A Framework for Workflow Management Systems Based on Objects, Rules and Roles

Gerti Kappel, **Stefan Rausch-Schott**, **Werner Retschitzegger**
Department of Information Systems, *University of Linz*
Altenbergerstraße 69, A-4040, Linz, Austria
Tel: ++43-732-2468-880
Fax: ++43-732-2468-9308
sec@ifs.uni-linz.ac.at

Abstract: The goal of this paper is to present an approach for the development of workflow management systems supporting both reusability and adaptability, i.e., customization due to frequently changing requirements in an organization. The principal contribution is the introduction of an object-oriented application framework for constructing such workflow management systems balancing between reusability and adaptability. The underlying techniques are an object-oriented workflow model, object evolution via an integrated role model, and the support of business policies via an integrated rule model.

Categories and Subject Descriptors: H.4.1 **[Office Automation]**: Workflow Management; D.2.11 **[Software Architectures]**: Domain-specific architectures

General Terms: Design

Additional Key Words and Phrases: Object-Oriented Frameworks, Event/Condition/Action Rule, Role Modeling, Context Dependent Behavior

INTRODUCTION

This paper examines the application of the framework idea to the development of workflow management systems. WorkFlow Management Systems (WFMS) have been introduced to support the design, execution and monitoring of generally long-lasting business processes. A business process, also called workflow, typically consists of various activities which have to be executed in some order involving multiple collaborating persons called agents in a distributed environment to fulfill a certain task in an organization [Jablonski and Bussler 1996]. Different workflows are based on recurring patterns such as similar organizational structures pointing to potential reusability.

At the same time, unpredictable situations and frequently changing requirements are prevalent in most organizations. In the following, we introduce the object-oriented application framework *TriGSflow* for constructing such WFMS balancing between reusability and adaptability [Demeyer et al. 1997]. To reach this goal, *TriGSflow* integrates three basic techniques [Kappel et al. 1995]. First, an object-oriented database system is used to build a generic workflow model providing both database functionality and possibilities for modeling, reusing, and customizing complex business domain objects. Second, to cope with changes in the personnel, a role model has been integrated into the object-oriented environment in order to decouple activities from particular persons. Third, Event/Condition/Action rules (ECA rules) are used to allow for a flexible coordination of activities as well as of resources needed to perform these activities.

All three techniques support both reusability by composition and adaptability by inheritance [Fayad and Schmidt 1997] (cf. Figure 1). The intrinsics of each model and their interplay are described in the following.

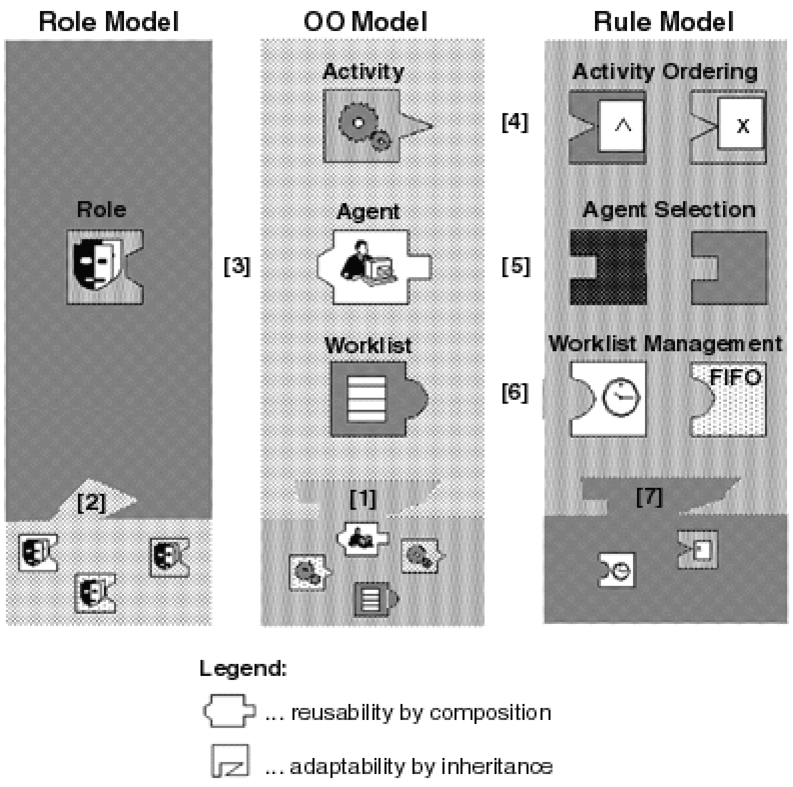

Figure 1. Interplay of Role Model, OO Model, and Rule Model.

