IFC Support for Model-based Scheduling

M. Weise & T. Liebich
AEC3, Munich, Germany

J. Tulke
HOCHTIEF AG, Corporate Headquarters, Essen, Germany

P. Bonsma
TNO-Netherlands Organization for Applied Scientific Research, Netherlands

ABSTRACT: Model-based scheduling is about linking information from three yet separated domains; (1) architectural design, (2) construction scheduling and (3) cost management & quantity take-off. It thus will benefit from the Building Information Modeling (BIM) approach followed by the international buildingSMART initiative and its product the IFC data model specification. On the basis of a novel rule-based linking approach, which is shortly explained in the beginning, the paper provides an overview of the work that has been done to apply the linking approach in conjunction with the IFC 2x3 release and available IFC-enabled design applications. The paper gives insights into the capabilities but also the gaps of IFC. However, the paper is not going into technical details, which are provided in a separate, freely available discussion paper. Instead it shows the way from the requirements definitions, to the gap analysis, suggested extensions and prototype implementations, namely the Scheduling Assistant and the IFC-interface plug-in for MS Project. It finally discusses the current status of our developments and gives an outlook for further developments within the European Research and Development project InPro, an Integrated Project under the 6th framework program that has supported this work.

1 INTRODUCTION

1.1 *Benefits and problems*

The benefits of model-based scheduling are widely recognized (Porkka & Kähkönen 2007). It is not only to have more accurate data, which for example in case of quantity take-off (QTO) data might be derived from the 3D geometry, but also to provide better decision support on the basis of 4D technology, i.e. the visualization of construction processes. Several research projects have proofed the advantages of integrated design, the so called Building Information Modeling (BIM). However, in real projects there are still problems using model-based scheduling. On one hand the BIM approach is not yet the typical way of working so that there is a lack of intelligent design data. According to Porkka & Kähkönen (2007) an important reason is the missing interoperability between available software tools. They furthermore argue that IFC is *"one of most promising candidates for open BIM standard"* and that *"IFC2x3 can be taken to a technological starting point; to be further equipped with 4D case definition. This case definition needs to present the new process taking advantage of the holistic IT infrastructure within construction process."* On the other hand the effort for integrating required domain data, i.e. architectural design data, cost & quantity information and construction schedules, is still high and requires a lot of experience (Heesom & Mahdjoubi 2004).

Whereas the issue of efficiency is addressed by the rule-based linking approach developed by Jan Tulke (Houttu et al. 2009) the focus of the paper is to discuss its implementation on the basis of the IFC2x3 release. We discuss the possibilities of IFC, required extensions as well as the prototype implementations that will be used in demonstration scenarios of the InPro project (www.inpro-project.eu). In our gap analysis we also considered the new IFC2x4 release, which already fixes some of the found issues.

The paper is a summary of results that are published in a discussion paper about IFC-based scheduling (Weise et al. 2009). This document tries to include and continue the work that has been done so far with IFC. It is meant to be an open document where everyone who is interested to contribute to these developments is kindly invited to extend and improve the discussed specifications and examples. The paper is available at www.inpro-project.eu.

1.2 *Principle of the rule-based linking approach*

The aim of the rule-based linking approach is to
- reduce the effort for the overall set-up process by using semi-automated linking mechanisms and

