model. With EPM Model Server, such operations are performed by check-out and check-in (see appendix XX for further description of the procedure).

In a previous step, some columns and beams were created in the model. Apparent, this was not performed correctly. One of the columns has bee placed in front of a door opening and should be moved sidewise to the wall and the beam should be extended accordingly.

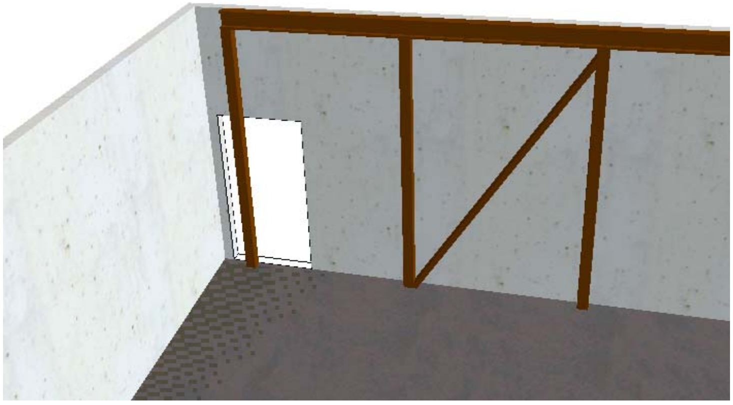
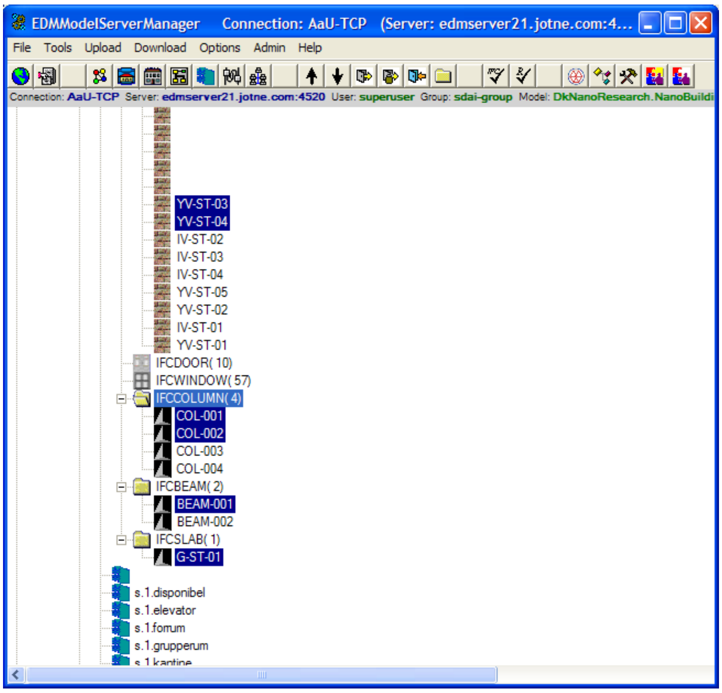


Figure 20 –Column and beam objects to be modified

In order to do this, a part of the model, the column, the beam and a few adjacent objects, are checked out from the model server with locking and saved in an IFC file.


EDMModelServerManager Connection: AaU-TCP (Server: edmserver21.jotne.com:4...
File Tools Upload Download Options Admin Help

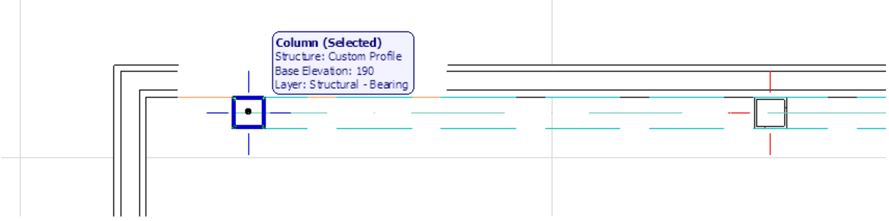
Connection: AaU-TCP Server: edmserver21.jotne.com:4520 User: superuser Group: sdai-group Model: DkNanoResearch.NanoBuildi

Figure 21 – Two columns, one beam, two walls and one slab objects are checked out

Part of the trace file from the model server shows the following messages:

```
Checking out IFCWALLSTANDARDCASE instance -1392426824:
Checking out IFCWALLSTANDARDCASE instance -1392427009:
Checking out IFCCOLUMN instance -1392415858:
Checking out IFCCOLUMN instance -1392415684:
Checking out IFCBEAM instance -1392415167:
Checking out IFCSLAB instance -1392414767:
Source instances #: 6
```

This file with the partial model is imported into ArchiCad.



Column (Selected)
Structure: Custom Profile
Base Elevation: 190
Layer: Structural - Bearing

Figure 22 – Partial model imported into ArchiCad. The ill placed column is selected.

With ArchiCad, the column is moved sidewise to the wall at the left side.

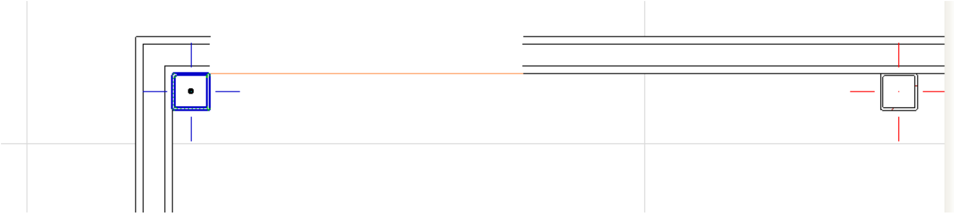


Figure 23 – The selected column is moved sidewise.

The beam is extended accordingly and the result can be shown in ArchiCad.

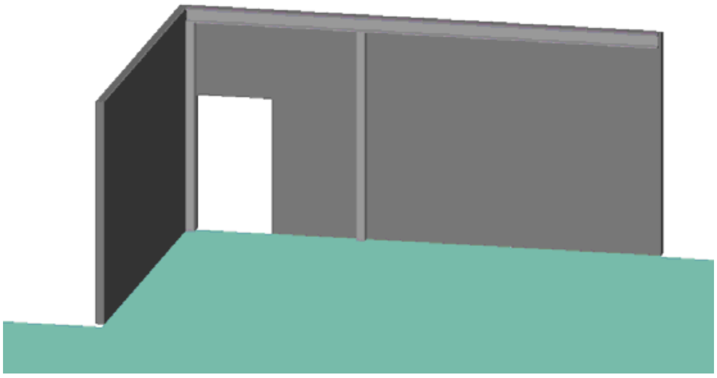


Figure 24 – The result of the changes made in ArchiCad

From ArchiCad, the partial model is exported to an IFC file and this file is then returned to the model server by a check-in operation to the same model from which it was checked out. By this operation, the objects form the partial model replaces the corresponding objects at the model server.

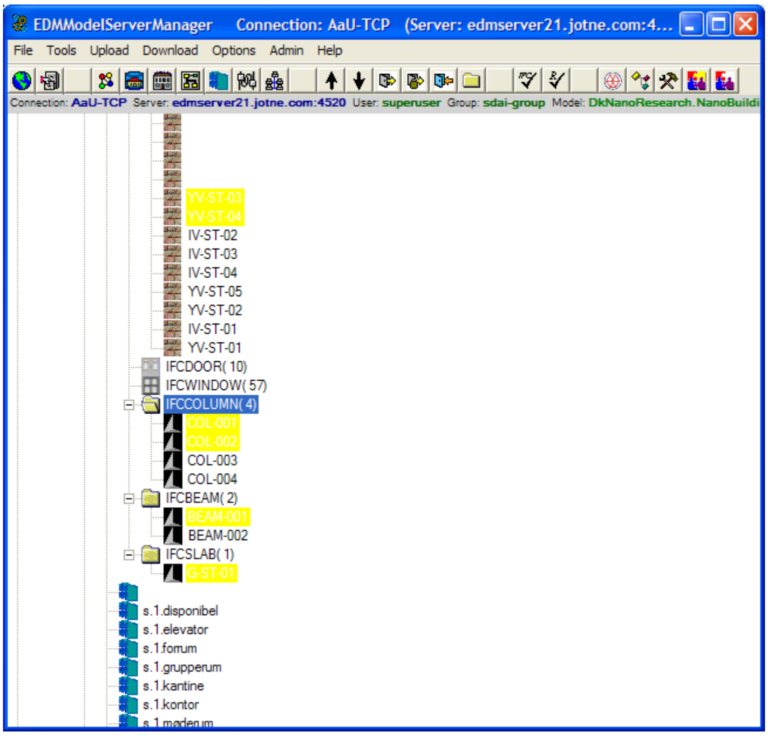


Figure 25 – The successful result of the check-in operation

After a successful check-in, the locks are released from the partial model in the model server.

From the model in the model server, the result can be visualised.

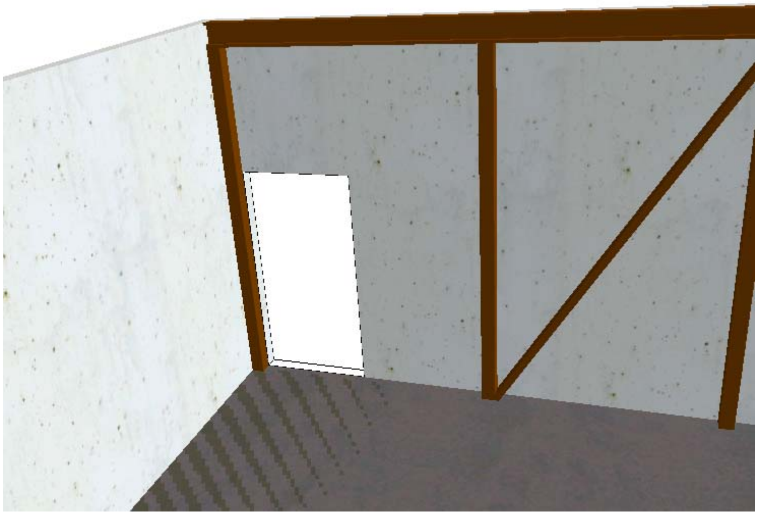


Figure 26 – The resulting model in the model server

6. Evaluation and Possible Enhancements

In section 5, it is demonstrated how collaboration between multiple parties of a building modelling process can take place by use of the EDM Model Server. In the demo, it is shown that the model server can store multiple versions of a model and how these models can be shared between multiple parties of a building construction project.

Full models can be uploaded and downloaded but, more importantly, it is demonstrated that work on partial models can also be supported by the model server. Different parts of the stored model can be selected and downloaded (checked out), modified by a modelling tool and reloaded (checked in) again. At check-in, the partial model is merged with the existing model in a special way so that re-entered objects replace the corresponding existing objects, new objects are added to the existing model and deleted objects are removed from the model.

In the following table, some evaluation statements are expressed.

Upload, download, check-in and check-out	These operations are performed via files, which are transmitted to and from the model server and imported to or exported from the modelling tools. For full models transmitted over the internet, this may give performance problems but, for partial models, it will normally work fine.
IFC interfaces	To enable the operations above to be performed correctly, the *IFC interfaces of the modelling tools* must work adequately. This means that nothing from a model or a partial model should be lost during the import-update-export steps with the modelling tool. Furthermore, when using the EDM Model Server a special custom property set must be handled correctly. In the demo, ArchiCAD was used and this tool is currently the only tool, which works adequately. A more detailed test of ArchiCAD interface has not been carried out.
Selection of model objects	As shown in the demo about partial models, *selection of model objects* are only made by means of the spatial structure hierarchy. For small models, where it is possible to get an overview over all the relevant objects, the current selection method is suitable. But for larger models, it is inappropriate. Other methods must be available and it should be possible to make selections based on a graphical visualisation.
Versioning	One of the features of model servers is *versioning*. With EDM, this is handled automatically, but only on model level. When a model with an identical name as an existing model is uploaded, a new version number is added to the name. Versioning on object level should be implemented [InPro 2008].
Object delete	As explained, the current check-in method operates with, i.e. objects, which are deleted by the modelling tool and thereby missing in the check-in file, is deleted in the model on the server. An alternative method would be to operate with explicit delete [InPro 2008], where objects may only be deleted by the MSM. If checked out objects are missing in the partial model at check-in, they are not deleted. With current standard of IFC interfaces, where there is a rather big risk that objects are lost in the update

	process, explicit delete would be an advantage because it will be easier to implement.
Access rights	Another feature of model servers is that it should be possible to grant different *access rights* to different users. This feature is partly implemented in EDM, but it has not been examined in this project.
Object locking	The locking facility represents the model server's ability to handle concurrent transactions and this makes very strict bindings on the modelling process. Building design consists of several parallel design processes and it is therefore most often inefficient to lock parts of the shared model.
Track and trace	Currently, EDM can provide tracking of transactions but only by means of a trace display at each i/o operation. Update history data are not added to the model.
Visualisation	EDM has integrated the Octaga model viewer, which make it possible to show selections of objects in a graphical view.
Report generation	A few standard reports are available for general use and it would be beneficial to have more reports available. There is no easy usable tool for creating ad hoc reports from models at the model server. Currently the only way is to develop scripts in the Express-X language.

7. Discussion

As previously described, this report focuses on changes in the way partners in building construction projects collaborate by use of building models and model servers based on the international standard Industry Foundation Classes (IFC). In this chapter, some issues are discussed, which are of importance to consider in the future development of model server supported building modelling.

