Aktuelles Schlagwort: Process Change Support Features

Manfred Reichert[1,2], Barbara Weber[3], and Stefanie Rinderle[1]

[1]Inst. Databases and Information Systems, Ulm University, Germany
{manfred.reichert, stefanie.rinderle}@uni–ulm.de
[2]Information Systems Group, University of Twente, The Netherlands
m.u.reichert@cs.utwente.nl
[3]Quality Engineering Research Group, University of Innsbruck, Austria
barbara.weber@uibk.ac.at

Abstract. To provide effective support, the introduction of process-aware information systems (PAIS) must not freeze existing business processes. Instead PAIS should allow authorized users to flexibly deviate from the predefined processes if required and to evolve business processes in a controlled manner over time. Many software vendors promise flexible system solutions for realizing such adaptive PAIS, but are often unable to cope with fundamental issues related to process change (e.g., correctness and robustness). In this paper we summarize a set of important change support features to foster systematic comparison of existing process management technology with respect to process change.

1 Introduction

More and more contemporary information systems (IS) have to be aligned in a process-oriented way. This new generation of IS is often referred to as Process-Aware IS (PAIS). To provide effective process support, PAIS should capture real-world processes adequately, i.e., there should be no mismatch between the computerized processes and those in reality. To achieve this, the introduction of PAIS must not lead to rigidity and freeze existing business processes. In domains like healthcare or automative engineering, for example, any PAIS will not be accepted by users if rigidity comes with it [1–3]. Instead 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) [4–7]. Such process changes should be enabled at a high level of abstraction and without affecting the robustness of the PAIS [5].

The increasing demand for process change support poses new challenges and requires the use of change enabling technologies. Many vendors promise flexible software solutions for realizing adaptive PAIS, but are often unable to cope with fundamental issues related to process change. So far, there exists no method for systematically comparing the change frameworks provided by existing process-support technologies. This, in turn, makes it difficult for PAIS engineers to assess

the maturity and change capabilities of those technologies. In [8] we have already suggested a set of *changes patterns* which allow for high-level process adaptations at the process type as well as the process instance level. This paper complements this work by summarizing a set of fundamental *change support features* needed to ensure that changes are performed in a correct and consistent way, traceability is provided, and changes at different levels are facilitated for users. Both change patterns and change support features are fundamental to make changes applicable in practice.

The remainder of this paper is organized as follows: Section 2 gives background information needed for the understanding of the further understanding. Section 3 summarizes 6 fundamental support features for process change. Section 4 concludes with a summary.

2 Backgrounds

A PAIS is a specific type of information system which allows for the separation of process logic and application code. At run-time the PAIS orchestrates the processes according to their defined logic. Workflow Management Systems (e.g., Staffware [9], ADEPT [4], and WASA [7]), for example, provide one of the technologies enabling PAIS.

For each business process to be supported a process type represented by a *process schema* S has to be defined. In the following, a process schema is represented by a directed graph, which defines a set of *activities* – the process steps – and control connections between them (i.e., the precedence relations between these activities). Activities can either be atomic or contain a sub process (i.e., a reference to a process schema S') allowing for the hierarchical decomposition of a process schema. In Fig. 1a, for example, process schema $S1$ consists of six activities: Activity `A` is followed by activity `B` in the flow of control, whereas `C` and `D` can be processed in parallel. Activities `A` to `E` are atomic, and activity `F` constitutes a sub process with own process schema $S2$. Based on a process schema S, at run-time new *process instances* $I_1, \ldots, I_n$ can be created and executed. Regarding process instance I_1 from Fig. 1a, for example, activity `A` is completed and activity `B` is activated (i.e., offered in user worklists). Generally, a large number of process instances might run on a particular process schema.

PAIS must be able to cope with change. In general, changes can be triggered and performed at two levels – the process type and the process instance level (cf. Fig. 1b) [5]. Schema changes at the type level become necessary to deal with the evolving nature of real-world processes (e.g., to adapt to legal changes). Ad-hoc changes of single instances are usually performed to deal with exceptions, resulting in an adapted *instance-specific* process schema.

