FlowOpt: Bridging the Gap Between Optimization Technology and Manufacturing Planners

Roman Barták[1*], Milan Jaška[1], Ladislav Novák[1], Vladimír Rovenský[1], Tomáš Skalický[1], Martin Cully[2], Con Sheahan[2], Dang Thanh-Tung[2]

[1] Charles University, Faculty of Mathematics and Physics, Malostranské nám. 25, Praha, Czech Republic

[2] Entellexi Ltd., National Technology Park, Limerick, Ireland

[*] bartak@ktiml.mff.cuni.cz (contact e-mail)

Abstract

FlowOpt is an integrated collection of tools for workflow optimization in production environments. It was developed as a demonstration of advancements in the areas of modeling and optimization with the focus on simplifying the usage of the technology for end customers. The system consists of several interconnected modules. First, the user visually models a workflow describing the production of some item. Then the user specifies which items and how many of them should be produced (order management) and the system automatically generates a production schedule. This schedule is then visualized in the form of a Gantt chart where the user can arbitrarily modify the schedule. Finally, the system can analyze the schedule and suggest some improvements such as buying a new machine. Constraint satisfaction technology is the solving engine behind these modules.

Introduction

One of the biggest problems of today`s advanced technology is its limited accessibility to users working in a given domain but not necessarily being experts in the used technology. Apple's iPhone is a great example how advanced technology can be made accessible to regular users. With the tradeoff of slightly limited functionality, it provides a user interface to very advanced techniques such as Q&A (question and answering) that anyone can immediately use without the hassle of long training.

FlowOpt is a system that attempts to address the above problem and bridges the gap between advanced optimization technology developed at universities and practitioners from production planning. In particular FlowOpt is targeted to production planning in Small and Medium Enterprises. It covers modeling, optimizing, visualizing, and analyzing production processes in a streamlined feature-rich environment. FlowOpt is a student software development project at Charles University in Prague (Czech Republic). The software itself is a collection of closely interconnected modules that are plugged into the enterprise performance optimization system MAK€ from Entellexi Ltd. (Ireland).

FlowOpt Functionality

FlowOpt covers almost the complete production-planning cycle. It allows users to describe visually and interactively the process of producing any item in the form of a nested workflow with alternatives. After specifying what and how many items should be produced, the system generates a production plan taking in account the existing resources in the factory. The plan is visualized in the form of a Gantt chart that uses information about workflows and allows users to arbitrarily modify the plan by selecting alternative

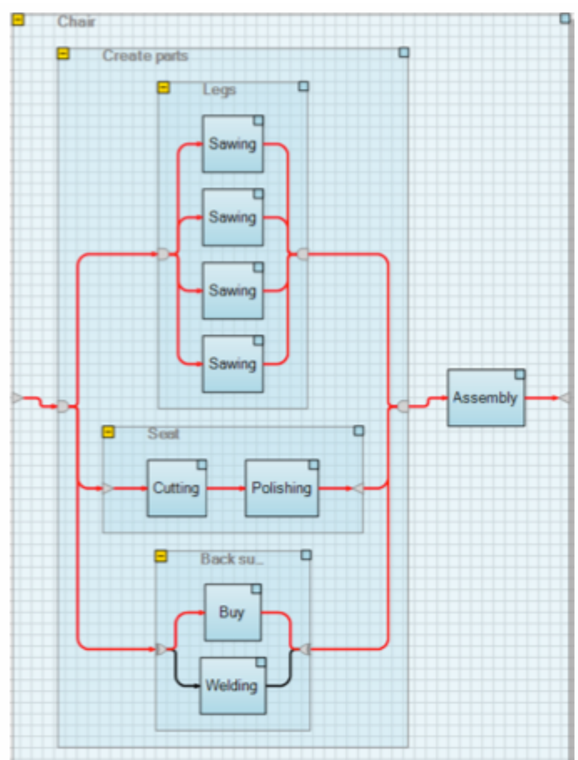

Figure 1. Visualization of a nested workflow in the FlowOpt Workflow Editor (from top to down there are parallel, serial, and alternative decompositions)

processes or allocating activities to different times or resources. Finally, the schedule can be analyzed, the bottleneck parts are highlighted and some improvements are suggested to the user. We will now introduce the functionality of individual modules.

