SUPPORT FOR MODELING, ENACTMENT AND MONITORING OF ENGINEERING DESIGN PROCESSES

Thomas Heer and René Wörzberger

Informatics 3 – Software Engineering,
RWTH Aachen University
{heer, woerzberger}@i3.informatik.rwth-aachen.de

Abstract: The controlled enactment and continuous improvement of engineering design processes needs on the one hand explicitly defined, consistent process models and on the other hand software tools for monitoring and controlling the processes during runtime. In the research project T6 *Dynamic Process Management based upon Existing Systems* tool support for both tasks is provided. In one part of the project, a modeling tool has been developed which allows for correctness checks of a process model against formal constraints. In the other part of the project, a process management environment has been built based upon a commercial software tool for process lifecycle management. This solution provides process support on different levels of granularity: from personal workflows of individual engineers over cooperation support for engineers in a development team to monitoring of the project as a whole.

Keywords: Engineering Design Process Management, Progress Measurement, Project Status Analysis, Correctness Checks

1. INTRODUCTION

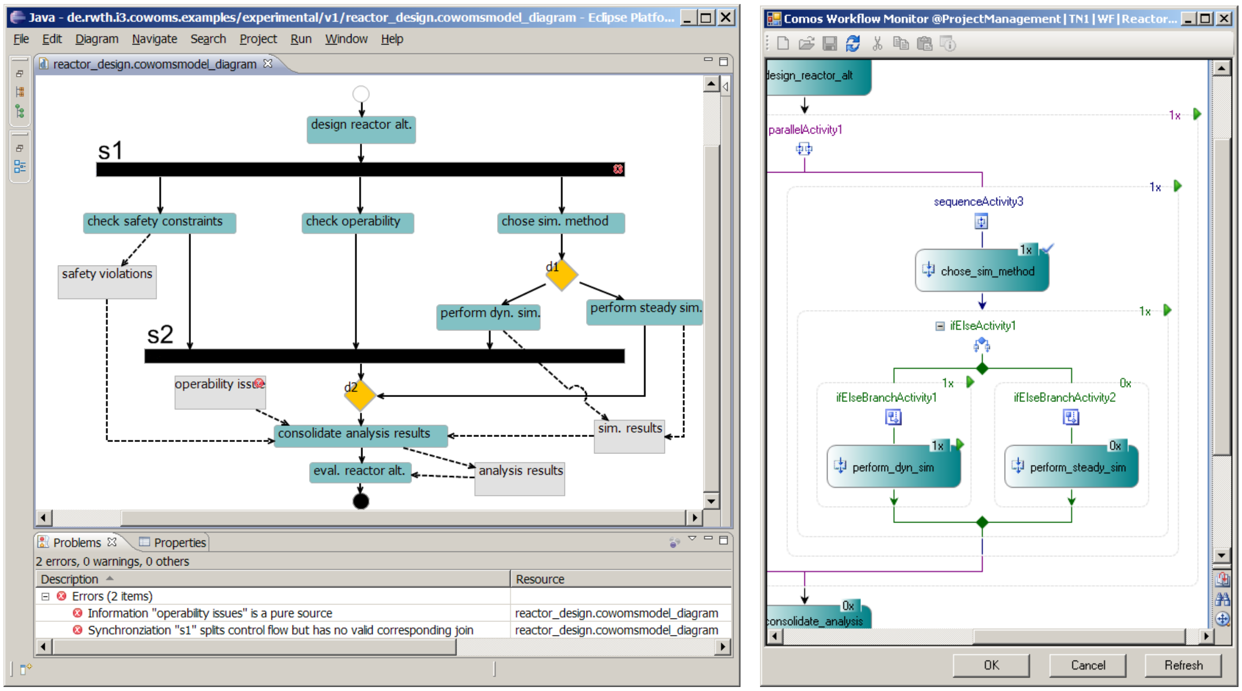
At RWTH Aachen University there has been the Collaborative Research Center 476 IMPROVE in which several departments and institutes from the areas of computer science, chemical engineering and ergonomics have worked together to provide substantial design process support (Nagl and Marquardt., 2008). In the past AHEAD project (Heller *et al.*, 2008) at the Department of Computer Science 3, innovative concepts for tools supporting the management of complex, highly dynamic development processes were developed. The objectives of the current research project T6 (Dynamic Process Management based upon Existing Systems) are transfer of research results from the AHEAD-project to industrial practice, further research regarding time management and progress control of dynamic development processes, and broadening the scope of application domains for process management tools. The latter objective led to a cooperation with an IT service provider from the insurance industries. In the context of this cooperation, a dynamic workflow management system has been realized based on the static IBM Websphere Process Server (Wörzberger *et al.*, 2008a). Moreover, a tool for checking correctness and compliance of diverse insurance work process models was developed. A similar tool is also useful for analyzing work process models in chemical engineering (cf. Sect. 2), which can be used afterwards for process enactment and monitoring (cf. Sect. 3).

