Advanced Engineering Informatics 24 (2010) 379–388

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

Bridge construction schedule generation with pattern-based construction methods and constraint-based simulation

I-Chen Wu [a,*], André Borrmann [b], Ulrike Beißert [c], Markus König [d], Ernst Rank [b]

[a] Department of Civil Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan
[b] Chair of Computation in Engineering, Technische Universität München, Munich, Germany
[c] Chair of Construction Engineering and Management, Bauhaus-University, Weimar, Germany
[d] Institute for Computational Engineering, Ruhr University Bochum, Bochum, Germany

ARTICLE INFO

Article history:
Received 18 June 2010
Received in revised form 5 July 2010
Accepted 7 July 2010
Available online 31 July 2010

Keywords:
Construction method
Constraint-based simulation
Monte-Carlo method
Process pattern
Schedule generation

ABSTRACT

This paper presents a novel methodology which assists in automating the generation of time schedules for bridge construction projects. The method is based on a simulation of construction works, taking into account the available resources and the interdependencies between individual tasks. The simulation is realized by means of a discrete-event simulation software originally created for plant layout in the manufacturing industry. Due to the fact that the fixed process chains provided are too rigid to model the spontaneous construction task sequences, a constraint module that dynamically selects the following task has been incorporated.

Constraint module input data is formed by activity packages comprising of the affected building element, the required material, machine and manpower resources, as well as the technological pre-requisites of the activity to be performed. Since manual creation of the large set of activity packages is laborious and error-prone, a 3D model-based application has been developed which allows the interactive assignment of construction methods to individual building elements. To facilitate this process, a level-of-detail approach has been implemented which allows the user to successively refine both the process model and the corresponding product model.

The discrete-event simulation system uses all the given information to create a proposal for the construction schedule automatically, which may then be refined using standard scheduling software.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Scheduling a construction project means to coordinate resources of workers, machines and materials in a time-efficient way in order to realize a construction project within the projected time and costs.

Traditionally, construction schedules are manually specified using Gantt chart techniques and the critical path method (CPM). A number of commercial management software solutions in the industry use these two concepts. However, the software is unable to assess schedule correctness, especially of process duration for a given amount of available resources, as well as its inability to optimize the schedule according to total costs or total duration work against the application of these methods within more complex scheduling tasks.

The simulation of construction processes has proven to be a suitable approach for detailed investigation of construction schedules [1]. In this case, individual activities, their dependencies and the availability of resources are taken into account. However, preparing the input data for such a simulation is a time-consuming and error-prone process. This paper introduces a new methodology which is based on interactively refining both the building model and its corresponding process model. It guides the scheduler and dramatically facilitates the generation of input data for the process simulator. The result of the interactive process is a large set of '*activity packages*' which combine atomic activities with the required resources, such as labor, material, and equipment, as well as establish links to the preceding activities.

These process components cover all information required to run the simulation. In the presented approach, the constraint-based simulation is employed, which exhibits the necessary flexibility to model construction processes with greater realism.

2. Related work

Since the 1960s, it has been recognized that discrete-event simulation (DES) provides a powerful tool to model and evaluate construction processes, including the overall project duration as

* Corresponding author. Tel.: +886 7 381 4526; fax: +886 7 383 1371.
E-mail addresses: kwu@cc.kuas.edu.tw (I.-C. Wu), borrmann@bv.tum.de (A. Borrmann), ulrike.beissert@uni-weimar.de (U. Beißert), koenig@inf.bi.ruhr-uni-bochum.de (M. König), rank@bv.tum.de (E. Rank).

1474-0346/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.aei.2010.07.002

well as the utilization of resources. The link node model developed by Teicholz [2] is the earliest known method for construction simulation. Subsequently, researchers employed general purpose DES programs such as GPSS [3,4] to simulate construction processes.

To give the user the possibility of focusing on construction-specific processes, the domain-specific simulation programs CYCLONE [5] and ICONS [6] have been developed. Their successors INSIGHT [7], MicroCYCLONE [8], DISCO [9], STROBOSCOPE [10] and SIMPHONY [11] further facilitated the preparation of the simulation engine's input data and the interpretation of its output.

Discrete-event simulation helps in analyzing defined workflows (usually represented by directed graphs) and to identify possible bottlenecks by providing the means to study resource utilization. By adapting the amount of resources employed, the user is able to carry out what-if analyses [12]. However, finding the optimal amount of resources requires a vast amount of simulation runs [13]. Researchers have therefore proposed the integration of DES with heuristics [14] and sophisticated optimization techniques such as Tabu search [13], genetic algorithms [12,15], simulated annealing [16], and Particle Swarm Optimization [17,18].

However, most of these approaches do not consider the optimization of the topology of the activity graph. This refers to the question of determining which activity to start first if a specific resource is required by several activities, where its availability is limited in amount or capacity. It is well known that for more complex construction processes, this "resource flow" [17] is the main source for optimizing the overall construction process.

Using the constrained-based simulation approach introduced by Beißert et al. [19,20], valid execution schedules considering the resource availability can be generated automatically using DES. Contrary to CPM and other simulation frameworks, like CYCLONE or SIMPHONY, this approach does not require an explicit process chain (i.e. activity graph) to be modeled. Instead, conditions for executing an activity, such as technological preconditions and the availability of certain resources, are modeled locally as constraints on the respective activity. Consequently, the constraint-based approach guarantees a high flexibility of modeling construction processes, if additions or new pre-requisites in processing occur. The model can be easily adapted by simply adding or removing certain constraints. During runtime, the DES system checks for activities where all constraints have been fulfilled. It randomly selects from these activities the next activity to be executed as long as sufficient resources are available. On the one hand, this approach dramatically increases the solution space, since it accounts for all variations of the activity sequence. On the other hand, it facilitates schedule calculation, because the creation of a global model of the entire construction process is not required. Furthermore the scheduling decisions become more transparent to other involved persons of the project.

Depending on the problem definition, the solution space may be extremely large. To find a good solution, a Monte-Carlo analysis may be applied, generating a significant set of solutions which can be later analyzed against project objectives. The constraint-based simulation approach can be coupled with Greedy randomized adaptive search procedures [21] or simulated annealing [22] to reduce the number of simulation runs, thereby speeding up the search for global optima.

The methodology presented in this paper makes use of the constraint-based simulation approach while focusing on the preparation of the input data. For most simulation systems, the task of generating input data remains tedious, time-consuming and error-prone. This has been identified by many researchers as the main cause of the slow adaptation of simulation technology in the construction industry. This is especially true for small scale projects where low budgets prevent long and laborious preparation phases.

As a possible solution, the integration of DES with 3D product models has been proposed. For example, the simulation system for heavy earthmoving operations presented in AbouRizk and Mather [23] has been integrated with a 3D CAD model. In this case, simulation models are automatically generated from the CAD model. A similar approach is followed by Chahrour and Franz [24]. Wang et al. [25] developed a 4D management system for construction planning and resource utilization, where a 3D geometrical model is linked with resources to compute the resource requirements. However, all of these approaches rely on the utilization of predefined CAD components whose definition includes a description of the construction processes required to build them.