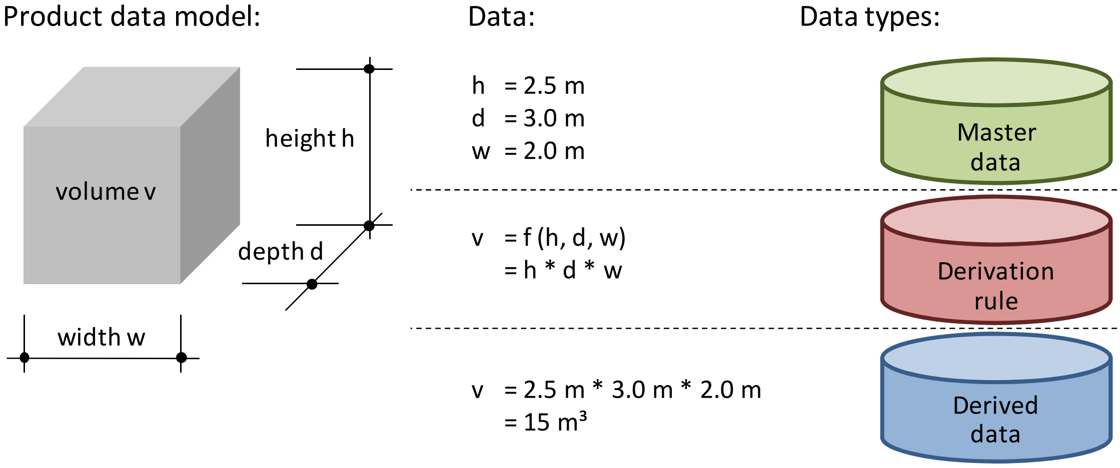

Figure 1: Example showing the three principle data types of product model data

– improve the management of data dependencies to react more efficiently on design changes.

The linking rules not only specify the dependencies between domains but also include the knowledge about refinement of BIM data, which in particular is needed to make a breakdown of building elements such as slabs and walls so that they can be linked with work tasks. The advantage of the rule-based linking approach is based on the hypothesis that only few decisions such as the definition of construction zones enable to derive a lot of design data and, once these decisions have been made, can easily be re-used in case of design changes.

In context of the suggested rule-based linking approach and in particular when dealing with design iterations there are three types of design data that are of interest for discussing the mapping to an IFC model (see Figure 1):

1 Master data, which is the "input" for discipline specific design processes.
2 Derivation rules, which are "design decisions" made within design processes.
3 Derived data, which is the "output" of design processes, e.g. the activity durations in construction schedules, related quantities and costs.

This explanation gives a basic understanding but is also misleading. In fact it is not really a differentiation between input and output of design processes, it is a differentiation based on data dependencies as illustrated in Figure 1. For that example the differentiation of the product model into the three data types is:

1 Master data = the dimension of the cube
2 Derivation rules = the function that calculates the cube volume
3 Derived data = the calculated cube volume

Please note that above differentiation is not prescribed. If the cube volume is more important than the cube dimension it could be changed so that for instance the master data are the volume v, the height h and the depth d. The derivation rule would then calculate the width w = f(v,h,d) = v/(h*d).

Plus, master data is not only the input but is also supplemented within discipline specific design processes. If this additional master data is the basis for decisions within subsequent design processes, it becomes equally important in case of design iterations.

1.3 *Scope of IFC*

Based on the three data types there are two options to capture results of design processes. It is possible to store master data either with derivation rules (1+2) or with derived data (1+3). In the first case we are working with unevaluated models, which have advantages in case of design iterations. But they typically require a noteworthy computational power since derivation rules have to be evaluated for getting required results, for example the quantities of building elements, analysis results of FEM calculations, etc. It is not (yet) the way of working in the AEC industry and in particular it is rarely used for the purpose of data exchange (Koch & Firmenich 2006).

Instead of storing design decisions, IFC is focused on design results that might hold a link to master data but without further details about used derivation rules. This principle works fine for storing a particular design solution but provides no support if design changes have to be revised and matched with previous design decisions. Consequently, the IFC data structure is primarily focused on master and derived data (1+3), whereas design decisions (2) have to be handled by responsible actors themselves.

According to the approach from Tulke et al. (2008) design decisions should be captured by a "linking language", which is out of scope for IFC extensions. Nevertheless, the aim of supporting model-based scheduling based on an open BIM standard is to capture the link between derived data and externally stored derivation rules. Accordingly, for an IFC-based implementation of the discussed approach the following questions have to be answered:

– How to capture required data, in particular what extensions are necessary?

