Agile Workflow Technology and Case-Based Change Reuse for Long-Term Processes

Mirjam Minor, University of Trier, Germany
Alexander Tartakovski, University of Trier, Germany
Daniel Schmalen, University of Trier, Germany
Ralph Bergmann, University of Trier, Germany

ABSTRACT

The increasing dynamics of today’s work impacts the business processes. Agile workflow technology is a means for the automation of adaptable processes. However, the modification of workflows is a difficult task that is performed by human experts. This article discusses the novel approach of agile workflow technology for dynamic, long-term scenarios and on change reuse. First, it introduces new concepts for a workflow modeling language and enactment service, which enable an interlocked modeling and execution of workflows by means of a sophisticated suspension mechanism. Second, it provides new process-oriented methods of case-based reasoning in order to support the reuse of change experience. The results from an experimental evaluation in a real-world scenario highlight the usefulness and the practical impact of this work.

Keywords: adaptation of workflows; agile workflow; case-based adaptation; workflow modeling language; workflow similarity measure

INTRODUCTION

“We could not employ a workflow system that is not adaptable to changes.” states a chip design expert from Silicon Image GmbH, Hannover (S. Rackow, personal interview, October 25, 2006). Chip designers are used to dealing with the dynamics that result from the evolution of technology as well as from changes in the market. The increasing dynamics of the workflow is a phenomenon that affects the production processes within the high-tech industry: Software developers have to be flexible when the customer requirements change. Health care professionals must react to side-effects and to other complications during the treatment of patients. What these examples from various domains have in common is that they cause major deviations from the usual business pro-
cesses at run time. Furthermore, the ongoing processes need refinement after several weeks or months of running. Workflow technology supports business processes (Workflow Management Coalition, 1999). However, traditional workflow management systems, those described in Leymann and Roller (2000), are not able to deal with adaptable processes. Consequently, there is a need for agile workflow technology (Weber & Wild, 2005), that is, a workflow technology that allows the late-modeling and structural adaptation of ongoing workflows. Change reuse is essential to these difficult modeling tasks.

Workflows are “the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules” (Workflow Management Coalition, 1999). A workflow management system “defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications” (Workflow Management Coalition, 1999). A workflow enactment service is “a software service that may consist of one or more workflow engines” (Workflow Management Coalition, 1999).

All agile workflow approaches require the application of changes concurrently to normal process execution. In contrast to the existing approaches, very large process graphs are dealt with. They require a suspension mechanism in case of delayed decisions as they do occur in practise. Due to the highly dynamic environments of the workflows, the approach is aware of the context of influence factors. As other agile approaches, reusing experience from the changes of workflows in the past for the adaptation of an ongoing workflow are aimed at. In contrast to existing change reuse approaches, the retrieval of past workflows should consider the workflows’ structure directly rather than solely operating on additional information. In this article, a new approach of agile workflow technology which includes new basic methods for the representation, execution, and partial suspension of adaptable workflows as well as more sophisticated methods for the support of change reuse by means of case-based reasoning is described. The following hypothesis that will be investigated empirically in this article is stated: The evaluation of structural differences of workflows provides a good approximation for the usability of past modifications for current change requests. The purpose of the work is to contribute new workflow concepts and methods to solving the practical but difficult problem of handling business processes in very dynamic, long-term scenarios.

LITERATURE REVIEW

The workflows to be interpreted in conventional workflow management systems have a fixed structure. In the literature, a number of approaches exist that aim at making workflows more flexible. Some work on flexible workflows in the broader sense are first discussed and then the literature on agile workflows in the actual sense are briefly reviewed. The latter is closely related to the work. The literature on similarity assessment of workflows is also discussed and the position on context-awareness during change reuse is defined.

Some recent approaches (Luo, Sheth, Kochut, & Arpinar, 2003; Hwang & Tang, 2004; Russell, van der Aalst, & ter Hofstede, 2006; Adams, ter Hofstede, Edmond, & van der Aalst, 2006) deal with exception handling for “deviations from normal execution arising during a business process” (Russell et al., 2006). Luo et al. (2003) use exception-handling processes for the resolution of conflicts that “primarily arise due to failure of a task in workflow execution” (Luo et al., 2003). For instance, a time out exception may occur for a task and can be handled by a retry, recovery, or compensation process. Russell et al. (2006) and Adams et al. (2006) associate this task-oriented exception handling with a case handling approach (van der Aalst, Weske, & Grünbauer, 2005) that focuses on the whole process and its objectives. This means that the exception handling processes
Stream Processing of a Neural Classifier I
www.igi-global.com/chapter/stream-processing-neural-classifier/10435?camid=4v1a