The methodology proposed in this paper aims at enabling the scheduler to use any kind of 3D model (i.e. product model) and interactively build the necessary input data for a constraint-based simulation by assigning construction patterns to individual components. Using this methodology provides further flexibility to the schedulers. The construction patterns encapsulate basic knowledge of a construction method, such as the composing activities and their precedent relationships. Following the concepts of construction method modeling [26], the proposed methodology makes use of a hierarchical approach which allows the scheduler to subsequently refine both the product model and the assigned construction patterns.

The proposed methodology is illustrated using a bridge erection example. Currently, only a small number of researchers have applied simulation technology on bridge construction processes. In Huang et al. [9] the erection of cable-stayed bridges is simulated using DISCO, a graphical user interface for the MicroCYCLONE simulation engine. In Hong and Hastak [27], the application of fiber-reinforced composites for the rehabilitation of bridge decks is compared against precast concrete using CYCLONE methodology. In Zhang et al. [28], the advantage of cell-based representation and analysis of spatial resources is discussed using the example of re-decking works at a bridge in Montreal. Said et al. [29] compared the construction of bridge decks with balanced cantilevers cast in situ against one using precast cantilevers using the STROBOSCOPE simulation engine.

Besides discrete-event simulation, there are also other technologies used for generating schedules. This includes agents-based approaches [30,31], for example. Other researchers are tackling the complexity of coordinating schedules among a multitude of concurrent projects [32]. However, this is out of scope of the work presented here.

3. Constraint-based simulation

3.1. General approach

The traditional process simulation approach, where rigid sequences of activities are defined (i.e. the preceding and succeeding activities are specified in advance), is only suitable for processes which are mainly driven by machines, such as earthwork processes. However, most construction processes have dynamic and spontaneous sequences of activities.

The constraint-based simulation approach has been developed to overcome the limitations of fixed activity graphs and to realize greater flexibility [19]. In this case, the scheduling problem is described as a constraint satisfaction problem [33], i.e. for each construction activity, all requirements for its execution are captured as constraints. This includes the requisite preceding activities, equipment, manpower, materials and space [19,20]. The solutions to the constraint satisfaction problems become valid execution orders for construction activities when all the associated constraints are fulfilled.

The analytical solution of complex constraint satisfaction problems is extremely time-consuming. In contrast, simulation methods can be used to investigate and evaluate different scenarios very quickly. For this reason, the constraint satisfaction approach was incorporated into a discrete-event simulation application.

Every construction process can be decomposed into atomic activities, referred to as process steps. Each atomic activity has a status of execution and is performed without interruption or changes to its associated employees, working space, and other resources. Whenever an activity is finished, an event is triggered and all activities which are yet to commence are checked for the fulfillment of their constraints. From the resulting set of executable activities, one is randomly chosen for execution and the required resources are locked. The process of constraint checking and random selection is repeated, until no more tasks can be started at the current time step (event). Once the remaining time allocated for a certain work step has expired, the activity is marked as finished. The specifically reserved work force and materials are unlocked and can be employed for other construction activities.

All events, such as the starting and finishing of activities as well as locking and unlocking of resources, are recorded. Each simulation run produces one practical and valid execution schedule, its material flow, as well as its corresponding utilization of human resources and plant. The material flow, utilization of human resources, and total process time of the simulation run can then be analyzed.

To incorporate execution strategies, the constraint-based simulation concept has been extended by soft constraints. In contrast to hard constraints, which need to be fulfilled before a construction activity can be started, soft constraints specify functional conditions which can be violated within a certain range. For more detailed information, please refer to Beißert et al. [21].

The constraint-based simulation concept is implemented using the simulation framework Plant Simulation by Siemens PLM software and the Simulation Toolkit for Shipbuilding (STS). The STS was developed by Flensburgers shipbuilding, the SimCoMar cooperative agreement, the Bauhaus University Weimar and the Ruhr University Bochum. The components of the STS are presented in detail in [34].

3.2. Monte-Carlo analysis

The random selection of the activities to be executed in the following step can result in very different overall project durations (Fig. 1). Since real-world projects carry a very large number of possible sequencing configurations which cannot not be evaluated individually, we perform a Monte-Carlo analysis [35] for identifying the good solutions, where the constraint-based simulation is run with exactly the same input data a large number of times (>1000). Due to the randomness of activity selection, each simulation will result in a different sequencing configuration and thus produce different project durations and resource utilization.

Though we can state that the probability of finding a near-optimal solution increases with the number of simulation runs, finding the optimal schedule is not guaranteed. Other, more advanced optimization techniques, such as the Greedy randomized adaptive search procedures [21] and simulated annealing [22] have been combined with the constrained-based simulation approach to optimize the schedule.

4. Problem statement

The constraint-based simulation has proven capable of capturing the flexibility of construction processes. More importantly, it does not require explicitly modeled activity graphs. Instead, it relies on constraints which are defined locally for individual

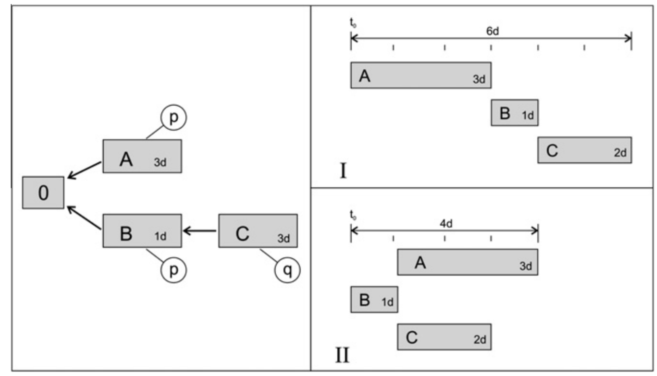

Fig. 1. Simplified example for sequencing configurations of activities resulting in the diverging overall project duration. The left hand side shows three different activities A, B and C, the resources p and q required to execute them and a precedence relationship stating that B has to be finished before C can be started. Assuming that p and q are available exactly once each, two possible sequencing configurations exist which are shown on the right hand side. As can be seen, the overall project time largely depends on the chosen sequencing configuration: it is 6 days for option I, but only 4 days for option II. Obviously the project duration is determined by the random decision of which tasks are started at time t_0.

activities. This advantage can turn into an issue, given that typical construction projects consist of thousands of activities. Defining these activities and their constraints manually is time-consuming and error-prone.

5. Proposed methodology

5.1. Overview

To facilitate the generation of the input data required for the constraint-based simulation, this paper introduces a methodology which is based on interactively refining both a building model and its corresponding process model. During the process, the scheduler selects one of the available construction methods applicable for a specific building part or component. This information is used for generating process steps on the next level-of-detail where the scheduler can choose among different construction methods. The process is repeated until the finest level-of-detail is reached, where each of the process steps corresponds with one atomic activity. The entire set of these atomic activities forms the input for the simulation. Since the chosen construction method also defines the required resources (material, labor, equipment) and precedence relationships between individual activities, the constraints can be created automatically.

The methodology is implemented in the software tool *Preparator*. Fig. 2 illustrates the overall workflow. The scheduler uses *Preparator* to interactively assign *construction methods* to individual building parts or components, refining both the product and process model. When reaching the finest level-of-detail, *Preparator* creates the activities and constraints which are used as inputs for the constraint-based simulation program. In contrast, when using CPM all the activities and their interdependencies have to be specified manually. Before starting the simulation, the user defines the resources available for the project. The constraint-based simulation is then repeatedly run, performing a Monte-Carlo analysis in order to find a good schedule. The resulting detailed schedule can be easily combined with the 3D model of the project to generate a 4D animation of the construction process.

To make the complex process of activity and constraint generation manageable for the user, the following concepts have been implemented:

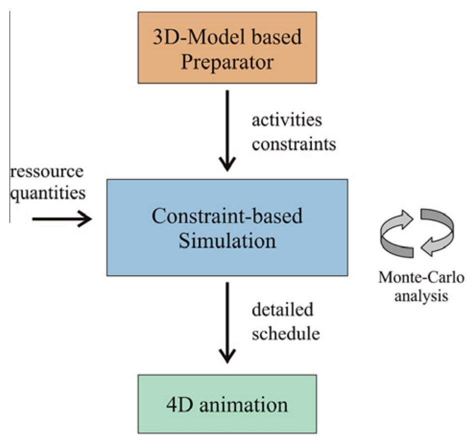

Fig. 2. Workflow of the presented methodology.

– interactive assignment of *construction methods* to individual building components using a 3D model
– successive refinement of both the product and process model traversing a level-of-detail hierarchy
– formalized construction methods encoded by means of process patterns
– generation of *activity packages* encapsulating atomic activities and all corresponding information (i.e. required resources, building component concerned and so on).

These concepts are explained in detail in the following subsections.

5.2. 3D model based activity generation

Preparator provides a 3D model of the building for which the schedule is generated in order to support the user's work in an intuitive way. All objects are clearly identifiable by their 3D representation.

The activities required for simulating the construction process are generated by interactively assigning *construction methods* to building components (Fig. 3). Construction methods define standard procedures, implying a fixed set of activities and precedence relationships among them. There are different construction methods available depending on the selected building component and the current level of the process model hierarchy. The developed application provides a set of standard construction methods for typical bridge components. This set can be easily extended by means of configuration files.

The 3D model-based application allows for the easy identification of components for which a construction method has not yet been defined by highlighting them on demand. Another important feature is that for most activities, a quantity take-off for the required materials and auxiliary equipment can be performed automatically from a 3D model.

As no standardized product model currently exists for bridges, *Preparator* makes use of pure 3D models. These 3D models are enhanced by semantics during the interactive assignment process. The resulting hierarchical product model strongly corresponds with the process model generated by applying the construction methods.

5.3. Level-of-detail approach

To further reduce the complexity of simulation input generation, we integrated a level-of-detail (LoD) approach into our methodology. The scheduler can then successively refine both the product model and its corresponding activities. By selecting a construction method for realizing a certain building component or subcomponent, its respective process model components on the next level-of-detail are generated, as well as the precedence relationships between them. At the same time, the process model components are linked to its corresponding building components in order to be able to perform a quantity take-off at a later stage.

The user starts at a very coarse level, looking at the entire project, and then interactively elaborates both the product model and the activities until a desired level has been reached. This approach is closely related to the concept of the construction method models introduced by Fischer et al. [26]. A hierarchical LoD approach reduces the complexity of the activity generation, because it allows the scheduler to concentrate on the appropriate level-of-detail.

When the scheduler assigns a construction method to a process component, the process components of the next level are automatically generated allowing the scheduler to proceed to the next level-of-detail. The choice of construction methods on a certain level automatically determines the available construction methods on the lower levels. These rules form an integral part of *Preparator* and can be easily adjusted by the user.

Fig. 4 provides an example of the hierarchical process and product model of a bridge construction which is generated during the interactive process. The planner starts on Level 1, selecting and applying a certain construction method for the high-level process *Construct Bridge*. From the available construction methods *Balanced Cantilever Method*, *Formwork Carriage*, and *In-Situ Casting on Standard Falsework*, the latter is chosen.

Based on this decision, the following components in the next level (LoD 2) of the process model are generated: *Construct Abutment*, *Construct Pier* and *Construct Superstructure*. Precedence relationships between these process components are also generated: the *Abutments* and the *Piers* must be finished before construction of the superstructure can start. This will be an important input for generating the precondition constraints (see Section 5.4).

The product model is simultaneously refined corresponding to this level of detail, i.e. the LoD 2 product model components *Abutment*, *Piers* and *Superstructure* are created. The user must now identify the corresponding objects in the 3D model. This is necessary for performing the quantity take-off at a later stage and to allow for the visual control of the process assignment status of individual building elements.

The planner then selects a construction method for each of the LoD 2 product model components. As an example, for *Piers*, the construction methods *Reinforced Concrete* and *Steel* are made available. If *Reinforced Concrete* is selected, the LoD 3 process model components *Construct subbasement*, *Construct basement* and *Construct Pier Shaft* including the precedence relationships are created, as well as the corresponding LoD 3 product model components.

Note that the same construction method is chosen by default for all instances of a given component type. However, they can be modified individually at any time.

After identifying the objects in the 3D model which correspond to the generated LoD 3 product model components, the planner selects a construction method for each of the LoD 3 components. In the case of the *Pier Shaft*, the choices available are *Precast* and *Cast-In-Situ*. If *Cast-In-Situ* is selected, the LoD 4 process components *Construct Formwork*, *Tier Rebar*, *Fill Concrete* and *Remove Formwork* are created. If the corresponding objects are available in the 3D model, they may be assigned to their process components accordingly.

In general, the number of LoDs varies with the chosen building type and applied construction methods. However, the process model components on the finest level-of-detail always represent atomic activities which form the basis for the subsequent discrete-event simulation. All additional necessary information is grouped with each of these activities within activity packages.

