

Aktuelles Schlagwort:

Process Life Cycle Management

Barbara Weber, Quality Engineering Research Group, University of Innsbruck

Contemporary information systems (IS) more and more have to be aligned in a process-oriented way. This new generation of information systems is often referred to as Process-Aware IS (PAIS) [DuHA05]. PAIS should capture real-world processes adequately to provide effective process support, i.e., there should be no mismatch between the computerized processes and those in reality. In order to achieve this goal, business processes must be supported in an integrated way.

For this purpose PAIS should allow authorized users to flexibly deviate from the predefined processes as required (e.g., to deal with exceptions) and to evolve PAIS implementations over time (e.g., due to process optimizations or legal changes). In addition, a framework is needed which enables the reuse of process knowledge when introducing new ad-hoc changes by learning from previous process instance changes. Moreover, the PAIS should support deriving optimized process models from them.

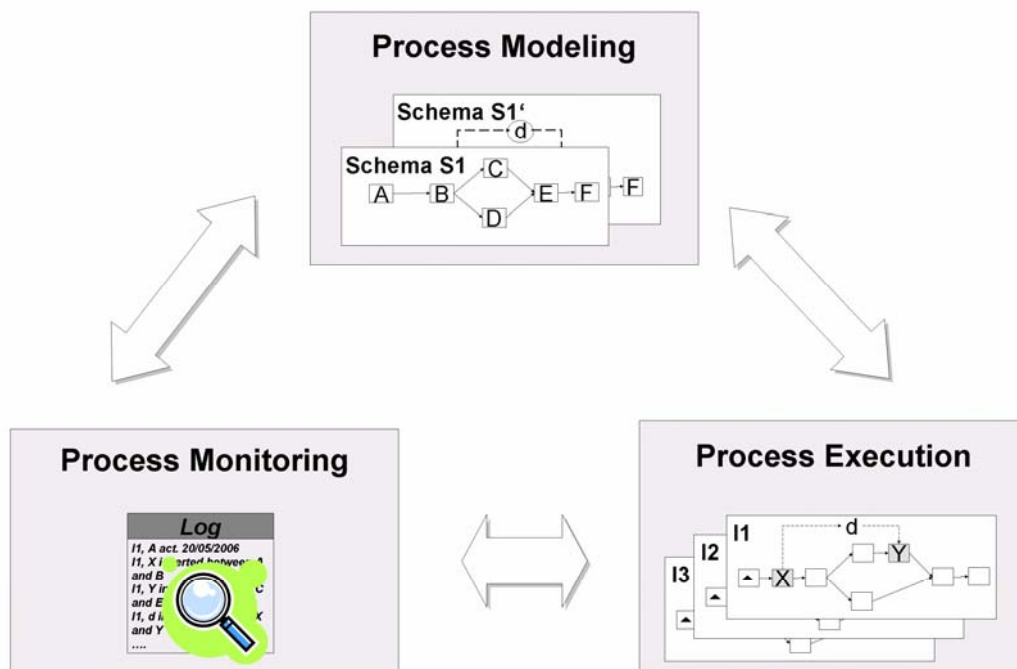
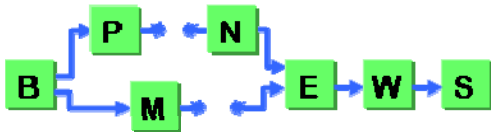


Figure 1: PAIS allow for the modelling, execution and monitoring of business processes



Process Modeling

PAIS enable users to model, execute, and monitor a company's business processes (cf. Fig. 1). In general, the orchestration of a business process is based on a predefined process model, called a *process schema*, consisting of the tasks to be executed (i.e., activities), their dependencies (e.g., control and data flow), organizational entities performing these tasks (i.e., actors), and business objects which provide or store data for the activities.

For each business process (e.g., booking a business trip or handling an order) a *process type* has to be defined for which different *process schema versions* may exist, reflecting the evolution of this process type. In Fig. 1, for example, process schemes *S1* and *S1'* correspond to two different versions of the same process type.

Process Execution

Based on a process schema new *process instances* are created and executed at run-time as specified in the underlying process schema. However, to deal with exceptions or unanticipated situations process participants must be able to deviate from the predefined execution path if required (cf. Fig. 2) [ReDa98]. The effects of such instance-specific changes should be kept local to the respective process instance, i.e., they must not affect other process instances of the same type. In Fig. 2, instance *I₂* has been individually modified by dynamically inserting two activities *X* and *Y* as well a data dependency between them. Thus the respective execution schema of *I₂* deviates from the original process schema *S1* this instance was derived from.

