



ELSEVIER

Decision Support Systems 35 (2003) 517–536

Decision Support
Systems

www.elsevier.com/locate/dsw

Component-based workflow systems development

Hai Zhuge*

*Knowledge Grid Group, Key Lab of Intelligent Information Processing, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, 100080, PR China*

Accepted 28 May 2002

Abstract

Component-based development is a promising way to promote the productivity of large workflow systems development. This paper proposes a component-based workflow systems development approach by investigating the following notions, mechanisms, and methods: workflow component, workflow component composition, reuse–association relationship between workflow components, and workflow component repository. The proposed approach is supported by a set of development strategies and a development platform. Through application and comparison, we show the advantages of the component-based workflow systems and the effectiveness of the proposed approach.

© 2002 Elsevier Science B.V. All rights reserved.

Keywords: Component; Component repository; Component reuse; Workflow

1. Introduction

A workflow is the computerised facilitation or automation of a business process, in whole or part [10,23]. A workflow can be abstracted as a network with task (i.e., activity) nodes and flows (i.e., transitions between task nodes). Domain business processes can be modelled through the execution of the network, and thus can be controlled and managed by incorporating the domain business application into the execution process of the network. The benefits of workflow-based applications, like flexibility, integration, and reusability, were discussed in Ref. [11].

Research works on workflow mainly concern the following aspects: (1) the extensions of workflow

models [30,37], e.g., topics on incorporating more application-relevant factors such as time restriction, resources scheduling, and cost-effective evaluation into the existing workflow model; (2) the establishment and updating of the workflow model standard [23–30], so as to build a common language and to improve the interoperability and the interchange between workflow products; (3) semantic research [1,6,8,21], e.g., topics on checking the deadlock and the loop of workflow execution, workflow equivalence, workflow optimisation, etc.; and (4) the workflow application system development [5,11,12] and application-relevant research, e.g., topics on transactional workflow, agent-based workflow, reflexive workflow, workflow interoperation, and enhancing the quality and productivity of large workflow systems development. The research scope of this paper is on the fourth aspect and focuses on the reuse-based workflow development methodology.

* Fax: +86-1062567724.

E-mail address: zhuge@ict.ac.cn (H. Zhuge).

A good workflow is not only a simple image of domain business process but also the abstraction of it. The current workflow system development approaches lack a built-in development methodology. The current WfMC standard only provides a workflow definition language [23], but it does not provide business analysis tools and the development methodology. To employ software development methodologies like OOM [18] (including UML [17]) for domain business analysis is a way to complement the lack of workflow development methodology [30]. OOM supports inheritance mechanism, but the class-based inheritance needs to be adjusted so as to meet the needs of the workflow process reuse. On the other hand, workflow system development is an experience-dependent process. To realise effective experience reuse, we need to use the abstraction and analogy method. Unfortunately, the existing workflow development approaches do not support abstraction and analogy. Each new workflow system development has to carry out from scratch and follows the stringent development steps even though the developers can fully understand the domain business and have the development experience. The efficiency and quality of workflow development vary with different developers' development skills and experiences. It is still troublesome to dealing with the correctness and the complexity in composing different workflows designed by different designers even though they are in a cooperative development team and work for the same project.

Component technology is a way to raise the efficiency and quality of system development. It is an interesting topic to apply the software component concept and method to workflow development so as to promote its development efficiency and quality. On one hand, the component-based development encourages the developers to make domain business analysis by referring to the analysis experience of similar domain business. On the other hand, the component-based reuse provides a possibility of reusing the existing workflow components to compose a new workflow. The motivation of this paper is to investigate the concept and mechanism of workflow components and to propose an approach that supports the component-based workflow development.

To realise the motivation, three main problems need to be solved: (1) the definition of workflow

component, an ideal workflow component should have an interface for encapsulating the workflow process and be able to be executed and used through the interface; (2) the approach for composing workflow components; and, (3) the establishment of an effective reuse mechanism, a set of reuse strategies, and the reuse-based workflow development process.

The semantic of event-driven workflow execution based on reactive components was investigated in Ref. [6], where the reactive component refers to the task (activity) entity that takes part in the workflow, and a broker mechanism is used to represent the component. The issue of the modular and the shared transactional workflow was discussed in Ref. [15]. However, the current research has neither given a complete definition of workflow component that has the advantages of a software component, nor a complete approach for component-based workflow systems development. The commercial workflow management products do not support the component-based workflow development.

A workflow system usually consists of two parts. First, a set of data definitions about: the workflow control data, the role model data, the work list, and the application invocation definition data. Second, a set of software components including: a process definition component, an enactment component (Engine), a work list handler, the user interface components, and the application components. Five kinds of interfaces between these components are defined by WfMC [23]. We define the workflow component by considering both the conceptual level and the system level. The conceptual level is separated from the system level by establishing an execution interface, which consists of the following four parts (subinterfaces): (1) an access interface, the input flow and the output flow of the workflow component by viewing the workflow component as a single task; (2) the process (control) interface; (3) the role interface; and (4) the restrictions.

To meet the needs of diverse application domains like e-commerce, we enable the flows of the discussed workflow model to reflect not only the execution dependence (i.e., control flow), but also the data dependence (i.e., data flow) and the temporal order relationship between tasks. Task execution can only be triggered by the condition satisfaction. To simplify the discussion of workflow component composition,

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات