zur Muehlen, Michael; Becker, Joerg: Workflow Management and Object-Orientation – A Matter of Perspectives or Why Perspectives Matter. (with J. Becker) In: Cummins, F. (Ed.): Proceedings of the 2nd OOPSLA Workshop on the Design and Application of Object-Oriented Workflow Management Systems, Denver, November 1st 1999.

Workflow Management and Object-Orientation - A Matter of Perspectives or Why Perspectives Matter

Michael zur Muehlen, Joerg Becker

University of Muenster, Department of Information Systems
Steinfurter Str. 109, D-48149 Muenster, Germany
Phone +49 (251) 8338100, Fax +49 (251) 8338109
{ismizu|isjobe}@wi.uni-muenster.de

Abstract. The application of object-oriented concepts to the design and implementation of workflow management systems promises improvements with regard to the distribution, scalability and adaptability of these systems. However, most of these advantages can be fully realized only in an environment that is built on an object technology foundation. We propose a classification framework for workflow applications that can serve as a transition roadmap for workflow application designers that strive towards an object-oriented workflow scenario and provides integrators and users with information about potential implementation problems and how these can be avoided.

The process is treatment of objects.
Therefore, it consists of activities and objects.
[Fritz Nordsieck, 1934]

1 Object-orientation and Workflow Management

A large part of workflow research in industry and academia is considering concepts of object-orientation for the development and implementation of workflow management systems [2]. Several of these approaches target an infrastructure that is built entirely on object technology, mostly a CORBA environment. Examples for such approaches are e. g. the prototypes $WASA_2$ [3] and $METEOR_2$ [4], that realize a workflow application based on distributed CORBA objects.

Consequently applied, the implementation of workfow management systems based on object technology promises improvements with regards to scalability, distribution and adaptability to different implementation situations [5,6]. This is a desirable situation for the future, however, at the current state of technology, workflow management systems that rely exclusively on a business-object environment face challenges that hinder their application, such as performance problems for enterprise-wide application [7] or the difficulty to efficiently retrieve status information from distributed objects for monitoring and controlling purposes [8]. While these problems are likely to be solved in the long run, new workflow management systems have to fit into the existing enterprise information systems infrastructure in the meantime. We argue that

- The implementation of workflow technology as the result of a business process reengineering effort suffers from structural problems due to the nature of BPR approaches. Current BPR approaches need to be enhanced with object oriented concepts to serve as a starting point for workflow implementations.
- The design of workflow applications from a technical approach often ignores the business objectives that motivate the introduction of this technology. This is one cause for the large number of non-productive implementations in the field.
- A classification framework based on the granularity of workflow components can help users and developers assess the necessary steps during the introduction of a workflow solution and point out potential problem areas in advance. The framework helps companies to structure their IT assets in order to facilitate an easier migration towards object-based workflow management systems.

2 A "Typical" BPR Project

We have conducted a business process reengineering (BPR) project within a large, nationwide operating facility management enterprise in Germany (an extensive report on this project can be found in [9]). Over the course of more than two years, 13 process modelers and more than 70 domain experts created app. 600 process models that were documented using event-driven process chains (cf. [10]). These models are now implemented in 12 district offices of the company, affecting the work of app. 10,500 employees. Due to regional differences, a large number of the process models have to be adapted to local requirements, leading to the situation that more than two years after the project start a significant number of the business processes are still under construction and have not been fully implemented. In the meantime, the technological foundation of the company has evolved, several large application system projects were started in parallel affecting the design of the processes in which these systems are used. A transfer of the workflow type inheritance described by BUSSLER [11] to the field of business process modeling as proposed by VAN DER AALST et al. for Petri-Nets [12] would greatly improve the maintainability and therefore the durability of the results of such a BPR project.