2. CORRECTNESS CHECKS OF WORK PROCESS MODELS

In (Theißen *et al.*, 2009), the semi-formal notation WPL (Work Process Language) for conceptual work process modeling is presented. WPL aims to achieve clarity about work processes, e.g. engineering design processes, from a professional perspective. Thus, work process modelers can and should concentrate on professional aspects of the work process in WPL-models. The modeling framework presented in (Theißen *et al.*, 2009) also covers exportability of WPL-models to other applications like PROCEED (cf. Sect. 3) or a simulation tool described in (Tackenberg *et al.*, 2008). The WPL modeling framework detects certain model flaws and aids completing underspecified models. Violations of certain application-specific formal restrictions in WPL-models are hard to capture by means of the employed meta-modeling language OWL and the OWL-reasoners. I.e., from the perspective of a certain application

WPL-models might be flawed. However, if WPL-models are supposed to be used as input for other applications, these flaws have to be removed. Otherwise, the WPL-model cannot be transformed into the application format at all, or the application might exhibit undesirable behaviour (e.g. deadlocks).

The same applies for work process models in other domains. We have studied dynamic work processes in the insurance domain. Here, we also find diverse models in different notations and of different precision each of which has to adhere to certain formal constraints or imposes constraints on other models. These adherence requirements have led to the realization of a prototypical editor for insurance work process models (Wörzberger *et al.*, 2008b). In a joint effort between researchers in chemical engineering and computer science we transferred concepts, which were implemented in the insurance work processes model editor, to the domain of chemical engineering. In particular, we realized a prototypical WPL-editor called *Constrained WPL-Editor (CWPLE)* that supports work process modelers in detecting formal flaws in WPL-models. In the sequel, we describe some kinds of these flaws.



2 errors, 0 warnings, 0 others

Description	Resource
Errors (2 items)	
Information "operability issues" is a pure source	reactor_design.cowomsmodel_diagram
Synchronziation "s1" splits control flow but has no valid corresponding join	reactor_design.cowomsmodel_diagram

Fig. 1: Screenshot of the Constrained WPL-editor (left side) and the PROCEED workflow monitor.

2.1. Generic Formal Constraints in Work Process Models

As presented in (Theißen *et al.*, 2009), conceptual WPL-models are represented in a diagrammatic way, meaning that the model elements are displayed as geometric figures like rounded boxes or arrows. These models have to adhere to certain generic, formal (syntactic) constraints. This is comparable to written English text, which has to adhere to certain generic spelling and grammar rules.

Fig. 1 depicts a screenshot of the CWPLE containing a simple work process model in the WPL-notation. It models work to be done during the design of a reactor (Eggersmann *et al.*, 2008). For reasons of brevity, just the analysis of reactor alternatives, consisting of parallel activities for safety and operability checks as well as dynamic or steady-state simulation, is detailed in this model. In the following, two types of formal flaws are exemplified by this model which impede export to other tools and yield deadlocks, respectively.