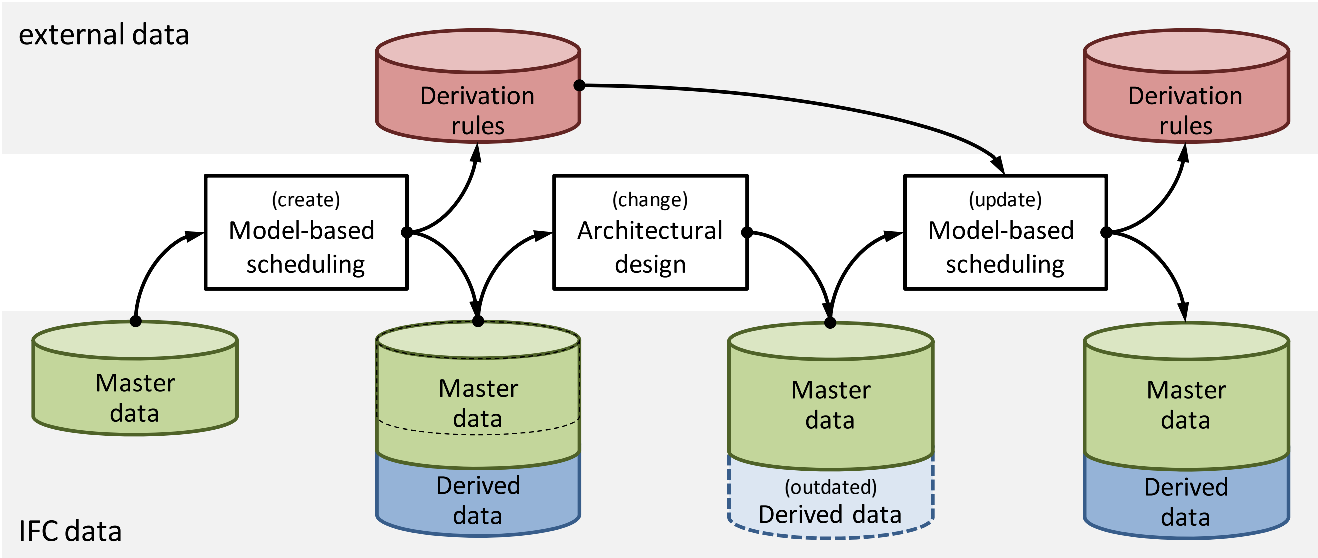

Figure 2: Iterative use case scenario and required data repositories for IFC-based scheduling

- How to differentiate between master data and derived data?
- How to link master data and derived data?
- How to set references to externally stored derivation rules?

The principle use case scenario of model-based scheduling is shown in Figure 2, which also describes the idea of model updates in case of design changes. It also shows the scope of IFC, which not only includes master data and derived data but also the link to externally stored derivation rules.

2 REQUIREMENTS FOR MODEL-BASED SCHEDULING

2.1 *Integration of three domains*

According to the requirements analysis from Tulke (Houttu et al. 2009) there are three domains that are of interest for model-based scheduling and 4D simulation:

- Architectural design
- Construction scheduling
- Cost management & Quantity take-off

Additionally, the interrelationships between these domains are an important issue as they enable the benefits of integrated design. This was also recognized in Staub-French & Fischer (2001), Tanyer & Aouad (2005) and others.

It is worth mentioning that domains do not imply ownership information or access right settings. For instance, we do not imply that elements of the architectural design domain such as walls, slabs, windows etc. are used by architects only. Instead, the differentiation into three domains is based on element types. A wall is for instance regarded as an element of the architectural design domain, even if "created" by the construction scheduler through element splitting as needed for 4D simulations or other purposes. Consequently, there is an overlap between these domains in terms of used element types, which also requires new tools for supporting integrated design.

2.2 *Requirements summary tables*

In the referred discussion paper data requirements are captured in four tables; one for each domain and one for the interrelationships. An example is given in Table 1. There are only two columns; the concept name and a short description. Thus the tables are not meant to provide a detailed definition of requirements. But they are detailed enough for a gap analysis and a principle discussion of extensions, which are using the same concepts and therefore can be compared very easily.

Table 1. Requirements table for the interrelationships between involved domains.

Cross-model links – requirements summary	
Concept	Description
Link between work tasks and building elements	It is needed for 4D simulation of the construction process and the calculation of quantities.
References to splitting rules/queries	Splitting rules are a way to speed up the definition of construction schedules and 4D simulations. Thus, they capture the design intend of the scheduler. In case of design changes they enable to redo the element splitting.
…	

3 IFC GAP ANALYSIS AND PROPOSED EXTENSIONS

3.1 *Related research and notations*

Main sources of our gap analysis have been the official IFC documentation from the IAI including the schema definition (IFC2x3 and IFC2x4), implementation guidelines (Liebich 2004), model view definitions (Hietanen 2006, MVD), implementation agreements (ISG), reported software certification results (Kiviniemi et al. 2008) and available information delivery manuals (Wix 2006, IDM). It also takes into account research results reported in various publications. But comparison of available work is sometimes difficult due to the use of different notations for the specification of required IFC subsets and implementation agreements. The following notations were found in the state-of-the-art analysis:

– Excel sheets, which are a straight forward documentation of agreements that were started with first IFC implementations and are still in use for the Coordination View (architectural domain).
– Various proprietary notations such as the VTT mapping definition that were used for describing project results and decisions (Serén & Karstila 2001).
– Aspect Cards (Karstila & Serén 2005), which is a notation that was developed in the ProIT project for supporting BIM implementations.
– Information Delivery Manual (Wix 2006), which is a BIM development methodology that puts special focus on processes and related data exchange specifications.
– Model View Definition format (Hietanen 2006), which is a notation for supporting the implementation of IFC data exchange scenarios. It comprises the definition of IFC subsets and further implementation agreements.

As the focus of our work was to provide solutions for identified gaps and to reuse existing agreements, a unified implementation guideline for IFC-based scheduling is still a pending task.

3.2 *Identified gaps*

The requirements summary tables like Table 1 provided the basis for the gap analysis. A first step was to map the concepts to the IFC data structure as defined in available IFC specifications. Table 2 shows the mapping of requirements from Table 1. It uses the same column for the concept names and a new column for the mapping definition. Again, it is not a detailed specification that enables to implement IFC interfaces. Further details, if available, can be obtained from the literature referenced in the discussion paper which provided the basis for the described concept mapping.

However, except for a completely missing mapping definition as it can be seen for the concept *"Reference to splitting rules/queries"* the mapping table does not enable to identify missing implementation agreements or conflicts with existing software interfaces. Thus the actual gaps are described in additional gap tables, which for each concept describes what is either missing in IFC or seems to be unclear for implementation. An example of such gap table is shown in Table 3, where for each concept further information about missing specifications is given. Finally, a summary table with a simple rating is given for each domain (Table 4), where the following marks are used:

– "++" no gap
– "+" supported by IFC , but not fully clear
– "o" implementation agreements needed
– "-" realizable, but with drawbacks or restrictions
– "--" not yet supported by IFC

Table 2. Mapping table for the interrelationships between involved domains.

Cross-model links – concept mapping	
Concept	IFC2x3
Link between work tasks and building elements	*IfcRelAssignsToProcess*
References to splitting rules/queries	-
…	

Table 3. Gaps table for the interrelationships between involved domains.

Cross-model links – Missing implementation agreements or IFC schema gaps	
Concept	Identified gaps
References to splitting rules/queries	Splitting of objects is defined through *IfcRelNests*, but is limited by implementation agreements of the Coordination View (see architectural design – element splitting) and does not define additional dependency management information.
…	

Table 4 provides the overall conclusion for dealing with interrelationships. Compared to other domains as shown in the tables 5-7 we can see many gaps, which are mainly due to yet missing implementation agreements. For many issues there exist alternatives for representing the information and thus the main question is how to implement a concept so that other software packages can deal with the information.

Table 4. Gap summary for the interrelationships between involved domains.

Cross-model links – gap analysis summary		
Concept	IFC2x3	IFC2x4 alpha
Link between work tasks and building elements	++	++
References to splitting rules/queries	-	-
Link between work tasks and quantity/cost information	o	o
Link between grids (axis) and construction zones	-	-
Containment of building elements in construction zones	-	o
Link between building elements and quantity/cost information	o	o

The architectural domain and IFC-enabled CAAD applications are seen as major sender of required data. Since these applications are implemented according to the Coordination View, which is a subset of IFC with additional agreements, the analysis was made not only for IFC but also for the Coordination View. This is important to know when using available CAAD applications, because they are typically based on the Coordination View.

The overall conclusion for the architectural domain is that there are only minor remarks. It therefore can be concluded that in particular the new IFC2x4 provides all main features for model-based scheduling.

Table 5. Gap summary for the architectural domain

Architectural design – gap analysis summary			
Concept	IFC2x3	IFC2x4 alpha	Coordination View (2x3)
Building structure	++	++	++
Building elements	++	++	++
Element splitting	+	+	o
Element representation	++	++	+
Element properties	++	++	-
Grids	+	+	+
Construction zones	--	+	--
Unique identification	+	+	o

The use of IFC for construction scheduling is discussed in several papers with very detailed mapping specification (Serén & Karstila 2001). Overall, there are no major gaps. However, there are issues about proper display of work tasks, e.g. to know about the work task order if shown in the list view of a scheduling software, and about the mapping between different ID concepts, e.g. to preserve proprietary IDs of objects. Furthermore, there are no agreements how to use available IFC functionality for definition of 4D visualization parameters.

