
ELSEVIER

Data & Knowledge Engineering 34 (2000) 271–304

DATA & KNOWLEDGE ENGINEERING

www.elsevier.com/locate/datak

A reflective infrastructure for workflow adaptability

David Edmond *, Arthur H.M. ter Hofstede

Cooperative Information Systems Research Centre, Queensland University of Technology, GPO Box 2434, Brisbane, Qld 4001, Australia

Received 4 May 1999; accepted 17 January 2000

Abstract

We present a flexible framework that enables workflow systems to adapt to changing conditions. The model is designed to reveal key aspects of the tasks involved in representing and enacting business processes. These fundamental characteristics are identified as state, behaviour, distribution, coordination and enactment. By isolating such core concepts in a way that allows them to be varied, we open up the general process of task coordination and execution, allowing for extensions in a *planned* way. By suitable manipulation of each of these aspects, at the appropriate level, a workflow system may be *extensively* modified in a way that minimises the effect of such change upon other aspects of the system.

Keywords: Workflow; Reflection; Adaptation

1. Introduction

Remaining competitive is almost a *raison d'être* for most organisations. They are involved in a restless and unceasing struggle to attain and retain competitiveness. In the case of a manufacturing company, for example, this may be carried out in a number of ways [21]:

- Production techniques can be restructured, leading to an increase in output or in capacity.
- Product characteristics – such as quality, design, delivery time and outlet distribution – may be adjusted.
- Product demand may be stimulated through advertising campaigns.
- Innovation in both the production and the design processes may arise from research carried out by the company.

All of these may be instrumental in enabling a firm to compete in a market: whatever the particular nature of an organisation, it will have, at its disposal, a variety of tools and techniques to be used in whatever tactics are employed to maintain market share and relevance. A particular set of tactics may be realised in the form of a *business process* (BP), which is defined as a group of tasks, the performance of which results in something of value to a customer [11].

* Corresponding author.
E-mail addresses: davee@icis.qut.edu.au (D. Edmond), arthur@icis.qut.edu.au (A.H.M. ter Hofstede).

0169-023X/00/$ - see front matter
PII: S0169-023X(00)00018-5

Belief in the importance of business processes has resulted in a great deal of attention being paid to the *workflow* systems that automate them. Business Process Reengineering (BPR), in particular, advocates the use of information technology as the means of driving the process, and thus achieving goals more quickly. This promotion of IT has led to a great deal of research into the concepts of workflows and workflow management. A *workflow* is the automation of a *business process*, with documents, information or tasks being passed from one participant to another for action, according to a set of procedural rules. Workflow (WF) technology is designed for building process-centred application systems. A *workflow management system* (WFMS) is a system that defines, creates and manages the execution of workflows by means of software which interprets the process definition, interacts with workflow participants, and invokes IT tools and applications [22]. Yet, as previously discussed, in many application domains, business processes are highly volatile. To meet the constraints and opportunities posed by new technology, new markets, and new laws, businesses must constantly refine their processes: unplanned deviations are the norm [7,19]. Organisations need to be able to adjust their business processes, and quickly adapt their software systems to match. Rigid process models do not allow for creativity because they curb flexibility [18]. The consistent and effective *evolution* of workflows is a basic step towards the adaptability that will be demanded of future workflow management systems [7]. Evolution in workflows gives rise to two related problems: sometimes, because of organisational and functional adaptations of the BP *model* itself, the corresponding workflow *schema* requires amending; but sometimes also, because of unplanned events or exceptional circumstances, individual BPs may require changes and dynamic extensions [12]. The next generation of workflow systems must provide primitives that allow the incremental modification of a workflow, without requiring that it be entirely rewritten; and they must provide mechanisms to handle running instances of a workflow schema that is undergoing modification [7]. However, changes must be controlled, and restrictions placed on change operators. Such restrictions must be based on a WF model that has a proper theoretical basis. Generally, it must be possible to add new tasks to a WF at any point of time during its execution, to work on an inserted task concurrently to other tasks, to synchronise the execution of an inserted task with those of other tasks, to insert tasks into WF regions that have not yet been entered, to dynamically map the parameters of the inserted task to existing or to newly generated data elements, and so on [12].

Ideally, process-centred applications should reflect changes of the BPs they support without delay. Presently, however, only if the BP to be supported is well-structured, may WFMSs be used reliably. Current WF technology handles well-defined sets of tasks, ones with fairly fixed execution sequences. It only provides rudimentary support for dynamic structural changes and dynamic extensions of WFs. This significantly limits its applicability [19]. Existing workflow systems have made various attempts to provide adaptability:

- Fujitsu's *TeamWARE Dolphin* and *TeamWARE Flow* both allow processes to be changed, even while they are in progress, allowing business processes to undergo *continuous* improvement.
- InConcert's *InConcert 2000* allows changes to an active process.
- *Verve Workflow* supports schema modification. The product distinguishes between the process *plan*, which is the model, and the *actual* process, which is an enactment of that plan. The plan can be re-fashioned without limit, thus modification of each individual instance is possible.