Towards Truly Flexible and Adaptive Process-Aware Information Systems*

Peter Dadam, Manfred Reichert, Stefanie Rinderle, Martin Jurisch, Hilmar Acker, Kevin Göser, Ulrich Kreher, and Markus Lauer

Ulm University, Institute of Databases and Information Systems,
James-Franck-Ring, 89081 Ulm, Germany
{peter.dadam,manfred.reichert,stefanie.rinderle, martin.jurisch,hilmar.acker,kevin.goeser,ulrich.kreher, markus.lauer}@uni-ulm.de

Abstract. If current process management systems shall be applied to a broad spectrum of applications, they will have to be significantly improved with respect to their technological capabilities. Particularly, in dynamic environments it must be possible to quickly implement and deploy new processes, to enable ad-hoc modifications of running process instances on-the-fly (e.g., to dynamically add, delete or move process steps), and to support process schema evolution with instance migration (i.e., to propagate process schema changes to already running instances if desired). These requirements must be met without affecting process consistency and by preserving the robustness of the process management system. In this paper we describe how these challenges have been addressed and solved in the ADEPT2 Process Management System. Our overall vision is to provide a next generation process management technology which can be used in a variety of application domains.

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) [1]. Recently, numerous technologies and paradigms have emerged in this context such as Workflow Management, Business Process Management, Enterprise Application Integration, and Service-oriented Architectures (SOA). They all focus on the realization of PAIS [1]. By offering system-based support for implementing business processes, these technologies aim at an increased efficiency and adaptivity of enterprises regarding their internal processes. By combining process management with SOA the interaction between enterprises and their customers and partners shall be improved as well.

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, PAIS enabling technologies must fulfill a number of requirements:

* The development of ADEPT2 was funded by the State of Baden-Württemberg as part of the AristaFlow project (see www.AristaFlow.de).

R. Kaschek et al. (Eds.): UNISCON 2008, LNBIP 5, pp. 72–83, 2008.

1. They must cover a broad spectrum of applications ranging from form- or document-centered workflows to complex production workflows (where application integration constitutes a major task).
2. They must allow for the rapid and cost-effective implementation of a large variety of business processes.
3. The implemented processes must run in a robust and stable manner. The overall objective should be "robustness by design".
4. PAIS must not lead to rigidity and freeze existing business processes. Instead, they must allow authorized users to flexibly deviate from the predefined processes as required (e.g., to deal with exceptions). Such ad-hoc process changes should be enabled at a high level of abstraction and without affecting robustness of the PAIS.
5. Due to process optimization or legal changes PAIS implementations evolve over time Respective process changes have to be accomplished in an easy and cost-effective way. For long-running processes the "on-the-fly" adaptation of already running process instances to the new process schema should be possible as well.

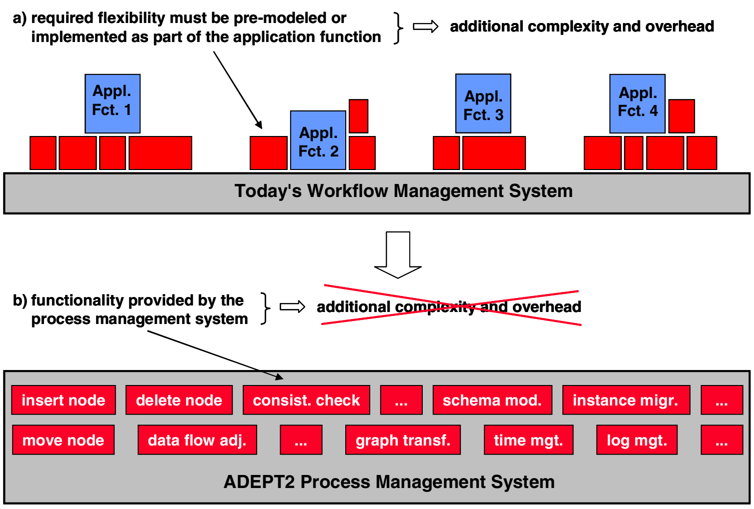


Fig. 1. Overhead caused by realizing system functions within the application programs is avoided by providing the required functionality as integral part of the ADEPT2 system

Off-the-shelf process management systems do not meet these requirements or offer restricted features [1,2]. Several vendors promise flexible process support, but are unable to cope with fundamental issues related to process change (e.g., correctness). Most systems, however, completely lack support for deviating from the predefined processes in an ad-hoc manner or for migrating process instances to a changed process schema. Thus, application developers are forced to "enrich" applications with respective process support functions to deal with these limitations. This, in turn, aggravates PAIS development and maintenance significantly and shifts the risk of errors and the task to deal with them to application developers or end-users.