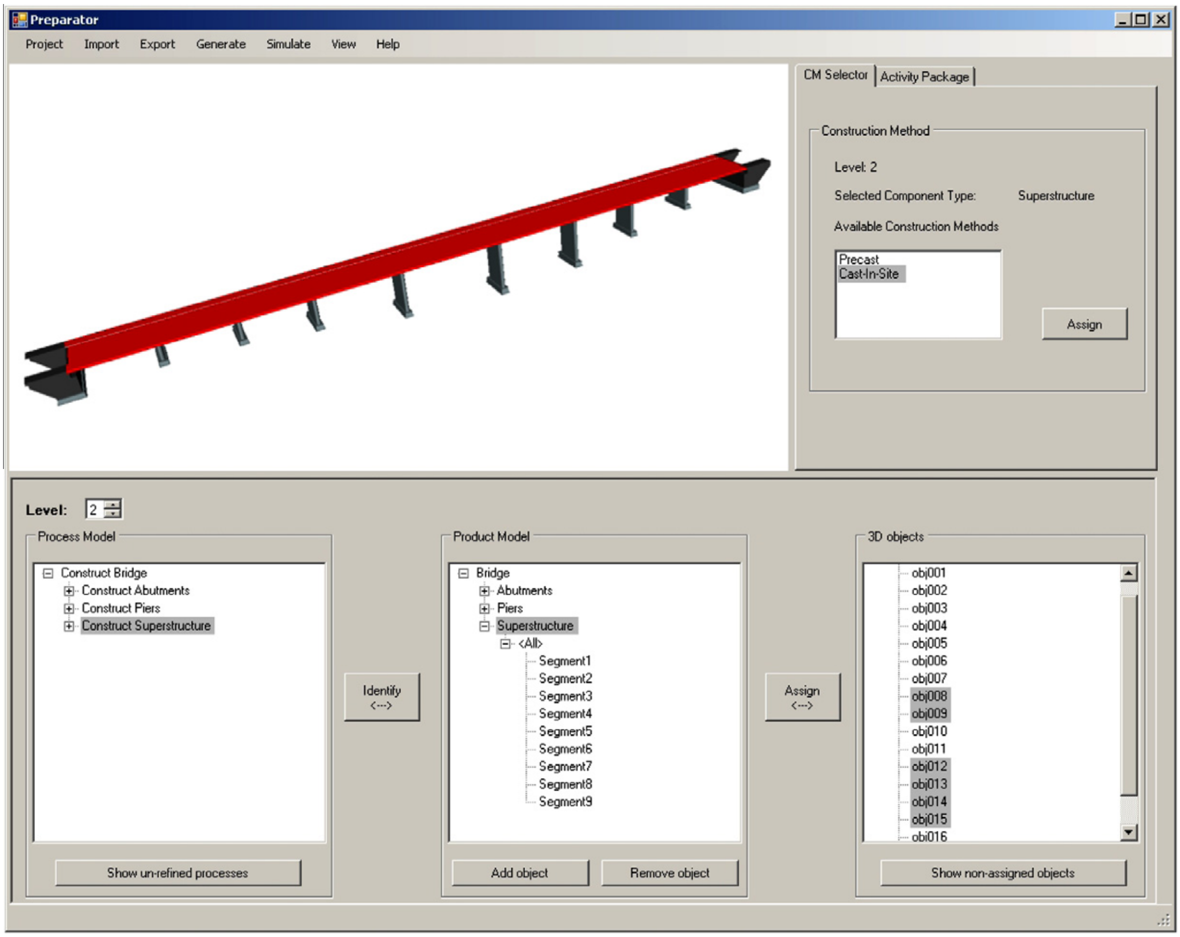

Fig. 3. *Preparator* enables the user to interactively assign construction methods to individual building components or groups of building components. In the example shown, the user has selected all pier shafts and now assigns the *Cast-In-Situ* construction method to them.

5.4. Formalizing construction methods by means of process patterns

The aforementioned construction methods are formalized and encoded in a computer-interpretable way using 'process patterns'. A process pattern combines a number of process components and their precedence relationships and thus represents a companies' knowledge of how to execute certain construction methods. This is used to generate the process components for the next level-of-detail.

Fig. 5 shows two examples of process patterns. The top example encodes the Level 3 construction method *Cast-In-Situ*, which is applicable for a large number of different building element types. In this case, we see a strictly serialized pattern, i.e. the sub-processes have to be executed one after the other.

The bottom example encodes the Level 1 construction method *In-Situ Casting on Standard Falsework*. It allows the parallel construction of the abutments and the piers, but enforces that these sub-processes must be finished before *Construct Superstructure* can be started.

Process patterns formally capture a companies' knowledge on the execution of construction methods. The entire set of defined process patterns is stored in a library. Using *Preparator*, they can be easily modified and newly created.

5.5. Activity packages

An activity package encapsulates an activity and all information associated with it, as shown in Fig. 6. It specifies the materials (type and quantity), laborers (qualification) and machines (type) required for completing the activity concerned. The quantity of the required material is derived from the geometry of the respective building component.

In contrast, the resource quantities, such as the number of available labourers and machines, are defined later, immediately prior to starting the simulation. This further increases the program's flexibility, since the scheduler can easily change the resource setup for the simulation without the need to re-generate the required input data.

In any case, a performance factor for each associated machine type and labourer qualification is required. This factor defines the ratio between the number of employed labourers or machines and the time they require to complete the activity. The performance factors will usually be taken from a companies' database and stored in *Preparator*. They form an integral part of an activity package, since they are required for computing an activities' duration.

Furthermore, the activity package also contains information on the preconditions, i.e. a list of the activities which have to be finished before the activity in question can be started. This information is taken from the construction method applied.

The resulting set of all activity packages generated by *Preparator* is exported into an XML[1] file. Depending on the project size, the number of activity packages and thus the entire data set can be very large. We developed a special XML schema which is able to represent all data related to the activity packages.

[1] Extensible Markup Language – W3C standard for storing structured textual information.

