Collaborative engineering with IFC: new insights and technology

L.A.H.M. van Berlo
Netherlands organisation for Applied Scientific Research TNO, Delft, The Netherlands

J. Beetz
Eindhoven University of Technology, Eindhoven, The Netherlands

P. Bos
ZEEP architects, Amersfoort, The Netherlands

H. Hendriks
deBIMspecialist, Nijverdal, The Netherlands

R.C.J. van Tongeren
ARCADIS Netherlands, Arnhem, The Netherlands

ABSTRACT: The concept of working in one central Building Information Model (BIM) is becoming increasingly popular. Reports in literature as well as industry practitioners describe the lack of good implementations for IFC import/export in current software tools. The so-called ‘round tripping’ of an IFC model cannot be performed without data loss, making the merging of data into one central data repository impractical. This feeds discussions about the workability of IFC in relation to a homogeneous software environment. In the Netherlands several experiments were conducted to research if IFC still meets the needs from the AEC industry. The observations and opinions from users refute current theories and perceptions on collaborations using IFC. This paper describes a collaboration process called ‘reference models’. User opinions from the research state that the use of IFC, in an suitable collaboration process, meets the needs of the industry even better than homogeneous proprietary software environments.

1 THE PROMISE OF A CENTRAL BIM

For many years the central BIM collaboration concept has been idealized. Some of the current problems of the industry that can be attributed to the lack of up-to-datedness of data can be addressed by using central data repositories [Hannus et al 2003 & Froese 2003]. The concept where all project partners work simultaneously in the same central building model has raised a range of new questions, both technical as well as legal and about process workflows.

1.1 *New concepts raise new questions*

The following scenario raises new questions every time it is discussed. The famous problem that always pops up is the one where an architect wants to change the placement and/or properties of a structural wall. The structural wall is ‘owned’ by the structural engineer who is responsible for it. Many systems these days ‘flag’ objects in a database to an owner. That way the architect cannot edit the structural wall, but has to place a change request to the structural engineer. More advances systems even distinguish properties of the wall to allow the architect to change specific parts like finishing, but not the material or placement.

Working in a central BIM model also raises a lot of legal questions like ‘who owns the data?’, ‘who owns the intellectual property?’ and ‘what happens when the structural engineer doesn’t reply to a request?’.

1.2 *History of the central model concept*

The concept of working in a central data model was introduced years ago. The famous picture with the central model in the middle was originally introduced as a concept for a ‘shared data model’. A data model in software engineering is an abstract model, that documents and organizes the business data for communication between team members and is used as a plan for developing applications, specifically how data is stored and accessed. A data model explicitly determines the structure of data or structured data. The most common data model in the AEC is the IFC (Industry Foundation Classes) model. The IFC data model is the central model that was meant to be in the middle of the circle because the standardized structure of the data is shared by all team members.

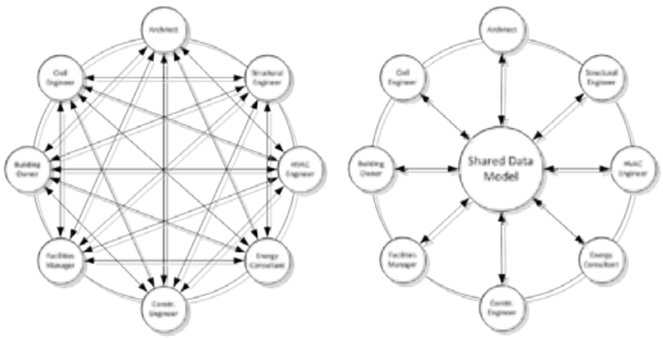


Figure 1. The well know conceptual picture of the Shared Data Model [Beetz, et.al 2011]

The sector however, driven by innovation, took the 'shared data model ' in the middle as 'shared data' or 'shared data repository'. This mix-up probably arose with the introduction of the term 'BIM' (Building Information Modeling or Building Information Model). The term BIM was used as a name for an instance of a data model. It is thought that non-technical users in the industry mixed up the 'Shared Data Model' (meaning IFC as a description of data structure) with an instance (data from a specific project) which was also called 'model' in the BIM-terminology.