This paper presents the ADEPT2 process management system – one of the leading technologies for realizing flexible and adaptive processes. Using ADEPT2, process-oriented applications can be composed out of existing application components in a plug & play like fashion, and then be flexibly executed at run-time. The ADEPT2 technology enables support for a broad spectrum of processes, ranging from simple document-centered workflows to complex production workflows, which integrate heterogeneous, distributed application components. We illustrate how ad-hoc changes of single process instances as well as process schema changes with (optional) propagation of these changes to the running instances are supported by ADEPT2 in an integrated, safe, and easy-to-use manner. In particular, application programmers and users of the ADEPT2 system are not confronted with the inherent complexity coming with dynamic changes (as indicated in Fig. 1a). Instead, this functionality is easy to use since it is an integral part of ADEPT2.

The remainder of this paper is structured as follows: Sect. 2 sketches how process composition is realized based on plug & play. In Sect. 3 we show how ad-hoc process adaptations can be accomplished by end users and how the interaction between the user and the ADEPT2 system looks like. In Sec. 4 we discuss process schema changes and the adaptations of already running process instances. Sect. 5 describes the current status of the ADEPT2 technology and Sect. 6 discusses current trends and related work. We close with a summary and an outlook in Sec. 7.

2 Process Composition by Plug & Play

A new process can be realized by creating a process template (also denoted as process schema). Such a template describes the planned order of the process steps (e.g., sequential, parallel, alternative paths, loops, etc.) as well as the data flow between them. It either has to be defined from scratch or an existing template is chosen from the process template repository and adapted as needed ("process cloning"). Afterwards application functions (e.g., web services, Java components, ERP functions, or legacy applications) have to be assigned to the process steps. When using the ADEPT2 process editor, these functions can be selected from the component repository and be inserted into the process template by drag & drop (cf. Figure 2). Following this, ADEPT2 analyzes whether the application functions can be connected in the desired order; e.g., we check whether the input parameters of application functions can be correctly supplied for all possible execution paths imposed by the process schema. Furthermore, additional checks are performed in order to exclude deadlocks, livelocks, etc. Only those process templates passing these correctness checks may be released and transferred to the ADEPT2 runtime system.

When dragging application components from the repository and assigning them to particular steps in the process template, the process designer does not need to have detailed knowledge about the implementation of these components. Instead the component repository provides an integrated, homogeneous view as well as access to the different components. Internally, this is based on a set of wrappers provided for the different types of application components. The chosen architecture will allow to add new wrappers if new component types shall be supported. Currently, the ADEPT2 Execution Environment Framework allows to integrate different kinds of application components like electronic forms, stand-alone executables, web services, Java library functions, and function calls to legacy systems. All these application components require different treatment when interacting with them.