Due to the chosen methodology, process type inheritance was not an option in our case, so other measures to secure the overall consistency of the process models had to be implemented. In particular we relied on two specific elements: We took a strict top-down approach for the modeling of business processes, starting with a framework of the company (a reference architecture) that was broken down into separate areas of the top level business process (providing facility services to customers). This way all interfaces between subprocesses could be documented and reviewed during the implementation of these processes. Secondly, we worked strictly with naming conventions that all domain experts agreed upon, eliminating potential homonym and synonym conflicts. These naming conventions not only determined the naming of objects such as orders, invoices or different types of buildings, but also described how activities and events had to be named (an activity was always named following the pattern ([verb] [process object]), e. g. create order, while states and events were named ac-

cording to the preceding activities following the pattern ([process object] [verb]), e. g. order has been entered).

Recently, the board of directors has shown interest to apply workflow technology to some of the company's business processes. With the help of the existing process models, several areas of interest could be identified, where workflow management technology could be used profitably. However, the existing process models need a thorough revision and refinement before they can serve as a requirements specification for a workflow project. This results partially from the fact, that the models were created with the perspective of organizational restructuring, thus data relevant for the implementation of a workflow process is not represented in the models (e. g. a detailed description of the information systems used within the execution of single activities). On the other hand the level of abstraction was - even at the present level of detail - too high in order to be of use for a technical implementation.

We can conclude, that the success of business process reengineering projects depends to a large extent on a sound framework that embeds the processes modeled within the project. In addition to this, the maintainability of the process models can benefit from object-oriented principles like inheritance. Nevertheless, even if BPR projects are carried out to a very fine level of detail, the resulting models have to be either refined or completely redone to serve as a foundation for software implementation. The use of a unified method could help to bridge this gap. Existing object-oriented modeling methods mainly focus on the support of software implementation and offer only little support for the various organizational perspectives of reengineering projects (organizational restructuring, ISO 9000 certification, activity based costing, to name a few). If an object-oriented workflow management system is to benefit from existing process knowledge in an enterprise, a continuous method is necessary that can accommodate these different perspectives. This remains a challenge for the forthcoming discussion about Version 2.0 of the Unified Modeling Language.

3 A "Typical" Workflow Project

In a different project, we were asked to model the invoice auditing process of an industrial enterprise to prepare the introduction of an ERP-based workflow system. The ERP system was already in place and it was a requirement by the user, that the proprietary workflow engine of the ERP software would be used. Furthermore, the system had to be integrated with an existing document management system, which had a proprietary interface to the ERP system. The meta model of the embedded workflow management system models a workflow process as a sequence of events that trigger the execution of methods on business objects within the system.

The problems that occurred within this project were of a different nature than the ones in the BPR project:

- The user was accustomed to the process modeling technique used for the customizing of the ERP software's reference models. Thus, the invoice auditing process should be documented using the same methodology. After the process had been modeled, it became apparent that the ERP system did not provide the necessary business objects by default, but they had to be devel-

oped using the internal 4GL programming language of the ERP system. No part of the existing process documentation could be used at this point.

- During one part of the process, an invoice should be signed by the department requesting the goods ordered. The relevant recipients could only be determined on the instance level and – even worse – most of the affected departments were not equipped to be part of a distributed workflow environment. These facts had never been an issue during any of the previous reengineering projects of the company.

The implementation of the workflow scenario finally succeeded, but using a slightly different workflow model from what the user expected and with a greater effort than anticipated by the user. This situation is not unusual within workflow projects. It has been noted previously that the number of productive workflow installations is still low compared to the total figure of systems installed (about 30%, cf. [13]). On the one hand, this is caused by the complexity of workflow management projects, that touch numerous areas of an enterprise. On the other hand, the integration of a workflow management system into an existing information system infrastructure is a costly and time-consuming task. Some analysts predict, therefore, that workflow management systems will cease to exist as stand-alone products and will ultimately be fully integrated into business applications [14], [15]. With regard to this workflow management systems can be seen as sophisticated CASE-Tools, that are used to enhance traditional application systems with process-oriented functionality such as the forwarding of process objects, monitoring capabilities or the logging of audit trail data. The tendency of workflow vendors to supply their systems as OEM components to independent software vendors (ISVs) or value added resellers (VARs) is a further indication of this development.