7.1 Competences in collaboration

A few hundred years ago, most of the needed competences in connection with design of buildings were placed in one head, namely the Building Master's head. As knowledge about functionalities of the underlying building component systems increased, this knowledge was diffused to the separate more specialised brains of architects, engineers, etc. This also meant that the prerequisites for competence utilization fundamentally changed. Now each competence would use

- deep discipline domain knowledge usually unknown to other disciplines
- discipline oriented languages to some extent
- different representations for supporting (digital) building models
- slower knowledge transfer mechanisms (long transactions between different brains)

On the other hand the design team would compared to a single Building Master

- have access to a wider knowledge domain
- expand the design solution domain
- design solutions that to a higher degree met complex client requirements and needs
- have possibilities to produce creative and innovative solutions besides pure routine design solutions.

7.2 Collaboration in physical and virtual workspaces

Collaboration in the building process is undergoing big changes depending on introduction of new ICT supported collaboration tools. These tools support personal as well as team work in physical and virtual environments [Fischer 2002]. New effective ways for communication and collaboration through *multimedia interfaces* may be introduced supported by the *underlying models* formalising different building processes and building products under design, construction and use.

Collaboration and communication between design process participants can take place in real time face to face, over telephone, in virtual room settings or in asynchronous mode over e.g. email, annotated building models, and annotated drawings (red-lining).

Virtual Workspace, VW [Christiansson 2001], is considered the new design room designed to fit new and existing design routines. VW may well be a mixed reality environment. The VW will host all design partners from project start with different access and visibility (for persons and groups) in space and time to the project, and will promote building up shared values in projects. The VW thus acts as a communication space with project information support in adapted appearances. VW gives access to general and specific ICT-tools.

Collaboration takes place in a context that will put requirements on underlying models and user collaboration tools. Some important parameters are [Christiansson 2001] (see also Figure 27)

- *number* of participating persons
- Collaboration *subjects* such as design synthesis, analyses, simulation, design review, model annotations, planning, co-ordination, evaluation, purchase, learning, training;
- *Form of interaction* such as presentation, brainstorm, negotiation, consultation, discussion, decisions, documentation, sketching;
- Communication *information content* to support interaction; e.g. speech, sound, images, music, video, whisper, body language, 3D objects, control information;
- Meeting *spaces and room definitions*; physical, virtual, static, dynamic, mobile and combinations. (Intra personal workspaces, personal work spaces, team work spaces, linked spaces or one common space for all collaborators, spaces for personal and team annotations of models, etc.);
- Simultaneousness; *synchronous* and *asynchronous* meetings, time stamped activities;
- Collaboration *artifacts*

1. *Communication channels* e.g. displays, glasses, haptic devices, positioning devices (Haptics[10] is the science of applying touch (tactile) sensation and control to interaction with computer applications.)
2. *Control and access mechanisms* (function and form of ICT tools for search, navigation, time browsing, annotation, information storing and access, space connection and overlap, x-ray and see through, model and application sharing and handling etc.),
3. User applications and information *containers* (e.g. modelservers, Cad-systems, databases, data warehouses, simulation programs, planning programs, and other external resources).

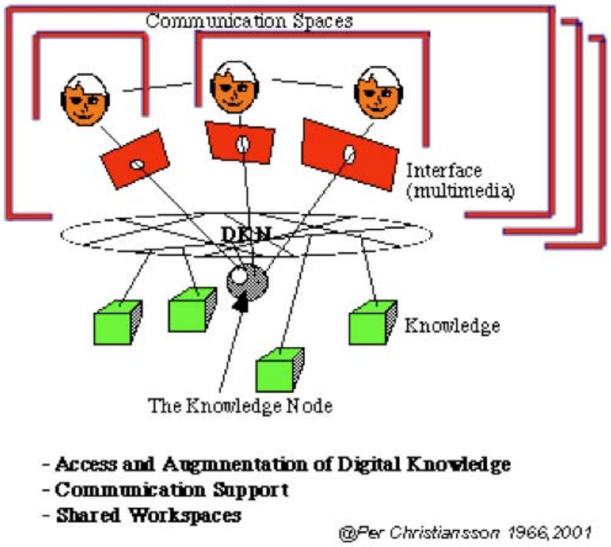


Figure 27 – Information and communication tools (ICT) support communication between persons in defined spaces [Christiansson 2001]

7.3 Collaboration on digital building models

In the report, a number of issues are raised concerning model redundancy, integrity, consistence, accessibility and selection/deletion of model objects in connection with use of model servers. Use of model servers will open up for more successful collaboration in design with efficient building models handling but also put new constraints on work coordination and models coordination. We list some crucial *activities* performed by man and supported by computer systems

- model redundancy check and reduction
- model integrity check and maintenance
- model consistency check and maintenance
- access to adapted model views

[10] http://whatis.techtarget.com/

- model versioning and secure storage
- models update and merging

It should be emphasized here that these activities might be more or less difficult to perform depending on degree of formalisation of building object and working routines. A relatively standardised building, stored in a parametric model, will be less open for inconsistencies partly depending on the fact that this model already has reached a rather detailed stage in its building component configuration.

Organisation of design, construction, operations and maintenance processes will be affected. Building model growth will be influenced on how, by whom and when buildings are documented and in what form the documentation is delivered. For example use of object based model servers is well in line with the vision of the Danish National Digital Construction program (Det Digitale Byggeri, DDB)[11] founded by The National Agency for Enterprise and Construction in Denmark. The public clients can since January 2007 put requirements on delivery of IFC based building models for use in for example early visualisations and digital delivery of information for the subsequent use in the O&M phase of the building life cycle. See also figure 28. The building model may well be delivered via a ProjectWeb containing metadata to point to e.g. IFC sub-models and documents such as 2D drawings and O&M instructions.

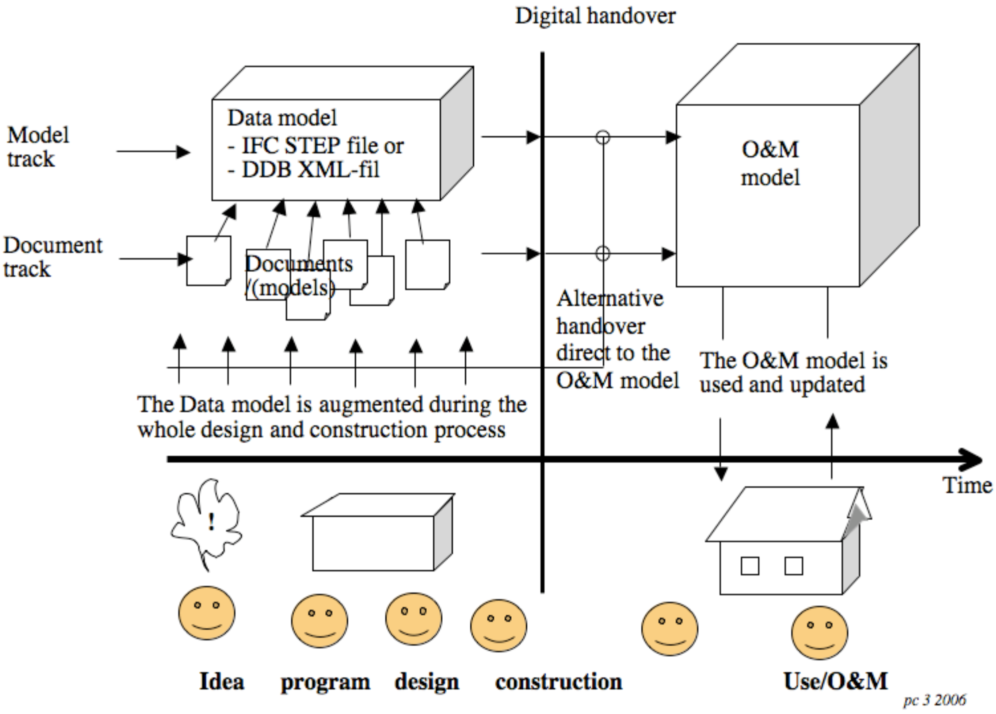


Figure 28 – Digital delivery [Sabroe 2006], [EBST 2007]

[11] www.detdigitalebyggeri.dk (Danish)

7.4 ICT supported building design modelling systems

A few typical usage scenarios for the use of ICT supported design tools and model storage tools can now formulated. In Figure 29, five cases for information flow has been sketched.

Cad systems are the main design tool today. Cad systems locally store building models in different representations and have different abilities to export and import models, or part of models, in predefined formats. One such format is IFC. In Figure 29, we have sketched Cad systems used by different disciplines like architect, structural engineers, and ventilation designers.

Today we often, due to inadequate computer resources, extract information to produce simplified lightweight models here called (Virtual Reality) VR-models. These are extracted to enable real time interaction in big building models e.g. during walk-through in visualization proposals or during design reviews.

The successes of merging discipline models are highly dependent on the relations between the sub-models extracted from the IFC server. If the extracted sub-models not have common components or component properties there are no problems. Though in most cases, they have at least some common properties or building components, for example insulation capacity on walls or a wall separating two work domains. In the later cases the models may be annotated to mark the changes (compare to red-lining of 2D drawings) or separate change records may be delivered to persons involved in model consolidations.

The 5 different scenarios of Figure 29 are

- A) Today's storage in Cad systems, where building models are developed and stored in Cad systems and transferred between Cad systems typically used by different disciplines

- B) The ideal case, where discipline models can be merged into the common IFC Building Model either direct (simultaneous work on he building model) or via model file transfer

- C) A possible situation of today, where building sub-models are extracted from he model server, checked and stored locally by e.g. Solibri modelchecker,

- D) A rare situation today, where even changes on simplified VR-models (often surface models) can be transferred back to discipline models in Cad systems and further to the IFC Model server for merging. VR models are made due to lack of computer resources and low network bandwidth to allow direct interactive work on large building models. These VR-models may typically be used for design brief and design review.

- E) Is the same as D) but updates on the VR-model has to be manually transferred from VR-model to discipline models.

Practical experiences on how to organise and carry through collaboration on digital building models are accumulated in practice and will contribute to the future implementation of tools for collaborative design using digital building models [NIBS 2007].

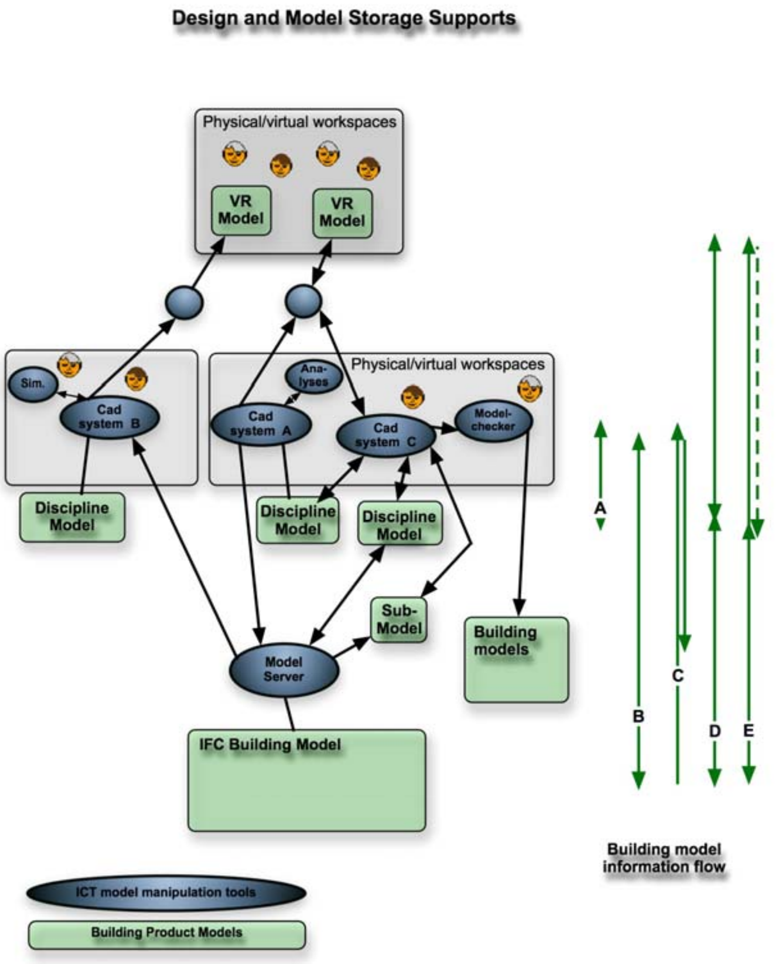


Figure 29 – Illustration of a number of different scenarios for use of ICT supported design tools.

8. Conclusion

This report deals with IFC based building model servers with the aim to evaluate the possibilities for practical use. The primary purpose of model servers is to provide support for collaboration in building modelling, i.e. that building models or partial building models can be stored and manipulated on model servers and so that different users can access the model separately. It is thereby expected that new collaboration scenarios can be developed and that many problems with current modelling scenarios can be solved.

Because IFC based building model servers can store any model, which is validly represented with IFC, it is obvious that model servers are ready for simple use (model publish), where models are uploaded and where users can be granted access to the models and download them when needed. This use of model servers may give some value to modelling partners but may be more important for other partners. However, much more support for selection of model objects, visualisation, analysis, simulation, reporting, etc. should be developed before sufficient benefit from this use of model servers can be perceived.