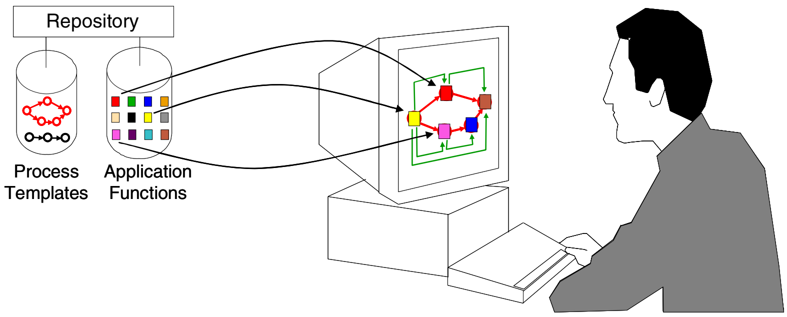


Fig. 2. Composition of correct processes using plug & play

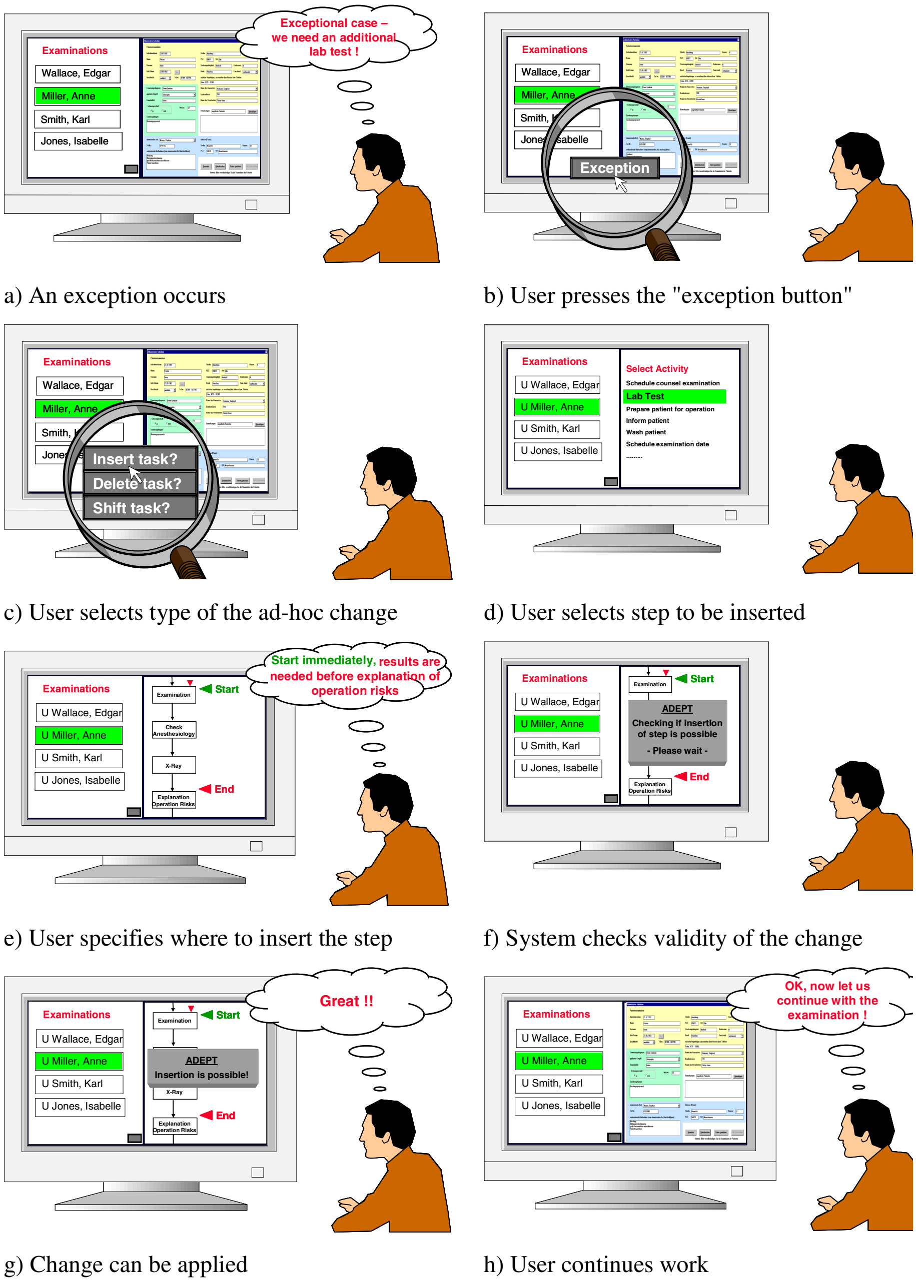
3 Support of Ad-Hoc Adaptations

Composing processes in a plug & play like fashion is very useful for developers since it allows for rapid implementation of new processes. However, composition support alone does not constitute a big technological progress when compared to the state-of-the-art. If process management technolgy shall become applicable to a broader spectrum of applications than today, it must allow for ad-hoc deviations from the pre-defined process schema as well. Such runtime changes must not violate workflow correctness. Further, authorized users should be able apply ad-hoc changes in an intuitive way to.

