Chapter VIII
Enabling Adaptive Process-Aware Information Systems with ADEPT2

Manfred Reichert
University of Ulm, Germany

Peter Dadam
University of Ulm, Germany

ABSTRACT

In dynamic environments it must be possible to quickly implement new business processes, to enable ad-hoc deviations from the defined business processes on-demand (e.g., by dynamically adding, deleting or moving process activities), and to support dynamic process evolution (i.e., to propagate process schema changes to already running process instances). These fundamental requirements must be met without affecting process consistency and robustness of the process-aware information system. In this chapter the authors describe how these challenges have been addressed in the ADEPT2 process management system. Their overall vision is to provide a next generation technology for the support of dynamic processes, which enables full process lifecycle management and which can be applied to a variety of application domains.

INTRODUCTION

In today's dynamic business world the economic success of an enterprise increasingly depends on its ability to quickly and flexibly react to changes in its environment. Generally, the reasons for such changes can be manifold. As examples consider the introduction of new regulations, the availability of new medical tests, or changes in customers' attitudes. Companies and organizations therefore have recognized business agility as prerequisite for being able to cope with changes and to deal with emerging trends like business-on-demand, high product and service variability, and faster time-to-market (Weber, Rinderle, & Reichert, 2007).

Copyright © 2009, IGI Global, distributing in print or electronic forms without written permission of IGI Global is prohibited.

Process-aware information systems (PAISs) offer promising perspectives in this respect, and a growing interest in aligning information systems in a process-oriented way can be observed (Weske, 2007). As opposed to data- or function-centered information systems, PAISs separate process logic and application code. Most PAISs describe process logic explicitly in terms of a *process template* providing the schema for handling respective *business cases*. Usually, the core of the *process layer* is built by a process management system which provides generic functions for modeling, configuring, executing, and monitoring business processes. This separation of concerns increases maintainability and reduces cost of change (Mutschler, Weber, & Reichert, 2008a). Changes to one layer often can be performed without affecting other layers; e.g., changing the execution order of process activities or adding new activities to a process template can, to a large degree, be accomplished without touching the application services linked to the different process activities (Dadam, Reichert, & Kuhn, 2000). Usually, the process logic is expressed in terms of executable *process models*, which can be checked for the absence of errors already at buildtime (e.g., to exclude deadlocks or incomplete data flow specifications). Examples for PAIS-enabling technologies include workflow management systems (van der Aalst & van Hee, 2002) and case handling tools (van der Aalst, Weske, & Grünbauer, 2005; Weske, 2007).

The ability to effectively deal with process change has been identified as one of the most fundamental success factors for PAISs (Reichert & Dadam, 1997; Müller, Greiner, & Rahm, 2004; Pesic, Schonenberg, Sidorova, & van der Aalst, 2007). In domains like healthcare (Lenz & Reichert, 2007; Dadam et al., 2000) or automotive engineering (Mutschler, Bumiller, & Reichert, 2006; Müller, Herbst, Hammori, & Reichert, 2006), for example, any PAIS would not be accepted by users if rigidity came with it. Through the described separation of concerns PAISs facilitate changes. However, enterprises running PAISs are still reluctant to adapt process implementations once they are running properly (Reijers & van der Aalst, 2005; Mutschler, Reichert, & Bumiller, 2008b). High complexity and high cost of change are mentioned as major reasons for not fully leveraging the potential of PAISs. To overcome this unsatisfactory situation more flexible PAISs are needed enabling companies to capture real-world processes adequately without leading to mismatches between computerized business processes and those running in reality (Lenz & Reichert, 2007; Reichert, Hensinger, & Dadam, 1998b). Instead, users must be able to deviate from the predefined processes if required and to evolve PAIS implementations over time. Such changes must be possible at a high level of abstraction and without affecting consistency and robustness of the PAIS.

Changes can take place at both the *process type* and the *process instance* level. Changes of single process instances, for example, become necessary to deal with exceptional situations (Reichert & Dadam, 1998a; Minor, Schmalen, Koldehoff, & Bergmann, 2007). Thus they often have to be accomplished in an ad-hoc manner. Such *ad-hoc changes* must not affect PAIS robustness or lead to errors; i.e., none of the execution guarantees ensured by formal checks at buildtime must be violated due to dynamic process changes. *Process type changes*, in turn, are continuously applied to adapt the PAIS to evolving business processes (Casati, Ceri, Pernici, & Pozzi, 1998; Rinderle, Reichert, & Dadam, 2004b; Pesic et al., 2007). Regarding long-running processes, *evolving process schemes* also require the migration of already running process instances to the new schema version. Important challenges emerging in this context are to perform *instance migrations* on-the-fly, to guarantee *compliance* of migrated instances with the new schema version, and to avoid performance penalties (Rinderle, Reichert, & Dadam, 2004a).

Off-the-shelf process management systems like *Staffware*, *WebSphere Process Server* and

29 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/enabling-adaptive-process-aware-information/19692

Related Content

A Link-Node Nonlinear Complementarity Model for a Multiclass Simultaneous Transportation Dynamic User Equilibria
Mohamad K. Hasanand Xuegang Ban (2013). *International Journal of Operations Research and Information Systems (pp. 27-48).*
www.irma-international.org/article/link-node-nonlinear-complementarity-model/76671

Strategies for the Near Term
(2019). *Strategic Management of Business-Critical Information Assets (pp. 92-105).*
www.irma-international.org/chapter/strategies-for-the-near-term/225446

Dual-Level Evaluation Framework for Airport User Satisfaction
Dimitrios Dimitriou, Maria Sartzetakiand Iliana Kalenteridou (2021). *International Journal of Operations Research and Information Systems (pp. 17-30).*
www.irma-international.org/article/dual-level-evaluation-framework-for-airport-user-satisfaction/268351

Modeling the Skunkworks Technology: Sharing Experience to Make the Difference
Sylvia C. Mupepi, Kuda B. Mupepiand Tatenda Mupepi (2019). *Strategic Collaborative Innovations in Organizational Systems (pp. 80-99).*
www.irma-international.org/chapter/modeling-the-skunkworks-technology/218702

Modeling Adoption of a Successive Generational Product in Multiple Markets
Adarsh Anand, Gunjan Bansaland Mohini Agarwal (2018). *Handbook of Research on Promoting Business Process Improvement Through Inventory Control Techniques (pp. 427-441).*
www.irma-international.org/chapter/modeling-adoption-of-a-successive-generational-product-in-multiple-markets/198702