Driven by these pictures, and the similarity between building model and data model, the industry was flooded by the concept of working inside a central data repository.

2 MOTIVATION AND GOAL OF THE RESEARCH

Lately, there seems to be a split between BIM user groups when it comes to the central model concept in the Netherlands. Roughly speaking there is, on the one hand, a group that strongly believes in working with a central data repository based on a single homogeneous software environment. On the other hand hand there are the believers of freedom for a project partner to choose its own software tools. This group also tends to believe in a shared data repository, but finds this has to be based on an open data model like IFC. We call these groups 'homogeneous software environment' and 'plural data environment'. Both propagate arguments on different levels of technology, usability and efficiently. The most frequently heard arguments are discussed in the next two chapters.

These discussions were usually about finding solutions for issues that were only raised by working with a central data repository. This raised the question to research if open BIM data standards like IFC meet the needs of the industry. To answer this question, multiple experiments were conducted in a practical environment. We specifically choose this research question because software providers are conducting their own research about user needs for working with their central data repository within their own single environment.

3 LEVELS OF BIM

The use of BIM in daily practice is very differentiated. Some users are very advanced, others are on a lower competence level [Sebastian & Berlo 2010]. This may influence the needs of users under different circumstances.

To distinguish different situations of BIM use, we created a framework of BIM levels. This framework is decomposed from the perception of BIM levels in the Netherlands, based on 'BIG BIM, little bim' [Jernigan 2008]. Many Dutch BIM users see the use of BIM within the influence boundaries of a single company as 'little bim'. As soon as a collaboration between actors outside of the influence of the company is established it is called 'BIG BIM'. In most situations this happens when collaboration is established between multiple (project) partners. Often a consortium is created for a specific project, consisting of parts of multiple companies. This situation occurs a lot for large infrastructural projects. Because the consortium has significant influence on the actors in the project, this is still classified as 'little bim'.

Another ongoing debate evolves around the term 'openBIM'. The various definitions of openBIM are very differentiated. Some users call the exclusive use of BuildingSMART standards for exchange of information in a project 'openBIM'. Others focus on the freedom for all project actors to choose their own software tools 'openBIM'. Whatever the exact definition is, users seem to agree that 'openBIM' only has an added value on projects that exchange data between users that use different software environments.

There are many discussions on the position of working with a central data repository in our framework. Nour [2009] suggested the use of a private repository that enables the stakeholders to keep their unshared information and local versions of the design within the boundaries of their organizations. This enables them to use any type of software or developing platform and use a data repository in a homogeneous software environment within their 'little BIM'. At certain development stages of the design, a release version can be exchanged with the 'BIG BIM' repository [Nour 2009]. Most discussions compromise that open BIM standards do not add significant value in a 'little bim' environment. The relation between the use of open BIM standards and 'BIG BIM' collaboration seems to be obvious for everyone.

This rough framework is used in the research, the experiment and the presentation of the results in this paper.

4 RESEARCH QUESTIONS FROM COMPARISON BETWEEN CONCEPTS

Before starting the experiment, several research questions were formulated. Most of the questions are derived from the discussions described earlier. The most common and fierce discussed issues are transformed to research topics in this experiment. In this section we provide an overview of these questions.

4.1 *Who is responsible for transformations of the data (and resulting consequences)?*

One of the most debated questions is: "what happens, or should happen, when an architect wants to change the placement and/or properties of a structural wall?". The structural wall is 'owned' by the structural engineer who is responsible for it.

4.2 *How significant are the workflow changes to be able to use a concept at its optimum?*

Most of the issues that arise when working with a central data repository, can be dodged by using good agreements in process and workflow. But there seems to be an optimum in changing a process to fit the new technology, with respect to the theoretical optimum of technology.

4.3 *What data needs to be shared/exchanged?*

Recent research showed that BIM might cause costs of failure, when used in the wrong way [Hartmann 2010]. Many users seem to have an urge to create a theoretical perfect building model. This is not always desirable, depending on the phase of the process and stage of the adulthood of the project.