Figure 3a – h illustrate how the interaction between the ADEPT2 system and the end user may look like. In this example it is assumed that during the execution of a particular process instance (e.g., the treatment of a certain patient under risk) an additional lab test becomes necessary. Assume that this has not been foreseen at buildtime (cf. Fig. 3a). As a consequence, this particular process instance will have to be individually adapted if the change request is approved by the system. After the user has pressed the "exception button" (cf. Fig. 3b), he can specify the type of the intended ad-hoc change (cf. Fig. 3c). If an insert operation shall be applied, for example, the system will display the application functions that can be added in the given context (cf. Fig. 3d). These can be simple or complex application components (e.g., "write letter" or "send email" vs. application services), interactive or automatic functions, or even complete processes.

Now the user simply has to state after which step(s) in the workflow the execution of the newly added activity shall be started and before which step(s) it shall be finished (cf. Fig. 3e). Finally, the system checks whether resulting process instance adaptations are valid (cf. Figure 3f and Figure 3g).

In this context, the same checks are performed as during the process design phase (e.g., absence of deadlocks or validity of actor assignment expressions). In addition, the current process instance state is taken into account when the instance is modified.



a) An exception occurs
b) User presses the "exception button"
c) User selects type of the ad-hoc change
d) User selects step to be inserted
e) User specifies where to insert the step
f) System checks validity of the change
g) Change can be applied
h) User continues work

Fig. 3. Executing an ad-hoc modification from the end user's point of view

On the one hand this allows for modifications which would not be valid at design time (e.g., due to uncertainty at design time which execution branches will be taken). On the other hand the process state also restricts possible changes (see [3] for details).

All implemented change operations are also available via the ADEPT2 application programming interface. Furthermore, changes can be specified at a semantically high level of abstraction (e.g. "Insert Step X between Node Set 1 and Node Set 2"), which eases change definition significantly [2]. All these operations are guarded by preconditions which are automatically checked by the system when the operation is invoked. The related post conditions guarantee that the resulting process instance graph is again "problem-free". Users or application programs only interact via these high level API functions with ADEPT2. They never do have to manipulate system-internal states directly (see [4] for details).

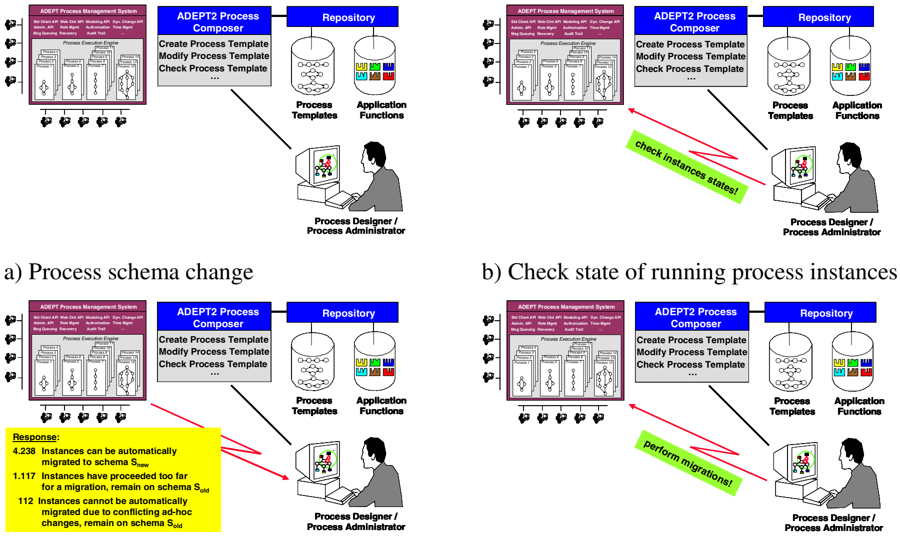
For several reasons (e.g., change traceability) ADEPT2 stores process instance changes within so called change logs. Together with the enactment logs, which capture the execution information of process instances, the structure and state of a particular process instance can be reconstructed at any time. This log information is also a valuable source for process optimizations because repeatedly performed ad-hoc modifications may be an indicator that the process has been not optimally designed [5].