Pure information sources. WPL provides elements for modeling information and information exchange within a work process. We call an Information-element a *pure information source*, if it does not originate from any activity, i.e. if it is not target of any InformationFlow-connector. Pure information sources might be reasonable in conceptual work process models as they imply the existence of a rather technical database-retrieval activity, which has been deliberately omitted in the conceptual model. However, most applications require that every Information-element originates from some activity. Thus, in a formally correct WPL-model every Information-element must be target of an InformationFlow-connector coming from an Activity-element. In the exemplary model of Fig. 1, Information-element operability issue does not originate from any activity and thus violates the aforementioned constraint.

Synchronization-Splits and -Joins. Parallel execution of work activities can be modeled in WPL using Synchronization-elements (black bars). In the example model, activity paths starting from check safety constraints, check operability and chose sim. method are executed in parallel. Eventually, parallel paths must be synchronized again. This is reasonably done before consolidate analysis results, because this activity depends on all results of the preceding activities. Therefore, the purpose of s2 is to join the incoming executions paths to just one single outgoing path, i.e., activities succeeding s2 will not start before each execution path running into s2 actually reaches s2. However, the model is formally incorrect inasmuch the execution of the rightmost path will never reach s2 if execution of activity perform steady sim. is chosen in Decision d1.

The screenshot in Fig. 1 exemplifies the core functionality of the CWPLE. Detected flaws are enlisted in a problems-table (s. bottom of the figure). Furthermore, the respective elements are marked with a red marker within the diagram in order to ease their localization. In this way, the work process modeler is not just informed about a flaw but also lead to the element which is incorrect of or at least involved in the defect.

2.2 Usage in the Overall Engineering Design Process and Tool Infrastructure

CWPLE helps preparing conceptual WPL-models, for subsequent export to other applications, like PROCEED. It complements the WPL-editor WOMS+ (Theißen *et al.*, 2008) inasmuch as WOMS+ particularly supports rapid modeling during the conceptual work process design phase. Later on, WPL-models can be analyzed and corrected by CWPLE before exporting them to other applications. In Fig. 1 on the right, the workflow monitor of the PROCEED system (cf. Sec. 3) is depicted. It shows a part of the workflow, which resulted from the WPL process model, during its execution. It can be seen that the requirement for the WPL model to be well-formed regarding the block structure of splits and joins is a necessary constraint for transforming the model into a workflow definition. This is because all workflow definitions in PROCEED are always block-structured.

To conclude, the constraint checks of CWPLE are generic and of rather technical nature as they merely regard the syntax of a WPL-model. Therefore, they complement the domain specific checks of professional constraints, i.e., checks of WPL-models versus knowledge stored in OWL-ontologies, which is described in (Theißen *et al.*, 2008).

3. PROCESS ENACTMENT AND MONITORING

The engineering software Comos developed by Comos Industry Solutions GmbH is a well known product in the plant engineering industry. It allows for the management of engineering data throughout the whole lifecycle of a chemical plant. However, in the current version Comos does not support the explicit management of the engineering design process. The explicit management of tasks, resources, products and their interrelations can significantly improve the productivity in an engineering project and it is the prerequisite for project monitoring. Therefore, in the course of the transfer project T6, Comos has been extended by the module PROCEED (PROCess management Environment for Engineering Design processes). The extended system now allows for the integrated management of tasks, resources and engineering data. The engineering design process is supported on three different levels of granularity. First, personal workflows of individual engineers are managed by a workflow engine. Second, the cooperation of engineers in a development team is supported by the core process engine of the PROCEED system. Third, a dedicated user interface supports graphical project status analysis. The project status can be analyzed in the monitoring GUI with regard to different aspects like e.g. workload distribution between different technical crews, engineering progress of different plant parts and the delay of certain milestones. Furthermore, project planning is supported by a commercial project management tool, which has been coupled with PROCEED. In the following subsections, the process management support on these three levels of granularity is described.