THE OBJECT-ORIENTED MODEL

The basic model used in *TriGSflow* to represent a business process including *activities*, *agents* and *worklists* comprises a number of predefined abstract and concrete object classes. These classes constitute a generic workflow model in the sense that workflows for different kinds of application domains, such as application for scholarship or reordering of goods, are modeled by simply inheriting and customizing the corresponding predefined classes (cf. [1] in Figure 1).

THE ROLE MODEL

To cope with changing personnel, WFMS require the separation between activities and agents performing them. For this, the *role model*, which is based on the work of [Gottlob et al. 1996] , has been integrated into our framework.

Traditional class-based object-oriented systems are characterized by an immutable linkage between an object and its object class, i.e., an object always belongs to exactly one most specific class. In order to extend or change the behavior of an object a new instance of the corresponding class has to be created and the attribute values have to be transformed from the old to the new instance. The concept of roles eliminates these drawbacks by allowing object evolution during runtime. Objects can learn and forget roles dynamically, they can play several roles at the same time, and they even can play the same role simultaneously several times, as is required by real-world agents. In *TriGS$_{flow}$* an agent and its roles are represented as instances of several distinct object classes (cf. Figure 1). Whereas role classes may be specialized by inheritance (cf. [2] in Figure 1), an agent may acquire new roles by composition (cf. [3] in Figure 1).

THE RULE MODEL

In WFMS the existence of flexible mechanisms for realizing policies coordinating various activities as well as resources needed to perform these activities plays a central role. This need for flexible coordination mechanisms emerges from the fact that workflow applications are subject to frequent changes caused by the business environment [Ellis et al. 1995] . Flexibility and adaptability is required in different directions ranging from a dynamic evolution of existing coordination policies to proper reactions to unpredictable situations.
Active object-oriented database systems along with their basic mechanism of ECA rules seem to be a promising technology in order to cope with these requirements [Kappel et al. 1994] . ECA rules allow for an event-driven realization of context-dependent and time-dependent behavior by monitoring a situation represented by an event together with a condition and executing the corresponding action when the event occurs and the condition is true. By encapsulating coordination policies within ECA rules, organizational knowledge can be represented independently of specific business processes. ECA rules are captured within a class hierarchy and thus subject to inheritance (cf. [7] in Figure 1). This both eases customization of coordination policies and enhances their reusability. Furthermore, ECA rules in *TriGS$_{flow}$* are first class objects thus coordination policies may be adapted even on the fly. *TriGS$_{flow}$* uses ECA rules to realize coordination policies mainly in three different areas of the WFMS framework:

(1) Activity Ordering. Activities which are part of a certain workflow have to be coordinated with respect to their execution order by means of *activity ordering policies*. The basic activity ordering policies which are supported can be classified along two orthogonal dimensions, namely kind of control structure and kind of dependency. Whereas the former dimension comprises sequencing, branching and joining, the latter defines the starting points of the successor activities depending on the end of the predecessor activity or the start thereof. By combining these two dimensions different activity ordering policies emerge. Figure 1 exemplarily shows two of them (cf. AND icon and exOR icon). Furthermore, these basic activity ordering policies can be combined to form some higher-level coordination policies. For the basic activity ordering policies appropriate parameterized ECA rules are provided which can be simply applied to a particular workflow by plugging in the domain dependent parameter values, e.g., an appropriate instance of the class Activity being part of the object-oriented model (cf. [4] in Figure 1). Furthermore, if the order of activities is to be changed the corresponding rule can be changed even dynamically, i.e., during workflow execution.

(2) Agent Selection. For each activity one or more agents have to be selected by means of *agent selection policies*. An example of an agent selection policy is that agents being on vacation are not allowed to be selected. The decision which agent(s) to select for a specific activity is based on information concerning both the history of and the documents processed by the actual workflow. Moreover, the roles defining an agent's capabilities and responsibilities can be incorporated into this decision process, too (cf. [3] and [5] in Figure 1). In order to keep the agent selection process flexible with respect to changing requirements these policies again are encoded as ECA rules. In this way, immediate reactions to an agent's state changes are allowed. Additionally, different selection policies can be realized and switched between dynamically.

