CoPMod: Support for
Construction Process Modeling *

Elisa Marengo*, Werner Nutt*, Matthias Perktold+

Faculty of Computer Science
Free University of Bozen-Bolzano, Italy
*`firstname.lastname@unibz.it`
+`matthias.perktold@hotmail.com`

Abstract. CoPMod is a proof-of-concept tool for supporting a collaborative construction process modeling approach that has been developed in a research project and tested in real construction projects. In the tests, the approach was applied using a semi-formal graphical modeling language and drawing models on whiteboards. CoPMod is the first IT solution to support this approach, for which a formal definition is provided in [4]. CoPMod represents the first step in the context of the research project COCkPiT, which aims to develop IT-tools for construction process modeling and automatic scheduling and analysis.

Keywords: Construction process modeling, Location-based dependency modeling, Consistency checks

1 Introduction

Construction process models are meant to define the coordination among several construction companies simultaneously present on site, which have to perform different tasks in shared locations. The coordination should specify an agreement on the task execution such that workers do not obstruct each other and the prerequisites to perform a work are all satisfied when it has to start. For instance, the *window installer* and the *floor layer* should agree whether windows must be installed before laying the floor or the other way around.

The specification of temporal requirements is at the basis of most of the existing process modeling languages (such as BPMN, Petri Nets, Declare [1]). However, in the construction domain, in order for the coordination to be effective a model should also specify details such as the *locations* where tasks occur and to take them into account when specifying the coordination (does a precedence apply floor by floor, room by room, or everywhere?).

Traditional approaches in construction are the Critical Path Method (CPM), the Program Evaluation and Review Technique (PERT) and Gantt charts. All of them abstract from some crucial details that make them difficult to be used for

---

execution and process control. In particular, they abstract from locations and from location-based relationships [3]. This has several effects, such as: *i)* companies may have different interpretations of a model; *ii)* it is difficult to recognize variants (e.g., in performances and duration) in the schedule and resources; *iii)* activity duration estimates are not precise; *iv)* duration estimates do not depend on the quantity of work to be performed in a location and do not account for the expected productivity there. Recently, Building Information Modeling (BIM) has emerged as an approach to coordinate the evolution of a building during its entire life. BIM tools are very powerful but also difficult to manage, requiring training and dedicated resources to be used and synchronized with the construction site.

We defined CoPMod, a proof-of-concept tool aiming at supporting the *collaborative* definition of a construction process model. Collaborative means that project managers and foremen from the different companies meet and actively take part in the modeling. The foundational aspects of the approach are described in [4]. CoPMod has been developed within the research project COCk-PiT [2] following an agile methodology, aiming at showing to the companies how IT-Tools can be used to support a collaborative process model definition.

2   Innovations

We describe CoPMod by means of a simplified example for the construction of a hotel. We consider two phases of a construction process *execution*, the skeleton and the interior construction, and we consider three companies: `Sk` responsible for the skeleton, `WI` for the window installation and `FL` for laying the floor.

To maximize their productivity and reduce waste, the three companies have to define a coordination plan in executing their activities, which we call *process model*. The `Sk` company comes first and has to make the excavations, secure the area and build the foundations. The hotel consists of two separate buildings: the main building (B1) and a complex of luxury rooms and wellness center (B2). It must be clear where the work needs to be finished before the other companies can start. EXCAVATION and SECURE AREA must be performed everywhere before any other work can start. However, the construction of the foundations can be performed independently in the two buildings, and once it is finished in one building the other companies can start working there.

The `WI` and `FL` companies have to agree on how they want to proceed in putting the windows. The `FL` company would like them to be installed first so that wooden floors are not damaged by putting the windows. At the same time, the `WI` company would like the wooden windows to be installed later, in order not to damage them when putting the floor.

The tool[1] consists of a *configuration* and a *flow* part.

