

# Incorporate Components into Workflow Application Systems\*

Wenhu Tian, Ge Yu, Guoren Wang and Baoyan Song  
Dept. of Computer Science and Engineering  
Northeastern University, Shenyang, China  
{yuge, wanggr}@mail.neu.edu.cn

## Abstract

*In the workflow domain, many researches have been focused on such aspects as workflow system architecture, workflow task scheduling, transactional workflow, etc. Few have been concentrated on the workflow application model, design and implementation. In this paper, we firstly discuss the problems related to the individual workflow application in current workflow management systems, then show a component-based method to resolve these problems.*

**Keywords:** Workflow, Component, CORBA

## 1. Introduction

Workflow is a business process consisting of a serial of activities aimed at the same goal. Workflow management systems (WFMSs) provide an automated framework for managing intra- and inter-enterprise business processes [1].

Current workflow systems usually only allow modeling and executing business processes, but have little support on the modeling aspect of workflow applications. Generally, a workflow developer does not care much about the details of the application programs used in workflow activities. He/She thinks the applications are pre-existing and ready for being integrated into the workflow system. Furthermore, most of these applications are obtained by enwrapping the existed legacy applications. Such an application, if used in a workflow schema, might result in a fixed process lacking flexibility and adaptability. When a revision of an application is demanded, it is more difficult to carry out this job because the change requires that the user

work on the source code level, but not on a more abstract design level like workflow process design. Even worse, the workflow developer does not have enough programming skill to make this revision because it requires too much programming skill.

The concept of component comes from software industry. Components are reusable software units that can be assembled or disassembled in an application without violating its integrity. The term component-based application refers to the application which is built out of components and which is adopted to distinguish itself from those traditional application programs [2]. By use of components, application developers will feel easy when creating or modifying component-based applications. In design time, more consideration about flexibility and adaptability can be taken into the component-based development.

A natural idea to resolve the problems resulting from workflow applications is to use component-based programming for the development of workflow applications. Because of the easiness and promptness by using components to build a workflow application, developing a new workflow application becomes simple and transparent. In this paper, we firstly talk about the problems resulting from traditional workflow applications, and then present a solution to these problems by using a component-based development method.

This paper is organized as follows: major problems resulting from workflow applications are discussed in section 2. Subsequently in section 3, a component-based way for building workflow applications is proposed to resolve these problems. In section 4, a language for

---

\* This work is supported by 863/CIMS, grant No: 863-511- 9841-010

describing both workflow and component-based applications is introduced. At last, Section 5 gives the conclusion.

## 2. Problems Related to Workflow Applications

Today, most of the workflow applications are not originally designed for the purpose of workflow. So when they are used in a workflow domain, problems will emerge.

- **Insufficiently adaptable for workflow applications**

Workflow environment is a capricious environment, which requires the task applications have enough ability to adapt to such a change. Paradoxically, most of the task applications in today's workflow systems are built on existing applications that do not have enough knowledge about the environmental changes. So most of the adaptability have to be put on the workflow level, however, almost few workflow systems have considered to provide the adaptability in the workflow applications.

- **Expensive to build a new workflow application**

For a business process, building a new workflow sometimes is indispensable. Few of current workflow tools provide support to design such a new workflow application. So the business process expert who generally has little programming skill must resort to the software vendor to customize the business application. This procedure is surely expensive and time-consuming.

- **Insufficiently flexible in workflow applications**

Dynamically changing workflow allows rapid reactions to the environmental changes, which is important for an organization to improve its market competitiveness. In current workflow systems, support for configuring or dynamically changing workflow application is not available yet because most of these applications are monolithic and can not be divided into parts. Therefore, such applications can not be applied in a situation where load balance is required.

- **Platform-dependent applications**

Generally speaking, workflow systems are used in a very heterogeneous environment. Although workflow specification has nothing with the operation platforms, the running workflow applications are often dependent on the platforms such as O.S., programming languages. Hence, the workflow application must be realized each on a different platform. For a workflow developer, this

requirement may be too critical, too costly.

- **Separation between design of workflow applications and of workflow**

Ordinary users of workflow systems can only define and modify the workflow-level specification. That is, when building a new workflow application, few of current workflow systems have the ability to implement the workflow applications. Hence, the procedures of designing workflow and designing workflow applications are separated.

## 2. Using Components to Build Workflow Applications

In this section, we show how to build workflow applications using the component technology in order to resolve the problems given in section 2.

- **Incorporate adaptability into workflow applications**

The process for building workflow applications by use of components can be seen as a mini-workflow. By identifying the relationship among the composite components, we can incorporate much experience useful in designing workflow into the workflow applications. Therefore, the adaptability of the workflow is proportionally maximized.

