Available online at www.sciencedirect.com
ScienceDirect
Decision Support Systems 42 (2006) 844–858
Decision Support Systems
www.elsevier.com/locate/dsw

Merging workflows: A new perspective on connecting business processes

Shuang Sun [a,*], Akhil Kumar [b], John Yen [a]

[a] *School of Information Sciences and Technology, Pennsylvania State University, University Park, PA 16802, United States*
[b] *SMEAL College of Business, Pennsylvania State University, University Park, PA 16802, United States*

Received 18 May 2004; received in revised form 17 June 2005; accepted 3 July 2005
Available online 1 August 2005

Abstract

This paper describes the concept of workflow merge and methods for merging business processes. We grouped merges in four categories according to the type of merge: *sequential*, *parallel*, *conditional*, and *iterative*, and describe the corresponding algorithms for performing these operations. We give results that allow us to determine whether a merge operation is sound. It is shown that to avoid invalid merges, one should choose merge points between which a sub-workflow, called a merge region, is well structured. These findings can provide useful guidance for future workflow merge research. We also raise issues of more complex merge problems, such as merge conflicts, semantic ambiguities and workflow splits.

Keywords: Workflow management; Workflow merge; Workflow modeling; Business process reengineering; Sequential; Parallel; Conditional; Iterative merge

1. Introduction

For agile business operation, modern corporations must make frequent business process changes as well as organizational changes through mergers and acquisitions. In 2001, Hewlett-Packard Company and Compaq Computer Corporation announced a merger agreement to create an 87 billion dollar global technology leader. The merged company offers the most complete set of products and services in the IT industry with expected cost savings of approximately 2.5 billion dollar a year [10]. Many important issues arise in integrating the two giant organizations, one of them being how to integrate their business processes. Since frequent changes in business processes and operations are becoming increasingly common, both through internal reorganizations and through mergers and acquisitions, we have conducted preliminary research on how workflows can be modified dynamically to adapt to such changes. In addition, our research provides support for complex process composition, i.e., creating complex workflow processes from simple ones.

A workflow may be modified at a *schema* level that defines a workflow process or at an *instance* level

* Corresponding author.
E-mail address: ssun@ist.psu.edu (S. Sun).

0167-9236/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2005.07.001

that represents a *specific instance* of an already defined process [12]. For example, a two-step workflow, "place order," then "deliver," represents a simple business process at the schema level. Within this process there may be orders or instances of a process "order" for customer "Sue". In this paper, we define the workflow concept and methods focused at the schema level. This research topic is important and has not been fully addressed by previous research efforts. Process reengineering and evolution have been studied from different perspectives such as: business process integration [5,17]; using generic workflow modeling methods to ensure flexibility [2]; describing methods for workflow evolution [12]; improving workflow interoperability [3,9,11,15]; and enhancing exception handling capabilities [14,19]. Dealing with more than one process makes a workflow merge different from other problems that commonly assume a single process, and makes existing methods inefficient for addressing workflow merging issues. Therefore, this topic does not fit well into the existing research frameworks. For example, the classic process integration methods [5,17] collaboratively bridge, adapt, and exchange information without actually modifying the processes of the business partners. Naturally, when companies merge, both process integration and merging of processes are necessary for streamlining their operations.

In the remainder of the paper, we first introduce a workflow modeling method with Petri nets. Next, in Section 3, we introduce the workflow merge concept and validate our approach. Section 4 introduces notions of sound and unsound merges, and gives two results related to soundness. Then, in Section 5, we discuss merge point detection and other issues such as conflicts, semantic ambiguities, and impact of merges on organizational roles and resources. Finally, Section 6 gives brief concluding remarks.

2. Workflow modeling

Many research efforts have investigated methods for modeling workflow processes, which define the steps of business operations. Dumas and Hofstede tried to specify workflows with activity diagrams of the Unified Modeling Language (UML) [6]. They demonstrated that activity diagrams can provide the expressive power that is required by most applications, and showed that an activity diagram is more powerful for express processes than most of the languages found in commercial workflow systems. A recent study by Aalst and Kumar [3] has demonstrated that the Extensible Markup Language (XML) can be used to model inter-organizational workflows. The main contribution of that research is to support process exchange through the internet. Aalst [1] has mapped the workflow concepts into *Petri nets*, giving a more formal way to represent and verify processes. In this paper, we also use *Petri nets* to specify workflows and related concepts. Dussart et al compared several workflow modeling methods such as Petri nets, WfMC, UML, ANSI, and EPC [7] on criteria such as formal basis, executability, ease of visualization, etc. Their study showed that Petri nets satisfied most of the criteria, and were therefore desirable. However, the merge concept and algorithms are independent of modeling techniques, and not limited to Petri nets. We choose Petri nets mainly because they offer a formal basis that helps to determine soundness of a merge. In this section, we briefly introduce important Petri net concepts and how to use Petri nets to represent a workflow.

Definition 1 (*Petri net*). A Petri net is a triple (P, T, F):

- P is a finite set of places,
- T is a finite set of transitions $(P \cap T = \phi)$
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (flow relation)

A place p is called an *input place* of a transition t if and only if there exists a directed arc from p to t. Place p is called an *output place* of transition t if and only if there exists a directed arc from t to p. We use $\bullet t$ ($t\bullet$) to denote the set of input (output) places for a transition t, while $\bullet p$ ($p\bullet$) is the set of transitions sharing p as an input (output) place. Note that we restrict ourselves to arcs with weight 1. In the context of workflow procedures it does not make much sense to have other weights because places correspond to conditions.

Definition 2 (*WF-net [1]*). A Petri net PN $= (P,T,F)$ is a WF-net (WorkFlow net, or workflow for short) if and only if

(i) PN has two special places: i and o. Place i is a source place: $\bullet i = \phi$; place o is a sink place: $o\bullet = \phi$.