Table 6. Gap summary for the scheduling domain

Model-based scheduling – gap analysis summary	
Concept	IFC2x3
Work task	++
Construction schedule	++
Time constraints	++
Logical dependency of work tasks	++
Hierarchical refinement of work tasks	++
4D visualization parameter and templates	o
Display and update mechanism	o

There is not much literature about the QTO domain. It is only the IFC documentation, an IDM specification and some definitions from the Aspect Card Library (Karstila & Serén 2005) that were available for the gap analysis. It therefore is a surprise that there are no gaps from the perspective of model-based scheduling. But to our knowledge there are no commercial or prototype applications that are using IFC for cost management & cost related quantity take-off. However, there are CAAD applications that are able to export so called "base quantities" of building elements. Such quantities provide a basis for cost-related quantities, which might be grouped to cost elements, multiplied with various impact factors or otherwise processed according to local cost management rules.

Table 7. Gap summary for the QTO domain

Cost management & QTO – gap analysis summary	
Concept	IFC2x3 (IFC2x2)
Cost value	++
Cost quantity	++
Cost impact factor	++
Cost group	++
Cost group properties	++
Cost item	++
Cost item properties	++
Cost classification	++
Cost item hierarchy	++
Cost schedule	++

3.3 *Suggested extensions and provided example*

As shown in the gap analysis most of our requirements are already fulfilled by IFC. Only minor extensions are needed to support construction scheduling, in particular 4D simulation. Most questions arise for implementation of "cross-domain links".

For the missing properties a proposal is made on basis of IFC property set definitions, which basically means to agree on the property name, its data type and the entities that are allowed to use the property. This solution is very flexible, does not require any change to the IFC schema and fulfils the main requirements. However, property sets cannot be assigned to relationship objects (*IfcRelationship* and specificly its subtype *IfcRelAssignsToProcess*) that in particular for 4D visualization properties would have been the preferred solution. Instead, these properties must be assigned to tasks, which have to be subdivided in case assigned building elements need different visualization parameters. It thus would mean that a m:n relationship between tasks and building elements (with m relationship objects) must be broken into m subtasks.

The definition of "cross-domain links" is facing a quite typical problem of rich and flexible product models; it is the question how to use available functionalities. IFC is an object-oriented model making intensive use of the inheritance concept in order to be maintainable and flexible. The drawback of such modeling style is surprisingly the flexibility that on one hand enables to deal with new requirements very easily but on the other hand offers several options for representing the data. But the options that are capable to cover our requirements might come with drawbacks, require additional implementer agreements or even require minor changes to the IFC schema. Thus, in order to understand the impact of our decisions for using or even extending the IFC schema we tried to explain the possible options with their advantages and disadvantages. An example is shown Table 8 that introduces possible solutions for representing the concept *"References to splitting rules/queries"* in IFC. A more detailed discussion with instantiation diagrams is given in our discussion paper. On that basis we finally made a decision for one solution, which was used with our prototype implementations and a detailed example. Furthermore, as our developments coincided with the IFC2x4 review phase we were able to give feedback to suggested extensions.

Table 8. Extension table discussing alternatives for identified gaps.

Cross-model links – proposed extensions	
Concept	Possible solutions
References to splitting rules/queries	There are several options that require further discussion: 1) Use references to external documents (*IfcDocumentReference*, *IfcRelAssociatesDocument*) 2) If element splitting is defined by construction zones it is possible to implicitly describe the splitting rule through assignment of building elements to construction zones via *IfcRelAggregates*, *IfcRelReferencedInSpatialStructure* or *IfcRelContainedInSpatialStructure*. 3) Use of constraints, which are defined by *IfcConstraint* and can be linked to IFC objects via *IfcRelAssociatesConstraint*.

4 PROTOTYPE IMPLEMENTATIONS

In order to demonstrate the current maturity level of the IFC data model with respect to model based scheduling, a workflow scenario using a model server was set up. In addition two prototype client software packages were developed which supplement existing project management software with BIM related viewing, linking and data exchange capabilities based on IFC.

