René Wörzberger: **Management of dynamic business processes based on static process management systems, doctoral thesis**

Promotion: RWTH Aachen, Fakultät für Mathematik, Informatik und Naturwissenschaften
Erstgutachter: Prof. Dr.-Ing. Manfred Nagl, RWTH Aachen
Zweitgutachter: Prof. Dr. Gregor Engels, Universität Paderborn
Drittgutachter: Prof. Dr. rer. nat. Bernhard Rumpe, RWTH Aachen
Datum der Prüfung: January 11, 2010
Veröffentlichung: Shaker Verlag, AIB-SE 2, 305pp, 2010

Kurzfassung:

Process management systems support business processes. They supply process participants, i.e., clerks or software services in an insurance company, with the right data at the right point in time.

However, complex business processes cannot be entirely planned before execution. Unexpected events demand for deviations from the plan during process execution. This inherent property of business and design processes is named dynamics .

Common process management systems cannot properly cope with dynamics as they require a static model of the respective business process before its execution and do not allow for deviations from that model during process execution. This restriction often leads to highly complex process models, which are hard to maintain yet fail to cover every reasonable process run. If a business process deviates from the process model, the process management system is detached from the process and cannot support it any more.

This work describes a process management system which particularly supports dynamics in business processes. Instead of deviating of an executed process model, a process participant can structurally modify a process model and thereby adapt it to the unforeseen situation.

In contrast to related works, the described system has not been realized from scratch. Instead it extends the widespread commercial process management system IBM WebSphere Process Server (WPS), which is particularly used by Generali Deutschland Informatik Services GmbH (GDIS) our partner in industry.

The following achievements have been made:

The work extends the incomplete modeling system of WPS. Process definition models, which are expressed in the standard language WS-BPEL, are complemented by process instance models that carry additional information about the current execution state of a process as well as information about the process history. Process knowledge models contain abstract regulations about runs of processes. This includes regulations about permitted activity sequences, activity frequencies and mutual exclusions of activities in certain processes. The process knowledge models syntax and semantic have been formally defined using meta models and temporal logic formulas which are applied to graph rewriting systems. A process model editor has been developed that supports editing of all three kinds of process models.

The semantics of process definition models and process instance models constitutes the formal foundation for the implementation of a dynamic layer that extends the WPS. The dynamics layer simulates dynamic changes which cannot be directly applied to the WPS. The simulation is hidden from process participants who perceive dynamic changes as real structural changes to process instance models within the process model editor.

Dynamic changes in the process model editor are restricted by technical and professional constraints. A syntax-based checking tool ensures that these constraints are not violated by some dynamic change. In contrast to related works, checks of technical and professional constraints are based on a single, holistic approach, which utilizes the Object Constraint Language (OCL). Thus, all kinds of process models can be explicitly checked for (inner) technical correctness and process definition and instance models for professional compliance with regard to some process knowledge model.

Explicit checks are complemented by checks against implicit process knowledge. Implicit process knowledge is knowledge about activity frequencies and orderings. Another checking tool leverages this implicit knowledge by revealing inconsistencies among process definition and instance models. These inconsistencies are detected in graph transitions systems which constitute the simultaneous run of two process instance or definition models.