4 On Supporting Process Schema Evolution

Though the support of ad-hoc modifications is very important, it is not yet sufficient. In the context of long-running business processes, it is often required to adapt the process schema itself (e.g., due to organizational changes in the company or because of business process optimizations). In this case all process instances based on this process schema may be affected by the change. If the processes are of short duration only, already running process instances usually can be finished according to the old schema version. However, this strategy will be not applicable for long running business processes. Then the old process version may no longer be applicable, e.g., when legal regulations have changed or when the old process reveals severe problems. One solution would be to individually modify each of the running process instances by applying corresponding ad-hoc changes (as described in the previous section). However, this would be too expensive and error-prone if a multitude of running process instances had been involved. Note that the number of active process instances may become very large; i.e., change propagation must be accomplished in a very efficient manner for hundreds or thousands of process instances.

An adaptive process management system must be able to support correct changes of a process schema and their subsequent propagation to already running process instances if desired. In other words, if a process schema is changed and thus a new version of this schema is created, process instances should be allowed to migrate to the new schema version (i.e., to be transferred and re-linked to the new process schema version). In this context, it is of particular importance that ad-hoc changes of single process instances and instance migrations do not exclude each other since both kinds of changes are needed for the support of long-running processes!

The ADEPT2 technology implements the combined handling of both kinds of changes. Process instances which have been individually modified can be also migrated to a changed process schema if this does not cause inconsistencies or errors in the sequel. All needed correctness checks (on the schema and the state of the



a) Process schema change

b) Check state of running process instances

c) Result of checks

d) Execute instance migration

Fig. 4. Process schema evolution

instances) and all adaptations to be accomplished when migrating the instances to the new process schema version are performed by ADEPT2. The implementation is based on a comprehensive formal framework [3, 6]. Based on it, ADEPT2 can precisely state under which conditions a certain process instance can be migrated to the new process schema version. This allows for checking the compliance of a collection of process instances with the changed schema version in an efficient manner. Finally, concurrent and conflicting changes at the process type and the process instance level are managed in a reliable and consistent manner as well. In particular, long-running processes will benefit from this close integration of the different change levels.

Figure 4a – Figure 4c illustrate how such a process schema evolution is conducted from the user's point of view in ADEPT2. The process designer loads the process schema from the process template repository, adapts it (using the ADEPT2 process editor), and creates a new schema version (cf. Figure 4a). Then the system checks whether the running process instances can be correctly migrated to the new process schema version (cf. Figure 4b and Figure 4c). These checks are based on state conditions and structural comparisons. Furthermore, the system calculates which adaptations become necessary to perform the migration at the process instance level. The ADEPT2 system analyzes all running instances of the old schema and creates a list of instances which can be migrated as well as a list of instances for which this is not possible (together with a report which explains the different judgments). When pressing the "migration button" the system automatically conducts the migration for all selected process instances (see Figure 4d).

5 On Transferring the ADEPT2 Technology to Business

The vision of enabling ad-hoc modifications within the process management system in a correct and consistent manner was the starting point for our research and implementation work done within the ADEPT project more than 10 years ago [4]. The resulting technology has been integrated in the experimental ADEPT1 system which, to our best knowledge, is still leading in the field of adaptive process management today. The ADEPT1 technology has enabled ad-hoc deviations in a controlled, secure, and user-friendly manner. Unforeseen exceptions can be handled within the PAIS and not by bypassing it as often required when using commercial process management systems. The ADEPT1 system has been used in several national and international research projects.

From these projects we gained valuable insights into the practical needs for process management technology on the one side, and we learned many lessons about implementing a complex system such as ADEPT1 on the other side. Partly these "lessons learned" have been published covering topics like log management, system-internal representation of process schemas and process instances, and the transfer of selected implementation concepts to other application areas such as data warehouses [7-9]. One important perception is that the system design and architecture should offer powerful functionality at a semantically high level and hide the inherent complexity as much as possible from the user.

To transfer the ADEPT2 technology into industrial usage and business, the AristaFlow GmbH was jointly founded by members of our institute and industrial partners (see [10] for details). The major focus of this company is to implement a robust and scalable commercial version of the ADEPT2 system. ADEPT2 includes all functionality of the old ADEPT1 system. Ad-hoc flexibility is now based on a broader range of supported operations, however. In addition, ADEPT2 realizes the composition of processes based on plug & play techniques as sketched in Section 2. Furthermore, it implements the theoretical framework for process schema evolution as outlined in Section 4. Altogether, the design and implementation of such a powerful and innovative system constitutes a big challenge. However, we can now exploit our lessons learned from implementing the ADEPT1 system. Finally, we benefit very much from the cooperation with our industrial partners and the University of Mannheim within the AristaFlow project (see [10] for details).