Configuration Part. CoPMod configuration part defines: *i)* the structure of the building, *ii)* the crafts involved, and *iii)* the activities to be performed.
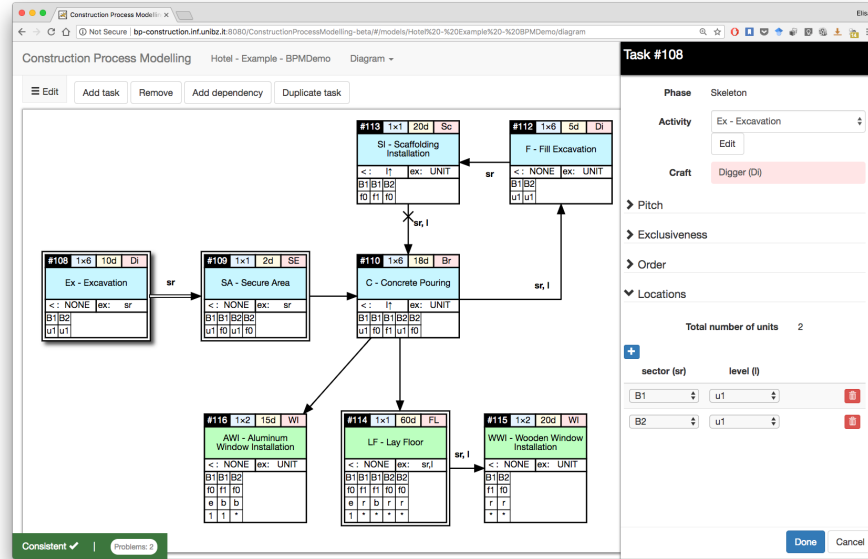
---

[1] https://www.inf.unibz.it/~emarengo/copmod-tool.html

Fig. 1. Flow Part.

The representation of the building structure can be uploaded as a JSON file. It consists of a hierarchy of attributes (e.g. sector, level), which define the locations (e.g., one can say that sectors are B1 and B2 and that B1 is composed of three floors, while B2 of two floors). The tool supports the definition of different construction phases (e.g., skeleton and interior) and a different representation of the building can be specified for each of them. It is indeed the case that skeleton uses a coarser representation for the locations than interior: skeleton identifies locations by sector and level, while interior also uses the section to identify the technological content of an area (e.g. room, bathroom) and the unit number. The companies can insert the activities they are responsible for and associate to them a responsible craft by means of a graphical interface.

Flow Part. The flow part supports the companies in drawing a coordination diagram consisting of tasks, locations, and location-based dependencies among them. The definition is expected to be orchestrated by a moderator (e.g., the project manager responsible for the project) who is expected to be familiar with the modeling language supported in CoPMod [4].

Figure 1 reports the coordination for the hotel construction. First the moderator inserts the Excavation activity which belongs to the skeleton phase (represented with the blue color). The activity is represented as a box, where in the bottom part the locations where to execute it can be inserted. Locations are specified according to the representation defined for the construction phase in the configuration part. In this way, the Sk company can specify that Excavation is performed in both buildings at the underground level (u1). The top of a task box shows: *i)* a task id (automatically assigned); *ii)* the number of

crews currently assigned by the company to the task and the number of workers needed to form a crew (e.g., 1×6); *iii)* the expected duration to execute the task (e.g., 10 days); and *iv)* the acronym of the assigned craft (e.g., Di for digger). The expected duration can also be automatically computed by the system if the crew productivity (e.g., pieces per day) and the total quantity are provided.

In order to specify that the EXCAVATION cannot be interrupted by other tasks, the moderator has to insert an *exclusivity constraint*. This type of constraint can be specified by means of the *exclusiveness* menu (see Fig. 1). The moderator can also specify that exclusivity applies at the level of sector (i.e., for each building separately). This constraint is graphically visualized with a double border for the task box with the label 'ex: sr' for the scope of the exclusivity constraint. When the excavation is finished in one sector, the next task to be performed is SECURE AREA. No other tasks can be performed in between the two. This is expressed by inserting a dependency between the two tasks and selecting as a dependency type the "chain" constraint (double arrow) and sector as scope.

In the diagram, green colored tasks belong to the interior phase. Arrows represent precedences between tasks and can be labeled with the scope (no label indicates that all of the first task must be finished before the following one can start). The dependency between SCAFFOLDING INSTALLATION and CONCRETE POURING is an *alternate* precedence, meaning that the scaffolding must be installed in one level before the concrete is poured, and in order to proceed to the next level, the concrete must have been poured in the previous one.