- **Easy to customize a new workflow application**

When developing a new task application is needed, it will be much easier for a workflow developer to implement such an application since there already exist many off-the-shelf components. Building a component-based application is just like piling up the toy bricks. By specifying the necessary components and identifying the control flow among these composite components, even a business specialist who do not have much programming skill can create a desired workflow application. So he does not have to implement them and the new task application can be constructed rapidly.

- **More flexible in workflow applications**

Through the separation of business logic from implementation logic, component-ware provides good flexibility to upgrade because the co-relativity between the change of implementation logic and of business logic is relatively low in actual applications[3]. So in a component-based workflow system, the user can add, replace, remove a member component or modify the control flow among those components.

- **Platform-independent workflow applications**

The problem of platform dependency can be overcome if the components for building workflow applications are implemented in a platform-independent programming language such as Java. Fortunately, Sun Inc. provides two Java-based component models, which are JavaBeans [4] and EJB [5], one is for client-side and another is for server-side. What's more, even heterogeneous components can inter-operate very well through distributed object technology, such as OMG/CORBA. All platform-dependent details are hidden in the workflow system infrastructure, the workflow developers do not need to care about them.

- **Integrating design of workflow applications with workflow**

According to the characteristics of component systems, we regard a component system as a simplified and specialized mini workflow system within the activities. So we can use the method in workflow systems to define component applications. On the other hand, the pre-existing component applications can be used in workflow too. The workflow system is responsible for the process-oriented part of the business process and coordinates the sequence of activities in a distributed, heterogeneous environment. The component-oriented system is responsible for the function-oriented part inside an activity and coordinates relationship among the composite components.

#### 4. Managing Component-based Workflows Using ScopeWork

In order to use components in workflow applications, a component modeling language is desired. If it can be combined with workflow modeling language in a homogeneous style, a great deal of burden can be reduced for the workflow developers.

Up to now, different models are used in the design for the modeling of workflow and for the modeling of component-oriented applications. But few of them can be used in the unified way. Therefore, we propose a language named SWDL[6], which is used in ScopeWork [7] workflow system. Fig.1 is the architecture of ScopeWork system. This language is extended from IDL. It can describe not only the workflow processes, but also the architecture of components and the relationship among them as well. SWDL adopts IDL to specify the attributes and behavior of components. Dependencies

between component objects are expressed by SWDL's extensions.

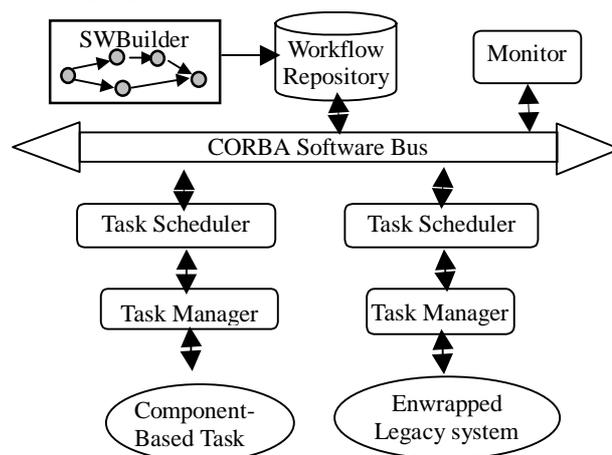


Figure 1 ScopeWork Architecture

#### 5. Conclusion

In this paper, some problems in current workflow systems are discussed, and then a way for resolving these problems is proposed. By building configurable, component-based applications, the way of workflow specification can also be applied into the workflow application level. Doing like this makes the creation or modification of a component-based application easier, even for a user without much programming skill.

#### References

- [1] D.Hollinsworth, The Workflow Reference Model, Technical report TC00-1003, Workflow Management Coalition, Dec. 1994.
- [2] David Garlan, Mary Shaw, Software Architecture: Perspectives on an Emerging Discipline, Prentice-Hall International Inc., 1996.
- [3] R. Schmide, Component Architecture for Business Application Systems, The 1998 International Workshop on Component-Based Software Engineering, available from <http://www.sei.cmu.edu/activities/cbs/icse98/papers/>
- [4] Javasoft: Java Beans Specification 1.0 A, <http://www.javasoft.com/>.
- [5] Javasoft, Enterprise JavaBeans Specification 1.1, <http://www.javasoft.com/>.
- [6] Y. Li, C. Zeng, B. Song, et al., Design of ScopeWork's Workflow Definition Language-SWDL, Journal of Lanzhou University, Vol.35, 1999.
- [7] W. Tian, G. Yu, B. Song, et al., Design of a CORBA-Based Workflow Management for Advanced Applications, Proc. of ICYCS'99, Nanjing, Aug. 1999.