Next-generation Process Management with ADEPT2 *

Kevin Göser[1], Martin Jurisch[1], Hilmar Acker[1], Ulrich Kreher[1], Markus Lauer[1], Stefanie Rinderle-Ma[1], Manfred Reichert[2], and Peter Dadam[1]

[1] Institute of Databases and Information Systems, University of Ulm, Germany, {kevin.goeser, martin.jurisch, hilmar.acker, ulrich.kreher, markus.lauer, stefanie.rinderle, peter.dadam}@uni-ulm.de
[2] University of Twente, IS Group, 7500 AE Enschede, The Netherlands, m.u.reichert@cs.utwente.nl

Abstract. Short time-to-market, easy adaptation to changes in business environment, and robustness of processes are key requirements in today's business world. In the IT area of Business Process Management (BPM), solutions claim to satisfy these new demands, but are still not sufficient. In this paper we present a short overview on how these challenges are tackled by the ADEPT and AristaFlow projects and demonstrate a prototypical implementation.

1 Introduction

Today's enterprises are facing increasingly fast changing requirements in the business world and permanently growing market pressure. Questions like "How fast can we deploy new processes?" or "How fast can we react to market changes and customer requirements?" become essential. Existing Business Process Management (BPM) tools can not fulfill the real-life requirements and particularly fail to quickly react to changes. Quick deviations from the predefined workflow while still guaranteeing the absence of execution failures (e.g., due to missing input parameters when invoking application functions) in the sequel is hardly achieved, if at all.

We recognised this weakness of academic and commercial BPM systems already in the middle of the 90's and addressed it in the ADEPT project [1, 2] and in the implemented ADEPT system (referred to as "ADEPT1" in the following) whose first version became available in 1998. ADEPT1 mainly focused on the demonstration of ad-hoc flexibility [1] and the distributed execution of processes [3]. The existing concepts have been extended, especially in the area of *process schema evolution*, which allows for automatic propagation of process type-specific changes to already running instances [4].

In order to incorporate existing and new results of our research (e.g., process composition by plug-and-play [5], semantic constraints [6]) as well as the aspect

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

of extensibility in a comprehensive system, we started the design and implementation of ADEPT2 as part of the AristaFlow project* in 2004. The remainder of this paper is organised as follows: Sect. 2 addresses the aspect of fast business process implementation, Sect. 3 deals with flexibility, and Sect. 4 describes the system architecture of ADEPT2.

2 Business Process Implementation

The implementation of business processes comprises two important aspects: modelling the control and data flow, on the one side, and the integration of application functions (services), on the other side. Both parts need to be supported, such that the user can implement and deploy the process much faster than today, and can rely on the robustness of the composed process at runtime.

In ADEPT2, processes are modelled by applying high-level *change operations* starting from an initial (empty) process graph. These change operations, e.g., `insertNode`, have formally well-founded preconditions and postconditions which ensure the structural correctness of the resulting process graph with regard to the control flow [1]. This approach enables a fast, syntax-driven modelling of the control flow and provides guidance to the process designer.

Application integration in ADEPT2 works in a plug-and-play fashion by selecting activities (e.g., operations of a component) from a repository and drop them onto steps in the process. A little wizard supports the semi-automatic mapping of the parameters of the activity to the data flow of the process. Formal checks verify the mapping as well as the correctness of the overall data flow [1].

3 Flexibility

Incorporating all possible exceptional situations in a process template is hardly to achieve and would also lead to very complex processes. Therefore, modern PMS have to support deviations from the predefined workflow, i.e., it must enable authorised users to perform individual *ad-hoc modifications* to running instances. This, in turn, demands for an adequate user-friendly interface.

In ADEPT2, these requirements are met by using the same high-level *change operations* for ad-hoc modifications as are used for process composition (cf. Sect. 2). They are only enriched with constraints concerning the execution state of the process instance. This approach allows offering a semantically high-level API to users which makes ad-hoc modifications nearly trivial in many cases.

Beside ad-hoc deviations, the overall business process itself may evolve over time [4] (e.g., due to changes in business strategy, laws or policies) and, therefore, may require process changes at template level. In such cases, it is often necessary to propagate the changes to already running instances of the process (*instance migration*). ADEPT offers formal criteria for checking the feasibility of a migration, which also consider the interplay of ad-hoc modifications and template changes [4].

4 Architecture

The preceding sections already give an impression of the complexity which comes along with building a BPM system satisfying today's needs. However, since the BPM area is still developing, a BPM system must be open for future requirements, which implies a clean and extensible architecture.
ADEPT2 features a layered architecture including a strong separation of duties, as depicted by Fig. 1. The first layer is a thin abstraction on SQL, featuring database implementation independent persistence. The second layer is responsible for storage and locking of different entities (e.g., process instances) in the system. The third layer encapsulates the essential process support like process execution and change management. The topmost layer establishes the connection to the user, including worklist management and user interfaces.