6 Current Trends and Related Work

This section discusses current trends and approaches from literature and relates them to ADEPT2:

Plug & Play: Recently, lots of attention has been paid to the area of (dynamic) web service composition [11,13]. In particular, the emergence of WS-BPEL (Business Process Execution Language) has resulted in many research activities (e.g., on match making [12]) as well as new process composition tools and service flow engines (e.g., IBM WebSphere Process Server or SAP Netweaver). How to provide intelligent support for finding the right web services or partners, however, has not been a research topic in ADEPT2 so far. Nevertheless, the basic functionality for composing services in a process-oriented way is already provided by ADEPT2 as described in Section 2. In particular, ADEPT2 is able to invoke any kind of standard component (including web services and Java components); it further offers powerful interfaces to integrate arbitrary application components with little programming effort.

Process Choreography & Orchestration. Though a distributed variant of ADEPT1, which supports distributed process execution, has been developed [14, 15] and some attention was paid to inter-workflow coordination [16, 17], the focus of ADEPT2 is on orchestration; i.e., on the coordination and enactment of (business) processes from the viewpoint of one company. Opposed to that, conversation languages like WS-CDL or WSCI have been designed for defining the choreography of different partner processes (i.e., for global processes where each partner runs an internal process and provides a public view on it based on which it exchanges messages with partners).

A promising integration variant of ADEPT2 with such web service standards is conceivable: While ADEPT2 defines and manages internal (i.e., private) processes, WS-BPEL can be used for describing public process views and WS-CDL for defining the choreography of partner processes based on these public views (i.e., the global protocol based on which the processes of the different partners communicate). The advantage of this approach would be that with ADEPT2 any application component and particularly interactive process steps can be called within the internal processes. This is useful since internal processes typically comprise a mix of interactive and automated activities as well as a variety of different application functions such as Java components, Web services, but also complex ERP system functions. Using the ADEPT2 technology the correctness of the modeled (plugged) processes can be guaranteed based on the formal checks on, for example, control and data flow.

Adaptive Process Management: The flexible support of business processes has been a hot topic in research for a long time. Mostly, approaches either deal with ad-hoc deviations at process instance level or process schema evolution [3, 18, 19]. The same applies for commercial systems offering some, but very limited flexibility (e.g., Staffware or Ultimus Workflow). Only few approaches and prototypes [20, 21, 22] allow for both kinds of process changes, but in an isolated manner. To our best knowledge, ADEPT2 is the only adaptive PMS which supports ad-hoc deviations, process schema evolution, and their interplay based on a sound theoretical framework and within one implemented system.

Finally, several approaches exist that allow to define "placeholder activities" in a process model for which a concrete sub-process can be bound or modeled during runtime. Representatives of this system category include PocketsOfFlexibility [23] and Worklets [24]. Typically, late modeling has to be finished before creating the corresponding sub-process instance. Though late binding and late modeling increase process flexibility (see [2] for a detailed discussion), they do not allow for structural changes of already running instances. Recently a list of change patterns and change support features have emerged, which allow to systematically compare change and flexibility support in existing PAIS [2].

7 Summary and Outlook

The ADEPT2 technology meets major requirements claimed for next generation adaptive process management systems: it provides advanced functionality to support process composition by plug & play of arbitrary application components, it enables ad-hoc flexibility for process instances without losing control (i.e., without causing process execution errors or inconsistencies), and it supports process schema evolution in a controlled and efficient manner. As opposed to many other approaches all these aspects work in interplay as well. For example, it is possible to propagate process schema changes to individually modified process instances or to dynamically compose processes out of existing application components. All in all such a complex system requires a sound theoretical framework in order to avoid incomplete solutions and implementation gaps [3-5]. Finally, it is important to mention that most of our theoretical results have not been just kept "on paper", but have been implemented within the process management systems ADEPT1 and ADEPT2 respectively. As ADEPT2 additionally provides powerful tools and application programming interfaces its transfer to and its use in practice will be further eased.