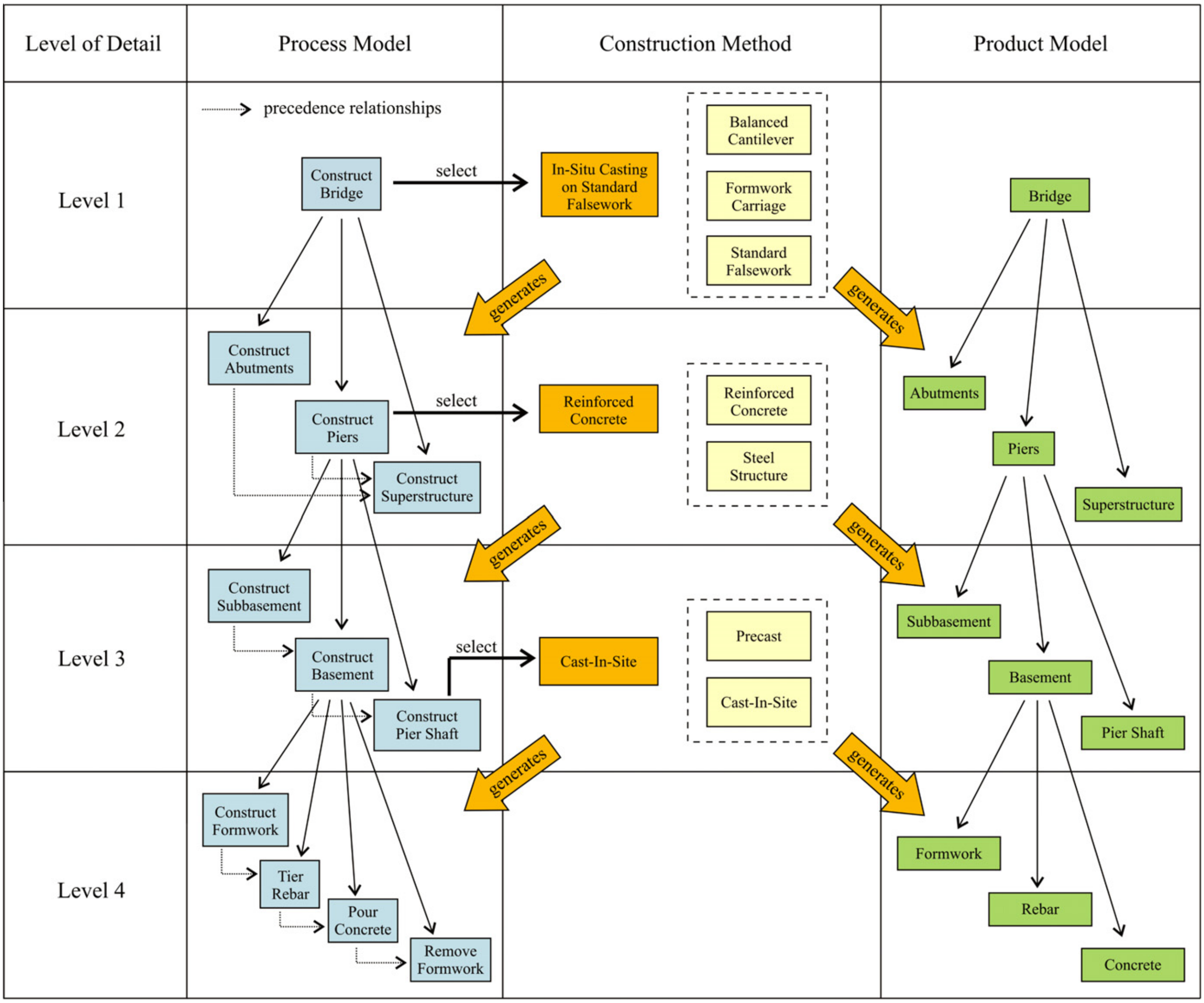

Fig. 4. Level-of-detail hierarchy of the process and the product model. By selecting a construction method for realizing a certain process, the process components for the next level-of-detail are automatically generated.

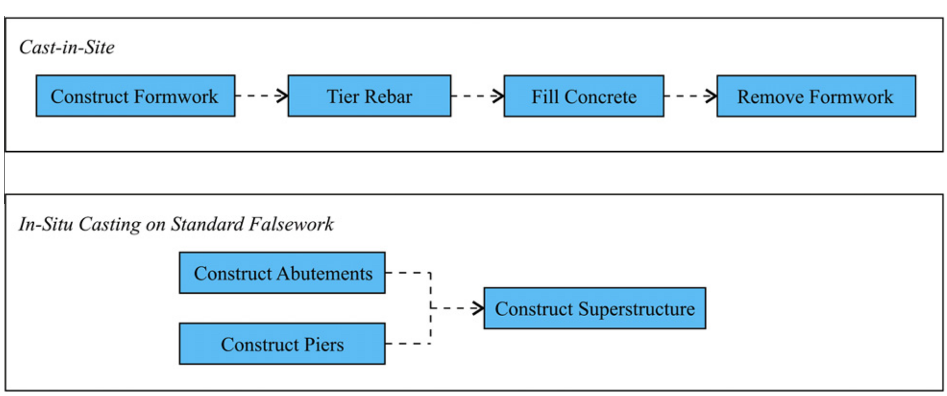

Fig. 5. Two examples for process patterns.

The XML file is read-in by the discrete-event simulation engine. Afterwards, the user defines the quantity of the available resources and the simulation is started, following which the Monte-Carlo analysis is performed.

6. Demonstration

A simple bridge project with the construction method *In-Situ Casting with Standard Falsework* was used as an example to demon-

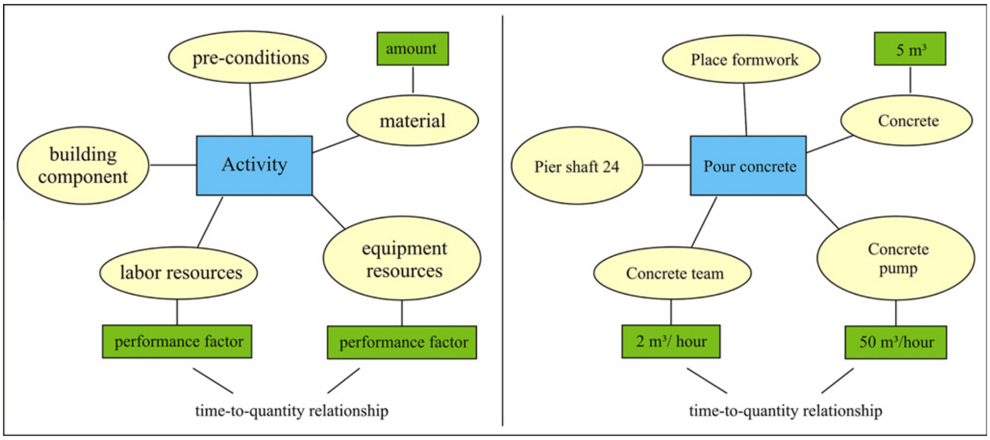

Fig. 6. Activity packages encapsulate an atomic activity, the required materials (type and quantity), laborers (qualification) and equipment (type). Left: general scheme. Right: example for the activity *Pour Concrete*.