Besides the technical challenges workflow designers face today, they have to manage the expectations of their customers accordingly. Users perceive workflow technology too often as “just another BPR tool” and expect the implementation of a workflow application to be no more than just the graphical design of a workflow process and the naming of the associated application systems and users or roles respectively. While this may be true for some light-weight workflow implementations esp. in the area of document routing or form-based workflow processes, these are generally not the mission critical applications where workflow technology can provide a substantial competitive advantage for an enterprise. The framework presented in the next section can help users and designers to develop a migration strategy towards a workflow environment that takes full advantage of object technology.

4 A Framework for Different Levels of Workflow Components

In the following sections we analyze three different scenarios for component granularity in workflow management systems with respect to their economic potential and the implications for the workflow management system used (for a detailed discussion cf. [16]). The coarse level of granularity (*rocks*) is formed by workflow activities that call upon complete application systems. The invocation of single modules and procedures of application systems form a medium level of granularity (*stones*),

whereas the coordination of elementary functions by the workflow management system itself forms the finest level of granularity (*sand*).

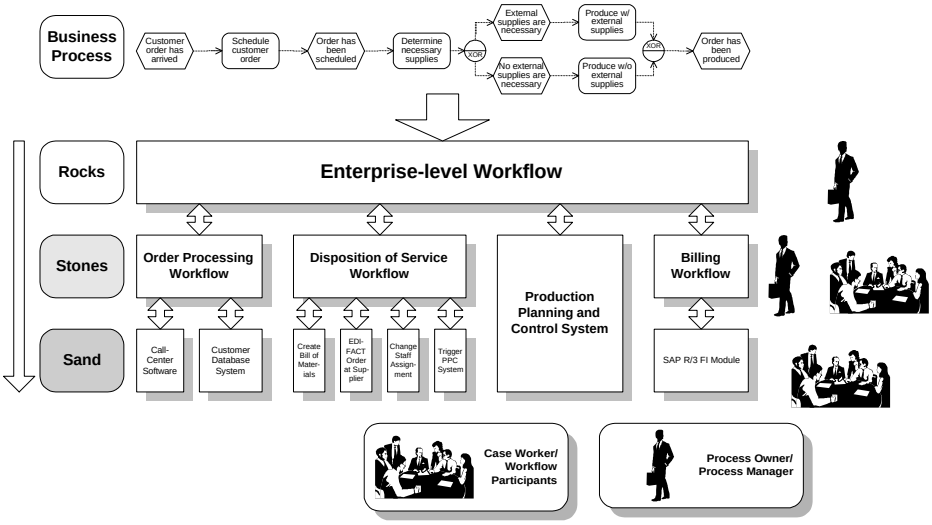

Figure 1. Different Granularity of Workflow Realization

4.1 Rocks

A coarse level of granularity can be found if a workflow management system implements top-level enterprise processes. In this case the invoked application systems are triggered as a whole while only few application data is exchanged between the workflow management system and the application systems. The complexity of the business process model is low, because at this high level of abstraction the process model usually consists of a small number of activities.

Possible applications that are likely to be linked at a coarse level of granularity are e. g. legacy applications, production planning and control-systems (PPC-systems) and other workflow management systems that control specific parts of a business process. For the supervising workflow management system some parts of the typical workflow functionality are less important, e. g. the process of staff resolution. In the "rocks“-scenario the staff resolution mechanism can be used to assign process managers to different sections of an enterprise-wide process. The information about process managers is relevant in case of escalations and can also be used for controlling purposes, e. g. the implementation of an activity-based costing system.