Another use of model servers has been the primary subject of this project. In that scenario, model servers are used to support collaboration on shared models. Instead of each partner creating one or more separate models, one shared model is created and all partners will work together on maintaining this model. By storing the shared model on a model server, all model designers should be able to perform the necessary update operations on the model at the model server. Possibly, modelling tools will be able to work directly with the model server but, currently, the currently most used method is to exchange files between the model server and the relevant software applications. The model server should provide basic features as model sharing, model protection, access control, transaction processing, model versioning, data extraction, etc. The primary part of the report is a presentation of a demo, which should illustrate how such a collaboration scenario would look like.

Based on the performed work with the EPM model server, a number of conclusions can be drawn.

- Creating users on the server and granting access rights to users works adequately but there are internationally no agreements on how this should be conveyed to objects of the building model.
- Upload and download of models is typically performed via the internet and, for large models, this may give performance problems.
- Support for selection of model objects, visualisation, analysis, simulation, reporting are mostly limited.
- Locking of objects on the model server is necessary in order to control concurrent update operations but it may become a serious binding on the modelling processes.
- The quality of IFC import/export interfaces of building modelling tools is generally insufficient. They must work correctly, i.e. IFC representations of building models must be interpreted correctly when imported to a tool and generated correctly when exported from a tool.
- It is possible to merge models on the server but only by addition (i.e. objects are simply added together in a model). There are currently very few? functionalities for merging multiple models to a single consistent model.
- The model server provides merging of partial models, i.e. when a partial model is extracted form the server (checked out) and modified by a modelling tool, it can be re-entered in the full model (checked in) and merged with the existing model objects. However, EPM has developed the merge operation with the assumption of implicit delete, which means that object missing in the checked in partial model will be deleted from the full model. This requires that no accidental los of objects must take place during the update operations and major constraints are set on the quality of IFC import/export interfaces of the modelling tools. Alternatively, if missing objects were not deleted but only explicitly marked objects were deleted, it would be much easier to get modelling tools to work adequately with model servers.

The concluding statements above clearly indicate that there are still a number of problems related to the use of model servers. Some problems relate to the model servers, some problems relate to building modelling tools and others relate to modelling collaboration in practice. These problems still represent serious limitations for the use of model servers, especially when modelling on shared models. However, the development of model servers has gained good momentum and it is becoming more evident that the import/export interfaces are the current primary obstacles for use of model servers in real building construction projects.

References

[Bakkmoen 2007]

Bakkmoen, Kjell Ivar: IFC-prosjekt Nye Ahus (Norwegen). C.F.Møller, v.1, december 2007.

[Chistiansson 2001]

Christiansson P: "Experiences from Using Internet Based Collaboration Tools". 'Konference om Arkitekturforskning og IT'. Proceedings Conference on Architectural Research and Information Technology. Nordic Association for Architectural Research. Aarhus School of Architecture 27.-29. april 2001. (pp. 103-112). http://it.civil.aau.dk/it/reports/r_aaa_2001.pdf

[EBST 2007]

EBST: "Vejledning til bygherren og rådgiveren i anvendelse af IKT" EBST 30.1.2007 (116 pp.).

[Erabuild 2008]

Review of the Development and Implementation of IFC compatible BIM. Erabuild 2008 http://www.ebst.dk/file/9498/Review%20of%20the%20Development%20and%20Implementation%20of%20IFC%20compatible%20BIM.pdf

[Fischer 2002]

Fischer, M.; Stone, M.; Liston, K.; Kunz, J.; Singhal, V.: Multi-stakeholder collaboration: The CIFE iRoom. In: Proceedings from CIB w78 conference, pp. 6-13, 2002, in Aahus, Denmark.

[Fox 2007]

Fox, Stephen; Hietanen, Jiri: Interorganizational use of building information models: potential for automational, informational and transformational effects. In: Construction Management and Economics, vol. 25 (March), pp. 289-296, 2007.

[InPro 2008]

InPro: Open Standards for Interoperability between Applications in Early Design, Deliverable 6, 2008. http://www.inpro-project.eu/docs/InPro_D6_SpecificationsForAnOpenICTPlatform.pdf

[Jørgensen 2007]

Jørgensen Kaj A., Skauge J: Building Models and Building Modelling, Aalborg University and Aarhus School of Architecture 2007 http://www.iprod.auc.dk/bygit/Web3B/index.htm

[NBS 1988]

NBS-DATA: "Information Technology in the Building Process. Development Trends in the USA 1988"". (Editor Per Christiansson). NBS-DATA, Nordiska Byggforskningsorganens Samarbetsgrupp 1988. (83 pp.) http://it.civil.aau.dk/it/reports/nbs_data_usa_1988.pdf

[NIBS 2007]

National Institute of Building Sciences: United States National Building Informatyion Modeling Standard – Overview, principles and methodologies. Version 1 – part 1, 2007.

[Plume 2007]

Plume, Jim; Mitchell, John: Collaborative design using a shared IFC building model – Learning from experience. Automation in Construction, vol. 16, no. 1, January 2007, pp. 28-36, Elsevier.

[Sabroe 2006]

Sabroe H, Johansen J, Fage N, Christensen L, Buchardt L, Emborg J , Christiansson P, Carlsen H, Jensen P A: Byggherrekrav - Digital Aflevering. Kravspecifikation - revision 2/final. Det Digitale Byggeri. Erhvervs- og byggestyrelsen. Marts 2006. (42 pp).
http://it.civil.aau.dk/it/reports/2006_03_kravspec_dacapo_final.pdf

[Senate Properties 2007]

Senate Properties´ BIM Requirements for Architectural Design, 2007.
http://www.senaatti.fi/tiedostot/BIM_Requirements_2007.pdf

[Wright 2003]

Wright R: "Building and Fire Research at NBS/NIST 1975-2000" NIST BSS 179, 2003.
http://www2.bfrl.nist.gov/info/bfrl_history/ (Chapter 8)

Appendix – User guide to EPM Server Manager

Kaj A. Jørgensen – John Mittchell

Industry Foundation Classes (IFC)

IFC is the international standard[12] for sharing building model data, which allows collaboration between all disciplines involved on a building project. This exchange of building information not only supports the intensive design and construction phases but also allows owners, clients, occupier, asset and facility managers to access appropriate data for their use, from the inception phase to operations and demolition.

When multiple AEC applications are IFC compliant, the data can be integrated in a single building model database – a model server.

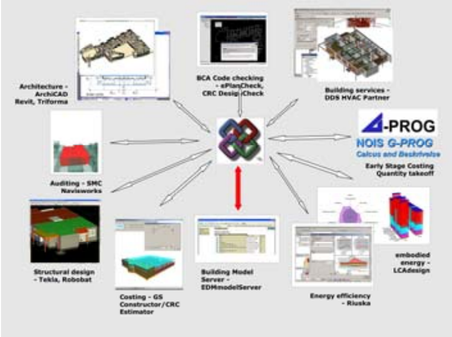

Architecture - ArchiCAD Revit, Triforma
BCA Code checking - ePlanCheck, CRC DesignCheck
Building services - DDS HVAC Partner
∆-PROG
NOIS G-PROG
Calcus and Beskrivelse
Early Stage Costing Quantity takeoff
Auditing - SMC Navisworks
embodied energy - LCAdesign
Structural design - Tekla, Robobat
Costing - GS Constructor/CRC Estimator
Building Model Server - EDMmodelServer
Energy efficiency - Riuska

***Figure 1**: Shared IFC model Environment*

EPM Model Server

The EPM model server comprises a building model database and applications, which enables building information to be shared using the IFC data model.

Model Server Functions

A model server, storing and manipulating building data in IFC format, potentially provides high level functions of coordination and project management, life cycle, and operational data management.

Key aspects of building model servers are

- Discipline (partial models/views), and ad hoc queries
- Merge function

[12] See http://www.iai-interoperability.org for more details

- Concurrent usage
- Team member's rights / security
- Speed/performance/integrity
- Version control
- Transaction processing
- Audit (user's roles, decisions, and issue tracking)
- Data protection (mirroring/backup)
- Storage

The server can operate over LAN networks, using either TCP or HTTP (web) protocols. Potential benefits to be gained from a building server environment are:

- Flexible multi-disciplinary management
- Database sharing
- Closer mapping between discipline/application data
- Large project scalability
- Wider & sufficient building structure/services engineering support for real projects
- Full life cycle support
- Ownership & security system
- Any selection of data
- Long transactions and auditing

It is anticipated a server solution will suit many project/client situations:

- Large project DBMS solution -> for large institutional, design, contracting and FM companies
- Single building type/focused solution -> for smaller custom integration e.g. domestic housing,
- Project hosting service -> as a model management server, such as design coordination, whole project data management = project management, complex alterations e.g. hospitals, laboratories, etc.

EPM Model Server Manager

The server provides an integrated source of model data that allows multiple users concurrent access. Users will typically be distributed geographically, not only on a project location (i.e. in a city) but also nationally and internationally.

The server provides users with the *IFCModelServerManager*, which runs on project team users’ computers over a LAN or internet connection.

An EDM Server *database* is a container for named *repositories*.

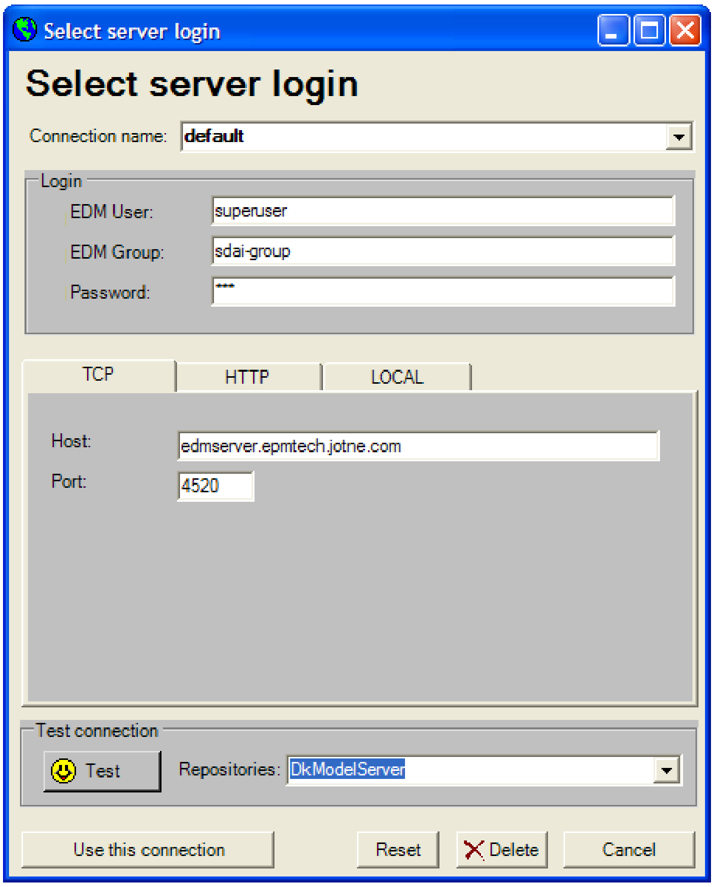
A user may create any number of repositories, provided such access right is granted (by the super user).

A data *repository* is a container for named *models*.

Connect and Login (project dependent)

When Model Server Manager is used for the first time, parameters for connection and login must be set. (Starting the manager with wrong parameters, will currently take very long time!!) The default connection name 'default' can be used or a new name can be created.

Two connect protocols can be used: TCP connection and HTTP connection.



Select server login

Connection name: default

EDM User: superuser

EDM Group: sdai-group

Password: ***

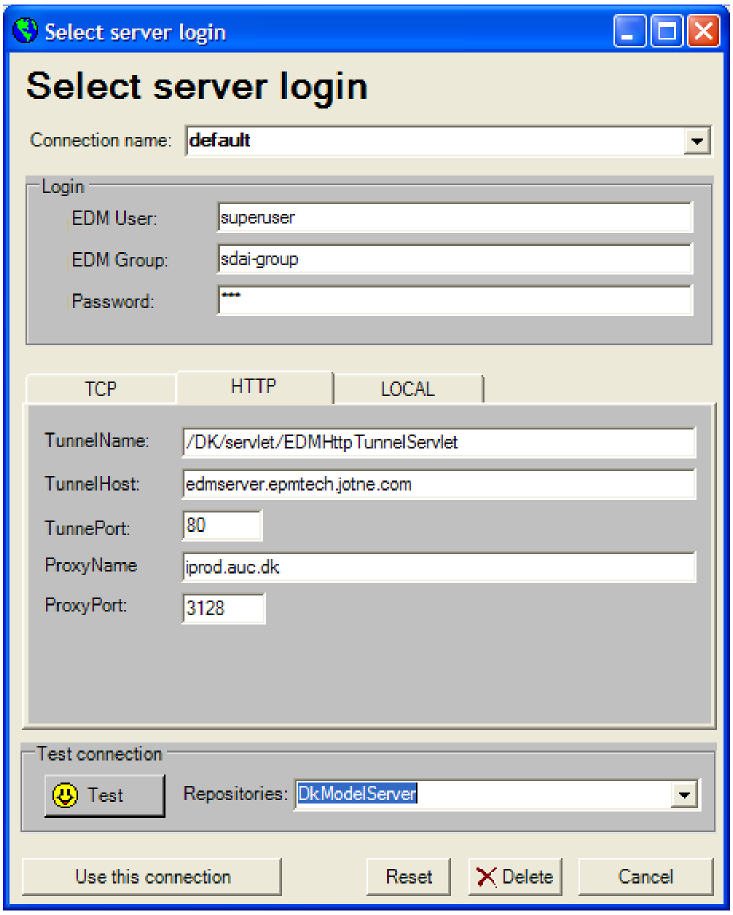
Host: edmserver.epmtech.jotne.com

Port: 4520

Repositories: DkModelServer

Connect and login window – TCP connection

Some local networks do not allow use of the TCP connection, but must connect via a proxy server. In this case the slightly slower HTTP connection can be used.