4.4 *What is the preferred function of locking and releasing of objects?*

When working in a central data repository, a commonly used solution to avoid conflicts is the 'locking' of objects in the data repository. By locking an object, no other user can change that object during the time it is locked. This solution increases the chance to preserve integrity of data in the central repository.

4.5 *How to prevent loss of data during parsing or transformation?*

When data is used from different sources it has to be transformed to the single data model of the central repository. In both concepts external data has to be transformed or parsed into a chosen data model. Users try to find different solutions to prevent data loss during this transformation.

4.6 *Can all actors be forced to use the same software tool to do their work?*

This research question only applies to the homogeneous software environment concept. It is taken into account in this research because it holds a veto on that approach.

Of course there are also other issues in the comparison of the two concepts. This research has limited its efforts to the topics that seem to have the most divergent opinions.

5 EXPERIMENTS

To gain practical insights in working with a central data repository, a national Dutch modelserver pilot was started. The experiment was initiated in 2010 by TNO and nine Dutch companies participated. The group of companies represented a cross-section of the industry, containing architects, clients, BIM process consultants, software vendors and engineers from multiple disciplines. Goal of the pilot was to test different workflows for BIM collaboration.

Around the same time, a few real-world projects were started by project teams that committed themselves to experimenting with central data repositories. Goal of the experiment was to compare the efficiency of the collaboration process in two different cases: (1) using a central data repository where all actors use the same software tool, and (2) using open standards like IFC to exchange data because all actors used different software tools.

In 2011, ARCADIS Netherlands started an internal project, researching scenarios of usage of data servers for BIM.

All of the above projects consulted TNO to support them. The results of these projects are also incorporated in the research results described in this paper.

The research consisted of observing users during the projects. Also the data and model instances were analyzed during and after the experiment. Researchers had full insight in the data repositories during the whole life cycle of the project.

Furthermore the research consisted of in-depth interviews of the stakeholders and users both during and after the experiments. Performance of different workflows and concepts was valued and measured by the level of practical use. The interviewers were focused on filtering parroting of marketing flyers from actual practical experiences.

From the experiments, all project managers and several project members (at least one from all disciplines involved) were interviewed and observed.

All users (respondents) were experienced craftsmen. All of them had multiple (>5 years) of experience in engineering in the AEC industry. Also all respondents were advanced BIM users (>3 years FTE experience).

Some of the questions that were asked during the interviews were: How is the project organized and how are the responsibilities organized?, How is information from different disciplines aggregated?, Are there moments in the process where reviewing of data is taking place?, What is the expected output of a discipline?, What would improve the current collaboration process?, What are the top 5 bottlenecks in the collaboration process? What would be a good moment to aggregate, compare and evaluate data from more than one discipline?

6 PROCESS OBSERVATIONS

During the experiments in the Netherlands some big issues were raised. The following pages describe the strongest opinions of the users during and after the experiments.

6.1 *Locking and releasing of objects*

The sequential workflow of asking permission for a change weakens the flexibility of the engineering- and design process. The AEC industry is by nature an industry that works in parallel. For more than 25 years now, ICT tools that lock parts of data for some partners in a project are not accepted by the industry. Established software tools that facilitate locking of objects are widely used, but we observed that most users use full access privileges to overrule locks.

6.2 *Real time working in one data repository*

A main reason for most users to work in one homogeneous software environment is the live synchronization of all the data. The previously described solution of 'flagging' objects in a central data repository (most of the time to identify the ownership of an object), actually creates several different instances in the data repository. Some interviewed users even called them 'different projects living their own life in a single database'. In principle, this is conceptually not much different from using several different databases and bringing the data together into one single repository at a given time. The only difference is the real time synchronization. However, 'real-time' does not really exist. There is always a delay and synchronization time, even when several users connect to one database. The question should be what synchronization time is desired by the users. There is fundamentally not a great difference between using one single database where everybody connects, or a set of (distributed) databases that synchronize at a given time.