4.1 *Scheduling Assistant*

The Scheduling Assistant (see Figure 3) is a complete 4D simulation package written in Java and Java3D which provides a fully featured IFC interface. It allows to import complete IFC models including geometry, properties, quantities, costs, scheduling and 4D information and provides comprehensive 3D, tree and list views with corresponding navigation and selection functionalities.

To ease the process of linking schedule tasks with other information the software provides an interpreter for linking rules including the ability to automatically split geometrical objects according to construction sections defined by a grid or zones. Once a 4D simulation is defined, it can be provided to other project members based on IFC for review.

The editing of schedule information is intended to be done outside in a feature rich and commonly used project management software package. MS Project is used for this purpose. To enable a quick, interactive and bidirectional communication with MS Project, the Scheduling Assistant utilizes the COM-interface to MS Project's proprietary data model. Thus data updates in both directions can be triggered by a single click.

Current implementation work focuses on a new functionality which allows to derive a construction schedule from an existing IFC building model as published in Tauscher et al. (2009). This semi-automatic process includes the creation of links between tasks and corresponding building elements as needed for 4D simulations.

The purpose of the development of the Scheduling Assistant was to provide an open, IFC based 4D software package as basis for research which allows full access to the source code and thus gives the possibility to easily adjust the data mapping and add new advanced functionalities not provided by commercially available applications. It is intended as platform for IFC based prototype developments for other domains, too. Internally it consists of four packages all of which are written in Java and in-tended to be published as open source on the website www.openifctools.eu.

On the lowest level an IFC Toolbox provides read and write capabilities for IFC STEP physical files as well as data management functionalities.

On the second level there is a Java3D loader package which interprets the geometry information of an IFC model and returns a scene graph ready for visualisation. In order to cope with CSG operations, it utilizes a Boolean Modeler package developed based on the algorithm given in Hubbard (1990). This package also provides the object splitting algorithm as published in Tulke et al. (2008).

On the third level there is the 4D user interface. The first three packages are designed in a way that they can be used stand alone, too. All packages were tested with real project data and are scheduled to be published under the given website until spring 2010.