3.1 Workflow Management

The workflow engine of the PROCEED system provides support for predefined, highly structured processes in an engineering project. These workflows can range from the revision of a single document to a complete change management process. For each workflow type, an executable workflow definition is created. Based on this definition, arbitrary many workflow instances can be started in an engineering project. The workflow definition contains control structures like loops or alternative branching constructs, whose conditions are evaluated at runtime of the workflow instance. The execution of tasks in the workflow depends on this evaluation. When a task is activated in the workflow instance, the assigned engineer is informed about this work item.

The workflow management functionality is tightly integrated with the engineering data in the Comos database. For example, when an engineer inserts a new device into a flow diagram, then a workflow instance is automatically created, which defines the necessary steps for specifying the device. Another engineer can accept this task and set the device's property values. During the execution of the workflow instance the workflow engine observes the state of the device object in the database. Whenever required property values are set, the workflow proceeds to the next step. In this way, the workflow engine supports engineers to comply with the standard procedures and monitors their progress without impeding their work.

Although workflows are already flexible by design through the use of control structures, it is sometimes necessary to deviate from the workflow definition. Example cases are the insertion of an additional step for the revision of a document or an additional evaluation during a change management case. The workflow engine and the provided monitoring tool allow for such dynamic changes at workflow runtime. However, there are several restrictions for dynamic changes to guarantee the consistency of the workflow data.

The workflow engine was developed based on the Windows Workflow Foundation (WF) by Microsoft. The WF provides a data format for serializing workflow definitions and workflow runtime data to XML. All the workflow data is stored in the Comos database, including the current state of all workflow instances and the tracking data of their past execution. The WF provides libraries for developing tools for editing workflows in a graphical representation and for connecting workflows to other applications like Outlook or Sharepoint Server. All this has been exploited to build a full-fledged workflow management system.

However, this workflow management system is only one part of the PROCEED prototype. The personal workflows of individual engineers, the multiple instances of change management processes and the like have to be managed as parts of the overall design process. For this reason, the workflow engine is complemented by the PROCEED process engine which provides an execution context for each workflow instance.

3.2 Management of Dynamic Development Processes

For the management of engineering design processes the workflow approach is not suitable. Design processes are very flexible and cannot be completely predefined in advance. Furthermore, the focus lies on the cooperation of engineers instead of standard procedures for individuals. Therefore, an innovative approach for task management in development processes was developed in the AHEAD research project (Nagl and Marquardt, 2008). The process engine of the PROCEED system comprises the main functionality of the AHEAD research prototype. The data structure for task management is a so-called dynamic task net. It is a network diagram with task nodes and control flow edges extended by task states and dataflow relationships. The control flow relations may have a less strict semantics than the sequential execution of tasks like e.g. end-to-end relationships, which is important for the simultaneous execution of dependent tasks. A dynamic task net instance allows for fallbacks to redo earlier, already completed steps in the process. The task net can be modified during execution, so that additional tasks and dependencies can be inserted in the process while others are already executed. All these aspects are necessary requirements for an adequate support for engineering design processes. Furthermore, a dynamic task net is hierarchically structured. Each task can define a whole sub process consisting of several subtasks. The integration of all workflow instances, which are managed by the workflow engine, into the overall development process is achieved by the identification of tasks in the hierarchical task net with workflow instances. Thereby, the task net defines an execution context for each workflow instance. The approach is described in detail in (Heer *et al.*, 2008).