During the ARCADIS experiment it was found that real-time capabilities of software systems were usually turned off and replaced by manual synchronization. More and more engineers in the AEC industry prefer a synchronization that is not done in real-time, but once a day, or even once a week [Arcadis 2011].

The AEC industry is a fragmented industry with specialists working on a specific task. It is the responsibility of a specific actor to perform a task within given conditions. A single actor only needs a specific subset of all available data to perform the task. Working in a central database gives a lot of data overhead that is not necessary to perform the task, and will reduce performance and effectiveness of the specific task to be carried out [Kvan 2000, Kalay 1998].

Furthermore, with the growing complexity of buildings and the associated growth of data volume, the hardware is not always capable of synchronizing the full database. And certainly not in a real-time concept. Therefore users prefer to extract a subset of the data, work with that and place their work back into the central data repository. They prefer a subset of the data that only contains information they need at that time. Most users that collaborate often with BIM cannot wait for the IDM concept to work in practice.

6.3 *Exchanging data versus information*

Most BIM users prefer to receive data that only contains information they need at a given time. In the experiments, no clear definition of what (semantic) information was needed to perform a specific task was at hand. During the experiments, is was unclear what the exact information need from engineers was. However, when asking respondents if all the data they needed was in the exchanged IFC file, all respondents answered positive. Respondents think the lack of information agreements is becoming a crucial threshold to overtake in the transformation of the industry to use BIM in its full potential.

Respondents believe the IDM and MVD concept [BuildingSMART 2011] could solve this issue. Most users that collaborate on a regular base with BIM cannot wait for these concepts to work in practice and they urge the BuildingSMART organization to accelerate both the development and the practical implementations.

6.4 *Forcing software use*

Experiments where project partners voluntarily or involuntarily selected the same software tool have been aborted early by the stakeholders. There seems to be an agreement among the interviewed users, that (software) tools should be chosen based on the task performed by the expert, not based on the ability to share data. All interviewed users believe that an expert should be able to choose the tools (software) that fits his needs to perform the job for which he is asked to participate in the project.

6.5 *The myth of the roundtrip*

The next issue in the workflow is the problem of data loss during import and export of data in modeling software and a central repository. Many publications discuss the so called 'roundtrip' where an IFC instance is imported and exported in a chain of software applications [Kiviniemi, 2009]. When all the import and export implementations of the software tools are fully functional, the IFC instance at the end of the chain will be identical to the one at the beginning. Because the IFC data model is not always restrictive, and software implementations are never 100% correct, this roundtrip will probably never work to its full extent [Banerjee et al 1987, Lerner and Habermann 1990, Eastman 1992, Zicari 1992, Amor 1997, Atkinson et al 2000, Amor and Faraj 2001, and Grundy et al 2004]. Many researchers and software vendors claim that this will stop the adoption of IFC in the industry. However, during our experiments, users dodged this issue in a practical manner. Their claim is that nobody needs the whole model and everybody takes responsibility for their own data. This way, a user never works in or from data in a central repository, but only with their own database or model instance. All users takes responsibility for their own data, which is (conceptually) a part of the central data repository. By sharing and synchronizing a subset of their own model instance in a central data repository the users do not aim at creating a perfect BIM model, but focus on doing a good engineering task.

The workflow of *import, add data, export and send to next user* is therefore not being used. This avoids the problems of data loss due to invalid import/export implementations in software, but also gives the ability to work in parallel.

Users of IFC dislike the fact that the myth of the roundtrip is a constantly recurring story that has to be refuted in every discussion. IFC users in daily practice have to defend themselves against the negative arguments that some researchers (and sometimes even BuildingSMART members) keep feeding.

7 TECHNOLOGY OBSERVATIONS

Of course, to perform a specific task in the AEC industry a user needs to get data from other project partners. When using a homogeneous software environment this is not an issue. When using a pluralistic software environment the data is most often exchanged using the IFC data model. Several publications report on the difficulties on data merging using IFC [Jørgensen et al, 2008]. During the experiments we expected the team to encounter these issues. However, during the experiments the project teams never seemed to appreciate a (theoretical) single merged IFC model but were more concerned about architectural and engineering issues. The following sections provide an overview of observations about merging, model fusion and change finders made during the research.