An invocation of applications at a coarse level of granularity is useful for those applications whose elementary functions cannot be coordinated by a workflow management system sufficiently. For example the PPC-functionality of ex-ante capacity planning and real-time-control is supported only to a little extent in current workflow management systems, because these mostly perform an ad-hoc allocation of resources to workflow activities at runtime. Therefore, it is easier to integrate the PPC-system as

a “black box” into the workflow model, exchanging only start/stop-information, than to emulate these functions using the built-in functionality of a workflow management system.

4.2 Stones

A medium level of granularity can be observed if the workflow management system invokes parts of application systems, e. g. transactions of an ERP-software, modules of a legacy system or activity specific parts of a standard software application. The granularity of the workflow components is determined by the size of the functions of the surrounding system (modules, functions, methods or procedures). This is especially true if the workflow management system is integrated into the components of an application family (so-called *embedded* workflow management systems).

Data consistency is crucial issue at a medium level of granularity, since references to application data may be exchanged more frequently than at the “rocks”-level and the workflow enactment service may be used to pass data between different application systems. With the increasing number of activities the complexity of the workflow model increases which in turn leads to an increasing number of control data entities. If the components invoked in the workflow activities do not provide dedicated interfaces for the import and export of data, the workflow management system may have to perform data conversions as well. The management of data integrity is shifted from the application level towards the level of the workflow management system.

Due to the increasing complexity at a medium level of granularity the importance of activity coordination increases as well as the importance of the coordination of application systems. The assignment of actors to activities is determined by the division of labor within the single modules, thus, the importance of actor coordination varies. If the an activity is performed by a single actor, traditional staff resolution concepts can be employed. If several actors are involved in the execution of an activity, e. g. if the application system can only be triggered as a whole, the workflow management system can only assign a responsible actor to the activity. This can be problematic with respect to the authorization concept. The assignment of an activity to a specific actor does not necessarily ensure that this actor has the necessary privileges at application level to perform the activity in a proper way, e. g. access rights to certain fields in a database management system. The workflow modeler has to ensure that both the role model of the workflow management system and the access model of the invoked applications fit together. This may lead to a redundant maintenance of access privileges in both systems.

4.3 Sand

The finest component granularity can be found, if a workflow management system controls the application logic at an elementary level, e. g. by invoking the methods of a business object. In this case the control flow of existing application systems is bypassed by the workflow management system, making it no longer part of the process logic. A precondition for this is the accessibility of the invoked applications at an elementary level, which can be assumed, if the enterprise information infrastructure

builds on top of elementary business objects. The integration of existing applications, however, is just one alternative in the "sand“-scenario. Another alternative is the use of a workflow management system as a CASE-tool for the design of process-oriented applications. Depending on the process modeling paradigm required, this can either be realized through the development of activity specific application components around a central workflow engine, or through a distributed enactment of workflow process objects that encapsulate the relevant routing information themselves [17]. Regarding security requirements, the workflow management system has central control over the access privileges of the users, therefore, the access control can be embedded in the role concept of the workflow management system.

5 Conclusions

From a workflow designer’s point of view the ideal workflow scenario involves a homogeneous granularity of application logic within the enterprise. In this case, the workflow management system can take over all relevant coordination functions and maximize the benefits that the automation of coordination between business objects and enterprise resources promises. However, most enterprises cannot offer this ideal environment, but have the desire to implement workflow technology.

A separation of the information technology infrastructure found in an enterprise into areas of rocks, stones and sand can provide a valuable insight where object-oriented workflow technology can be applied profitably. Moreover, the separation enables us to use different workflow paradigms (such as speech-act based modeling vs. input-process-output based workflows as classified by CARLSEN (cf. [18])) in different parts of the enterprise, where the elementary business processes require different paradigms. The OMG workflow facility provides a mechanism to link together such a distributed workflow infrastructure [19]. The defined mechanisms for interoperability between the different workflow process managers can be applied to integrate workflow applications on different levels of abstraction. This way, a stepwise migration towards a business-object-based workflow environment is possible without forcing users to change their existing infrastructure at once.