3 Process Change Support Features

In general, a number of process change support features must be considered to enable flexible PAIS (cf. Fig. 2). Relevant change support features include *pro-*

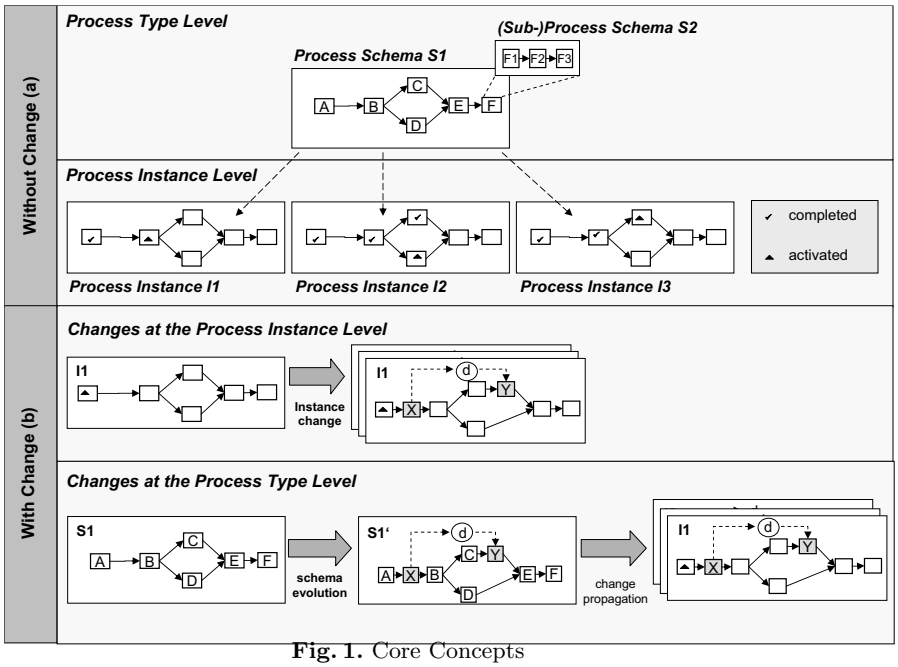

Fig. 1. Core Concepts

cess schema evolution and *version control*, ad-hoc changes, change correctness, change traceability, access control, and change reuse[1]. As illustrated in Fig. 2 the described change support features are not equally important for both process type level and process instance level changes. Version control, for instance, is primarily relevant for changes at the type level, while change reuse is particularly useful at the instance level (i.e., when conducting ad-hoc changes for single process instances) [12].

3.1 Schema Evolution, Version Control and Instance Migration

To support changes at the process type level, version control for process schemes should be supported (cf. Fig. 2). In case of long-running processes, in addition, controlled migration of already running instances, from the old process schema version to the new one, might be required [6, 5]. In this subsection we describe different existing options in this context (cf. Fig. 3).

If a PAIS provides no version control feature, either the process designer can manually create a copy of the process schema (to be changed) or this schema is

[1] We restrict ourselves to the most relevant change support features here. Additional change support features not covered here are change concurrency control and change visualization (cf. [11])

Change Support Features			
Change Support Feature	**Scope**	**Change Support Feature**	**Scope**
F1: Schema Evolution, Version Control and Instance Migration	T	2. By change primitives	
		F3: Correct Behavior of Instances After Change	I + T
No version control – Old schema is overwritten		**F4: Traceability & Analysis**	I + T
1. Running instances are canceled		1. Traceability of changes	
2. Running instances remain in the system		2. Annotation of changes	
Version control		3. Change Mining	
3. Co-existence of old/new instances, no instance migration		**F5: Access Control for Changes**	I+T
4. Uncontrolled migration of all process instances		1. Changes in general can be restricted to authorized users	
5. Controlled migration of compliant process instances		2. Application of single change patterns can be restricted	
F2: Support for Ad-hoc Changes	I	3. Authorizations can depend on the object to be changed	
1. By change patterns		**F6: Change Reuse**	I

T … Type Level, I … Instance Level

Fig. 2. Change Support Features

overwritten (cf. Fig. 3a). In the latter case running process instances can either be withdrawn from the run-time environment or, as illustrated in Fig. 3a, they remain associated with the modified schema. Depending on the execution state of the instances and depending on how changes are propagated to instances which have already progressed too far, this missing version control can lead to inconsistent states and, in a worst case scenario, to deadlocks or other errors [5]. As illustrated in Fig. 3a process schema $S1$ has been modified by inserting activities X and Y with a data dependency between them. For process instance $I1$ the change is uncritical, as $I1$ has not yet entered the change region. However, $I2$ and $I3$ would be both in an inconsistent state afterwards as instance schema and execution history do not match (see [5]). Regarding $I2$, worst case, deadlocks or activity invocations with missing input data might occur.