7.1 *Reference models*

An IFC dataset does not hold all data from an original model. While importing an IFC instance data can get lost because of an invalid IFC implementation in the software [Amor et al 2007].

In all experiments and subsequent interviews we found that both of these issues do not cause any problems in daily practice. It is true that IFC will not contain the full dataset of the original, native database from the source software, but often the receiving user does not require all details. Occasionally some data became unreliable (objects are misplaced or gone) during import, but the overall quality of contemporary implementations was considered satisfying. The imported data is used as a reference during engineering. For example the MEP engineer uses some data from the structural engineer to design the location of the piping. The piping is not added to the original dataset of the structural engineer, but exported to IFC as a new dataset that is send to the central data repository.

This concept is not new and often referred to as the use of 'reference' models, 'discipline' models or 'aspect' models (in all cases models meaning instances) [Lundsgaard 2008]. The concept is illustrated in figure 2.

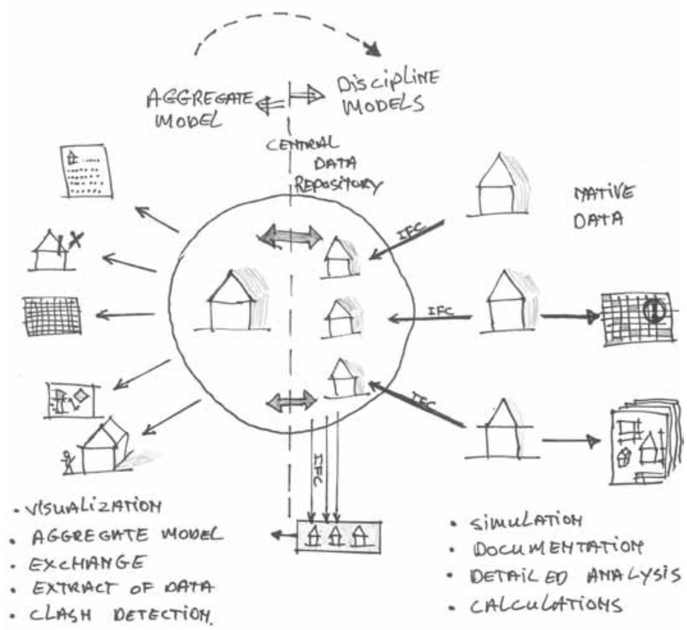


Figure 2.The concept of reference models in the context of a central data repository [loosely based on Lundsgaard 2008].

During the experiments the concept was generally used in the following way: on the right hand side of figure 2, the native BIM instance models are located. In most situations, these were models from the architect, construction engineer and MEP engineer, but they could be instances from arbitrary team members in the project. Each model instance manager creates an IFC instance model from the native data (at any given time) and sends that IFC instance to a central (IFC) data repository. In all experiments during this research this central data repository was called ‘bimserver’. In almost all situations, the software used to facilitate the central data repository in the experiments had the ability to keep revisions of the data from the discipline models [BIMserver.org 2011]. On the left hand side of the dotted line in figure 2, a revision is recorded every time a single change is made to one of the discipline models. The system allows the creation of a hierarchy of discipline models adding up to a complete model on several levels.

In any AEC project, a large variety of different engineering tasks have to be performed. Depending on the task at hand, the project members or project manager extract the needed data from the project. This could be a combined model, an instance of a single discipline model, or the native data instance in its original (proprietary) format. During the experiment it was found to be logical to perform engineering tasks within the boundaries of a single discipline of a project member, with its native data. For example, construction analyses are done by the construction engineer using native data. Same goes for other simulations and documentation of the project. Tasks like visualization, data sharing, exchange and clash detections have a logical place on the left side of the conceptual picture. These tasks are performed using a fused version of IFC discipline models. Notice the direction of the different arrows. In practice, users don’t work with the previously described roundtrip principle, but only use IFC to send data downstream to other users. When results from analyses on a fused model (like clash detection) need an adaptation of their native data, they change their own native model without using import of (a proposed instance model in) IFC from other users. The later does not mean that users never import IFC. On the contrary: IFC instance models from other disciplines are imported in native models very frequently. The reason to import other user’s IFC data is not to edit it, but to use it as a reference (in interviews something refered to as ‘background model’) to create and edit their own native data.