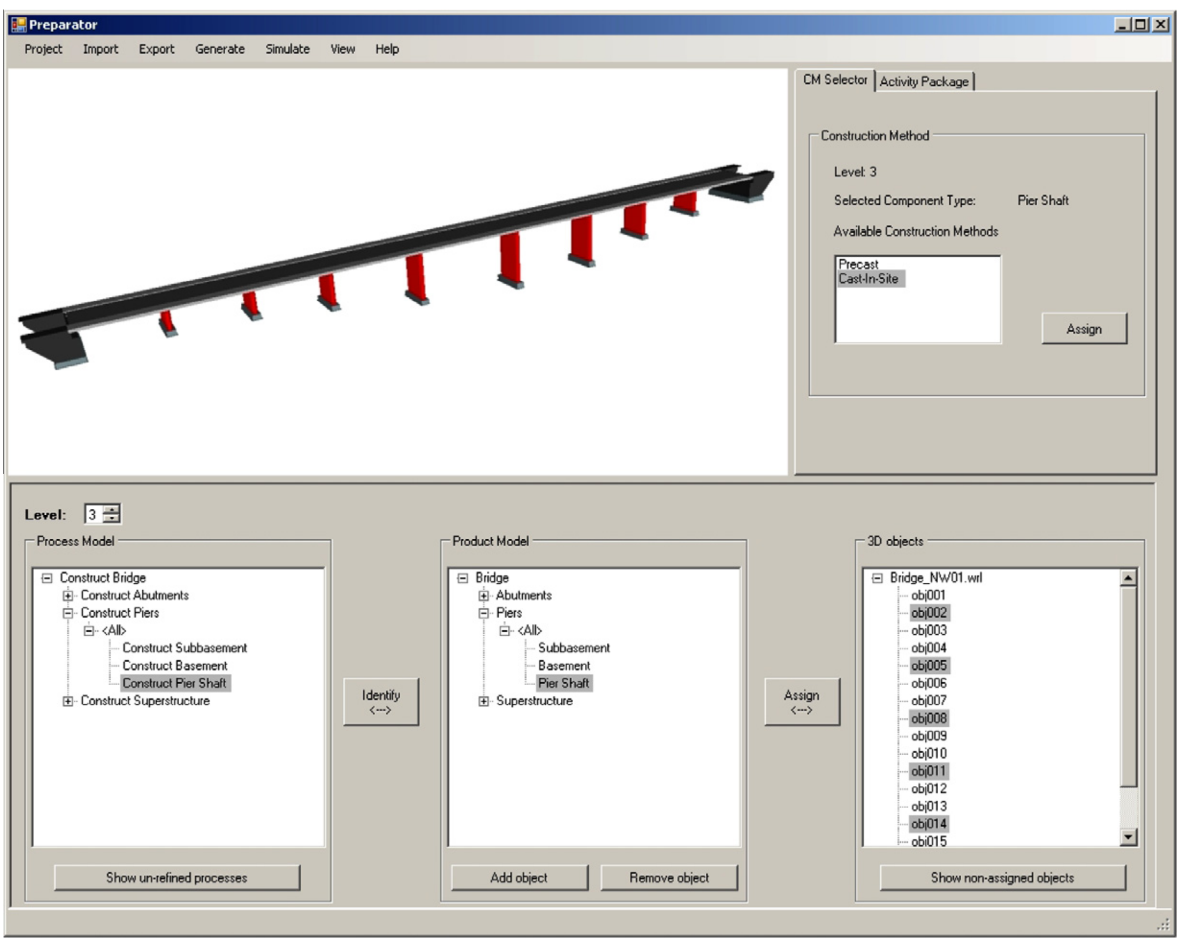

Fig. 7. Screenshot of the *Preparator* application. The user is assigning the LoD 3 construction method *Cast-In-Situ* to the previously selected process *Construct Pier Shaft*. This result in the generation of the corresponding LoD 4 process and product model components (see Fig. 8).

strate the advantages of our approach. The bridge consists of two abutments, eight piers and nine superstructure segments.

The application scenario is presented in two parts according to the software tools employed: *Preparator* and the *Constraint-based Simulation*. *Preparator* is used to generate activities and the corresponding constraints, i.e. the resources required for their execution and the precedence relationships among them.

6.1. Preparator

Fig. 7 shows the graphical user interface of *Preparator*. The upper part of the user interface shows the 3D model of the respective building. The lower part consists of hierarchical views of the process model and the product model, as well as a list of all 3D objects that the model is composed of. Here, product model

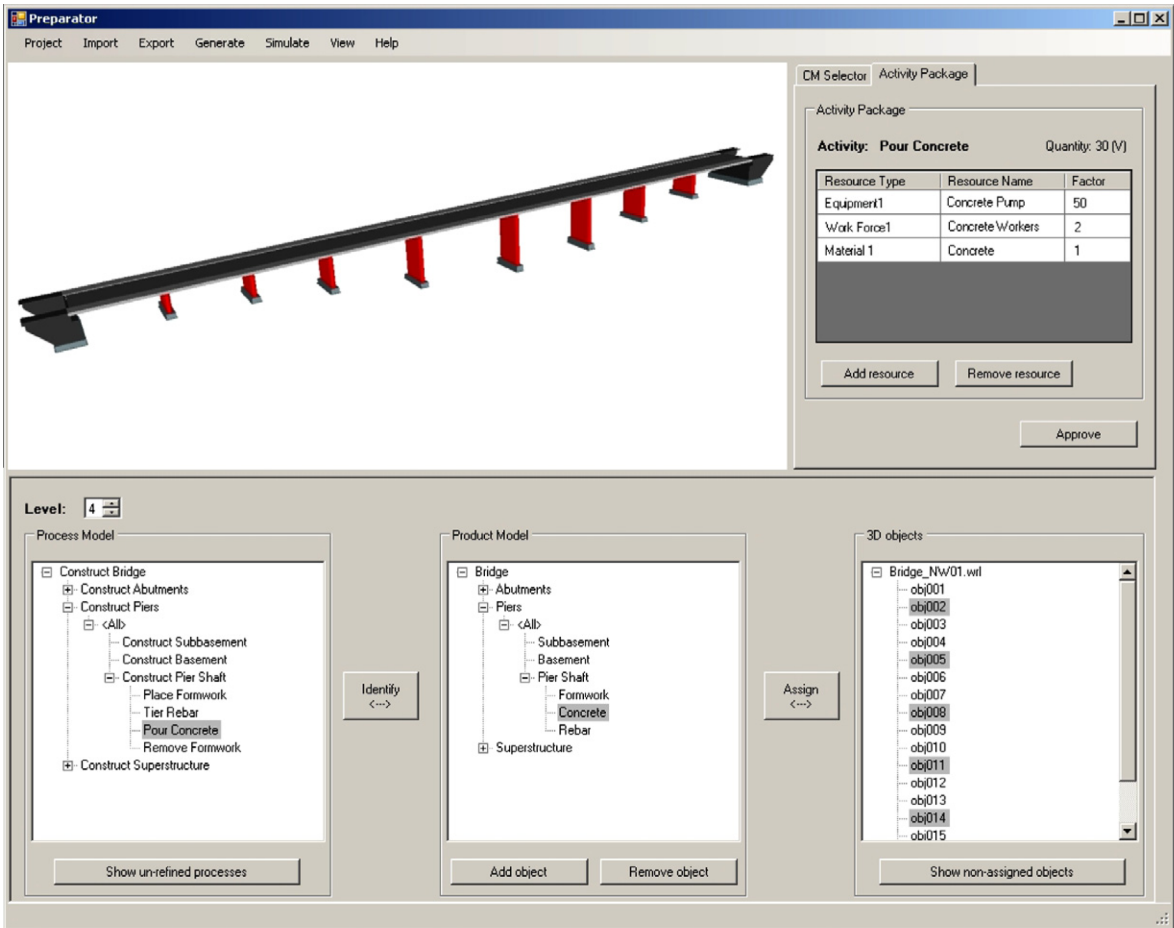

Fig. 8. Screenshot of the *Preparator* application. After the user has assigned the LoD 3 construction method *Cast-In-Situ* to the pier shafts, the LoD 4 process components *Place Formwork*, *Tier Rebar*, *Pour Concrete* and *Remove Formwork* have been generated. Since these represent atomic activities, the user is now able to specify the necessary resources, thus defining the activity package.

Table 1
Simulated resources variations.

	Number of workers	Number of concrete pumps	Number of falsework equipments
Experiment 1	10	1	1
Experiment 2	20	2	2
Experiment 3	20	3	3
Experiment 4	20	4	4
Experiment 5	15	3	2

components can be assigned to their corresponding 3D objects. The part on the right is used to assign construction methods to the currently selected process component, or to define the necessary resources within an activity package.

Figs. 7 and 8 show two different stages of the refinement process. In Fig. 7 the user is assigning a construction method to a LoD 3 process component. The generation of the corresponding LoD 4 components and how an activity package is fed with additional resource data is shown in Fig. 8.

Using the *Preparator* the necessary information for running a constraint-based simulation for this example can be generated very efficiently. In total, 126 activity packages were generated comprising atomic activities and the corresponding constraints. These packages were exported into XML format and read-in by the constraint-based simulation.

6.2. Constraint-based simulation

In the constraint-based simulation system, the quantity of the available resources was specified (i.e. the number of workers, the number of concrete pumps and the number of falsework equipments). We analyzed five different resource configurations, as shown in Table 1.

For each resource configuration, a Monte-Carlo analysis consisting of 1000 simulation runs was performed, with each run resulting in a different schedule. The results are depicted in Fig. 9. The minimum and maximum net working times as well as the average are shown in Table 2.

Within these simulation runs, the same activities, constraints and material are used. For each simulation run, the work step schedule and the workload of employees was recorded and evaluated afterwards. In this example, the shortest net working time is given for the assignment of 20 workers, 4 concrete pumps and 4 substructure equipments (Experiment 4).

