Model-based Scheduling for Construction Planning

Veronika Hartmann & Karl E. Beucke  
Bauhaus-Universität Weimar, Germany

Kateryna Shapir & Markus König  
Ruhr-Universität Bochum, Germany

Abstract

Proper and effective construction scheduling is an essential task in the context of the realization of any construction project. For this purpose, Building Information Modeling (BIM) is used more and more to support the project manager in this complex as well as time- and cost-consuming assignment. The main idea is to use the link between a schedule and a BIM model not only for the visualization of the building processes, but to use the model as a base for creating the schedule. This paper presents an extended BIM-based scheduling concept.

Planning processes not only involve building elements, but also constraints like construction approvals, weather conditions, funding etc. These elements play an important role in time and risk management and therefore must be considered in the scheduling phase. However, currently they are not modelled in standard BIM models based on the Industry Foundation Classes (IFC). The first extension presented in this paper is the introduction of so called abstract elements that are not part of the IFC model. They are integrated into the BIM-based scheduling logic.

A further extension is the detailed definition of states of elements. The realization of an element usually involves several processes. In a detailed schedule each process has to be taken into account independently. This is described by assigning different states to the considered element with the states containing information about the related process. To reflect interdependencies between processes that are partly parallel in time, intermediate states are provided. Thus, the start of one process can be defined related to any point of progress within another process. The considered process is automatically split accordingly. Finally, a hierarchical data structure is provided to organize the different levels of detail within one schedule. To illustrate the introduced concept the results of a case study are presented.

Keywords: BIM, scheduling, construction, IFC.

1 Introduction

The realization of construction projects involve numerous experts providing a huge quantity of information documented in models, design documents, technical drawings or reports. In the context of construction scheduling more and more BIM is used to support the project manager to organize a project efficiently. In general, BIM-based scheduling tools enable the linkage of building elements of a 3D model with the corresponding construction processes. This allows for the visualization of the defined processes which helps the project manager to check for completeness and correct order of the
schedule. This idea is carried further by not only linking a schedule with a BIM model, but using the model as the source of information to create the schedule.

Creating a schedule is a time-consuming task and thus the reuse of data is of interest. The reuse of a complete schedule without any adaptions is normally not possible due to the fact that the realization of a building is usually a unique process. Adapting complex data always bears a great risk for mistakes. Nevertheless, the reuse of data is possible and sensible. Therefore, the base of the work presented in this paper is an existing BIM-based scheduling tool that links an IFC model to a Case Based Reasoning (CBR) system. Information about the objects that are to realize, the way they can be realized as well as all constraints that need to be taken into account is contained in cases. Cases are stored in the database of the CBR system and can be retrieved, adapted and reused. Thereby the constraints which are essential for the creation of correct schedules are described by the involved elements in certain states. This paper focuses on the elements as well as their states.

Current BIM models usually only contain information about building elements of a project. However, they are not always sufficient to describe all constraints required to model construction processes. To create a schedule additional constraints have to be taken into account, e.g. approval, funding or weather conditions. In this work new concepts are presented that enable the user to define additional elements. They reflect non-structural constraints, which are integrated into the BIM-based scheduling logic.

Different states of an element reflect different processes that are necessary for its realization. As soon as a more detailed schedule is created more than one process will be assigned to one element. This paper presents an approach that allows for a flexible definition of detailed states. It also considers the fact that often processes start before all of their predecessors are completed. To describe the required interdependency, intermediate states are introduced. Therefore, a hierarchical data structure is provided to organize the different levels of detail within one schedule.

The paper will give an overview of the developed data structure for defining and handling elements and their states. It will summarize the theoretical background and methods used to realize the scheduling tool. Finally, the results of a case study are presented to illustrate the introduced concept.

2 State of the art

Proper and effective construction scheduling is an essential task in the context of the realization of any construction project. Currently, existing scheduling tools do not provide adequate support for this complex as well as time- and cost-consuming assignment. There are two main types of software in civil engineering used for construction scheduling: project management software and 4D simulation software.

2.1 Project management

The standard practice in modern construction-process planning is the Precedence Diagram Method (PDM), including the well-established methods such as Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT) (Winch and Kelsey, 2005; Besner and Hobbs, 2008). The usage of these techniques allows for the calculation of the duration of a project and the identification of potential risks, e.g. processes in the construction sequence that are critical in time. However, the definition of processes and their dependencies is performed manually, resulting in imprecise and error-prone schedules due to the high complexity of the projects. During construction frequent changes occur but updating a schedule without adequate computer support is extremely demanding and associated with eventual loss of accuracy. Additionally, each schedule is created from scratch despite of their similarity to former projects. These and other deficiencies of the PDM incited many researchers to explore the automatic support for the generation of schedules. Thereby, the main focus
of the recent research is virtual representation of the construction-process, i.e. 4D animation or simulation.