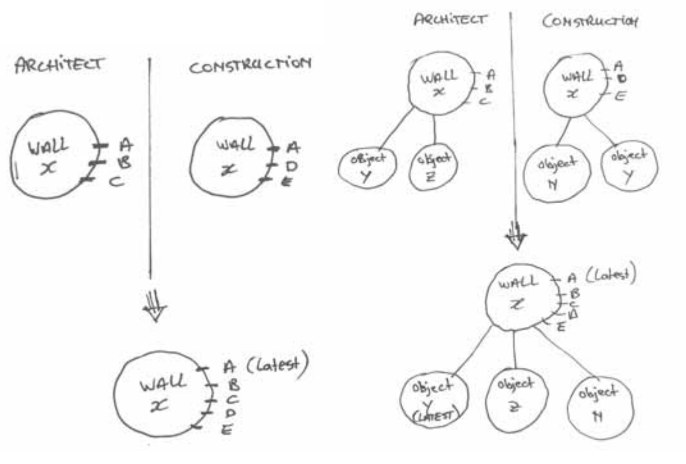
7.2 *Non-intrusive merging*

The central database where all data is collected, was nevertheless the backbone of all projects during the experiments. This database performs specific tasks needed to support the workflow. These tasks are e.g. querying, user management, filtering, sending out change notifications to subscribers and of course merging. Merging can be done in two ways. The ‘classic’ way of merging comes from the original concept of a central data repository. In this concept a central model is being changed by consecutive calls like add, delete and change. This way only changes (deltas) are communicated to the central data repository. This modus requires a high quality of import and export.

Merging data instances using IFC is done by getting the different model instances together and fuse (merge) them into a single instance. This process is less intrusive than the classic way. The most important issue to solve in this matter are the duplicate objects in the data repository. A side effect of using ‘reference’ models is the likelihood of duplicate objects. For example, the structural engineer has an instance of a structural wall in his database, while the architect also has the wall in his database (with the same IFC identifier). This is justified because both users need to edit their own data. During a model fusion the fusion algorithm has to detect those duplicates and fuse (merge) them into one single object instance. Conceptually, the same issue occurs in a homogenous software environment with live synchronization.

7.3 *Fusion algorithms*

During the experiments several model fusion algorithms have been developed and tested. Object fusion based on object identifiers IFC GUID, and the IfcName string have been tested. Tests were also conducted on fusing relations and references to and from an object into a single instance.



Figures 3 and 4. Principle explanation of the fusion algorithm concept.

In figure 3, the basic concept of model fusion is illustrated. The circle ‘wall x’ defines a specific wall with an identification. Using the concept of reference models, it can happen that two discipline models both store the (semantically equal) wall x.

The architect owns a wall x instance including properties a, b and c. The structural engineer also owns an instance of wall x', including properties a, d and e. In this scenario both wall instances x and x' have the same identifier. When these two discipline models are fused into one model instance, there will be a wall x, including the latest version of property a, and properties b, c, d, and e. What happens to objects that are related (or are referenced to) the wall object, is illustrated in figure 4. The same principle for merging properties is used for relations and references. Object y and object z have a relation r to wall x in the instance model of the architect. The wall x from the construction engineer has references and relations to object n and object y (note that object y is again, semanticly the same as object y in the instance model of the architect). After the fusion algorithm, the result is also shown in figure 4: a wall x, with references and relations to the latest version of object y, object z and object n.

During the experiment users had diverging opinions about model fusion. Some users did not want software to fuse model instances, but only expose the overlap and abnormalities. These users were all users of software tools like Solibri Model Checker [Solibri 2012]. Using Solibri Model Checker, the overlap of two instances (a semantically single wall, that has a structural instance from the construction engineer, and an architectural instance from the architect) is exposed by the software. The users wanted to be able to analyze these overlaps to control the data.