Select server login

Connection name: default

Login

EDM User: superuser

EDM Group: sdai-group

Password: ***

TunnelName: /DK/servlet/EDMHttpTunnelServlet

TunnelHost: edmserver.epmtech.jotne.com

TunnePort: 80

ProxyName iprod.auc.dk

ProxyPort: 3128

Test connection

Repositories: DkModelServer

Connect and login window – HTTP connection

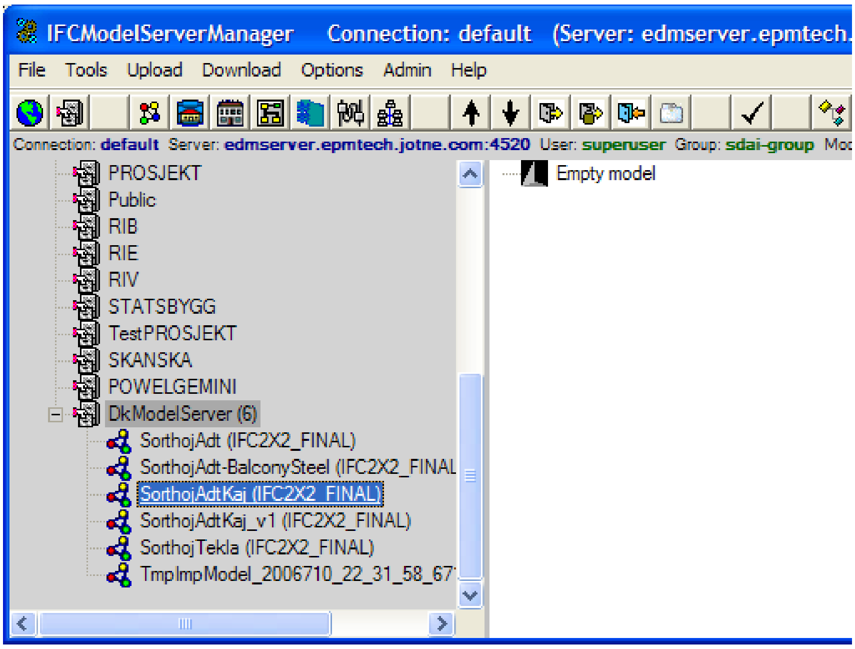
The same parameters must be used in HTTP.

When these parameters are entered, the connection can be tested with the 'Test' button. If the connection is approved, a smiley is shown. Otherwise, a sour smiley is shown.

Finally the 'Use this connection' button must be pressed in order to login to the model server.

Select Repository and Building Model

If the login is performed successfully the previously used model on this connection will be shown in the IFC browser. To see list of repositories, press the second left icon or use menu: File | Repository.



IFCModelServerManager Connection: default (Server: edmserver.epmtech.

Connection: **default** Server: **edmserver.epmtech.jotne.com:4520** User: **superuser** Group: **sdai-group** Mo

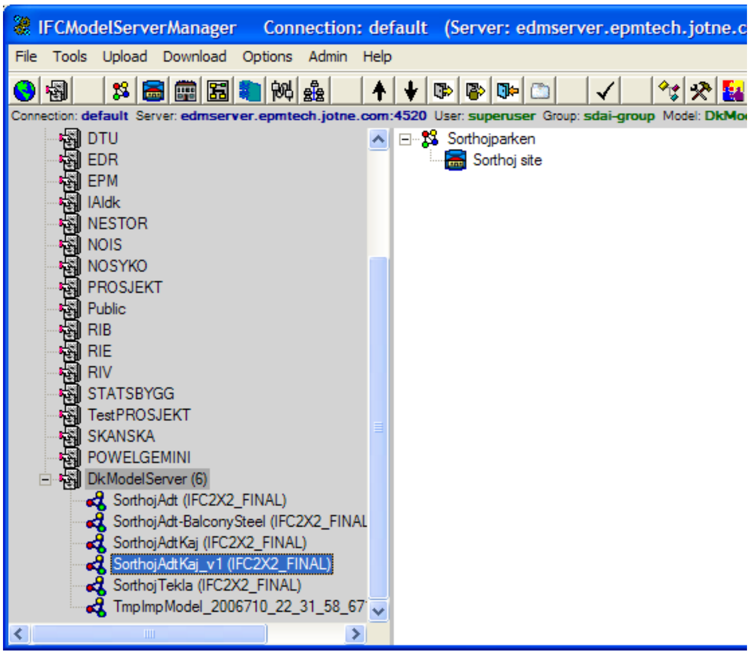
PROSJEKT
Public
RIB
RIE
RIV
STATSBYGG
TestPROSJEKT
SKANSKA
POWELGEMINI
DkModelServer (6)
SorthojAdt (IFC2X2_FINAL)
SorthojAdt-BalconySteel (IFC2X2_FINAL
SorthojAdtKaj (IFC2X2_FINAL)
SorthojAdtKaj_v1 (IFC2X2_FINAL)
SorthojTekla (IFC2X2_FINAL)
TmpImpModel_2006710_22_31_58_67

Empty model

Repository and model selection window

Double-click on the repository to see the contained models and double-click on a model to select it.

If the model is not empty, the IFC project name is shown in the window pane to the right.


IFCModelServerManager Connection: default (Server: edmserver.epmtech.jotne.c
File Tools Upload Download Options Admin Help

Connection: default Server: edmserver.epmtech.jotne.com:4520 User: superuser Group: sdai-group Model: DkMo

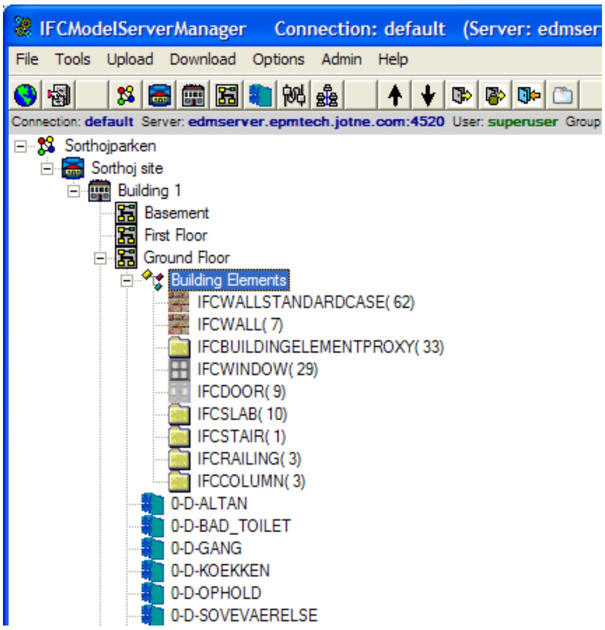
```
DTU
EDR
EPM
IAldk
NESTOR
NOIS
NOSYKO
PROSJEKT
Public
RIB
RIE
RIV
STATSBYGG
TestPROSJEKT
SKANSKA
POWELGEMINI
DkModelServer (6)
    SorthojAdt (IFC2X2_FINAL)
    SorthojAdt-BalconySteel (IFC2X2_FINAL
    SorthojAdtKaj (IFC2X2_FINAL)
    SorthojAdtKaj_v1 (IFC2X2_FINAL)
    SorthojTekla (IFC2X2_FINAL)
    TmpImpModel_2006710_22_31_58_67
```

```
Sorthojparken
    Sorthoj site
```

A non-empty model is selected

Navigate Through an IFC Building Model

The spatial structure of the building model can be exploded by subsequently double-clicking on IFC project, IFC site, IFC Building and IFC storey. For each storey, one node is shown with the IFC Building Elements and, below, the IFC space objects are shown. A double-click on the Building Elements node shows a list of the different types of building elements.

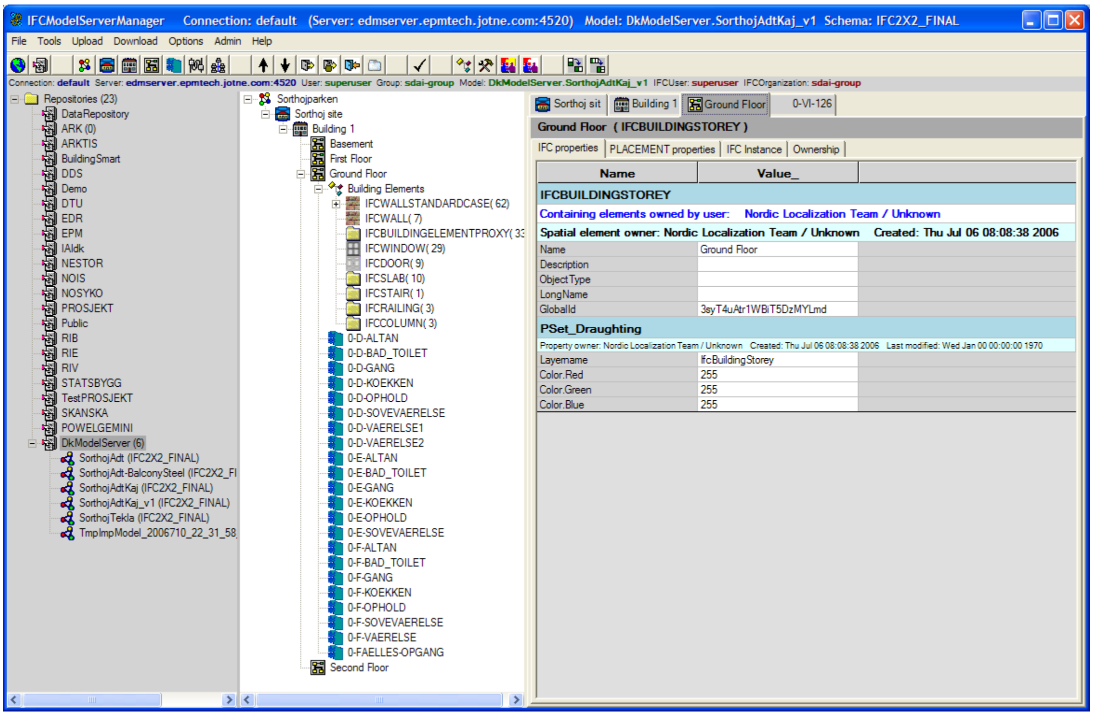

IFCModelServerManager Connection: default (Server: edmser
File Tools Upload Download Options Admin Help

Connection: default Server: edmserver.epmtech.jotne.com:4520 User: superuser Group

```
Sorthojparken
    Sorthoj site
        Building 1
            Basement
            First Floor
            Ground Floor
                Building Elements
                    IFCWALLSTANDARDCASE( 62)
                    IFCWALL( 7)
                    IFCBUILDINGELEMENTPROXY( 33)
                    IFCWINDOW( 29)
                    IFCDOOR( 9)
                    IFCSLAB( 10)
                    IFCSTAIR( 1)
                    IFCRAILING( 3)
                    IFCCOLUMN( 3)
                0-D-ALTAN
                0-D-BAD_TOILET
                0-D-GANG
                0-D-KOEKKEN
                0-D-OPHOLD
                0-D-SOVEVAERELSE
```

Building elements in the spatial structure

The manager is able to show the details about each object in a third window pane.



Connection: default Server: edmserver.epmtech.jotne.com:4520 User: superuser Group: sdai-group Model: DkModelServer.SorthojAdtKaj_v1 IFCUser: superuser IFCOrganization: sdai-group

Ground Floor (IFCBUILDINGSTOREY)

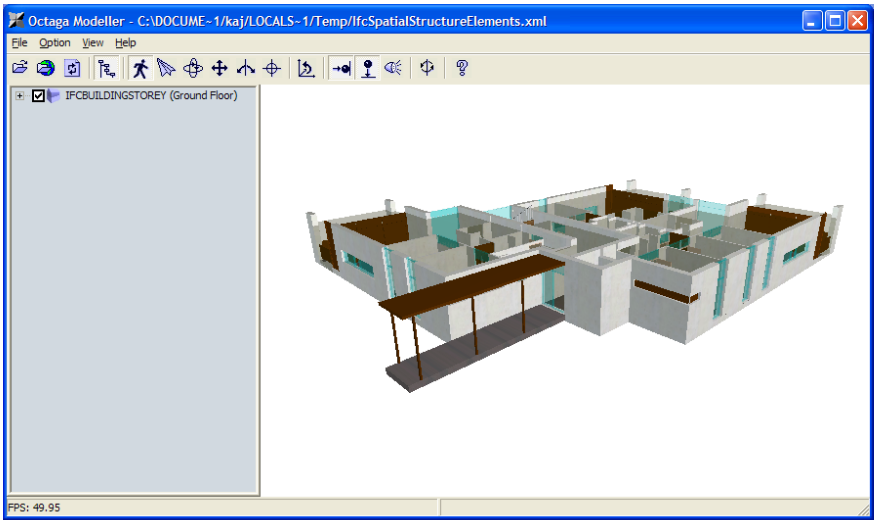
Name	Value_
IFCBUILDINGSTOREY	
Containing elements owned by user: Nordic Localization Team / Unknown	
Spatial element owner: Nordic Localization Team / Unknown Created: Thu Jul 06 08:08:38 2006	
Name	Ground Floor
Description	
ObjectType	
LongName	
GlobalId	3syT4uAtr1WBiT5DzMYLmd
PSet_Draughting	
Property owner: Nordic Localization Team / Unknown Created: Thu Jul 06 08:08:38 2006 Last modified: Wed Jan 00 00:00:00 1970	
Layername	IfcBuildingStorey
Color.Red	255
Color.Green	255
Color.Blue	255

Details about an IFC storey is shown in a separate window pane

(The use of the shown attributes is not explained.)