### 2.2 4D simulation

With the increasing availability of 3D building models, the scheduling process has been supplemented with animations of the construction progress. For this purpose building geometry is linked with temporal information and construction processes of the schedule. Because of the identified disadvantages of manual linking (Koo and Fischer, 2000), further research has investigated the automatic linking between product and process model (Staub-French et al., 2008; de Vries and Hanrik, 2007). Several rule-based, e.g. in Dzeng and Wang (2003), as well as constraint-based linking approaches, e.g. in Huhnt and Enge (2006), König et al. (2006), Tauscher et al. (2007) have been presented. The generated visualization of scheduled processes permits the verification of technical feasibility and completeness of schedule.

There is no definitive preference of the 4D simulation software in the context of construction scheduling. Nevertheless, BIM-based 4D software, especially IFC-based, are becoming increasingly important in research and practice (Howard and Björk, 2008; Watson, 2010). Some BIM-based scheduling tools integrate BIM into the design phase of construction projects and enable the reuse of data from previously executed projects (Tulke, 2008; Weise et al., 2009; Mikulakova et al., 2010). The advantage of IFC standard is that most of the required object-definitions are provided. However, within the computer-aided design (CAD) applications there is a lack of concepts to integrate non-building elements into an IFC model.

### 2.3 Reuse of process knowledge

One of the well-known and effective approaches for solving new design problems by reapplying and adapting the solutions from previous similar experiences is CBR. Every solved case is stored in a case base and is separated into two parts: a problem and a solution. Within the typical four phases of the CBR cycle (retrieve, reuse, retain and revise) the most similar solution for a new problem can be found, reused, adapted and stored in a case base for future disposal (Aamodt and Plaza, 1994). Despite the prevalence of CBR very few case-based expert systems have been developed for the construction planning domain, such as CasePlan (Dzeng and Tommelein, 2004), CBRidge Planner (Tah et al., 1999), CONPLA-CBR (Ryu et al., 2007) and Alice (Tauscher et al., 2007, Mikulakova et al., 2010). The concept of Alice is described in more detail in the following section, since it is the base of the work presented in this paper.

### 2.4 Alice

Alice is a CBR-based scheduling tool linked to a BIM. Cases stored in the database of the CBR system contain information about one or several objects to be realized, the way they can be realized and about constraints to be fulfilled. The constraints are described by building elements in certain states and are considered as prerequisites and results of processes. This concept distinguishes only between two states: non-existent and completed. The need for the definition of more detailed states has been recognized as well, but it is not implemented so far.

A case contains a problem part which describes the as-is state (prerequisites) as well as the target state (results). The solution part stores the process that transfers the as-is state into the target state. The tool automatically compares the building elements of the current IFC model with those in its database and suggests related existing cases. Once all elements have sufficient processes assigned, either by retrieve and adaption from the database or by new definition, the processes are put into a correct order so that all constraints are respected.
3 Extended BIM-based scheduling concept

This work presents an extended concept based on the research of Tauscher et al. (2007) and Mikulakova et al. (2010). The basic concept described in the subsection 2.4, is sketched in Figure 1. The highlighted boxes are the extensions outlined in this paper. The idea of describing constraints by means of elements in certain states is adopted. However, it is extended by the possibility of defining additional elements, which are not stored in currently used IFC models. These elements can also be used to describe constraints of processes. Additionally, more precise states can be defined in a transparent and user-friendly way, while detailing a schedule. Realistic schedules have to be able to reflect different levels of detail. Consequently, a hierarchical structure to store the related cases is introduced.

Figure 1. Extended BIM-based scheduling concept.

3.1 User-defined elements

For building elements all needed information can be extracted from the IFC model. As planning processes not only involve building elements so-called abstract elements are implemented. This extension enables the user to define any kind of element, such as documents, materials, resources or weather conditions. These elements can be organized in a hierarchical structure providing abstract element types and super types similar to the IFC object hierarchy.

Apart from the predefined attributes (id, name and type) of abstract elements it is possible to save any element-specific information (e.g. material, position) as properties. Due to a simple and uniform structure the abstract elements can be treated in the same way as the building elements of the IFC model. On the one hand this is relevant for the similarity measure calculation between the elements during the CBR cycle, c.f. Mikulakova et al., 2010. On the other hand it provides an opportunity to integrate abstract elements in the IFC file, using existent IFC objects, e.g. IfcActor, IfcResource, IfcDocumentType or IfcCost. Thus any constraint can be determined as a pair of element and state.

3.2 States of elements

Results and prerequisites of processes can be defined on the basis of the involved elements and the basic states non-existent and completed. The process building column, for example, has foundation in the state completed as prerequisite and column in the state completed as result. However, in more detailed schedules elements may have more than one process assigned. In order to ensure a non-ambiguous definition of prerequisites and results, the states have to contain information about the considered process. The states of a column can be e.g. formwork completed, reinforcement completed,
concreting completed. This idea is similar to the concept presented in Huhnt and Enge (2006). However, this approach is based on the idea that each element type has a predefined ordered set of status variables describing the construction processes. Within the paper at hand a different approach is adopted. Each element can have several states which are defined independently from each other. The construction process is broken down into basic processes. Each state is the result-state of such a basic process and can be a prerequisite-state of any other process. The interdependencies between the processes are described by prerequisites, e.g. a prerequisite of the process *placing reinforcement* is that the considered element has reached the state *formwork completed*. Consequently, the correct sequence of processes can be computed automatically. Furthermore, the user can assemble the single processes to more complex process patterns. This allows for more flexibility in the reuse of data than a predefined chain of processes. It also means that identical processes do not have to be defined several times for different types of elements. For example, the process of painting a wall is the same for a wall of concrete, brickwork or timber.