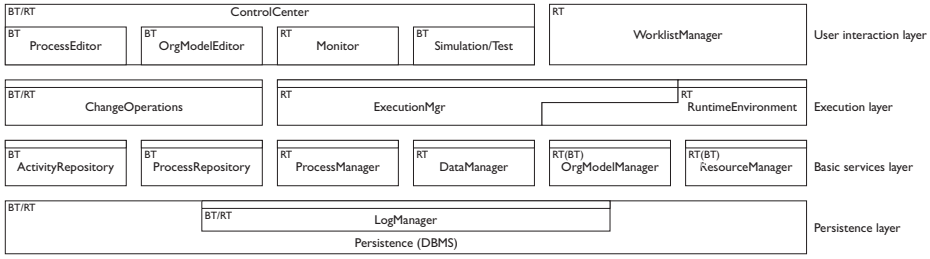

Fig. 1. Architecture of ADEPT2 (Overview)

All of the ADEPT2 core components are loosely coupled services, which enables the easy exchange of the implementation and allows for exchangeable communication techniques between the services. Apart from exchanging service implementations, further plug-in interfaces are provided all over the system, which allow for extension of the core, the data models, and the user interface.
The presented system is a premature version of the future ADEPT2 system and is based on this architecture. However, as the implementation of ADEPT2 is still ongoing, not all functions and components are completely implemented, yet.

5 Demonstration

We will present two parts of the user interface of the ADEPT2 prototype: the *ADEPT2 Process Template Editor* and the *ADEPT2 Test Client*, cf. Fig. 2.

The ADEPT2 Process Template Editor is the main component for modelling process templates, providing means from simple control flow construction to modelling processes for productive execution. The following features will be demonstrated:

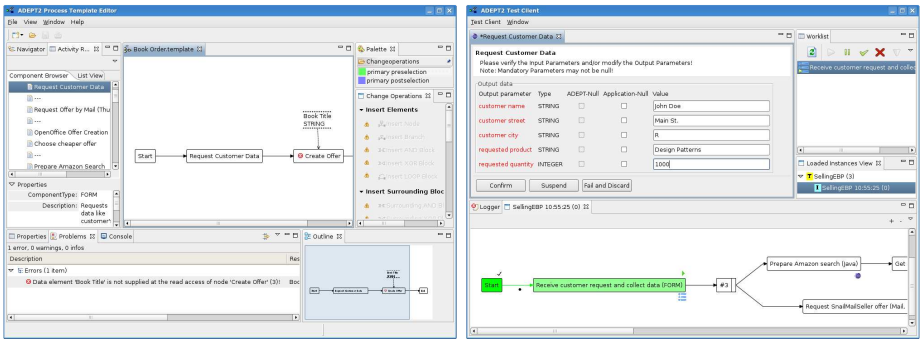

Fig. 2. Process Template Editor and Test Client of ADEPT2

– modelling process templates, correctness guarantees by construction
– on the fly checks of the data flow
– fast yet reliable application integration in plug-and-play fashion
– quick control flow modelling using activity templates
– creation of staff assignment rules

The ADEPT2 Test Client is a fully-fledged test environment for process execution. Unlike commonly known simulation tools, it runs on a light-weighted instance of the PMS itself. As such, various execution modes between pure simulation to production mode are possible. Demonstration will include:

– execution of arbitrary process templates, including sub-processes
– call of Java, WebServices, OpenOffice, and binary executables
– integration of software components with graphical user interfaces
– performing ad-hoc modifications

Screenshots of the applications can be found at `http://www.informatik.uni-ulm.de/dbis/bpm2007/`.

References

1. Reichert, M., Dadam, P.: ADEPT$_{flex}$ – Supporting Dynamic Changes of Workflows Without Losing Control. JIIS, Kluwer Academic Publ. Vol. 10, No. 2 (1998) 93–129
2. Dadam, P., Reichert, M., Kuhn, K.: Clinical workflows – The killer application for process-oriented information systems? In: Proc. BIS '00, Poland. (2000) 36–59
3. Bauer, T., Dadam, P.: Efficient distributed workflow management based on variable server assignments. In: Proc. CAiSE '00, Stockholm, Sweden. (2000) 94–109
4. Rinderle, S., Reichert, M., Dadam, P.: Flexible support of team processes by adaptive workflow systems. Distributed and Parallel Databases, Vol. 16, No. 1 (2004)
5. Acker, H. et al.: Aspects of Component-oriented Development of Adaptive Process-orientierted Enterprise Software. In: Proc. AKA '04, Augsburg, GI Lecture Notes in Informatics, P-57, Tagungsband 1. (2004) (in German).
6. Ly, L.T., Rinderle, S., Dadam, P.: Semantic correctness in adaptive process management systems. In: Proc. BPM 2006, Vienna, Austria. LNCS 4102 (2006) 193–208