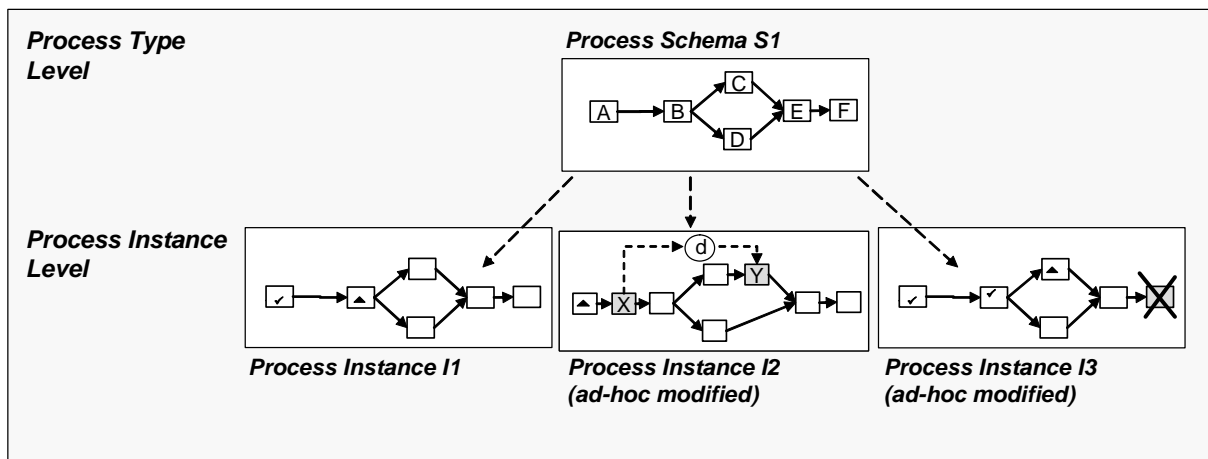
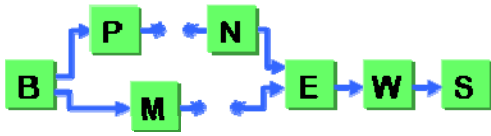


Figure 2: Individual Process Instances have to be modified due to Exceptions

Similar deviations can occur more than once. As it requires significant user experience to define changes from scratch, change reuse should be supported. In order to reuse changes they must be annotated with contextual information (e.g., about the reasons for the deviation) and be memorized by the PAIS. This contextual information can then be used for retrieving similar problem situations and therefore ensures that only changes relevant for the current situation are presented to the user [WWB04, RWRW05].



Process Monitoring

To ensure traceability the execution histories of process instances as well as changes are logged by the PAIS. This information can then be used to derive suggestions for process improvements [WRRW06, GRRR06]. To incorporate the respective process improvements into the process model, the PAIS must support process type changes.

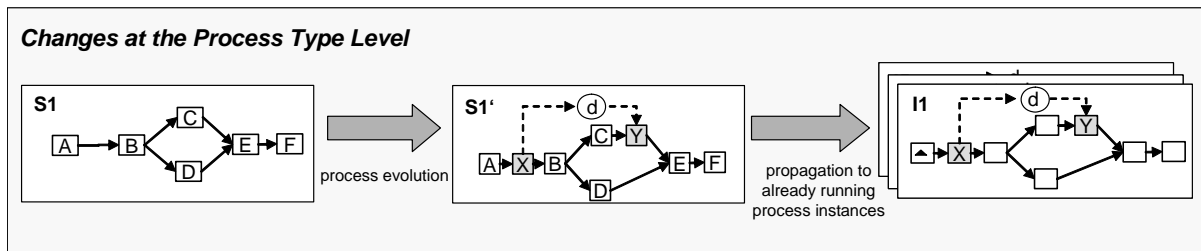


Figure 3: Business Processes Evolve over Time

These changes cover the evolution of real-world business processes and are performed by the process engineer [RRD04, CCP98, Wesk00]. As a result a new schema version of the same type is obtained (cf. Fig. 3). Usually, the execution of future process instances is then based on the new schema version. Regarding long running process instances, it might be required to additionally migrate them to the new process schema version [RRD04].

Integrated Process Lifecycle Support by Combining Business Process Management and Case-Based Reasoning

Fig. 4 shows how integrated process life cycle support can be achieved by combining process management technology and case-based reasoning [WRW06].