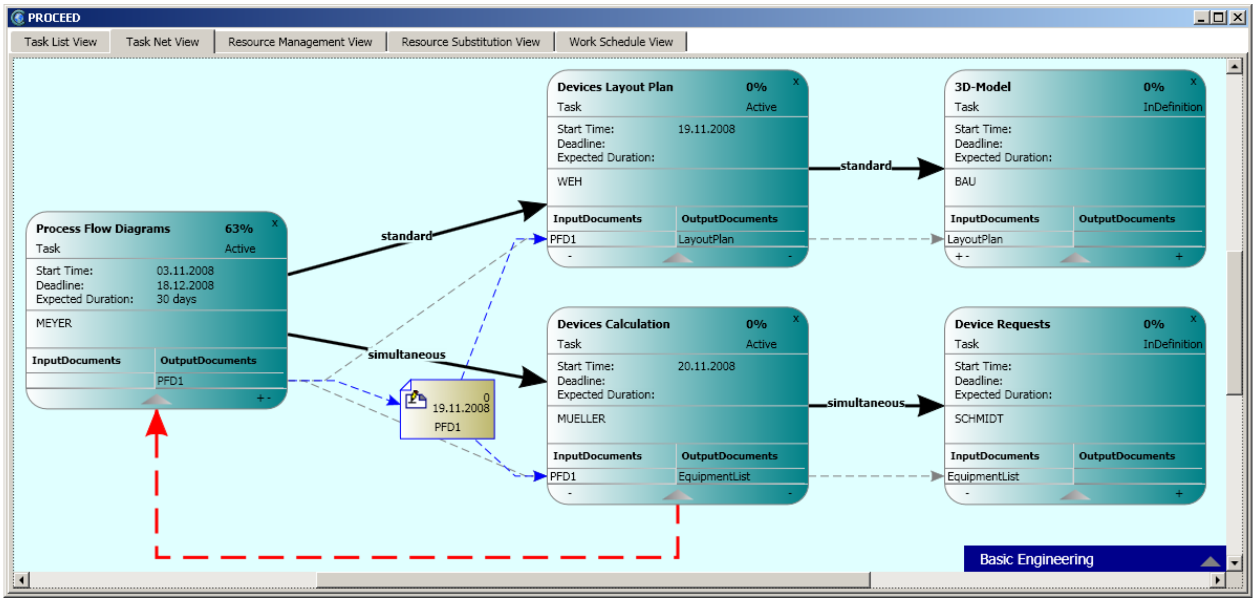


Fig. 2: Screenshot of the PROCEED Task Net View.

Just like workflows, dynamic task nets are tightly integrated with the engineering data in the Comos database. Tasks have input and output parameters which refer to documents or device objects in the database. Data flows are defined between these parameters and document revisions are associated with these data flows. A document revision produced in Comos is associated with one of the tasks, which have an according output parameter defined, and it can be consumed in a task, whose input parameter is connected with that output parameter by a data flow. This way, the PROCEED system keeps track of the information flow between all process participants. The process participants are managed by means of the resource management functionality of PROCEED, which complements the provided functionality for task and product management.

3.3 Project Status Control

A key feature of process management systems is the monitoring of running processes. Especially in plant engineering projects it is essential to have accurate information about the current project status. This information is even more valuable if it is available at short notice. The PROCEED system supports incremental progress measurement of plant engineering projects. At any time during the project, the current status can be retrieved, in which the most recent developments are reflected. The progress measurement largely relies on the automatic monitoring of workflow instances and the analysis of the engineering data. Hence, it reduces the use of estimated progress values to a minimum and at the same time does not impose too much extra work on the engineers. It facilitates qualitative statements about the actual project's progress and allows comparing these to the planned progress. Thereby, critical delays can be detected early. The use of task types and workflow types for process modeling allows to gather experienced data about their average required effort and to use these values as a reference for monitoring future task or workflow instances respectively.

The progress of the engineering project is measured across all three levels of granularity. The workflow engine measures the progress of running workflow instances. To gather reference values, the tracking data of previous instances of a workflow type is analyzed and for each activity in the workflow definition the average runtime is calculated. Besides human tasks, activities in a WF workflow can also be control flow blocks like a while loop or an alternative branching activity. Average execution times are also calculated for these control flow activities. During the execution of a workflow instance the actual duration of the workflow is compared with the expected overall runtime. The latter is based on the estimated time until completion, which is calculated by taking the control flow and the average runtimes of future activities into account. This avoids subjective estimation about the remaining execution time by the resource assigned to the respective workflow. The progress of monitored workflows is aggregated and propagated upwards in the hierarchical task net structure of the overall design process.