By contrast, if a PAIS provides explicit version control two support features can be differentiated: running process instances remain associated with the old schema version, while new instances will be created on the new schema version. This approach leads to the co-existence of process instances of different schema versions (cf. Fig. 3b). Alternatively a migration of a selected collection of process instances to the new process schema version is supported (in a controlled way) (cf. Fig. 3c). The first option is shown in Fig. 3b where the already running instances $I1$, $I2$ and $I3$ remain associated with schema S1, while new instances ($I4$-$I5$) are created from schema $S1'$ (co-existence of process instances of different schema versions). By contrast, Fig. 3c illustrates the controlled migration of process instances. Only those instances are migrated which are *compliant*[2] with $S1'$ ($I1$). All other instances ($I2$ and $I3$) remain running according to $S1$. If instance migration is uncontrolled (as it is not restricted to *compliant* process instances) this will lead to inconsistencies or errors. Nevertheless, we treat the

[2] A process instance I is compliant with process schema S, if the current execution history of I can be created based on S (for details see [5]).

uncontrolled migration of process instances as a separate design choice since this functionality can be found in several existing systems (like Staffware).

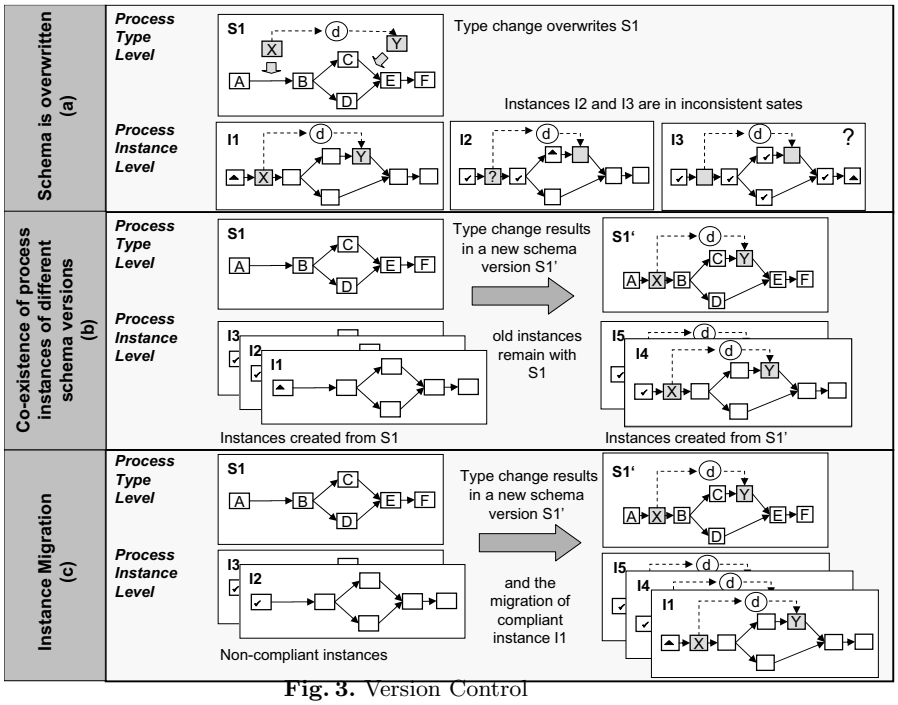

Fig. 3. Version Control

3.2 Other Change Support Features

Support for Instance-specific Process Changes: To deal with exceptions or unplanned situations the PAIS must support ad-hoc changes at the process instance level. Ideally such changes can be defined through high level adaptations in the form of change patterns (cf. [8, 4]). Examples include the dynamic insertion, deletion or movement of process activities. To deal with uncertainty, in addition, PAIS must allow keeping parts of the model unspecifieed during build-time and deferring the concretisation of the respective part to run-time. In both scenarios the effects resulting from the instance-specific process changes can be permanent or temporary. A permanent instance change remains valid until completion of the instance (unless it is undone by a user). By contrast, a temporary instance change is only valid for a certain period of time (e.g., the current iteration of a loop) [4].