Checking Functionality. By relying on a configuration part defining the building, the set of crafts and of activities, it is possible to reduce insertion errors (such as typos). Additionally, CoPMod checks for *structure warnings* and *errors* and *consistency errors*. Structure warnings show parts of the diagram that potentially can be rewritten in a better way, such as a location repeated in a task box, or cycles that do not create inconsistencies but that may mean that some constraint is redundant. Structure errors indicate that something is missing in the diagram (e.g., an activity is not associated to a construction phase). Consistency errors mean that the model cannot be satisfied by any execution. In Figure 2 several warnings and errors are signaled. For instance, the loop between ALUMINUM WINDOW INSTALLATION, LAY FLOOR and WOODEN WINDOW INSTALLATION cannot be satisfied by any execution. The loop between CONCRETE POURING, FILL EXCAVATION and SCAFFOLDING INSTALLATION, instead, is satisfiable and thus is not highlighted as a consistency error but as a structure warning.

3   Maturity

CoPMod has been developed starting from a collaborative process modeling approach. The approach was tested in real construction projects without the support of IT-tools (using a whiteboard). CoPMod's current version is a proof-of-concept to show to construction companies the benefit of graphical IT-based tools for construction process modeling. It has been presented to one construction company during the process modeling of a real construction project. A
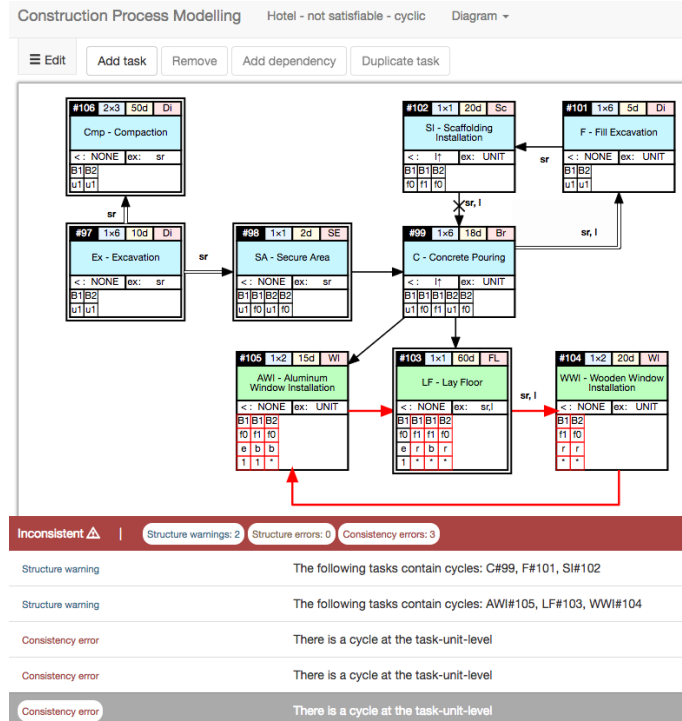
Fig. 2. Warnings and Consistency Checks.

moderator was replicating the modeling steps that were carried out by the project managers and foremen of the involved construction companies. For the automatic checks that it supports, we developed an effective ad-hoc checking algorithm, which outperforms off-the-shelf model checkers. The algorithm translates a diagram into a graph where nodes are activities to be performed in one location and checks the existence of a cycle-free path in the graph (resolving some non-determinism) [4]. A screencast of the tool is available at https://www.inf.unibz.it/~emarengo/copmod.html. The next step will be to leverage constraint programming to develop a tool for automatically schedule tasks at locations, taking into account the constraints expressed in the model.

References

1. van der Aalst, W.M.P., Pesic, M., Schonenberg, H.: Declarative Workflows: Balancing Between Flexibility and Support. Computer Science-R&D **23**(2) (2009)
2. COCkPiT. www.cockpit-project.com/
3. Kenley, R., Seppänen, O.: Location-Based Management for Construction: Planning, Scheduling and Control. Routledge (2006)
4. Marengo, E., Nutt, W., Perktold, M.: Construction Process Modeling: Representing Activities, Items and their Interplay. In: BPM 2018. LNCS 11080, Springer (2018)