Use Octaga Model Viewer

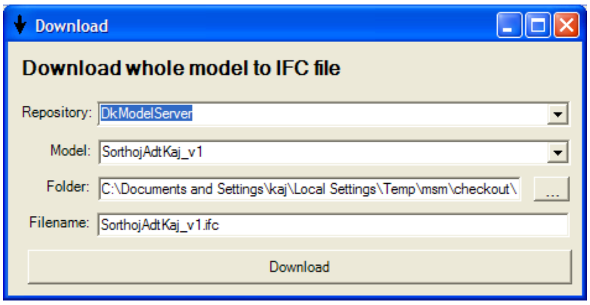
When an object is selected, e.g. IFC building, IFC storey or IFC Building Element, the selected objects can be shown in the Octaga Modeller IFC Viewer. Use the menu: Tools | Octaga Viewer


Octaga Modeller - C:\DOCUME~1/kaj/LOCALS~1/Temp/IfcSpatialStructureElements.xml
File
Option
View
Help
IFCBUILDINGSTOREY (Ground Floor)
FPS: 49.95

A selected IFC storey is shown in Octaga Model Viewer

Download Whole Model to IFC File

To start download a model to an IFC file, press the down arrow icon or select the menu Download | Download whole model as IFC file.



Download whole model to IFC file

Repository: DkModelServer

Model: SorthojAdtKaj_v1

Folder: C:\Documents and Settings\kaj\Local Settings\Temp\msm\checkout\

Filename: SorthojAdtKaj_v1.ifc

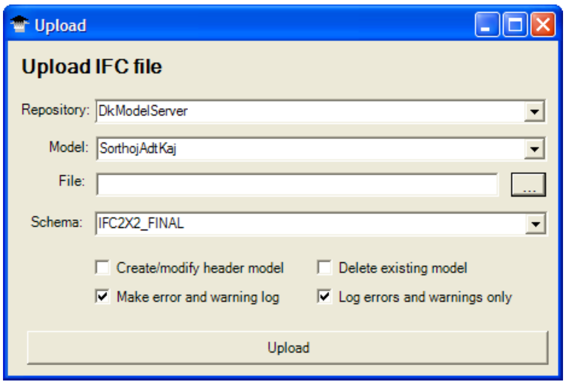
Download window

Select model, folder and file name.

It is also possible to drag the model node from the repository-model tree view to any windows application taking a file-drop. For instance to the desktop or a Windows File Explorer window. Keep the mouse in the position where you want the drop to occur until the drop is finished. An IFC file with the same name as the model will be created in the drop position.

Upload IFC File

To start upload a model from an IFC file, press the up arrow icon or select the menu Upload | Upload IFC file.



Upload IFC file

Repository: DkModelServer

Model: SorthojAdtKaj

File:

Schema: IFC2X2_FINAL

☐ Create/modify header model ☐ Delete existing model

☑ Make error and warning log ☑ Log errors and warnings only

Upload window

Select a model name or give the model a new name. If an existing model name is selected, a new model with same name but extended with a version number is created. Write or choose a file and select the IFC2X2_FINAL schema. The upload may take some time and, currently, there are no indications about the time, e.g. hour glass or similar.

If errors occur during upload, an error report window will open.

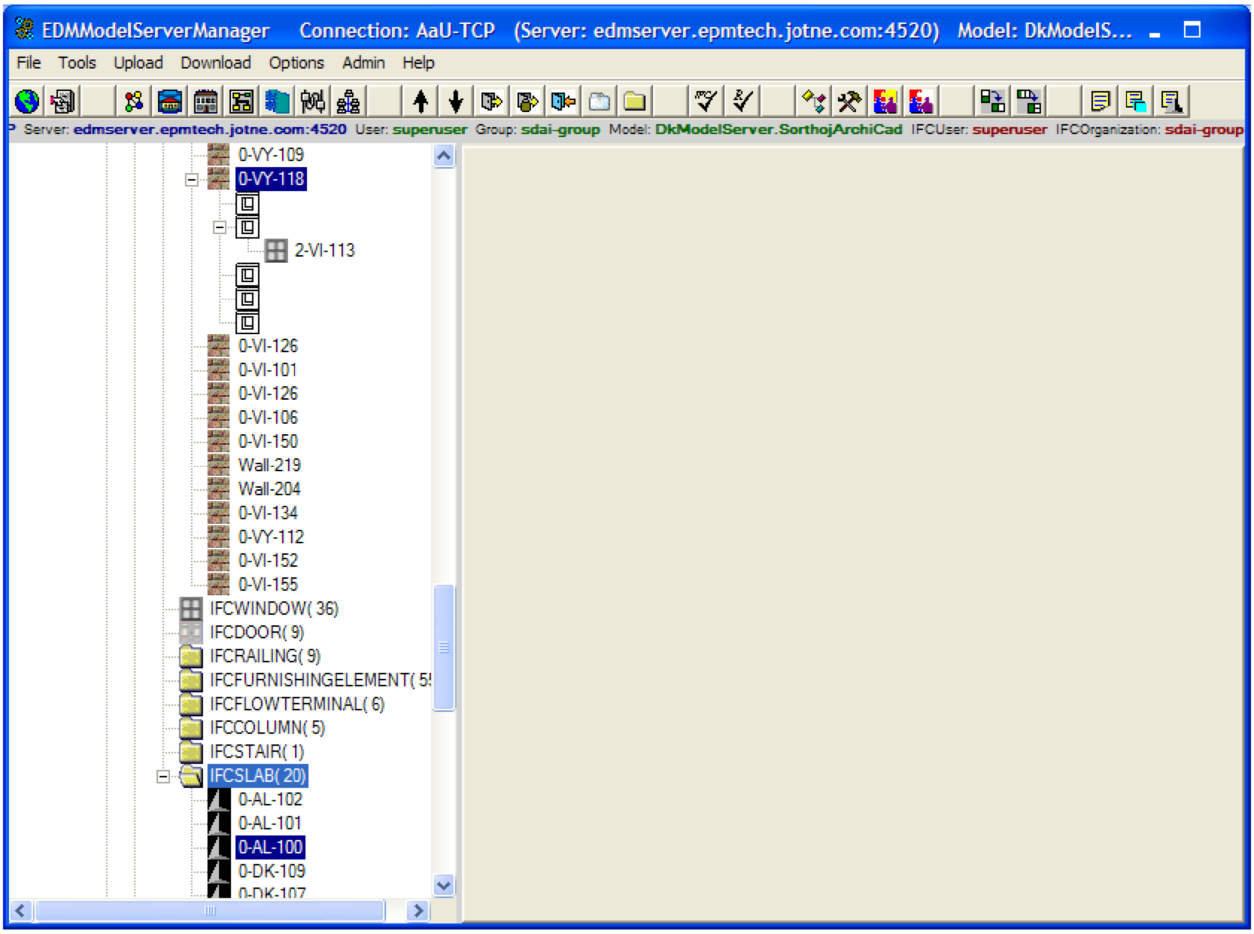
Upload can also be done with drag-and-drop from a Windows File Explorer window (desktop or any application creating a file-drop operation) to the repository-model tree view. A new model will be created with the same name as the filename possibly extended with a version number if the name already exists.

Check Out Selected Objects

The complete model or any partial model can be checked out by use of the check-out buttons or by drag-and-drop. There are two different check-out operations: with and without locking. Locking of objects means that only the user, which performed the check-out, can afterwards check-in the objects again or release the locks. A folder should be prepared in Windows Explorer.

Note: If a trace report should be generated, this must be marked in the 'Options' menu: 'Debug trace'.

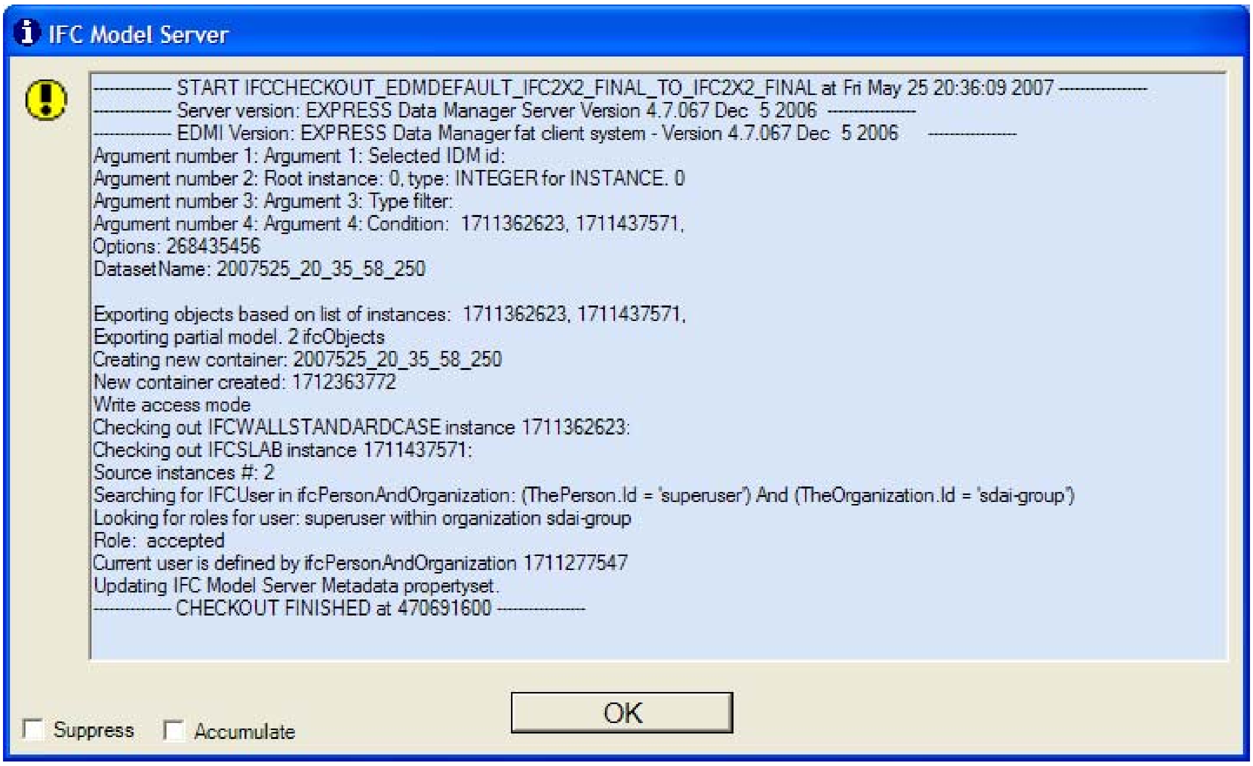
In the MSM tree-view, objects can be marked by holding down the <ctrl> key. In order to check-out the objects with locking, keep holding the key down, drag the objects to the folder and drop the selection in the prepared folder. As the IFC file will be named as the last object marked, it is convenient finally to mark an additional node for file naming (light blue), e.g. a parent node of the objects, IFCSLAB.


EDMModelServerManager
Connection: AaU-TCP (Server: edmserver.epmtech.jotne.com:4520) Model: DkModelS...
File
Tools
Upload
Download
Options
Admin
Help

Server: edmserver.epmtech.jotne.com:4520 User: superuser Group: sdai-group Model: DkModelServer.SorthojArchiCad IFCUser: superuser IFCOrganization: sdai-group

Dark blue objects are marked for check out and a light blue is marked for file naming

A possible trace report is shown in a separate window:



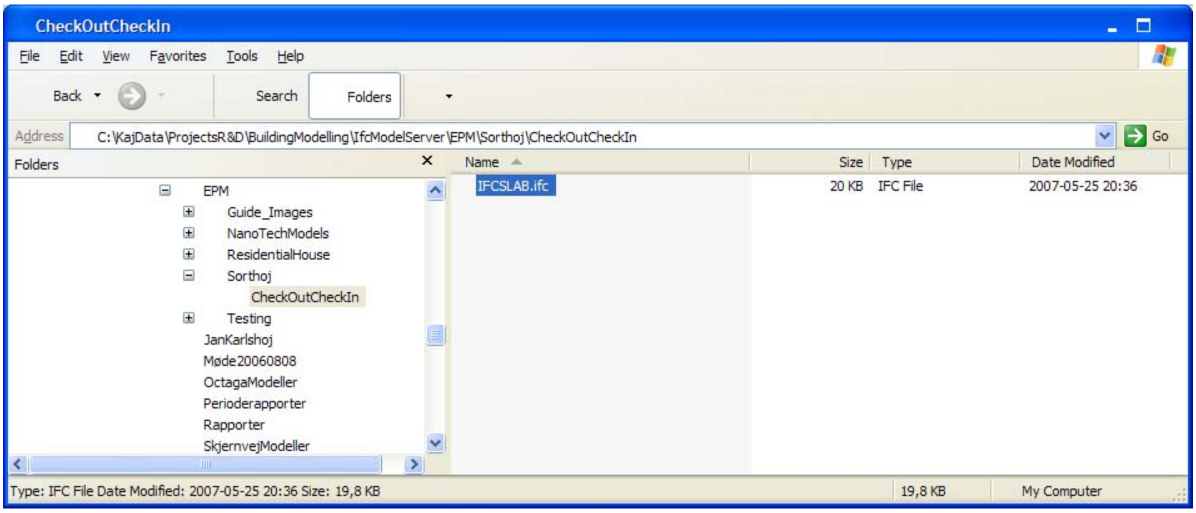
```
------------- START IFCCHECKOUT_EDMDEFAULT_IFC2X2_FINAL_TO_IFC2X2_FINAL at Fri May 25 20:36:09 2007 ---------------
------------- Server version: EXPRESS Data Manager Server Version 4.7.067 Dec  5 2006 ---------------
------------- EDMI Version: EXPRESS Data Manager fat client system - Version 4.7.067 Dec  5 2006     ---------------
Argument number 1: Argument 1: Selected IDM id:
Argument number 2: Root instance: 0, type: INTEGER for INSTANCE. 0
Argument number 3: Argument 3: Type filter:
Argument number 4: Argument 4: Condition:  1711362623, 1711437571,
Options: 268435456
DatasetName: 2007525_20_35_58_250

Exporting objects based on list of instances:  1711362623, 1711437571,
Exporting partial model. 2 ifcObjects
Creating new container: 2007525_20_35_58_250
New container created: 1712363772
Write access mode
Checking out IFCWALLSTANDARDCASE instance 1711362623:
Checking out IFCSLAB instance 1711437571:
Source instances #: 2
Searching for IFCUser in ifcPersonAndOrganization: (ThePerson.Id = 'superuser') And (TheOrganization.Id = 'sdai-group')
Looking for roles for user: superuser within organization sdai-group
Role:  accepted
Current user is defined by ifcPersonAndOrganization 1711277547
Updating IFC Model Server Metadata propertyset.
------------- CHECKOUT FINISHED at 470691600 ---------------
```

□ Suppress □ Accumulate

Trace report from a check-out of selected objects

The report shows:

- DatasetName – the name of the dataset at the server, which holds data about the check-out. Data about this dataset are included in the IFC file as a custom propertyset. It is important that this propertyset is preserved by the modelling application for later check-in.
- A list of the checked-out objects – in this case two.
- The name and role of the user. These values are added to the check-out file as an IfcPerson object, an IfcOrganization object and an IfcPersonAndOrganization object.



Address: C:\KajData\ProjectsR&D\BuildingModelling\IfcModelServer\EPM\Sorthoj\CheckOutCheckIn

IFCSLAB.ifc 20 KB IFC File 2007-05-25 20:36

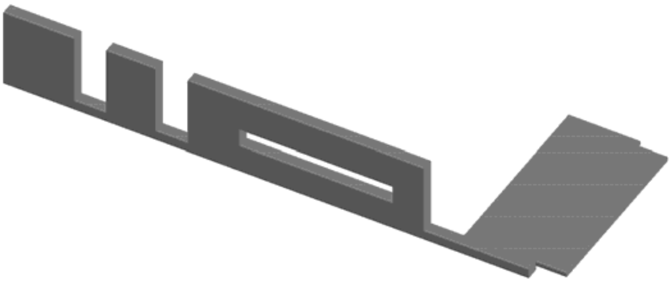
Type: IFC File Date Modified: 2007-05-25 20:36 Size: 19,8 KB 19,8 KB

The IFC file with checked-out objects

The objects, which have been checked out, are now locked. This can be verified by refreshing the tree-view.

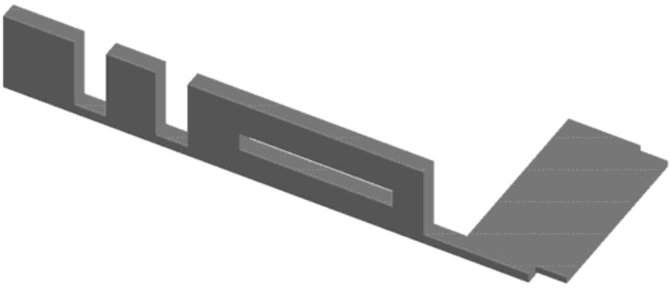
Modification of the Selected Objects

The file is imported into AC 10 with which some modifications are made.



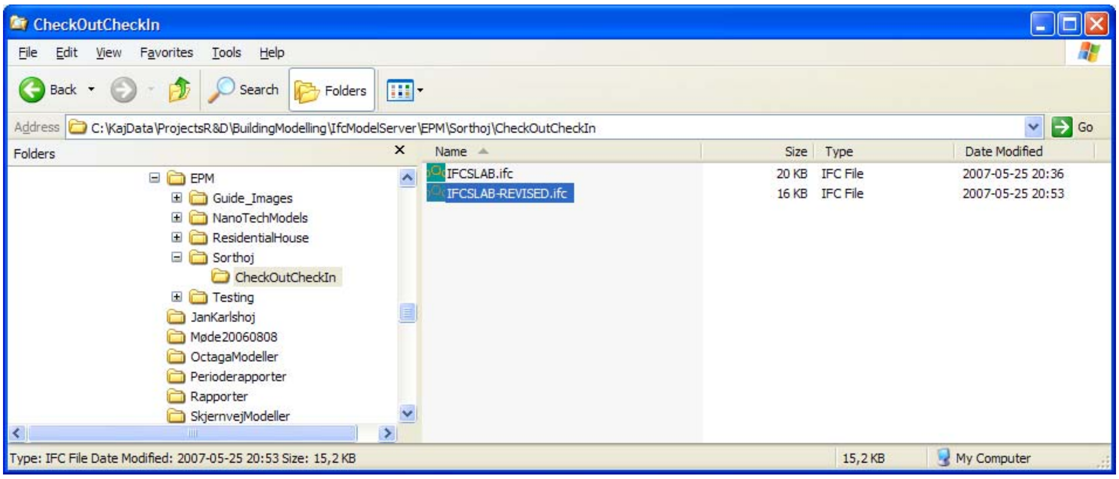
Objects imported into AC 10

In this example, some minor changes are made: the names of the two objects are changed by adding '***' to the names the thickness of the wall and the slab is changed.



Objects in AC 10 after modifications – thickness of wall and slab

The file with the revised partial model is exported from AC.



Address: C:\KajData\ProjectsR&D\BuildingModelling\IfcModelServer\EPM\Sorthoj\CheckOutCheckIn

Name	Size	Type	Date Modified
IFCSLAB.ifc	20 KB	IFC File	2007-05-25 20:36
IFCSLAB-REVISED.ifc	16 KB	IFC File	2007-05-25 20:53

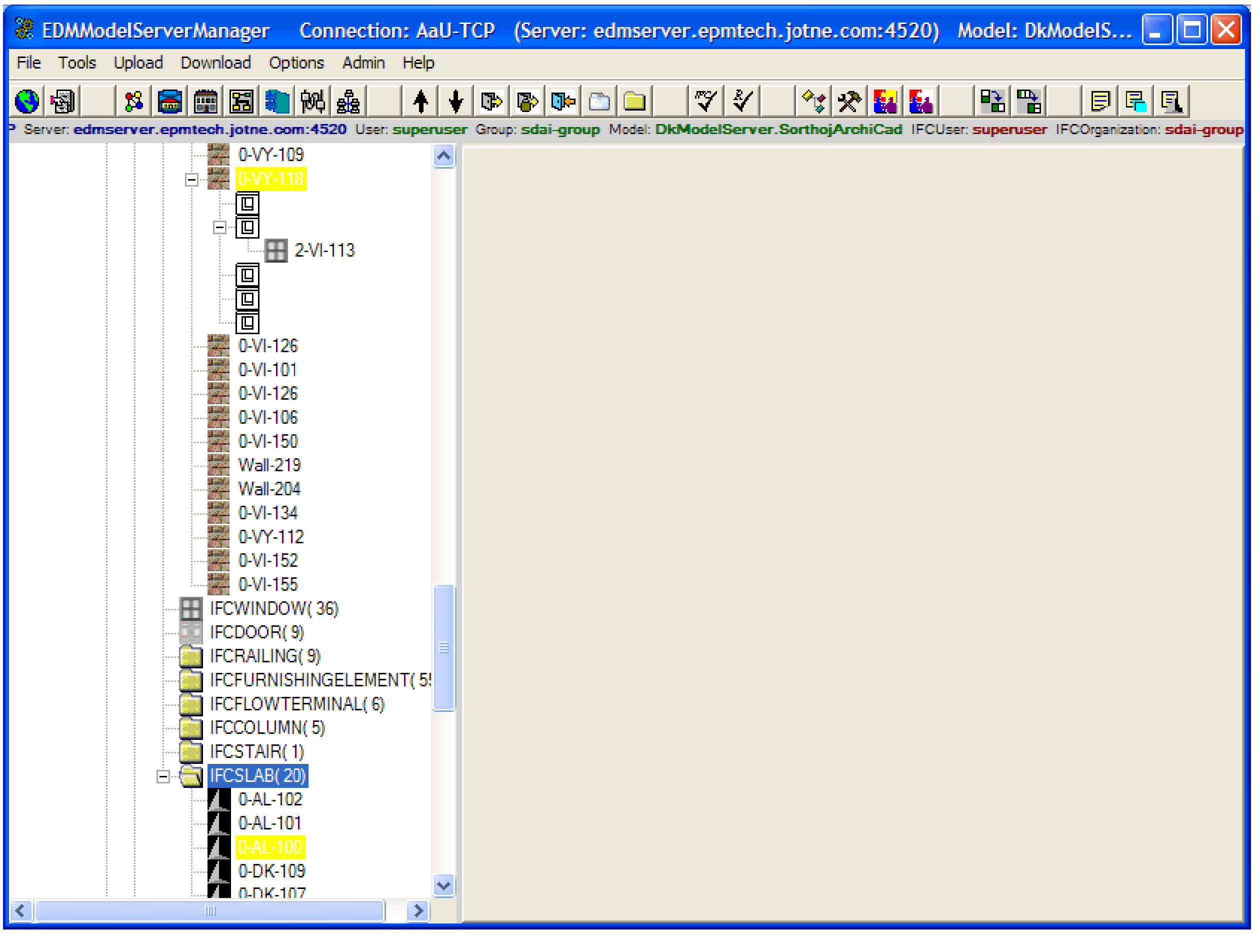
Type: IFC File Date Modified: 2007-05-25 20:53 Size: 15,2 KB 15,2 KB

IFC file with the revised partial model

Check-In from IFC file

Objects can be checked in from an IFC file and either added to a model or used for update of existing objects. In order to update, the existing objects must be locked in advance.

The easiest way to perform check in is by drag-and-drop similar to check out. A file can be selected in Windows Explorer and dragged into the tree-view. When the cursor passes over this window, the background will turn to light brown and, after drop, it will temporarily turn to light blue. If no errors occur, the result will look like this.



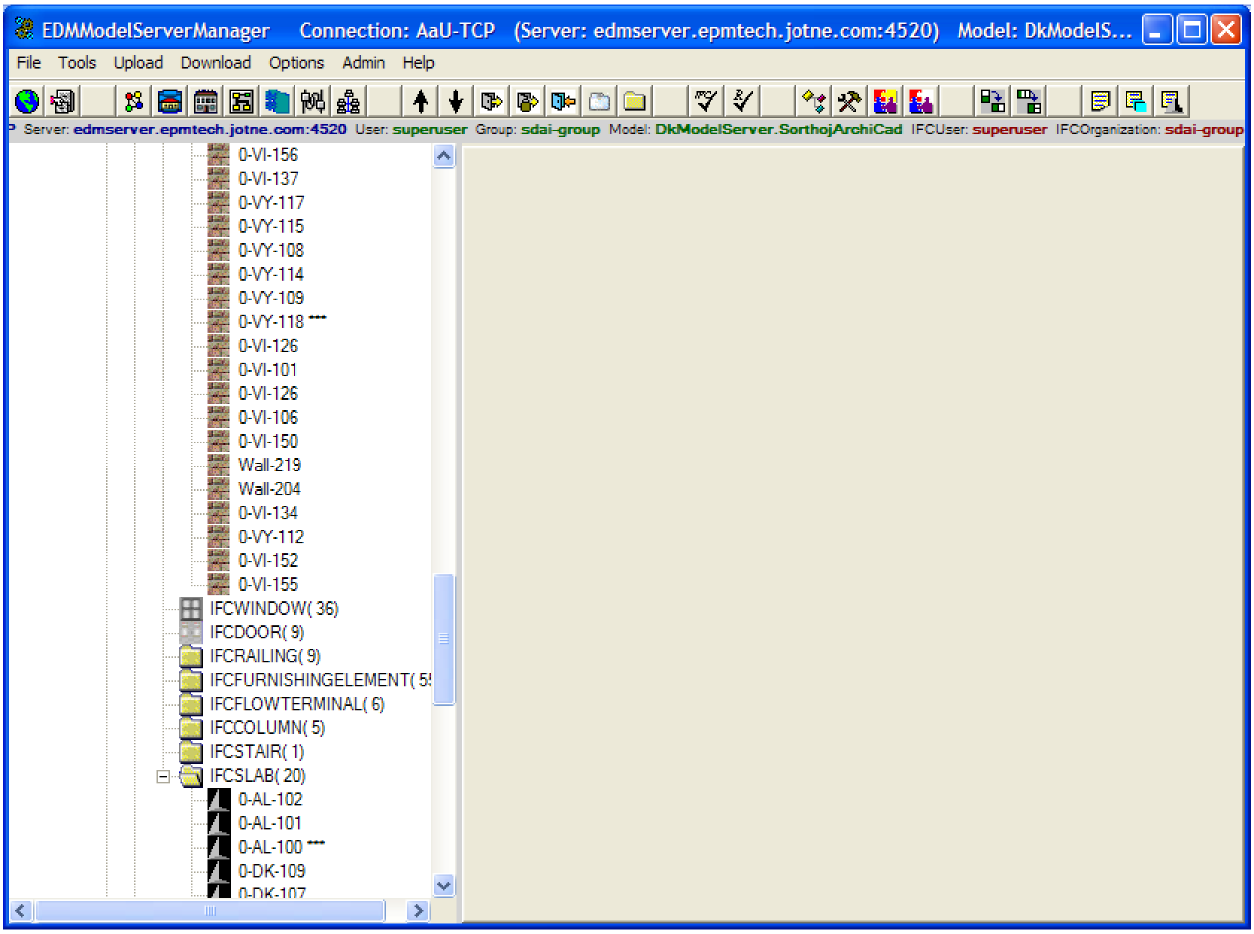
P Server: edmserver.epmtech.jotne.com:4520 User: superuser Group: sdai-group Model: DkModelServer.SorthojArchiCad IFCUser: superuser IFCOrganization: sdai-group

MSM presents a successful result in the tree-view in this way

The tree-view after check-in shows the checked-in objects in yellow. The grey objects are the objects in the spatial structure, which are included in the check-out file in order to be able to perform correct check-in.

