A Blackboard Architecture for Workflows

Stefan Kleine Stegemann, Burkhardt Funk and Thomas Slotos

Universität Lüneburg, Fakultät III, Volgershall 1
21339 Lüneburg, Germany
{stefankst,funk,slotos}@uni-lueneburg.de

Abstract. Most of today's business applications have to deal with automated workflows. In this paper, we argue that the blackboard pattern can be used to create an architecture for the development of such workflows. We illustrate how the fundamental building parts of a workflow are addressed in such an architecture. The development of a prototypical group calendaring application verified the proposed architecture. It also served as a basis for contrasting the approach with commonly used workflow control constructs.

1 Introduction

Automated workflows play an important role in contemporary business applications. One option for the development of workflows is the adoption of a workflow management system. However, such systems are often hard to customize and to extend which makes integration with existing applications difficult. Custom development is a commonly chosen alternative. For these situations, we suggest that metaphor and reference architecture would be helpful. Our work shows how the blackboard pattern can be used to establish such a workflow architecture.

In this paper, we first outline our primary goals. The blackboard pattern is briefly described and we illustrate how this pattern can be used to implement the different aspects of workflows. To verify our approach, we implemented a prototypical system and evaluated how commonly used control flow constructs are supported.

2 Goals

The work presented in this paper is motivated by the need for architectural guidance in workflow development. We aim to provide a workflow architecture with focus on the following aspects.

Workflow management systems often come in an all or nothing fashion with a variety of features [7, 8]. The heavyweight architectures of these systems makes the integration with existing applications difficult. Consequently, software developers are often forced to build home-made solutions [8]. We want to suggest an architecture that can be used in these situations.

The concept of a workflow commonly fits a model that consists of two tiers [7]. Business functionality is split into activities which reside in the *work tier*.

Coordination and execution of activities is a responsibility of the *flow tier*. The proposed workflow architecture must provide a solution for the implementation of these two tiers.

A workflow can be described from different perspectives [5, 1]. One of these perspectives is the control flow perspective which describes the initiation, sequence and dependencies of individual activities. In this context, van der Aalst et al. [2] identified a number of control flow patterns. For a workflow architecture to be successful, it has to permit the implementation of commonly used patterns. We therefore consider it important that a workflow architecture can at least represent the basic control patterns [2].

3 Blackboard Architecture for Workflows

The blackboard model emerged in the domain of artificial intelligence [9]. It is based on a metaphor where a group of specialists gathers around a blackboard and solve a particular problem in cooperation. Our workflow architecture leverages the blackboard model to implement business workflows. It is based on the blackboard pattern [3] and uses the elements defined by this pattern to implement the different aspects of a workflow.

The **blackboard** is a shared repository containing the data from various problem solving states [4]. In our workflow architecture, the blackboard is used to store the workflow data. Data on a blackboard is often organized in elaborate structures. For business workflows, we suggest to divide the blackboard into multiple disjunct zones where each zone holds the data from one workflow instance.

The domain knowledge is partitioned into **knowledge sources** which read data from and write data to a blackboard. Knowledge sources are independent because they don't communicate with each other directly or know what other sources are present [4]. In our approach, knowledge sources are used to implement activities in a workflow (work tier). Because the focus in workflows is more on task completion rather than on knowledge representation, we replace the term knowledge source with **action**.

The **control** is the component that rules the system. Depending on the situation on the blackboard, it selects appropriate actions and executes them. In a workflow architecture, the control corresponds to the flow tier. It inspects the blackboard and decides which activities have to be executed. For each activity, a corresponding action is selected and executed.

A major issue of the control component, as it is suggested by the blackboard pattern, is the intermixing of two tasks: selection of actions and tracking the flow. The blackboard based control plan [6] addresses this problem by establishing a clear separation of concerns. It defines a meta controller that follows steps in **control plan**. In the suggested workflow architecture, a control plan specifies the flow of work by defining the steps to be executed. A control component tries to complete such a plan by selecting and executing the appropriate actions for each step.

4 A Prototypical System

To verify the proposed architecture, a prototypical system has been implemented. The system can be considered to be part of a fictitious larger groupware system that provides a calendaring functionality. The scenario we have implemented covers scheduling of meetings among a number of participants. In particular, it provides a sophisticated conflict resolution strategy that applies in situations, where a meeting is scheduled and one or more participants are not available at the proposed date.