Another reason to take into account detailed element states is that in construction scheduling the definition of overlapping processes is often necessary. This means that processes can start before all of their predecessors are completed. Therefore, the definition of states is further detailed by including the progress of the process: e.g. instead of *excavating completed* the state is *excavating 100% completed* which enables the user to also define *excavating 80% completed* or any other intermediate state between *non-existent* and *completed*. Consequently, this leads to a hierarchical process and state definition.

### 3.3 Hierarchical structure

The simplest relationship between processes is a linear sequence, which can be computed automatically when prerequisites and results are defined non-ambiguously. Introducing the possibility of refining processes requires a different type of relationship. Cases describing different levels of details cannot be mixed in one schedule. Therefore, the cases are stored in a hierarchical structure as shown in Figure 1. If a schedule in a certain level of detail is requested, only the corresponding cases are used to generate it.

One typical example for refining a process is the detailing of a process by breaking it down into sub-processes. The process *built column* can be refined by the sub-processes *set formwork*, *place reinforcement* and *pour concrete*. A different demand for refinement is the consideration of detailed progress information and overlapping processes. If an element is required to be in a certain intermediate state for defining a prerequisite the relevant process needs to be split. The problem is illustrated in Figure 2 and explained in the following.

For the case *paving of the ground* the process *paving* shall start before the process *excavating* is fully completed. Its prerequisite is thus defined as e.g. *ditch* in the state *excavating 70% completed*. Therefore, it cannot be correctly linked to the case *excavating* as its result-state is *excavating 100% completed*. Asking the user to foresee all necessary intermediate states and define the cases accordingly is too demanding and complex. For this reason the automatic splitting of cases is implemented, which creates two new cases based on the original one. The first case has the same prerequisites as the original one and its resulting element is in the required intermediate state, here *ditch excavating 70% completed*. This is at the same time the prerequisite of the second newly generated case. Its result is the result of the original case, here *ditch excavating 100% completed*. Thus, the sequence can be computed automatically. The completion of the process *excavating* is ensured and the process *paving* can be inserted at the required point. As a consequence, the logical order can be produced by using start-end relations only. Obviously, the hierarchical structure is necessary to define the relationship between the original case and those generated by splitting.
4 Case Study

The presented concept is tested within an infrastructure project where an industrial zone is developed from a rural area. Several high voltage cables, pipes and roads have to be relocated. New infrastructure has to be constructed and rain detention basins are built. The realization of the latter has been chosen as example here. The basic processes for such a rain detention basin are: excavating, paving of the ground, planting greenery. All three processes are assigned to the element rain detention basin as the ground and the greenery are not modeled. The processes result in three different states: excavating 100% completed, paving 100% completed, planting greenery 100% completed.

This project deals mainly with processes that extend activity over a large area or distance. Hence, often different processes assigned to one element are running in parallel. The process paving starts before the process excavating is fully completed. To reflect this, the rain detention basin in an intermediate state of the process excavating is defined as prerequisite of the process paving. The exact value of the intermediate state depends on external constraints like resources and space and is to be defined by the project manager. Here it is excavating 80% completed. The case describing the excavation is split automatically.

Additional constraints like approval and funding play a key role and have significant impact on the determination of the start of a process. They can be taken into account as abstract elements and inserted as a prerequisite to further activities. The definition of such a process is shown in Figure 4. As soon as the definition of cases for all processes is completed, the schedule is generated automatically. The generated schedule for the example above with some further processes is shown in Figure 5.
5 Conclusion and outlook

The extensions shown in this paper transfer the former basic approach into a more advanced one. The presented introduction of abstract elements allows covering the whole range of objects involved in a building process. Due to the flexible definition of states, including intermediate states, all possible constraints can be taken into account. The automatic splitting of cases enables the user to define in a simple way processes that are partially parallel in time. The hierarchical structure of cases ensures the correct handling of different levels of detail within one schedule.

However, further investigation is needed to integrate information about quantities. Beyond this, an appropriate visualisation of user defined elements will be realized. It is also of interest to export the abstract elements defined by the user to the IFC model. Thus, all data could be stored in only one model and would be available for further applications. Since an important part of the CBR system is the similarity determination of cases, the existing feature has to be adapted as to integrate the presented extensions. Finally, the automatic detection of process patterns would be a useful extension of the presented concept. It requires a more advanced data structure and would further enlarge the possibility of reusing data in a sensible way.
Acknowledgements

The authors gratefully acknowledge the support of this project by the German Research Foundation (DFG) and OBERMEYER Project Management GmbH.

References