Besides the calculation of progress values, project status analysis can also be performed graphically by means of a dedicated user interface of PROCEED. The development of this specific GUI was motivated by the specific characteristics of plant engineering processes. First, there is a huge amount of rather small engineering tasks for specifying all the devices, pipes and measuring instruments of a chemical plant. Common net plans and Gantt charts quickly grow too big to keep an overview over the current status of the project. Second, there is a high rate of simultaneous engineering. The visualization of tasks in the form of a network diagram is not sufficient since most tasks are executed in parallel. Therefore, an innovative multidimensional visualization approach was developed to provide a condensed overview over all tasks in the project, their workload and their current status. A 2-dimensional pivot table is presented to the user. In this table the domains of the axes can be freely chosen from a given set containing task hierarchy, roles, plant parts, workflow types and timeline. The measures for the coordinates in the table can be actual or planned workload, number of tasks with a certain status, or progress values. In this way, the current project status can be analyzed with regard to different aspects like for example the workload per role or the status of all workflow instances of a certain type. Furthermore, views for analyzing the performance of individual tasks are provided. These serve on the one hand to identify delayed tasks and on the other hand to investigate the reason for the delays.

Besides progress measurement, the PROCEED system also provides task scheduling functionality. Scheduling can be performed either in the planning phase of the project or during project runtime in case of dynamic plan changes. Among other things, the process management tool guarantees the consistency of the planned dates with the process structure at any time. Currently, all possible implications of dynamics on the schedule and the progress of a running process are investigated. The goal is to achieve an overall conceptual framework for the planning estimation, monitoring and controlling of dynamic development processes in which the tool support plays an integral role.

4. CONCLUSION

In this paper we have presented ongoing research work providing support for design processes in chemical engineering. We have shown that the concepts for checking correctness of work process models, which are applicable in the domain of insurance work processes, can also be utilized in chemical engineering. This constitutes an example of successful interdisciplinary knowledge transfer. Furthermore, the transfer of concepts for design process management into an industrial environment has been successfully pursued in the PROCEED-subproject. In this subproject, further research regarding time management and progress monitoring in dynamic development processes has been undertaken and is still going on. The resulting tools of the T6 project will be of great use in industrial practice and across different application domains.

REFERENCES

Nagl, M. and W. Marquardt (2008). Collaborative and Distributed Chemical Engineering: From Understanding to Substantial Design Process Support. In: Lecture Notes in Computer Science, vol. 4970, Springer, Berlin.

Heller, M., D. Jäger, C.-A. Krapp, M. Nagl, A. Schleicher, B. Westfechtel and R. Wörzberger (2008). An Adaptive and Reactive Management System for Project Coordination. In: (Nagl and Marquardt, 2008).

Heer, T., C. Briem and R. Wörzberger (2008). Workflows in Dynamic Development Processes. In: Proceedings of the Workshops of the 6th International Conference on Business Process Management, Springer, Berlin.

Eggersmann, M., B. Kausch, H. Luczak, W. Marquardt, C. Schlick, N. Schneider, R. Schneider and M. Theißen (2008). In: (Nagl and Marquardt, 2008).

Theißen, M., R. Hai and W. Marquardt (2009). A Framework for Work Process Modeling in the Chemical Industries. In this volume (6 pages).

Wörzberger, R., N. Ehses and T. Heer (2008a). Adding Support for Dynamics Patterns to Static Business Process Management Systems. In: Software Composition (Pautasso, C., and E. Tanter (Ed)), pp. 84-91, Springer, Berlin.

Wörzberger, R., T. Kurpick and T. Heer (2008b). Checking Correctness and Compliance of Integrated Process Models. In Proceedings of the 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing.

Tackenberg, S., B. Kausch and C. M. Schlick (2009). Organizational Simulation of Complex Process Engineering Projects in the Chemical Industry. In this volume (6 pages).