References

1. Nordsieck, F.: Grundlagen der Organisationslehre. (*Foundations of Organizational Theory*). Stuttgart 1934. [in German]
2. Schulze, W.: Implementation and Application of Object Oriented Workflow Management Systems. In: Proceedings of the 1998 OOPSLA Workshop on the Implementation and Application of Object Oriented Workflow Management Systems. Vancouver 1998.
3. Weske, M.: Workflow Management Through Distributed and Persistent CORBA Workflow Objects. In: Jarke, M.; Oberweis, A.(Eds.): Proceedings of the 11th International Conference on Advanced Information Systems Engineering (CAiSE 1999). Springer, Heidelberg 1999, pp. 446-450.
4. A. Sheth, K. J. Kochut, Workflow Applications to Research Agenda: Scalable and Dynamic Work Coordination and Collaboration Systems, in: A. Dogac, L. Kalinichenko, M. Tamer

Ozsu, A. Sheth (Eds.), Workflow Management Systems and Interoperability, Springer, Heidelberg 1998.
5. Leymann, F.: Workflows Make Objects Really Useful. Proceedings of the 6th Workshop on High Performance Transaction Systems. Asilomar 1995.
6. Schmidt, R.; Assmann, U.; Biegler, P.; Kramer, R.; Lockemann, P. C.; Rolker, C.: The Interrelatedness of Component-oriented Systems And Workflow-Management - What they can do for each other. Proceedings of the Workshop on Compositional Software Architectures, Monterey, January 1998.
7. Alonso, G.; Agrawal, D.; El Abbadi, A.; Mohan, C.: Functionality and Limitations of Current Workflow Management Systems, IEEE Expert, Vol. 12, No. 5, 1997.
8. zur Muehlen, M.; Rosemann, M.: Workflow-based Monitoring and Controlling – Technical and Organizational Issues. To appear in: Sprague R. Jr. (Ed.): Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS 2000). Wailea 2000.
9. Becker, J.; Kugeler, M.; Rosemann, M. (Eds.): Prozessmanagement (*Process Management*). Springer, Heidelberg 1999.
10. Scheer, A.-W.: ARIS – Business Process Modeling. 2nd Edition. Springer, Heidelberg 1998.
11. Bussler, Chr.: Towards Workflow Type Inheritance. In: Proceedings of the 1998 OOPSLA Workshop on the Implementation and Application of Object Oriented Workflow Management Systems. Vancouver 1998.
12. van der Aalst, W.; Moldt, R.; Valk, R.; Wienberg, F.: Enacting Interorganizational Workflows Using Nets in Nets. To appear in: Proceedings of the 1999 Workflow Management Conference. Muenster 1999.
13. Delphi Consulting Group: Workflow Software Buyer's Perceptions Today. Volume WF96 – Number 1. Boston 1996.
14. Marshak, Ronni T.: Workflow: The Market That Never Happened. In: Workgroup Computing Report. Patricia Seybold Group. December 1996.
15. Abbot, Kenneth R.; Sarin, Sunil S. Experiences with Workflow Management: Issues for the Next Generation. In: Proceedings of the 1994 Conference on Computer Supported Cooperative Work (CSCW 94), Chapel Hill 1994, pp. 113-120.
16. Becker, J.; zur Muehlen, M.: Towards a Classification Scheme for Application Granularity in Workflow Management Systems. In: Jarke, M.; Oberweis, A. (Eds.): Proceedings of the 11th Conference on Advanced Information Systems Engineering (CAiSE '99), Heidelberg 1999, pp. 411-416.
17. Barbará, D.; Mehrothra, S.; Rusinkiewicz, M.: INCAs: Managing Dynamic Workflows in Distributed Environments. In: Journal of Database Management, 7 (1996) 1, pp. 5-15.
18. Carlsen, S.: Conceptual Modeling and Composition of Flexible Workflow Models. PhD Thesis, Information Systems Group, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim 1997.
19. Object Management Group: Workflow Management Facility. Document bom 99-03-01. Framingham, MA 1999.