(3) Worklist Management. Similar to activity ordering policies coordinating the execution order of activities, *worklist management policies* coordinate the execution order of several activities inserted into the worklist of a single agent (cf. [6] in Figure 1). Worklist management policies are necessary since business processes and therefore also single activities are often subject to temporal constraints [Ellis et al. 1995] . In fact, our framework supports three basic kinds of worklist management policies that support these temporal constraints. First, as soon as a new activity arrives at a certain worklist, the execution order of existing activities including the new one is re-scheduled on the basis of the temporal constraints. Second, the actual start and end of temporally constrained activities is monitored and reacted to in case of violations. This is done, for example, by reminding the agent if some time has passed (cf. clock icon in Figure 1), delegating to another agent, or notifying the workflow administrator [Kappel et al. 1998] . And third, the start of an activity to be executed by some software agent is enforced according to its temporal constraints. Again, all these policies are mapped to ECA rules in order to gain similar advantages as described for activity ordering and agent selection.

CONCLUDING REMARKS

The main goal of developing the *TriGS*$_{flow}$ framework was to support adaptability, i.e., to be able to cope with frequently changing requirements in an organization, and to support reusability, i.e., to provide predefined workflow components, which can be flexibely composed together. To reach this goal, we have integrated the concepts of objects, roles and rules into a framework for WFMS. A first prototype of *TriGS*$_{flow}$ based on the object-oriented database system GemStone running under Sun Solaris OS is operational. We are currently extending the framework to integrate advanced transaction mechanisms embedding the activities which have to be performed [Kappel et al. 1996] [Sheth 1993] .

REFERENCES

[Demeyer et al. 1997]
Demeyer, S., Meijler, T.D., Nierstrasz, O., and Steyaert, P. 1997. Design Guidelines for "Tailorable" Frameworks. *Communications of the ACM, Object-Oriented Application Frameworks*, 40(10), 60-64.

[Ellis et al. 1995]
Ellis, C.A., Keddara, K., and Rozenberg, G. 1995. Dynamic Change Within Workflow Systems. In *ACM Conference on Organizational Computing Systems*, N. Comstock et al. (eds.), ACM Press, Milpitas, 10-21.

[Fayad and Schmidt 1997]
Fayad, S., and Schmidt, D.C. 1997. Object-Oriented Application Frameworks. *Communications of the ACM, Object-Oriented Application Frameworks*, 40(10), 32-38.

[Gottlob et al. 1996]
Gottlob, G., Schrefl, M., and Röck, B. 1996. Extending Object-Oriented Systems with Roles. *ACM Transactions on Information Systems*, 14(3), 268-296.

[Jablonski and Bussler 1996]
Jablonski, S., and Bussler, C. 1996. Workflow-Management: Modeling Concepts, Architecture and Implementation. *International Thomson Computer Press*, 1996.

[Kappel et al. 1994]
Kappel, G., Rausch-Schott, S., Retschitzegger, W. and Vieweg, S. 1994. TriGS - Making a Passive Object-Oriented Database System Active. *Journal of Object-Oriented Programming (JOOP)*, 7(4), 40-51.

[Kappel et al. 1995]
Kappel, G., Lang, P., Rausch-Schott, S., and Retschitzegger, W., 1995. Workflow Management

based on Objects, Rules and Roles. *IEEE Data Engineering Bulletin, Special Issue on Workflow Systems*, 18(1), 11-18.

[Kappel et al. 1996]
Kappel, G., Rausch-Schott, S., Retschitzegger, W., and Sakkinen, M. 1996. A Transaction Model For Handling Composite Events. In *Proceedings of the Third International Workshop of the Moscow ACM SIGMOD Chapter on Advances in Databases and Information Systems (ADBIS '96)*, Moscow, 116-125.

[Kappel et al. 1998]
Kappel, G., Rausch-Schott, S., and Retschitzegger, W. 1998. Coordination in Workflow Management Systems - A Rule-Based Approach. In *Coordination Technology for Collaborative Applications -Organizations, Processes, and Agents*, W. Conen, G. Neumann (eds.), Springer LNCS 1364, 99-119.

[Sheth 1993]
Sheth, A.P., and Rusinkiewicz, M. 1993. On Transactional Workflows. *IEEE Data Engineering Bulletin, Special Issue on Workflow Systems*, 16(2), 34-40.