Correctness of Change: The application of changes must not lead to runtime errors (e.g., activity program crashes due to missing input data, deadlocks, or inconsistencies due to lost updates or vanishing of instances). In particular, different criteria [6, 5] have been introduced to formally ensure that process instances can only be updated to a new schema if they are compliant with it. Depending on the used process meta model, in addition, (formal) constraints of the respective formalism (e.g., concerning the structuring of the the process schemes) have to be taken into account as well when applying process changes to a particular process schema.
Traceability and Analysis: To ensure traceability of changes, they have to be logged. For adaptation patterns the applied changes have to be stored in a change log as change patterns and/or change primitives. While both options allow for traceability, change mining [13] becomes easier when the change log contains high-level information about the changes as well. Regarding patterns for predefined changes, an execution log is usually sufficient to enable traceability. In addition, logs can be enriched with more semantical information, e.g., about the reasons and context of the changes [12]. Finally, change mining allows for the analysis of changes (e.g., to support continuous process improvement) [13].
Access Control for Changes: The support of change patterns leads to increased PAIS flexibility. This, in turn, imposes security issues as the PAIS becomes more vulnerable to misuse. Therefore, the application of changes at the process type as well as the process instance level must be restricted to authorized users [14]. Access control features differ significantly in their degree of granularity. In the simplest case, changes are restricted to a particular group of people (e.g., to process engineers). More advanced access control components allow to define restrictions at the level of single change operations (e.g., a certain user is only allowed to insert additional activities, but not to delete activities). In addition, authorizations can depend on the object to be changed, e.g., the process schema.
Change Reuse: In the context of ad-hoc changes "similar" deviations (i.e., combination of one or more adaptation patterns) can occur more than once. As it requires significant user experience to define changes from scratch change reuse should be supported. 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 be used for retrieving similar problem situations and therefore ensures that only changes relevant for the current situation are presented to the user [15, 12]. Regarding patterns for predefined changes, reuse can be supported by making historical cases available to the user and by saving frequently re-occurring instances as templates.

4 Summary and Outlook

In this paper we summarized 6 fundamental support features for changing processes in PAIS. In [16, 11] we additionally provide a detailed evaluation of selected approaches and systems regarding the support of these and other change sup-

port features. We believe that both change patterns and change support features contribute to better comparability of existing process change frameworks. In combination with workflow patterns they will enable (PA)IS engineers to choose process management technologies which meet their flexibility requirements best (or to realize that no system satisfies them at all).

References

1. Lenz, R., Reichert, M.: IT Support for Healthcare Processes - Premises, Challenges, Perspectives. Data and Knowledge Engineering (2007) 39–58
2. Dadam, P., Reichert, M., Kuhn, K.: Clinical workflows – the killer application for process-oriented information systems? In: Proc. Int'l Conf. on Business Information Systems (BIS'00), Poznan, Poland (2000) 36–59
3. Müller, D., Herbst, J., Hammori, M., Reichert, M.: It support for release management processes in the automotive industry. In: Business Process Management. (2006) 368–377
4. Reichert, M., Dadam, P.: ADEPT$_{flex}$ – supporting dynamic changes of workflows without losing control. JIIS **10** (1998) 93–129
5. Rinderle, S., Reichert, M., Dadam, P.: Correctness criteria for dynamic changes in workflow systems – a survey. Data and Knowledge Engineering **50** (2004) 9–34
6. Rinderle, S., Reichert, M., Dadam, P.: Flexible support of team processes by adaptive workflow systems. Distributed and Parallel Databases **16** (2004) 91–116
7. Weske, M.: Workflow management systems: Formal foundation, conceptual design, implementation aspects. University of Münster, Germany (2000) Habil Thesis.
8. Weber, B., Rinderle, S., Reichert, M.: Process change patterns (aktuelles schlagwort). EMISA Forum **27** (2007) 45 – 51
9. Dumas, M., ter Hofstede, A., van der Aalst, W., eds.: Process Aware Information Systems. Wiley Publishing (2005)
10. van der Aalst, W., Weske, M., Grünbauer, D.: Case handling: A new paradigm for business process support. Data and Knowledge Engineering. **53** (2005) 129–162
11. Weber, B., Rinderle, S., Reichert, M.: Change Support in Process-Aware Information Systems - A Pattern-Based Analysis. Technical Report TR-CTIT-07-76, Centre for Telematics and Inf. Technology, University of Twente (2007)
12. Rinderle, S., Weber, B., Reichert, M., Wild, W.: Integrating process learning and process evolution - a semantics based approach. In: BPM 2005. (2005) 252–267
13. Günther, C., Rinderle, S., Reichert, M., van der Aalst, W.: Change mining in adaptive process management systems. In: CoopIS'06. (2006) 309–326
14. Weber, B., Reichert, M., Wild, W., Rinderle, S.: Balancing Flexibility and Security in Adaptive Process Management Systems. In: CoopIS'06. (2005)
15. Weber, B., Wild, W., Breu, R.: CBRFlow: Enabling adaptive workflow management through conversational cbr. In: ECCBR'04, Madrid (2004) 434–448
16. Weber, B., Rinderle, S., Reichert, M.: Change patterns and change support features in process-aware information systems. In: Proc. CAiSE'07. (2007) 574–588