As discussed in Section 3.2, the application of Monte-Carlo analysis does not guarantee the finding of the optimum solution. However, this approach generates a multitude of practical schedules that can be analyzed and visualized to identify good solutions manually. Planners can select the best solution according to the objectives of their particular project.

The selected schedule can subsequently be imported into standard scheduling systems for further modifications and evaluations. In Fig. 10, the Microsoft Project Gantt diagram of a possible

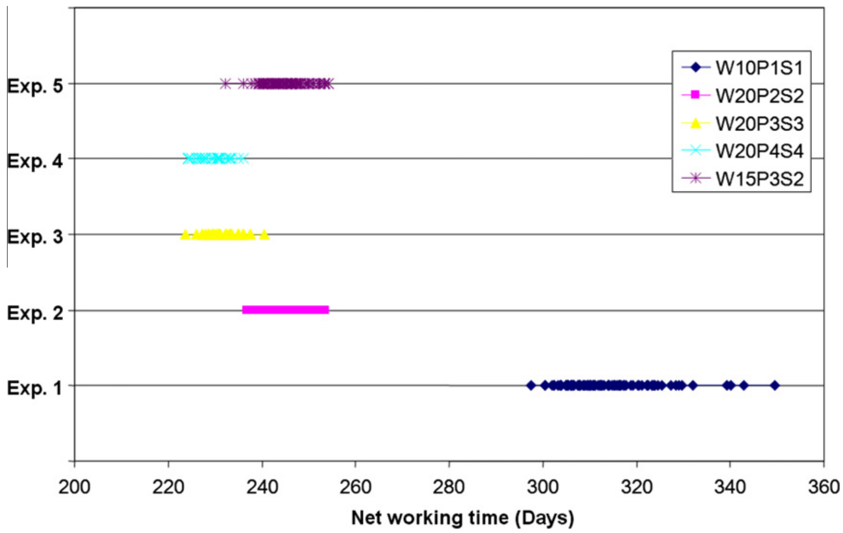

Fig. 9. The results of Monte-Carlo simulation.

Table 2
Net working time results from the experiments.

	Min	Max	Mean
Experiment 1	298 Days	350 Days	314 Days
Experiment 2	237 Days	254 Days	245 Days
Experiment 3	224 Days	240 Days	232 Days
Experiment 4	224 Days	236 Days	229 Days
Experiment 5	232 Days	254 Days	245 Days

schedule generated by the constraint-based simulation system is highlighted.

7. Conclusion

The scheduling of construction processes for building projects is extremely complex with a multitude of requirements, such as technological dependencies and resource capacities, to be taken into account, together with principal guidelines of project duration and available funds. The constraint-based simulation technique can generate effective schedules virtually automatically, but creating the necessary input data manually is tedious and time-consuming.

This paper has introduced a new methodology for creating input data for a constraint-based discrete-event simulation of

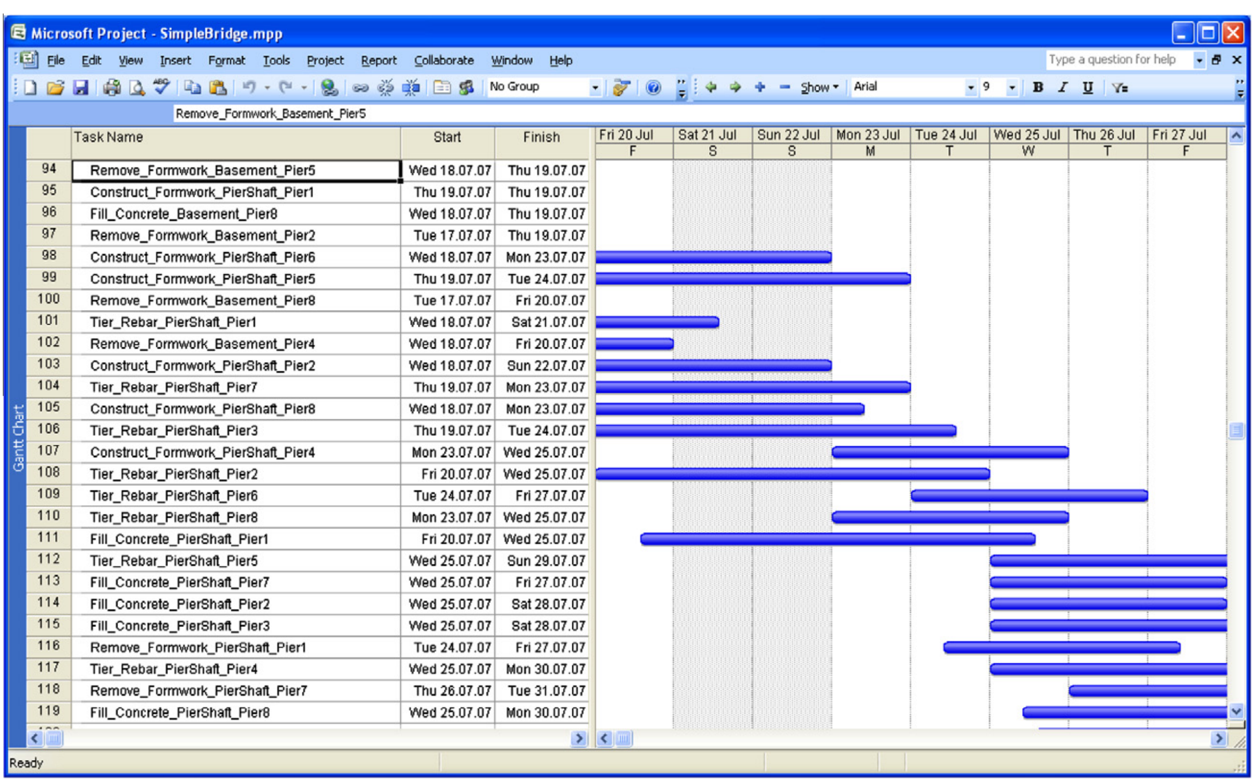

Fig. 10. A possible schedule.

construction processes. The methodology, implemented in a software application called *Preparator*, is based on an interactive process whereby a 3D model is used to generate a hierarchical process model and a corresponding product model. On each level-of-detail, the user assigns one of the available construction methods to individual building parts or components. Based on these decisions, process components and product components for the next level of detail are generated. Construction methods, which represent a companies' knowledge of construction process execution, are formalized by process patterns.

The methodology dramatically facilitates the generation of input data for a constraint-based simulation. By using a LoD approach, the complexity of creating a large set of fine-grained activities and corresponding constraints is reduced to a manageable size. The user is guided through the process by the application, focusing attention on the selection of suitable construction methods.

The end result of this interactive assignment and refinement process is a large set of activity packages which form the input for the constraint-based application, where each activity package combines an atomic activity with its requisite material, resources and its preceding activities. To determine the material quantities, *Preparator* automatically deducts the quantity required for each process (quantity take-off) on the basis of the 3D building model. The entire set of generated work packages forms the input for the constraint-based simulation of the whole construction process.