8 RESULTS

Based on the observations during the experiments and the interviews with users, the research questions can be answered by summarizing the results and observations. The following summary is focused on the use of IFC in a central data repository.

a. Using the concept of reference models with IFC exchange does not result in loss of data. The data is still kept in a native format. IFC is used to exchange subsets of the native data to communicate with team members. All detailed calculations, extractions, predictions, etc. are extracted from the native data source. The subset of IFC data that is shared with other partners seems to be detailed enough (but not too detailed) for project partners to be able to perform their required engineering tasks.
b. The best choice of concept to be used as central data repository, depends on the place in the BIM framework ('little bim', 'BIG BIM', 'open BIM'). In many situations it seems favorable to use a combination of homogeneous software environment ('little BIM' within a single company) and plural software environment ('BIG BIM', using 'open BIM' standards) in a multidisciplinary collaboration project.
c. Project team members should be free to choose their own software tools in order to achieve a higher performance in the execution of their engineering tasks.
d. Working with IFC in the reference model concept corresponds to the current ways of working in the AEC sector. The sector is fragmented with several experts that come together to engineer a building. All experts work in parallel on their own task. The result of individual tasks has to be synchronized. The synchronization interval depends on several factors such as: the preferences of the project team members, the project manager and the project phase. Users in the experiments in this research did not prefer a so-called 'live synchronization'. The median of synchronization time of model instances was 1 week (every Friday of every week was a widely used practice).
e. In a homogeneous software environment, users felt that all team members should have equal BIM software modeling expertise. When using the concept of reference models and IFC, not all project team members collaborating in a project need to have the same level of BIM expertise.
f. Considering the previous results, the additional conclusion can be drawn that project partners can be chosen based on competence in performing engineering tasks. The use of a reference model concept with IFC can lower the needed BIM competences for a project partner to be able to collaborate in a way that is sufficiently effective for the entire project team. All respondents in this experiment were strongly convinced that choosing project partners based on their competence of a specific software tool, prior to their engineering competence, is never preferred.

9 CONCLUSION

The conclusion of the Dutch experiments is that the IFC data standard, combined with smart process workflows creates a stable and usable collaboration environment in the AEC industry. The answer to the original research question if open BIM data standards like IFC meet the needs of the industry can be answered positive when used in a process of reference models (as described in this paper). The combination of product (IFC data model) and process is only successful in its combined use.

When using the concept of 'reference models' and IFC, not all project team members collaborating in a project need to be on the same level of BIM expertise. For most collaboration project this is crucial for successful engineering.

Another conclusion from the research is that most of the worries about interoperability in practice are not about the technological (semantic) interoperability of open BIM standards, but whether enough data is available to perform a specific engineering task. The industry is looking forward to IDM solutions to facilitate this. Experienced practitioners are not focused on creating a theoretical perfect data instance, but on engineering a high quality building.

BIM users are focused on BIM tools that support both their responsibility to perform an engineering task, as well as to effectively collaborate with project partners. Users are very sensitive for BIM software tools that need a (not preferred) process change that has no added value except creating a BIM instance on a purely theoretical level.

10 DISCUSSION AND FUTURE WORK

The results and conclusions in this research are solely based on interviews and observations. It is very difficult to objectively measure the collaboration performance of a project team. Therefore it should be noted that similar experiments in different setting, with different participants could result in completely different opinions and conclusions.

The experiments in this research are limited to only a few projects. To confirm the results, opinions and conclusions described in this paper should be checked again after a longer duration of experimenting.

Some results and conclusions state some process workflows and technology fits the need of the industry. This is an opinion of the current state of the industry. New technology that might seem undesirable today, might drive the industry to new innovations nevertheless.

11 REFERENCES

Alshawi, M. & Aouad, G. & Faraj, I. & Child, T. & Underwood, J. 1998, The implementation of the industry foundation classes in integrated environments, *Proceedings CIB W78 Conference*: 55–66.