Note: In this example, these objects were not marked for check-out because the model on the server is owned by another user. Therefore, they are not altered even though they may be changed in the check-in file.

After refreshing the tree-view, the changes made in the two selected objects are shown and the locks are released:



All objects are updated and locks are released. Note the new names of the objects.

At check-in, a trace report can also be generated with various messages about the check-in process. The first part of the file contains:

```
Target respoitory: DkModelServer, Target model: SorthojArchiCad
Query function IFCEditFunctions.MovePartialModelToModel: Start: Fri May 25 20:56:19 2007
IFCEditFunctions:MovePartialModelToModel(
    Extract = '',
    FromModel = 'TmpImpModel_2007525_20_56_5_890',
    ToModel = 'SorthojArchiCad',
    NewOwner = 0,
    OwnerType = '' )
Searching for IFCUser in ifcPersonAndOrganization : (ThePerson.Id = 'superuser') And (TheOrganization.Id = 'sdai-group')
Looking for roles for user: superuser within organization sdai-group
Role:  accepted
Current user is defined by ifcPersonAndOrganization 1711277547
Looking for dataset name in "IFC Model Server Metadata propertyset" of the ifcProject instance matching receieving models ifcProject - GlobalId.
Same project in source and target dataset. Project name: Sorthojparken, GlobalId: 2p0WZ4fsb8ZxqEYlg4yv8R
Dataset name: 2007525_20_35_58_250
Matching dataset container 2007525_20_35_58_250 found.
Locked container. Number of locked instances is 210
Merge all objects from the source
```

Note that the 'Target repository', the 'Target model', the 'ThePerson.Id' and the 'TheOrganization.Id' values are found. Note also that the 'IFC Model Server Metadata propertyset' is found with data about the check-out dataset.

Further in the trace report, it is listed that the building objects of the model is found and handled:

```
IFCSLAB, Name: 0-AL-100 ***, GlobalId: 0tW7a2$sz9zfl2zlXkGInX
  EXISTING OBJECT
    Replacing data of IFCSLAB instance
       Replacing relative placement
         Replacing representation
...

IFCWALLSTANDARDCASE, Name: 0-VY-118 ***, GlobalId: 23uUzWB$90x8zu$ki$4Rer
  EXISTING OBJECT
    Replacing data of IFCWALLSTANDARDCASE instance
       Replacing relative placement
         Replacing representation
```

If errors occur, they are also reported.

Annex – Model Server Usage Demo

Jørn Skauge

The purpose of this test has been to make a demo simulation of the situation where information from the engineer's construction and HVAC models are placed / merged into the architect model. The test focused on a few objects situated on the buildings ground floor. These objects were some exterior walls, a slab, some columns and a small steel construction consisting of columns and beam as well as some HVAC objects such as radiator and pipes.

1. Examination of the Architect Model and the Engineer Models

The architect model originally modelled in Architectural Desktop (ADT) had been saved in IFC2x2 format uploaded and placed on the DkNanoArkitekt repository on the model server. The architect model in IFC 2x2 format was downloaded and then opened in ArchiCAD for examinations. The original ADT model seems through this process to have been imported into ArchiCAD correctly with no faults. (fig1)

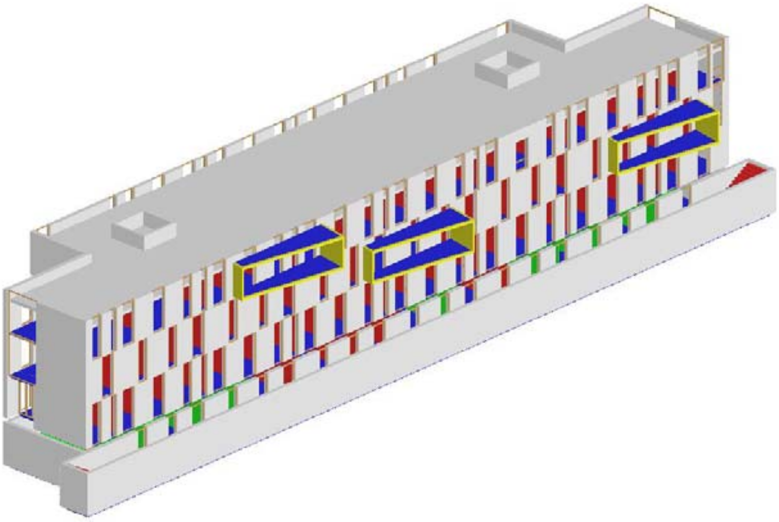


Figure 1. The architect model in ArchiCAD10 from ADT imported via IFC2x2 format.

The engineer's construction model originally modelled in Tekla had been saved in IFC2x2 format uploaded and placed on the DkNanoIngeniør repository on the model server. The engineers Tekla model in IFC 2x2 format was then downloaded and now opened in ArchiCAD for examination. The original Tekla model seems through this process to have generated some faults. The following examination of the ArchiCAD model showed that the material layers of the exterior walls did not match with the real dimensions originally modelled in Tekla. The exterior walls showed a total dimension of 495 mm, consisting from the outside of a concrete layer of 170 mm, an insulation space layer of 155 mm and a concrete wall layer of 170 mm. The real exterior wall dimension as they were modelled in Tekla is 450 mm consisting from the outside of a concrete layer of 90 mm, an insulation space layer of 200 mm and a concrete wall layer of 160 mm. The dimensions of the Slab had also changed from the thickness of 270 mm modelled in Tekla to 170 mm in the ArchiCAD file (fig 2).

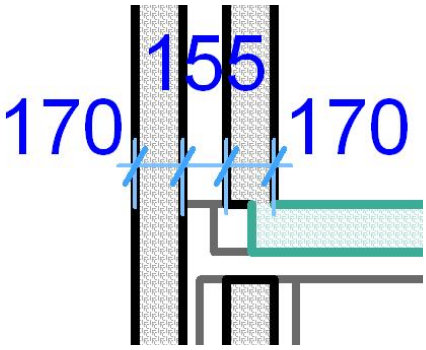
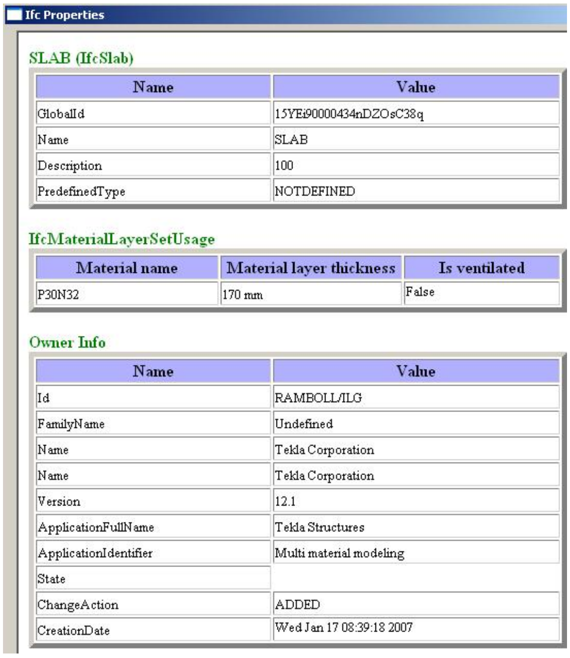


Figure 2. The dimensions of the exterior walls as they appeared after download and import into ArchiCAD10

It is not possible to tell how these faults have arisen but they might have happened through the process of exporting the Tekla file into the IFC 2x2 format. A check of this ifc2x2 model in the DDS Viewer shows a Wall layer thickness of 170 mm. In terms of the Floor Slabs they appear to be of a Floor Slab thickness of 170 mm where they were modelled in the original Tekla model to have a thickness of 270 mm. (fig. 3) In terms of columns and beams it appears that the dimensions of the columns (100 x 100) mm and the beam of (100 x 200) mm were correct. Fault and correct dimensions as they appear in the DDS Viewer is kept the same after importing the ifc2x2 file into ArchiCAD.



SLAB (IfcSlab)

Name	Value
GlobalId	15YEi90000434nDZOsC38q
Name	SLAB
Description	100
PredefinedType	NOTDEFINED

IfcMaterialLayerSetUsage

Material name	Material layer thickness	Is ventilated
P30N32	170 mm	False

Owner Info

Name	Value
Id	RAMBOLL/ILG
FamilyName	Undefined
Name	Tekla Corporation
Name	Tekla Corporation
Version	12.1
ApplicationFullName	Tekla Structures
ApplicationIdentifier	Multi material modeling
State	
ChangeAction	ADDED
CreationDate	Wed Jan 17 08:39:18 2007

Figure 3. Slab dimensions seen in the DDS Viewer

The engineers VVS model originally modelled in MagicCAD had been saved in IFC2x2 format uploaded and placed on the DkNanoIngeniør repository on the model server. The engineer's MagicCAD model in IFC 2x2 format was then downloaded and now opened in ArchiCAD for examination. All objects such as radiators and pipes seem to have been exported correctly (fig 4). However for some reasons the objects are all placed on a single storey instead of the five stories of the building (basement, ground floor, first floor, second floor and roof). When opened in ArchiCAD10 this one storey is automatically called Etage 0.

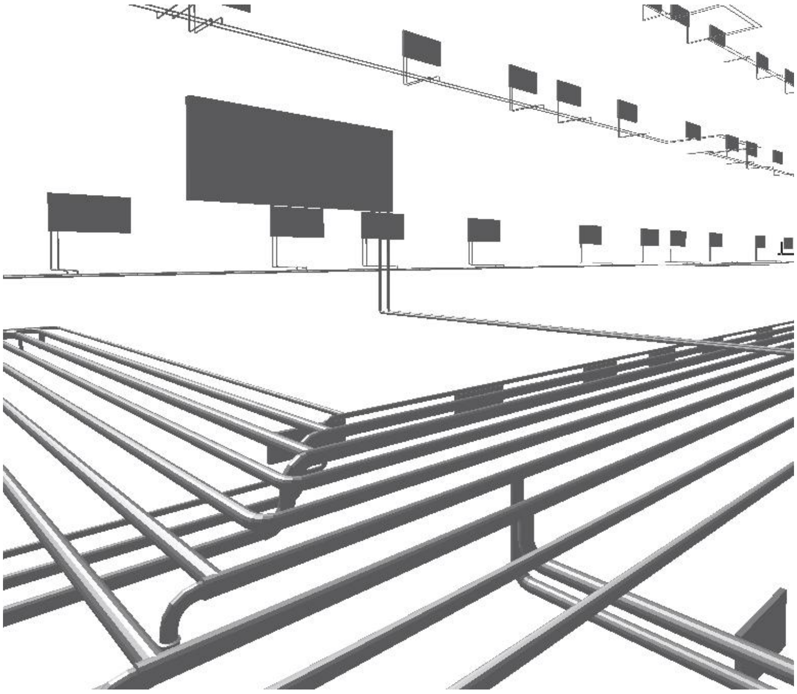


Figure 4. Part of the VVS model in ArchiCAD10 from Magic cad imported via IFC2x2

2. Preparation of the Usage Demo

2.1 The downloaded architect model in IFC 2x2 format was again opened in ArchiCAD. On the Ground Floor certain objects were selected and named for experimentation purposes. These objects were named so that they were easy to recognize during later examinations (fig 5).