In addition to the features presented in this paper, the ADEPT2 technology offers promising perspectives for process learning and continuous process optimization [5, 26]. In particular, audit data (i.e., process logs) become much more meaningful, since they do not only capture process execution events (e.g., start / completion of activities), but also contain insightful information on performed ad-hoc changes at the process instance level [8]. By mining the change logs related to a collection of individually modified process instances, we can (semi-)automatically derive potential process improvements (i.e., changes to be applied to the original process schema).

Recently, we have started working on process change mining, and first project results indicate the new opportunities emerging in this context [26]. Furthermore, process schema adaptations derived with respective mining techniques can be implemented with ADEPT2 in a much quicker and more cost effective way when compared to existing technology. Thus, continuous process evolution and full process lifecycle support become possible. Finally, other interesting perspectives arise when introducing adaptive process management to practice. This includes the support for emergent processes [30], the "outsourcing" of successfully applied exception handling procedures to knowledge management components [5, 27], the automatic, rule-based adaptation of process instances [28, 29], or the support of ad-hoc workflows.

In future work we will incorporate more semantic knowledge into the ADEPT2 framework, i.e., it shall become possible to specify semantical integrity constraints on business processes [25]. We also aim at providing efficient methods to check the validity of such semantic constraints when changing processes. Following such a constraint-based approach it becomes possible to not only guarantee that processes run correctly regarding their control and data flow, but also regarding the validity of the specified semantic constraints. As another important task we will elaborate the use of ADEPT2 in different practical settings and application areas (e.g., healthcare [31,32]).

References

1. Dumas, M., van der Aalst, W., ter Hofstede, A.: Process-aware Information Systems. John Wiley & Sons, Chichester (2005)
2. Weber, B., Rinderle, S., Reichert, M.: Change Patterns and Change Support Features in Process-Aware Information Systems. In: Krogstie, J., Opdahl, A., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 574–588. Springer, Heidelberg (2007)
3. Rinderle, S., Reichert, M., Dadam, P.: Correctness Criteria for Dynamic Changes in Workflow Systems - A Survey. Data and Knowledge Engineering 50(1), 9–34 (2004)
4. Reichert, M., Dadam, P.: ADEPTflex - Supporting Dynamic Changes of Workflows Without Losing Control. Journal of Intelligent Information Systems 10(2), 93–129 (1998)
5. Rinderle, S., Weber, B., Reichert, M., Wild, W.: Integrating Process Learning and Process Evolution – A Semantic Based Approach. In: van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds.) BPM 2005. LNCS, vol. 3649, pp. 252–267. Springer, Heidelberg (2005)
6. Rinderle, S., Reichert, M., Dadam, P.: Flexible Support Of Team Processes By Adaptive Workflow Systems. Distributed and Parallel Databases 16(1), 91–116 (2004)
7. Rinderle, S., Kreher, U., Lauer, M., Dadam, P., Reichert, M.: On Representing Instance Changes in Adaptive Process Management Systems. In: Proc. ProFlex 2006, First IEEE Workshop on Flexibility in Process-aware Information Systems, Manchester (2006)
8. Rinderle, S., Reichert, M., Jurisch, M., Kreher, U.: On Representing, Purging, and Utilizing Change Logs in Process Management Systems. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 241–256. Springer, Heidelberg (2006)
9. Rinderle, S., Jurisch, M., Reichert, M.: On Deriving Net Change Information From Change Logs - The DeltaLayer Algorithm. In: Proc. BTW 2007, Datenbanksysteme in Business, Technologie und Web, Aachen. LNI, vol. 103, pp. 364–381 (2007)
10. http://www.AristaFlow.com (AristaFlow GmbH), http://www.AristaFlow.de (AristaFlow project)
11. Khalaf, R., Mukhi, N., Weerawarana, S.: Service-oriented composition in BPEL4WS. In: Proc. WWW 2003, Budapest (2003)
12. Wombacher, A., Fankhauser, P., Mahleko, B., Neuhold, E.: Matchmaking for business processes based on choreographies. IJWS 1(1), 14–32 (2004)
13. Yi, X., Kochut, K.J.: Process composition of web services with complex conversation protocols. In: Proc. Conf. on Design, Analysis, and Simulation of Distributed Systems Symposium at Advanced Simulation Technology, pp. 141–148 (2004)
14. Bauer, T., Reichert, M., Dadam, P.: Intra-Subnet Load Balancing in Distributed Workflow Management Systems. Int'l J. of Cooperative Information Systems 12(3), 205–223 (2003)
15. Bauer, T., Dadam, P.: A Distributed Execution Environment for Large-Scale Workflow Management Systems with Subnets and Server Migration. In: Proc. CoopIS 1997, Kiawah Island, South Carolina, USA, pp. 99–108 (1997)
16. Heinlein, C.: Synchronization of Concurrent Workflows Using Interaction Expressions and Coordination Protocols. In: Meersman, R., Tari, Z., et al. (eds.) CoopIS 2002, DOA 2002, and ODBASE 2002. LNCS, vol. 2519, pp. 54–71. Springer, Heidelberg (2002)
17. Heinlein, C.: Workflow and Process Synchronization with Interaction Expressions and Graphs. In: Proc. ICDE 2001, Heidelberg, Germany, April 2001, pp. 243–252 (2001)
18. van der Aalst, W.M.P., Basten, T.: Inheritance of Workflows: An Approach to Tackling Problems Related to Change. Theoretical Computer Science 270(1-2), 125–203 (2002)
19. Casati, F., Ceri, S., Pernici, B., Pozzi, G.: Workflow Evolution. Data & Knowledge Engineering 24(3), 211–238 (1998)