Amor, R. 1997. A Generalised Framework for the Design and Construction of Integrated Design Systems. *PhD thesis, Department of Computer Science, University of Auckland, Auckland, New Zealand* (350).

Amor, R. & Faraj, I. 2001. Misconceptions about Integrated Project Databases, *ITcon journal* (6): 57-68.

Amor, R. & Jiang, Y. & Chen, X. 2007. BIM in 2007 - are we there yet? *Proceedings of CIB W78 conference on Bringing ITC knowledge to work*: 159-162.

Atkinson, M.P. & Dmitriev, M. & Hamilton, C. & Printezis, T. 2000. Scalable and Recoverable Implementation of Object Evolution for the PJama1 Platform, *Persistent Object Systems, 9th International Workshop, POS-9:* 292-314.

Arcadis, BIM working group. 2011, *Pilot open source BIMserver final results*. Internal document Arcadis Netherlands.

Banerjee, J. & Kim, W. & Kim, H. & Korth, H. 1987. Semantics and Implementation of Schema Evolution in Object-Oriented Databases, *Proceedings of the 1987 ACM SIGMOD international conference on Management of data*: 311-322

Beetz, J., Berlo, L. van 2011, Advances in the development and application of an open source model server for building. *Proceedings CIB W78 W102*: 117.

BIMserver.org 2011, www.bimserver.org, last visited March 2012.

BuildingSMART 2011, www.buildingsmart.com, last visited March 2012.

Eastman, C.M. 1992. A data model analysis of modularity and extensibility in building databases. *Building and Environment* 27(2): 135-148.

Froese, T. 2003, Future directions for IFC-based interoperability. *ITCon* 8: 231-246.

Grundy, J.C. & Hosking, J.G. & Amor, R.W. & Mugridge, W.B. & Li, Y. 2004. DomainSpecific Visual Languages for Specifying and Generating Data Mapping Systems. *Journal of Visual Languages and Computing* 15(3-4): 243-263.

Hannus, M. & Blasco, M. & Bourdeau, M. & Böhms, M. & Cooper, G. & Garas, F. & Hassan, T. et al. 2003. Construction ICT roadmap. *Public report of ROADCON project IST-2001-37278*, Deliverable WP5/D: 52, 30.

Hartmann, T. 2010. Detecting design conflicts using building information models: a comparative lab experiment. *Proceedings CIB W78*: 57.

Jernigan, Finith E. 2008, *BIG BIM little bim*, isbn 0979569923

Jørgensen, K.A. & Skauge, J. & Christiansson, P. & Svidt, K. & Sørensen, K.B. & Mitchell, J. 2008, *Use of IFC Model Servers*, Modelling Collaboration Possibilities in Practice.

Kalay, Y.E. 1998, computational environment to support design collaboration, *Automation in Construction 8 (1)*: 37–48.

Kiviniemi, A. 2009, IFC Certification process and data exchange problems. *Ework and Ebusiness in Architecture Engineering and Construction*: 517-522.

Kvan, T. 2000, Collaborative design: what is it? *Automation in Costruction* 9(4): 409–415.

Lerner, B.S. & Habermann, A.N. 1990. Beyond schema evolution to database reorganization, *Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*: 67-76.

Lundsgaard, T. et al 2008 (translated March 2009), BIPS C102 CAD manual.

Nour, M. 2009, Performance of different (BIM/IFC) exchange formats within private collaborative workspace for collaborative work. *Journal of Information Technology in Construction 14 (special issue)*: 736-752.

Plume, J. & Mitchell, J. 2007. Collaborative design using a shared IFC building model--Learning from experience. *Automation in Construction* 16(1): 28-36.

Sebastian, R. & Berlo, L. van 2010. Tool for Benchmarking BIM Performance of Design, Engineering and Construction Firms in The Netherlands. *Architectural engineering and design management* 6: 254–263.

Solibri Model Checker 2012, www.solibri.com, last visited March 2012.

Zicari, R. 1992. A Framework for schema updates in an object-oriented database systems. *Morgan Kaufmann Series In Data Management Systems*: 146-182.