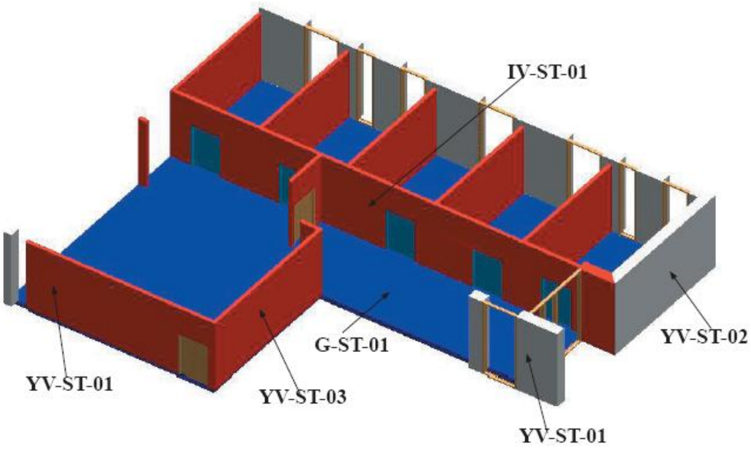
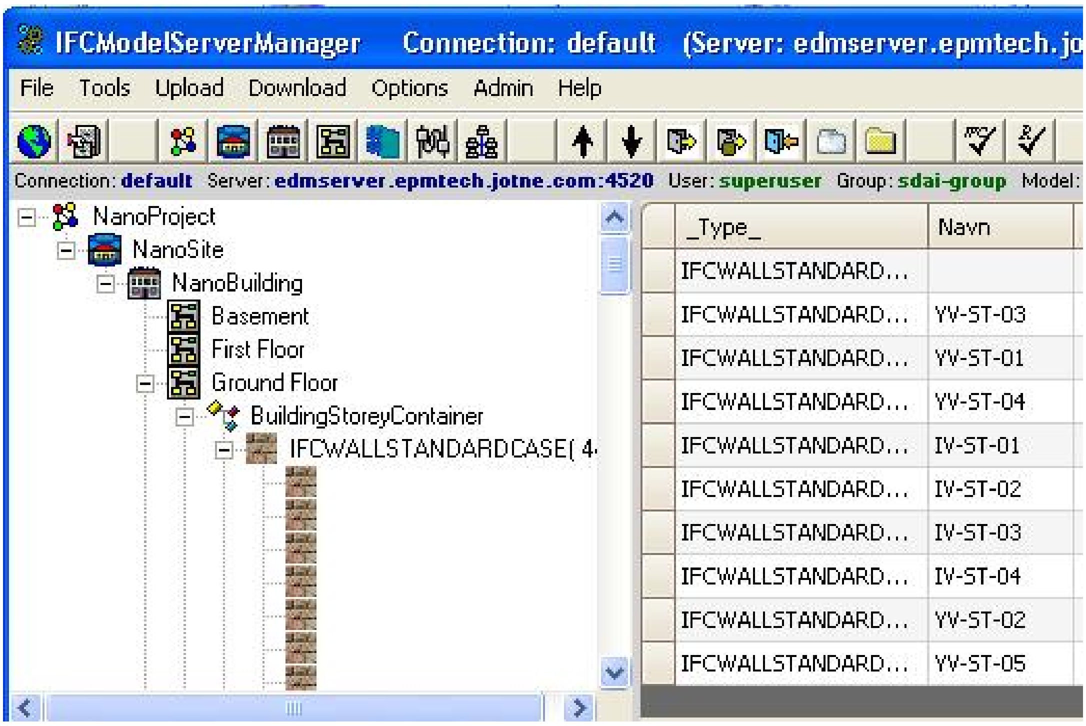


Figure 5. Objects named on the ground floor in the ArchiCAD10 file

The file was then uploaded on the model server and placed in the repository DK NanoResearch with the new name NanoBuilding. (fig 6)


IFCModelServerManager Connection: default (Server: edmserver.epmtech.jo
File
Tools
Upload
Download
Options
Admin
Help

Connection: default Server: edmserver.epmtech.jotne.com:4520 User: superuser Group: sdai-group Model:

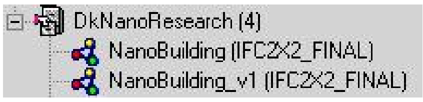
- NanoProject
  - NanoSite
    - NanoBuilding
      - Basement
      - First Floor
      - Ground Floor
        - BuildingStoreyContainer
          - IFCWALLSTANDARDCASE(4

Type	Navn
IFCWALLSTANDARD...	
IFCWALLSTANDARD...	YV-ST-03
IFCWALLSTANDARD...	YV-ST-01
IFCWALLSTANDARD...	YV-ST-04
IFCWALLSTANDARD...	IV-ST-01
IFCWALLSTANDARD...	IV-ST-02
IFCWALLSTANDARD...	IV-ST-03
IFCWALLSTANDARD...	IV-ST-04
IFCWALLSTANDARD...	YV-ST-02
IFCWALLSTANDARD...	YV-ST-05

Figure 6. Wall objects named as they appear on the server in the file called NanoBuilding.

2.2 The file NanoBuilding was then downloaded from the server and again imported into ArchiCAD. Now the dimensions were changed from the observed wrong dimensions under 1.2 to the correct dimensions on the named exterior walls (YV ST-01, YV ST-02, YV ST-03, YV ST-04) to a total of 450 mm (outside concrete layer of 90 mm, an insulation space layer of 200 mm and a concrete wall layer of 160 mm). The dimension was also changed for named floor (G ST-01) to the concrete total dimension of 270 mm.

Then the file was saved in IFC 2x2 format and uploaded on the Model server in the repository DkNanoResearch with a new name NanoBuilding_v1 (fig 7)



- DkNanoResearch (4)
  - NanoBuilding (IFC2X2_FINAL)
  - NanoBuilding_v1 (IFC2X2_FINAL)

Figure 7 The DkNanoResearch Repository

2.3 The file NanoBuilding_v1 was downloaded from the server and again imported into ArchiCAD. This time some of the steel construction elements from the construction model were added. These are 4 steel columns (Column ID: COL-001, COL-002, COL-003, COL-004) and two steel beams and 2 steel beams (Beam ID: BEAM-001, BEAM-002). (fig 8)

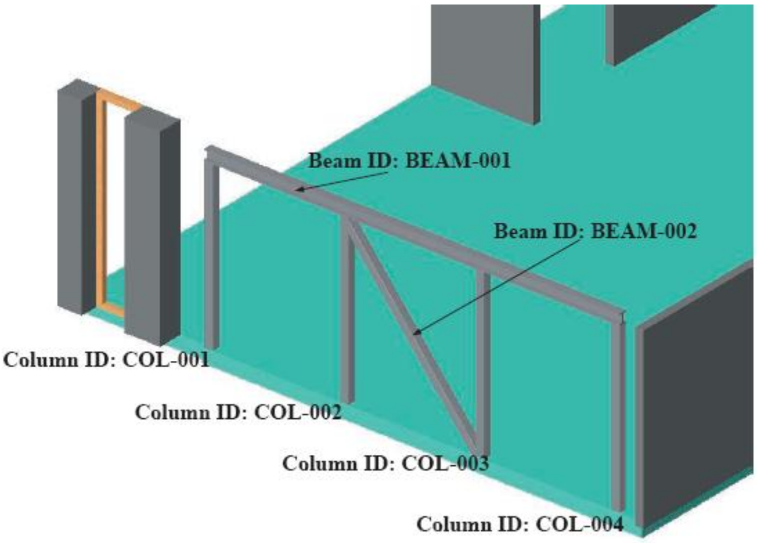
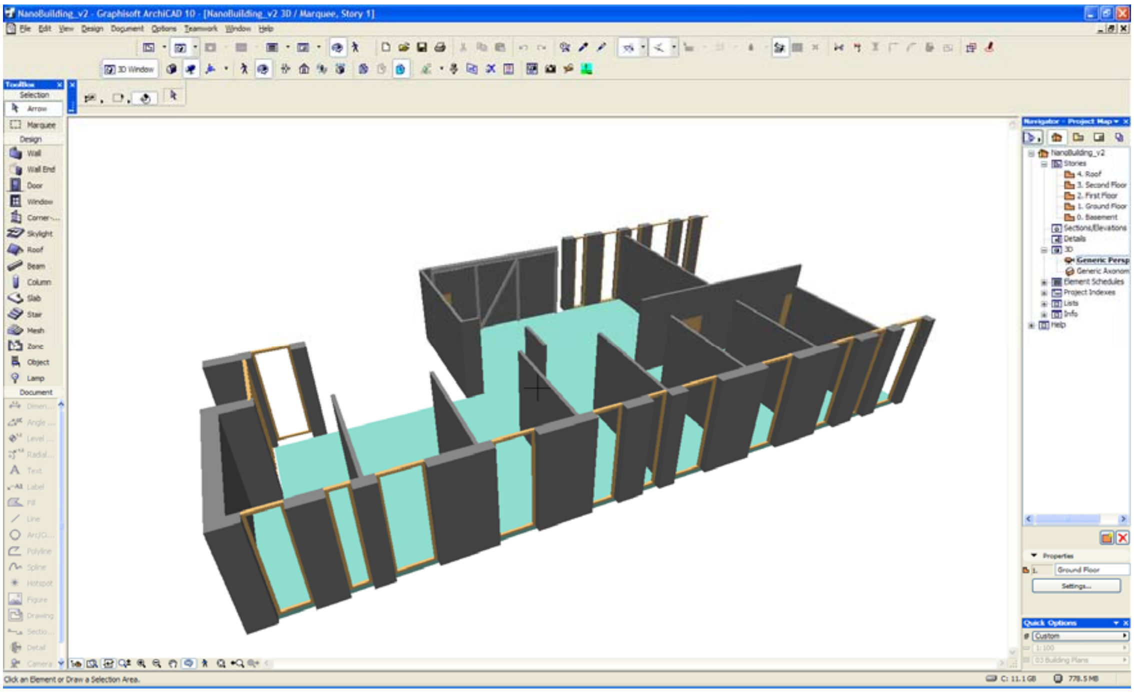


Figure 8. Names on Columns and Beams.

This is shown in the ArchiCAD model (Fig 9) and the steel construction in real is shown on photo (Fig 10 and 11) They were named and the file saved in IFC 2x2 format and uploaded on the model server in the repository DkNanoResearch with the name NanoBuilding_v2.

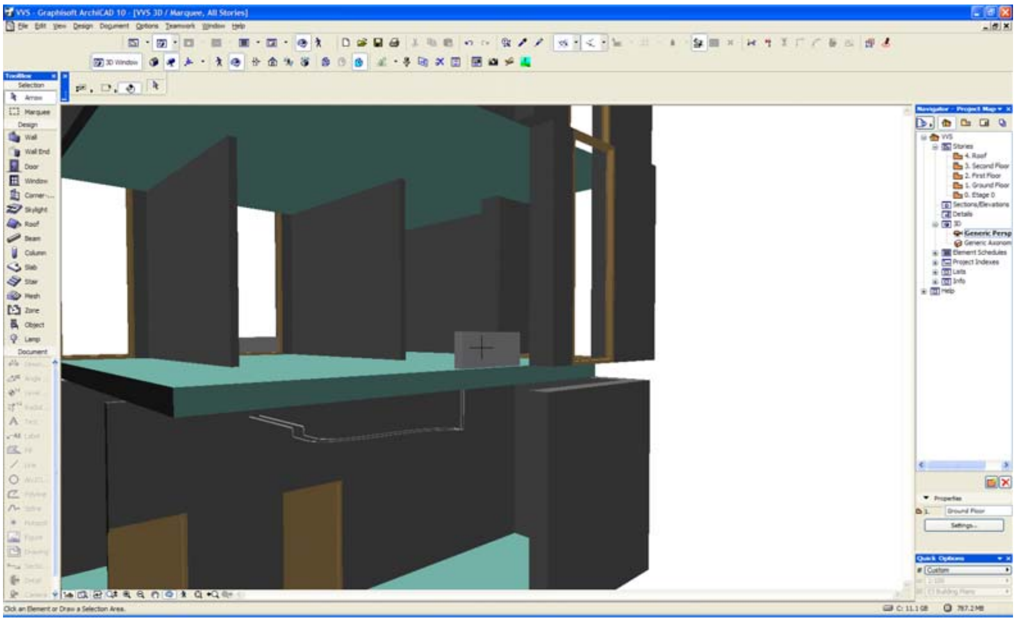


Fugure 9. The steel beam and the steel columns shown on the ground floor in the model.

Fugure 10. Photo (5 july07) of the steel construction in the building.

Fugure 11. Photo (5 july07) of the gable of the building.

2.4 The file NanoBuilding_v2 was then again downloaded from the server and imported into ArchiCAD. Further the engineers VVS file originally made in MagicCAD is downloaded from the server and opened in ArchiCAD. The VVS and pipe elements from the Magic cad file is the merged into the ArchiCAD file (fig 12). Finally the file is saved in IFC 2x2 format and uploaded on the model server in the repository DK NanoResearch with the name NanoBuilding_v3



Fugure 12. Convectors and pipes shown in the ArchiCAD10 model

3. Observations

No problems have been observed in naming individual objects in models (ArchiCAD10) and uploading these on the model server.

There seem to be great problems with Tekla since in the process of saving Tekla in IFC2x2 format uploading and downloading and importing the file in ArchiCAD10 some dimensions on objects had changed.

There has been no problems in placing new objects in ArchiCAD, or changing dimensions on existing objects in ArchiCAD and then saving the file in ifc2x2 and uploading. This indicates that it is possible to work with parts of objects make changes and upload them again.

Merging the VVS MagicCAD file turned out well. However the fact that the ifc 2x2 imported MagicCAD file showed all installations placed on one single storey and not on correct number of stories in the building is a great problem. It is possible to download a VVS model from the server, merge some of its objects into ArchiCAD10 and upload then again but there are problems in placing the objects on correct stories..

4. Other Observed Problems and Differences

When merging the architect and engineer construction model it was seen that the zero points were differently placed in the two models. In future modelling it is therefore important that the architect and the engineers decide about where to locate the zero point.

There were differences on where floor and ceilings should belong in relation to storeys. In the architect model the floor and the walls was defining the storey whereas in the engineer model the walls and the ceiling was defining the storey.