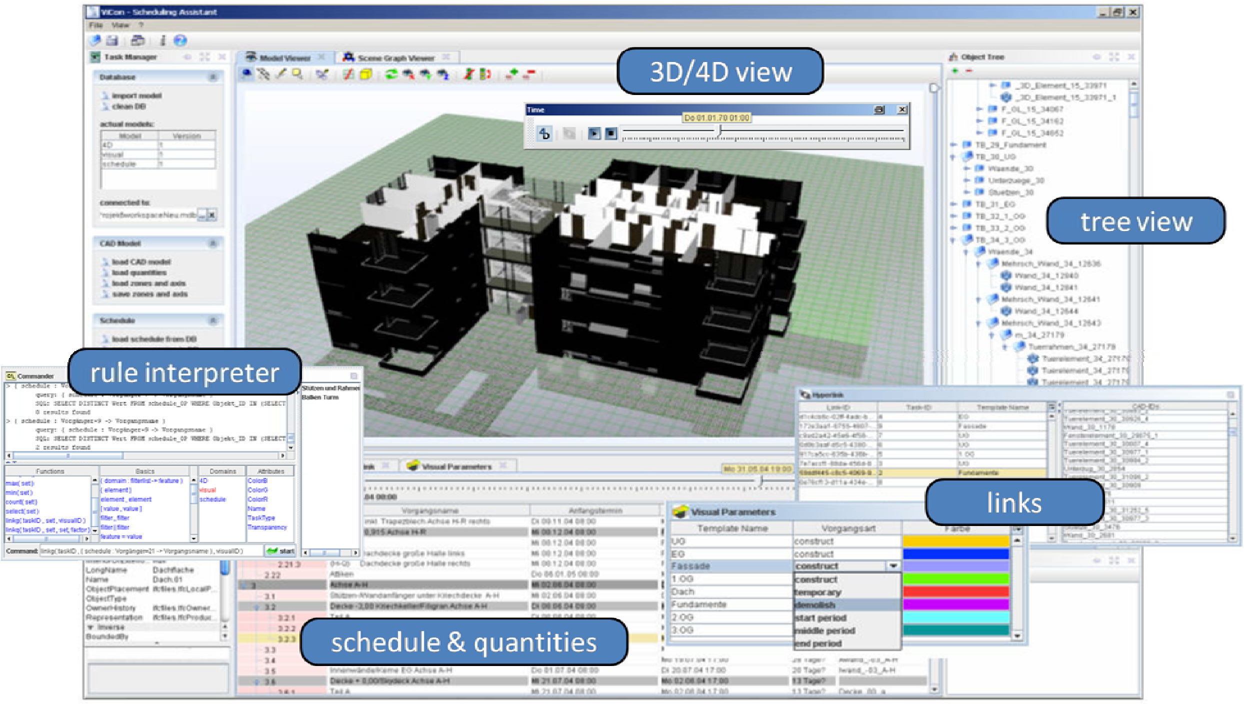

Figure 3: User Interface of the Scheduling Assistant

4.2 *IFC-interface plug-in for MS Project*

In order to make the functionality for storing scheduling information in the IFC format usable for schedulers from practice, it was decided to additionally create an extension of the Microsoft Project application.

The extension is developed as a ‘Microsoft COM Add-in’ application and written in MS Visual Basic 6.0. This means that the extension can be used by MS Project 2000 or any later version. The plug-in supports 4D visualization using the Microsoft DirectX 8.

MS Project can be used as usual but now offers three extra menu items. The first menu item (IFC import) enables data roundtrips that is to first import an IFC file, to create the construction schedule in MS Project and finally to write an updated IFC file without any data loss (second menu item – IFC export). Through further setting options it is also possible to create a new IFC file which then includes only the construction schedule. The third menu item (4D visualization) is interesting in case the imported IFC model contains a construction schedule which is already linked to building elements, so that the construction process can be visualized depending on the time line of the schedule. It is also possible to adjust imported and linked schedules so that it is very easy to create alternative construction schedules or to make changes at any stage within the process. The only limitation of the developed plug-in is that a construction schedule cannot be linked with building elements. This has to be done in the Scheduling Assistant.

The MS Project plug-in requires to read and write IFC files as well as to visualize 4D simulations. These functionalities are provided by the IFC Engine toolbox, which is available at www.ifcbrowser.com and can be used free of charge in case the application on top of this toolbox is distributed without costs. Since our plug-in is intended to be distributed as open source, there will be no license fee for using the IFC Engine toolbox, too.

At the time of writing this paper, the 4D visualization is not implemented with all features as our focus so far has been to provide full support for the IFC import and export. Currently the source code is available for our InPro project partners as well as for InPro Cluster members, so that the plug-in can be further adjusted to fulfil specific company’s requirements.

5 CONCLUSION AND FURTHER WORK

It has been shown that the IFC 2x3 schema specification provides a reasonable basis for the use case of model-based scheduling. Most of our requirements including the reference to externally stored linking rules are supported or can be handled by implementation agreements and property set definitions. However, starting our prototype developments required to review specifications from many different sources and to make decisions about unclear specifications or alternative solutions. The main reason for this consolidation work is that unlike to the architectural design domain, implementation of IFC interfaces for construction scheduling and cost management as well as QTO is not yet in focus of software vendors. But once these implementation agreements are available, IFC interfaces can be added to existing applications with reasonable efforts. This has been shown with the MS Project plug-in on top of the IFC Engine toolbox, which even supports 4D visualization.

The suggested linking approach has been implemented in the Scheduling Assistant tool, which enables to link architectural data with construction schedules. The tool also supports the splitting of building elements according to construction zones in order to match the granularity of construction processes. This kind of semi-automatism enables to save resources for integration of domain data. It also helps in case of design changes to update the links with only few user interactions instead of doing the whole linking and refinement work again. However, due to yet missing IFC data from the cost management & QTO domain it was not possible to test all aspects of the suggested approach.

For IFC-based data integration, the main task for future developments will be to implement interfaces to applications that do not have their focus on 3D geometry. This requires to agree on implementation guidelines and to provide a set of test examples as it has been started with our discussion paper (Weise et al. 2009). The more applications can provide different kinds of IFC data the better for testing more advanced integration concepts.

ACKNOWLEDGEMENT

We would like to thank the European Union for funding the InPro project (IP 026716-2) that enabled these developments.

REFERENCES

IFC Engine DLL (2009): http://www.ifcbrowser.com/ifcenginedll.html

Fijneman M., Matthyssen A., Verhofstad F. & Tulke J. (2008): D49B – Process Mapping for Key Process 4 – Construction Schedule. *Intermediate Deliverable of the NMP-EU project InPro (IP 026716-2)*, 8 August 2008.