The scheduling workflow was implemented with the blackboard architecture as described in the previous section. For the control plan, we chose a rule-based implementation. Each step in a plan is associated with a rule that is evaluated against the data on a blackboard. If a rule evaluates to true, it is desirable to perform the corresponding step. To map actions to steps, the control uses a creditability value that expresses the ability of an action to complete a particular step [6].

The OpenBBS framework [10] is an open-source project that was derived from the prototypical system. The framework has been used to implement the customer registration workflow in an online B2B trading application.

5 Support for Control Patterns

In section 2, support for commonly used control flow patterns has been defined as one goal for our approach. Based on the experiences with the group calendaring system, we evaluated if and how the five basic control flow patterns [2] are supported in a blackboard architecture with a rule based control plan.

Sequence: Add a step for each activity to the control plan and implement/assign an action for each step. Associate the rule for a step S2 that follows a step S1 with the existence of the output data of S1 on the blackboard. The action that performs step S1 must write the required output data to the blackboard.

Exclusive Choice: One of several possible branches is chosen. For every branch, add a step to the control plan. The rule for each step must check the condition for the branch in such a way that no or only one step is possible for a given situation on the blackboard.

Parallel Split: Multiple activities or sequences of activities have to be performed in parallel. Define a separate control plan for each sequence. Associate a rule with the initial steps for each plan. The rule evaluates to true if the data that has to be processed in parallel is written to the blackboard. For true concurrent execution, each control plan has to be executed by a dedicated control instance in its own thread. If concurrent execution is not required, a single control instance that executes all plans in the same thread is an alternative.

Synchronization: An activity must not be performed before multiple parallel activities or sequences of activities are completed. Add a step S to the control plan that represents the depending activity. The rule for S must not evaluate

to true before the steps which represent the parallel activities did complete. Assuming that each of this steps writes a result to the blackboard, the rule that is associated with S has to check for the existence of all results.

Simple Merge: An activity is performed after one of multiple possible branches has been executed. It is assumed that none of the alternative branches is ever executed in parallel. The implementation in the control plan is similar to synchronization. Define a step S for the depending activity. Associate a rule with S that evaluates to true if the result from at least one of the branches is present on the blackboard.

6 Conclusion

In this paper, we introduced a blackboard architecture for the development of workflows. We have shown how work- and flow-tier are addressed in this approach. Practicability has been verified through the prototypical implementation of a group calendaring system. Based on the experiences with this system, we evaluated the implementation of commonly used workflow control flow constructs with the proposed architecture. A complete evaluation of the full set of control patterns remains to be done. Such a study should investigate whether the blackboard approach is not only valuable as an architecture for the custom development of workflows but can also serve as a foundation for the implementation of extensible and customizable workflow management systems.

References

1. Van der Aalst, W.M.P.; Hee, K.: Workflow Management - Models, Method, and Systems. MIT Press, 2002.
2. Van der Aalst, W. M. P.; ter Hofstede, A. H. M.; Kiepuszewski, B.; Barros, A. P.: Workflow Patterns. In: Distributed and Parallel Databases, **14**(2003) 3, p.5-51.
3. Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P.: Pattern-oriented software architecture - a system of patterns. Wiley, Chichester, 1996.
4. Corkill, D. D.: Collaborating Software: Blackboard and Multi-Agent Systems & the Future. In: International Conference on Information Fusion (Fusion 2005), Philadelphia, Pennsylvania, 2005,
5. Jablonski, S.; Bussler, C.: Workflow Management Modeling Concepts, Architecture and Implementation. Thomson Computer Press, London 1996.
6. Lalanda, P: Two complementary patterns to build multi-expert systems. In: Pattern Languages of Programs (1997), Monticello, Illinois 1997.
7. Manolescu, D. A.: Micro-Workflow: A workflow architecture supporting compositional object-oriented software development. Ph.D. Thesis. University of Illinois, Urbana, 2000.
8. Manolescu, D. A.; Johnson, R. E.: A micro-workflow component for federated workflow. In: OOPSLA2000 Workshop on Implementation and Application of Object-Oriented Workflow Management Systems III, Minnesota, 2000.
9. Nii, H. P.: Blackboard Systems. In: A. Barr; P. Cohen; E. A. Feigebaum (Eds.): Handbook of Artificial Intelligence. Addison-Wesley, Boston, 1989.
10. OpenBBS Framework. Online. Available at http://openbbs.sourceforge.net/, accessed on 2007/10/03.