20. Kochut, K., Arnold, J., Sheth, A., Miller, J., Kraemer, E., Arpinar, B., Cardoso, J.: IntelliGEN: A distributed workflow system for discovering protein-protein interactions. Distributed and Parallel Databases 13(1), 43–72 (2002)
21. Weske, M.: Formal Foundation and Conceptual Design of Dynamic Adaptations in a Workflow Management System. In: Proc. HICSS-34, Maui, Hawaii, vol. 7, p. 7051 (2001)
22. Pesic, M., Schonenberg, M.H., Sidorova, N., van der Aalst, W.M.P.: Constraint-Based Workflow Models: Change Made Easy. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 77–94. Springer, Heidelberg (2007)
23. Sadiq, S., Sadiq, W., Orlowska, M.: Pockets of flexibility in workflow specifications. In: Kunii, H.S., Jajodia, S., Sølvberg, A. (eds.) ER 2001. LNCS, vol. 2224, pp. 513–526. Springer, Heidelberg (2001)
24. Adams, M., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.: Worklets: A service-oriented implementation of dynamic flexibility in workflows. In: Meersman, R., Tari, Z. (eds.) OTM 2006. LNCS, vol. 4275, pp. 291–318. Springer, Heidelberg (2006)
25. Ly, L.T., Rinderle, S., Dadam, P.: Semantic Correctness In Adaptive Process Management Systems. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 193–208. Springer, Heidelberg (2006)
26. Günther, C.W., Rinderle, S., Reichert, M., van der Aalst, W.M.P.: Change Mining in Adaptive Process Management Systems. In: Meersman, R., Tari, Z. (eds.) OTM 2006. LNCS, vol. 4275, pp. 319–326. Springer, Heidelberg (2006)
27. Weber, B., Wild, W., Breu, R.: CBRFlow: Enabling Adaptive Workflow Management Through Conversational CBR. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 434–448. Springer, Heidelberg (2004)
28. Müller, R.: Event-oriented dynamic adaptation of workflows. Ph.D. Thesis, University of Leipzig, Germany (2002)
29. Greiner, U.: Quality-oriented Execution and Optimization of Cooperative Processes: Model and Algorithms. Ph.D. Thesis, University of Leipzig, Germany (2005)
30. Jørgensen, H.D.: Interactive Process Models. Ph.D. Thesis, Norwegian University of Science and Technology, Trondheim, Norway (2004)
31. Lenz, R., Reichert, M.: IT Support for Healthcare Processes - Premises, Challenges, Perspectives. Data and Knowledge Engineering 61, 39–58 (2007)
32. Dadam, P., Reichert, M., Kuhn, K.: Clinical Workflows - The Killer Application for Process-oriented Information Systems? In: Proc. 4th Int’l Conf. on Business Information Systems (BIS 2000), Poznan, Poland, pp. 36–59 (2000)