Froese T. & Yu K. (1999): Industry Foundation Class Modeling for Estimating and Scheduling. *Proceedings of the CIB W78 conference*, 1999, Ottawa, Canada.

Heesom D. & Mahdjoubi L. (2004): Trends of 4D CAD applications for construction planning. *Construction Management and Economics, Volume 22, Number 2*, February 2004.

Hietanen J. (2006): IFC Model View Definition Format. International Alliance for Interoperability, April 2006, available at: http://www.iai-international.org/software/MVD_060424/IAI_IFCModelViewDefinitionFormat.pdf

Houttu J.-M. ed. (2009): D15 – Early Design Applications. *Restricted report of the NMP-EU project InPro* (IP 026716-2), to be published in 2009.

Hubbard P.M. (1990): Constructive Solid Geometry for Triangulated Polyhedra, *Technical Report CS-90-07*, Brown University, Department of Computer Science

Karstila K. & Serén K. (2005): IFC Aspect Card Library. Result of the Pro-IT project, Eurostep.

Kähkönen K. & Leinonen J. (2003): Visual product Chronology as a Solution for Accessing Building Product Model Data. *Proceedings of the CIB W78 conference*, Auckland, New Zealand.

Koch C., Firmenich B. (2006): Operative models for the introduction of additional semantics into the cooperative planning process. *Proceedings of the 13th EG-ICE Workshop "Intelligent Computing in Engineering and Architecture"*. Ascona, Schweiz, June 2006.

Kiviniemi A. Tarandi V. Karlshøj R. Bell H. Karud O.J. (2008): Review of the Development and Implementation of IFC compatible BIM. *Erabuild report*.

Liebich T. ed. (2004): IFC2x Edition 2 – Model Implementation Guide. Version 1.7, March 18, 2004, available at: www.iai-international.org.

Microsoft MS Project COM Add-in (2009): http://msdn.microsoft.com/en-us/library/aa164883(office.10).aspx

Porkka & Kähkönen (2007): Software Development Approaches and Challenges of 4D Product Models. *Proceedings of the CIB-W78 conference*, 2007, available at: http://itc.scix.net/data/works/att/w78-2007-013-096-Porkka.pdf

Serén K. & Karstila K. (2001): MS Project – IFC Mapping Specification. *Report from VTT and Eurostep for IFC1.5.1, IFC2.0 and IFC2x.*

Staub-French S Fischer M. (2001): Industrial Case Study of Electronic Design, Cost & Schedule Integration, CIFE, Stanford University, Technical Report 122

Tauscher E. Mikulakova E. Beucke K. König, M.(2009): Automated Generation of Construction Schedules based on the IFC Object Model. *2009 ASCE International Workshop on Computing in Civil Engineering*, Austin, Texas, U.S

Tanyer A. M. Aouad G. (2005): Moving beyond the fourth dimension with an IFC-based single project database; *Automation in Construction, Volume 14 page 15-32*

Tulke J. Nour M. and Beuke K. (2008): Decomposition of BIM objects for scheduling and 4D simulation. *Proceedings of the 7th European Conference on Product and Process Modelling in the Building and related Industries.* A. A. Balkema Publishers, September 2008.

Wix J. ed (2006): Information Delivery Manual: Guide to Components and Development Methods. Available at: http://idm.buildingsmart.no

Weise M., Liebich T., Tulke J. & Bonsma P. (2009): Disscussion paper: IFC support for model-based scheduling. *Public specification of the NMP-EU project InPro (IP 026716-2)*, March 2009, available at: www.inpro-project.eu

CITED SPECIFICATIONS

IAI, IFC	http://www.iai-international.org/
IDM	http://idm.buildingsmart.no/confluence/
IFC2x3	http://www.iai-tech.org/products/ifc_specification/ifc-releases/Ifc2x3-tc1-release/summary
IFC2x4	http://www.iai-tech.org/products/ifc_specification/ifc-releases/ifc2x4-release/alpha-release
IFD	http://dev.ifd-library.org/index.php/Main_Page/
ISG	http://www.iai.hm.edu/
MVD	http://www.blis-project.org/IAI-MVD/
MSG	http://www.iai-tech.org/