Workflow Editor allows users to create and modify workflows in a visual way. We use the concept of nested workflows that are built by decomposing the top task until the primitive tasks are obtained (Barták and Čepek 2008). Three types of decompositions are supported: either the task is decomposed into a sequence of sub-tasks which forms a *serial decomposition* or the task is decomposed into a set of sub-tasks that can run in parallel – a *parallel decomposition* – or finally, the task is decomposed into a set of alternative sub-tasks such that exactly one sub-task will be processed to realize the top task – an *alternative decomposition* (Figure 1). The final primitive tasks are then filled with activities defined in the MAK€ system (Barták, Sheahan, Sheahan 2012); each activity has a given duration and a set of unary resources necessary for its processing. The workflow can be built in the top-down way by decomposing the tasks or in the bottom-up way by composing the tasks. In practice the user can combine both approaches by decomposing any task or composing a collection of tasks to a form a new task that is then placed to the hierarchical structure of the root task. In addition to the core nested structure, the user can also specify extra binary constraints between the tasks such as precedence relations, temporal synchronizations (start-start, end-start, end-end), or causal relations (mutex, equivalence, and implication). Everything is done using an intuitive drag-and-drop approach. The system also supports import of foreign workflows and it has the function of fully automated verification of workflows (Rovenský 2011). The goal of verification is to find structural problems, namely to find tasks that cannot be part of any valid process due to workflow constraints. Figure 2 gives an example of output of workflow verification with highlighted flaws.

After defining the workflows for all items, this is the *modeling stage*, it is possible to start generating production plans. This is as easy as selecting the required items (workflows) in the **Order Manager**, specifying their quantities and required delivery date and starting the **Optimizer** by pressing a single button in GUI. The data about workflows, activities, and resources are automatically converted to the scheduling model and the system produces a schedule that is a selection of tasks from the workflows (if there are alternatives) and their allocation to resources and time. The Optimizer attempts to optimize both earliness and lateness costs that are derived from the delivery dates. Currently, the Optimizer supports only unary resources.

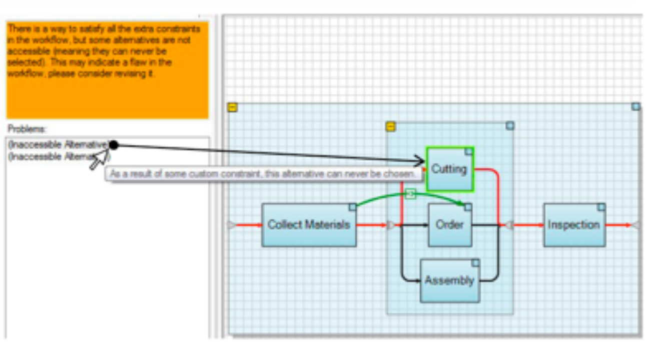

Figure 2. Highlighting the found flaws after workflow verification. The system shows all tasks that cannot be part of valid processes.

The generated schedule (production plan) can by visualized in the **Gantt Viewer**. This module provides both traditional views of the schedule, namely the task-oriented and resource-oriented views. Because the Gantt Viewer has full access to the workflow specification, it can also visualize the alternatives that were not selected by the Optimizer. The Gantt Viewer allows users to modify any aspect of the production plan using the drag-and-drop techniques. The user can move activities to different times and resources and change their duration. It is even possible to select another alternative than that one suggested by the Optimizer. Because the Gantt Viewer is aware about all the constraints originating from the workflow specification, it can also highlight violation of any of these constraints (Figure 3). Even more, the Gantt Viewer can automatically repair all flaws that were introduced to the schedule by the user's modifications (Barták and Skalický 2009).

The final module is **Analyzer** that is responsible for suggesting improvements of the production process. The Analyzer first finds bottlenecks in a given schedule, for example an overloaded resource. For each bottleneck, the analyzer suggests how to resolve it – this could be by buying a new resource or by decreasing the duration of certain activities (for example by staff training). The Optimizer then evaluates each such improvement and finds possible relations between the improvements, for example that applying two improvements together has more benefits than the sum of contributions of individual improvements. Finally the system selects a set of improvements such that their combination brings the best overall improvement of the production process under given constraints such as a limited budged to realize the improvements.

Technology Inside

The FlowOpt system is unique combination of modeling and optimization techniques. It is build around the concept of Nested Temporal Networks with Alternatives (Barták and Čepek 2008) that were suggested as a model of production workflows with hierarchical structure and alternative processes. In FlowOpt this concept was slightly modified and extended with additional constraints. These constraints may introduce flaws to the nested structure (for example a cycle) and hence novel verification techniques for