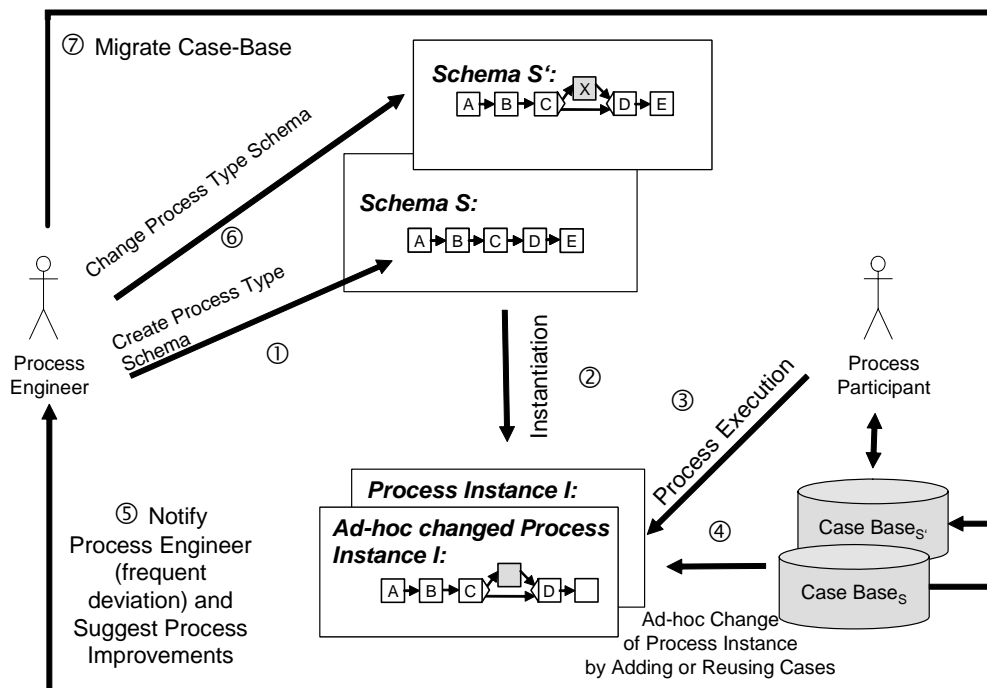
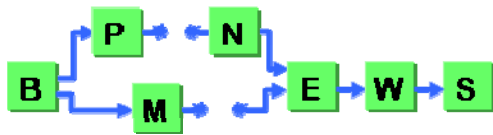


Figure 4: Integrated Process Lifecycle Support



At build time an initial representation of a business process is created either by process analysis or by process mining (i.e., by observing process and task executions) (1). At run time new process instances can then be created from the predefined process schema (2). In general, process instances are executed according to the process schema they were derived from, and activities are assigned to process participants to perform the respective tasks (3). However, when exceptional situations occur at the process instance level, process participants must be able to deviate from the predefined schema. Users can either define an ad-hoc deviation from scratch and document the reasons for the changes in the case base (CB), or they can reuse a previously specified ad-hoc modification from exactly this CB (4). The PAIS monitors how often a particular schema is instantiated and how often deviations occur. When a particular ad-hoc modification is frequently reused, the process engineer is notified that a process type change may have to be performed (5). The process engineer can then evolve the process schema (6). In addition, existing cases which are still relevant for the new process schema version are migrated to a new version of the CB (7).

References

- [DuHA05] Dumas, M., ter Hofstede, A., van der Aalst, W., eds.: Process Aware Information Systems. Wiley Publishing (2005).
- [CCPP98] Casati, F., Ceri, S., Pernici, B., Pozzi, G.: Workflow evolution. Data and Knowledge Engineering 24 (1998) 211–238.
- [GRR06] Günther, C., Rinderle, S., Reichert, M., van der Aalst, W.: Change Mining in Adaptive Process Management Systems. In: CoopIS'06. (2006) 309–326.
- [ReDa98] Reichert, M., Dadam, P.: ADEPTflex – Supporting Dynamic Changes of Workflows without Losing Control. JIIS 10 (1998) 93–129.
- [RRD04] Rinderle, S., Reichert, M., Dadam, P.: Correctness Criteria for Dynamic Changes in Workflow Systems - A Survey. Data & Knowledge Engineering 50 (2004) 9-34.
- [RWRW05] Rinderle, S., Weber, B., Reichert, M., Wild, W.: Integrating Process Learning and Process Evolution - A Semantics Based Approach. In: BPM 2005. (2005) 252–267.
- [Wesk00] Weske, M.: Workflow management systems: Formal Foundation, Conceptual Design, Implementation Aspects. University of Münster (2000) Habil Thesis.
- [WWB04] Weber, B., Wild, W., Breu, R.: CBRFlow: Enabling Adaptive Workflow Management through Conversational CBR. In: ECCBR'04, Madrid (2004) 434–448.
- [WRRW06] Weber, B., Reichert, M., Rinderle, S., Wild, W.: Towards a Framework for the Agile Mining of Business Processes. In: Proceedings Business Process Management Workshops, BPM 2005 International Workshops, BPI, BPD, ENI, BPRM, WSCOBPM, BPS (Revised Selected Papers) (2006) 191-202.
- [WRW06] Weber, B., Reichert, M., Wild W.: Case-Base Maintenance for CCBR-based Process Evolution. In: ECCBR'06, Fethiye, Turkey (2006) 106-120.