The constraint-based simulation technique has been selected since it allows the modelling of the highly dynamic processes encountered in the construction industry. Requirements can be easily defined or adapted by adding or removing constraints. The final outcome is a practical work schedule for executing construction projects.

Ongoing development work envisages the incorporation of soft constraints into our simulation approach [19]. Soft constraints represent conditions derived from execution strategies. Further development seeks to allow a constraint module to take soft constraints into account by ranking all executable tasks by their degree of soft constraint fulfilment when selecting the next activity for execution. Absolute compliance is not essential, although this will make schedule generation more realistic.

Acknowledgements

The research presented in this paper was carried out within the ForBAU project which is funded by the Bavarian Research Foundation (Bayerische Forschungsstiftung).

References

[1] T.M. Zayed, D.W. Halpin, Simulation as a tool for resource management, in: Proceedings of the 2000 Winter Simulation Conference, San Diego, CA, 2000.
[2] P. Teicholz, A simulation approach to the selection of construction equipment, Technical Report 26, Construction Institute, Stanford University, Palo Alto, CA, 1963.
[3] A. Gaarslev, Stochastic models to estimate the production of material handling systems in the construction industry, Technical Report No. 1 (11), Department of Civil Engineering, Stanford University, Stanford, CA, 1969.
[4] J. Willenbrock, Estimating costs of earthwork via simulation. Journal of the Construction Division, American Society of Civil Engineers 98 (1972).
[5] D.W. Halpin, CYCLONE – method for modeling job site processes, Journal of the Construction Division 103 (3) (1977) 489–499.
[6] R. Pilcher, I. Flood, The use of simulation models in construction, in: Proceedings of Institution of Civil engineers, Part 1, Design and Construction, ICE, London, vol. 76, 1984.
[7] B. Paulson Jr., W. Chan, C. Koo, Construction operations simulation by microcomputer, Journal of Construction Engineering and Management 113 (2) (1987) 302–314.
[8] D.W. Halpin, MicroCYCLONE User's Manual, Division of Construction Engineering and Management, Purdue University, West Lafayette, Indian, 1990.
[9] R. Huang, D.W. Halpin, Visual construction operation simulation: the DISCO approach, Microcomputers in Civil Engineering 9 (6) (1994) 175–184.
[10] J.C. Martinez, G.P. Ioannou, General purpose simulation with stroboscope, in: Proceedings of the 1994 Winter Simulation Conference, Orlando, FL, United States, 1994.
[11] D. Hajjar, S. AbouRizk, Unified modeling methodology for construction simulation, Journal of Construction Engineering and Management 128 (2) (2002) 174–185.
[12] L. Alberto, C. Azcarate, F. Mallor, P.M. Mateo, Optimization with simulation and multi-objective analysis in industrial decision-making: a case study, European Journal of Operational Research 140 (2) (2002) 373–383.
[13] F. Glover, J.P. Kelly, M. Laguna, New advances and applications of combining simulation and optimization, in: Proceedings of the 1996 Winter Simulation Conference, Piscataway, NJ, United States, 1996.
[14] H. Zhang, H. Li, Simulation-based optimization for dynamic resource allocation, Automation in Construction 13 (3) (2004) 407–418.
[15] R.J. Paul, T.S. Chanev, Simulation optimization using a genetic algorithm, Simulation Practice and Theory 6 (6) (1998) 601–611.
[16] M.A. Ahmed, T.M. Alkhamis, M. Hasan, Optimizing discrete stochastic systems using simulated annealing and simulation, Computers and Industrial Engineering 32 (4) (1997) 832–836.
[17] H. Zhang, C.M. Tam, H. Li, J.J. Shi, Particle swarm optimization-supported simulation for construction operations, Journal of Construction Engineering and Management 132 (12) (2006) 1267–1274.
[18] M. Lu, H.-C. Lam, F. Dai, Resource-constrained critical path analysis based on discrete event simulation and particle swarm optimization, Automation in Construction 17 (6) (2008) 670–681.
[19] U. Beißert, M. König, H.-J. Bargstädt, Constraint-Based Simulation of Outfitting Processes in Building Engineering, in: Proceedings of CIB w78 Conference, Maribor, Slovenia, 2007.
[20] U. Beißert, M. König, H.-J. Bargstädt, Execution strategy investigation using soft constraint-based simulation, in: Proceedings of the IABSE Conference, Information and Communication Technology (ICT) for Bridges, Buildings and Construction Practice, Helsinki, Finland, 2008.
[21] U. Beißert, M. König, H.-J. Bargstädt, Generation and local improvement of execution schedules using constraint-based simulation, in: Proceedings of the 12th International Conference on Computing in Civil and Building Engineering (ICCCBE-XII), Beijing, China, 2008.
[22] M. König, U. Beißert, Construction scheduling optimization by simulated annealing, in: Proceedings of ASCE Workshop on Computing in Civil Engineering, Austin, TX, 2009.
[23] S. AbouRizk, K. Mather, Simplifying simulation modeling through integration with 3D CAD, Journal of Construction Engineering and Management 126 (6) (2000) 475–483.
[24] R. Chahrour, V. Franz, Seamless data model for a CAD-based simulation system, in: Proceedings of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering, Montreal, Canada, 2006.
[25] H.J. Wang, J.P. Zhang, K.W. Chau, M. Anson, 4D dynamic management for construction planning and resource utilization, Automation in Construction 13 (5) (2004) 575–589.
[26] M. Fischer, F. Aalami, Scheduling with computer-interpretable construction method models, Journal of Construction Engineering and Management 122 (4) (1996) 337–347.
[27] T. Hong, M. Hastak, Simulation study on construction process of FRP bridge deck panels, Journal of Automation in Construction 16 (5) (2007) 620–631.
[28] C. Zhang, A. Hammad, T.M. Zayed, G. Wainer, H. Pang, Cell-based representation and analysis of spatial resources in construction simulation, Automation in Construction 16 (4) (2007) 436–448.
[29] H. Said, M. Marzouk, M. El-Said, Application of computer simulation to bridge deck construction: case study, Automation in Construction 18 (4) (2009) 377–385.
[30] H.W. Chun, R.Y.M. Wong, N*-an agent-based negotiation algorithm for dynamic scheduling and rescheduling, Advanced Engineering Informatics 17 (1) (2003) 1–22.
[31] D. Ouelhadja, S. Petrovica, P.I. Cowling, A. Meisels, Inter-agent cooperation and communication for agent-based robust dynamic scheduling in steel production, Advanced Engineering Informatics 18 (3) (2004) 161–172.
[32] M.K. Siddiqui, W.J. O'Brien, A formal mapping-based approach for distributed schedule coordination on projects, Advanced Engineering Informatics 23 (1) (2009) 104–115.
[33] F. Rossi, P. v. Beek, T. Walsh, Handbook of Constraint Programming, Elsevier Science, 2006.
[34] M. König, U. Beißert, D. Steinhauer, H.-J. Bargstädt, Constraint-based simulation of outfitting processes in shipbuilding and civil engineering, in: Proceedings of the Sixth EUROSIM Congress on Modeling and Simulation, Ljubljana, Slovenia, 2007.
[35] N. Metropolis, S. Ulam, The Monte Carlo method, Journal of the American Statistical Association 44 (247) (1949) 335–341.