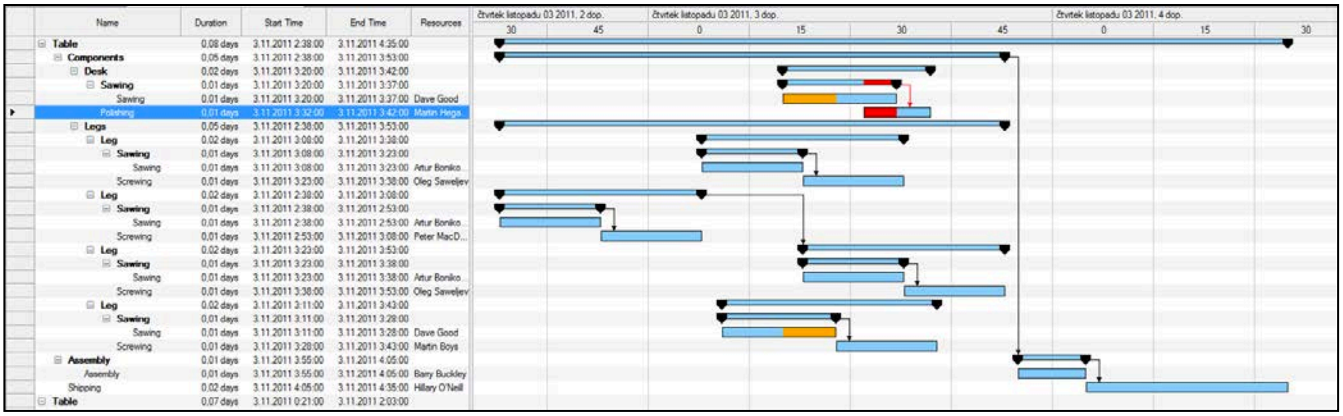

Figure 3. The task view of the FlowOpt Gantt Viewer with two highlighted constraints that are violated (exceeded capacity of resource Dave Good and broken precedence constraint).

workflows were proposed and implemented (Rovenský 2011). The general verification technique is based on modeling the problem as a constraint satisfaction problem and using advanced temporal reasoning techniques, namely IFPC algorithm (Planken 2008) to validate that there exists a feasible process for each task in the workflow. The information about workflows is combined with data about activities and resources to automatically build a scheduling model (Barták et al. 2010). Again, we use constraint satisfaction techniques to solve the scheduling problem; in particular, ILOG CP Optimizer is used to generate optimal schedules (Laborie 2009). The schedule is visualized in the form of a Gantt chart where the user can modify it. The Gantt viewer highlights constraints violated by user intervention, but it can also automatically repair these constraints using a technique of shifting activities locally in time (Barták and Skalický 2009). Again, constraint satisfaction techniques and IFPC algorithm (Planken 2008) are used in background. Finally, the Analyzer uses the idea of critical paths to discover weak parts of the schedule. Currently it uses ad-hoc rules to suggest some improvements (overloaded resource → buy a new resource). The improvements are then applied to the scheduling model and the Optimizer generates a new schedule whose cost is used to evaluate the improvement. Some interactions between the possible improvements are also discovered during this process. For example, the Analyzer can find that one improvement strengthens another improvement. From the set of possible improvements, a subset with the best overall cost is selected by using the techniques of project portfolio optimization. Again, the problem is modeled as a constraint satisfaction problem and ILOG CP Optimizer is used to solve it.

Demonstration Description

The complete process of generating a production plan will be demonstrated. First, we will present the design process of modeling nested workflows using decomposition and composition of tasks. We will also add extra constraints that introduce flaws to the workflow and then we will demonstrate the workflow verification procedure and its outputs. The schedule will be generated in real time and then the visualization capabilities of the Gantt Viewer will be presented. In particular we will show how the schedule can be modified and how the system can automatically repair the violated constraints. Finally, we will present the Analyzer, namely finding the bottlenecks, proposing and evaluating improvements, and selecting the best subset of improvements.

Acknowledgments. The research and development of the FlowOpt system was supported by the Czech Science Foundation under the contract P202/10/1188 and by EU Funding Scheme Research for the benefit of SMEs under the project ValuePOLE (contract 222218).

References

Barták, R. and Čepek, O.; 2008. Nested Precedence Networks with Alternatives: Recognition, Tractability, and Models. In *Artificial Intelligence: Methodology, Systems, and Applications* (AIMSA 2008). LNAI 5253, Springer Verlag, pp. 235-246.

Barták, R.; Little, J.; Manzano, O.; Sheahan, C.; 2010. From Enterprise Models to Scheduling Models: Bridging the Gap. *Journal of Intelligent Manufacturing*, **21**(1), 121-132, Springer Verlag.

Barták, R.; Sheahan C.; Sheahan, A.; 2012. MAK€ – A System for Modelling, Optimising, and Analyzing Production in Small and Medium Enterprises. In *Proceedings of 38th International Conference on Current Trends in Theory and Practice of Computer Science* (SOFSEM). LNCS 7147, Springer Verlag, pp. 600-611,

Barták, R. and Skalický, T.; 2009. A local approach to automated correction of violated precedence and resource constraints in manually altered schedules. In *Proceedings of MISTA 2009: Fourth Multidisciplinary International Scheduling Conference: Theory and Applications*, Dublin, Ireland, pp. 507-517.

Laborie, P.; 2009. IBM ILOG CP Optimizer for Detailed Scheduling Illustrated on Three Problems. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems* (CP-AI-OR), LNCS 5546, Springer Verlag, pp. 148-162.

Planken, L.R.; 2008. *New Algorithms for the Simple Temporal Problem*. Master Thesis, Delft University of Technology.

Rovenský, V.; 2011. *Workflow Modeling*. Master Thesis, Charles University in Prague, 2011.