Becker, J.; zur Muehlen, M.: Towards a Classification Framework for Application Granularity in Workflow Management Systems. In: Jarke, M. (Ed.): Proceedings of the 11th Conference on Advanced Information Systems Engineering 1999 (CAiSE 99), Heidelberg 1999, pp. 411-416.

Towards a Classification Framework for Application Granularity in Workflow Management Systems

Jörg Becker, Michael zur Muehlen

University of Muenster, Department of Information Systems
Steinfurter Str. 107, D-48149 Muenster, Germany
{isjobe|ismizu}@wi.uni-muenster.de

Abstract. The support of process enactment through the use of workflow management systems has gained considerable attention within the last few years. We develop a classification framework consisting of three different strategies for the use of workflow management systems and outline how these scenarios can be applied in practice. Following the definition of relevant terms in the we identify coordination techniques that are provided by workflow management systems in order to support the automation of business processes. These coordination techniques are evaluated with regard to their dependency on the granularity of applications using the classification framework. The results of this evaluation are compared and an outlook for future research is given.

1 Basic Workflow Geology

For the realization of integrated information systems, several integration perspectives have emerged in the past. Two of these perspectives are the *integration of data*, which is the subject of *data management*, and the *integration of functions* which is part of various *process management*-approaches. A major part of today's business applications satisfies the demands of process orientation only to a minor extent, because they are built as functional program hierarchies and, therefore, are oriented among the fulfillment of single functions. Workflow management systems provide the opportunity to integrate these functionally structured applications into process oriented applications. They guide the user along the business processes to be executed and provide the data and applications that are necessary for the execution of the partial functions.

A *workflow* is the part of a work process that contains the sequence of functions and information about the data and resources involved in the execution of these functions. It is described using a *workflow model*, which usually does not describe the behavior of the single functions or the data to be processed as long as these do not determine the flow of work itself [1]. A *workflow management system* (WfMS) is an information system that enables the execution, coordination and control of workflows. A *workflow application* is the combination of a WfMS and the invoked application systems that is used to support the execution of a work process based on a workflow model [2].

2 Coordination and Efficiency

WfMS support the execution of work processes by the provision of automated coordination mechanisms. Through the reduction of manual coordination and the migration of these coordination mechanisms from existing application systems into the workflow engine WfMS enable a more efficient management and adoption of control mechanisms in enterprises. The coordination mechanisms of a WfMS are outlined in table 1 (cf. [3]) while table 2 shows five efficiency criteria that can be used to measure the efficient execution of business processes (cf. [4]).

Table 1. Coordination Mechanisms of WfMS

Activities	The WfMS automates the transitions between single process activities. The implicit knowledge about the sequence of activities can be handed over to the workflow system entirely (transactional or production workflow) or it can be left in part to the discretion of the workflow participants (ad-hoc or collaborative workflow).
Actors	The WfMS supports the assignment of actors to single process activities according to a set of rules. The coordination instruments used here are the notification and synchronization mechanisms of the work lists.
Data	During the presentation of a workflow activity the WfMS provides the relevant data necessary for the fulfillment of the given task. The efficient supply of data is one of the most important economic arguments for the use of WfMS [5].
Application Systems	During the execution of activities the WfMS provides the actor with the tools necessary for the fulfillment of the given tasks and coordinates the appropriate application systems.
Monitoring and Controlling	WfMS foster the automation of the extraction, analysis and user-appropriate presentation historic data about workflow instances. This data is the main input for early warning mechanism and may serve as a foundation for continuous process improvement [6].

Table 2. Efficiency Criteria for Business Processes

Process Efficiency	Optimization of process criteria such as processing time (to be minimized) or faithfulness to deadlines (to be maximized).
Resource Efficiency	Efficient use of the resources (human resources as well as application systems) available for the execution of processes.
Delegation Efficiency	The proper positioning of the enterprise in its relation to market partners. This includes a reliable prediction of delivery times, transparent communication with suppliers and customers and optimized procurement and distribution processes.
Market Efficiency	Adequate use of the competencies of superior (greater scope of vision along the process) and subordinate (detailed knowledge about single activities) organizational units.
Motivation Efficiency	Motivation of staff to act in a way congruent to the business goals of the enterprise.

3 Application Granularity

An *Invoked Application* is an existing software solution, that is being integrated into one or more activities of a workflow model. The application is executed during the invocation of the workflow activity in order to provide the workflow participant with the tools necessary to fulfill the given task. In this course data can be exchanged between the WfMS and the invoked application. Data affecting the sequence of activities is called *workflow relevant data* while internal data of the single activities is called *application data* (cf. [7]). In most cases application data is not stored in the database of the WfMS, but it is handled by the application system itself. An exception from this approach can be found at some document centric workflow systems.

In the following sections we analyze three different sizes of invoked applications with respect to their economical potential and the implications for the WfMS used. The coarse level of granularity (*rocks*) is formed by activities that call upon complete application systems. The invocation of single modules and function parts of application systems form a medium level of granularity (*stones*), whereas the execution of elementary functions by the WfMS itself forms the finest level of granularity (*sand*).

3.1 Rocks

A coarse application granularity can be observed if a WfMS supports processes at the enterprise level. In this case the invoked application systems are triggered as a whole while only few application data is exchanged between the WfMS 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. The coordination of activities and the assignment of work to workflow actors play only a minor role within this scenario.

Possible applications that can be linked at a coarse level of granularity are e. g. legacy applications, PPC-systems and other WfMS that control specific parts of a business process. For the supervising WfMS 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. A linking of applications at a coarse level of granularity is useful for those applications whose elementary functions cannot be coordinated by a WfMS sufficiently. For example the PPC-functionality of ex-ante capacity planning and real-time-control is implemented only to a little extent in current WfMS. 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 WfMS.

Data consistency is easy to maintain at a coarse level of granularity, because only control information for the invoked applications have to be exchanged (start, stop etc.). Application data is exchanged only in few cases. Component designers for a coarse level of granularity have to provide interfaces that enable the invocation of the application by a WfMS, the querying of the program status and the passing of control data and application data and pointers respectively. The coordination functions of WfMS mentioned in section 2 are of varying importance in this scenario. While the coordination of activities has only little importance due to the few activities in the workflow model, the coordination of application systems is more important. The coordination of actors and the coordination of data are less important due to the high level of abstraction. Moreover, the level of abstraction fosters the use of the audit trail data for controlling purposes at the management level. In order to identify weaknesses in the process this data is less useful, because important information about the elementary functions is missing. Since the WfMS does not deal with application data, security aspects such as the control of data access and the permissions of workflow actors have to be maintained by the application systems themselves.

3.2 Stones

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

Data consistency is much more important at a medium level of granularity than at the “rocks”-level since references to application data may be exchanged more frequently. 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 invoked applications do not provide dedicated interfaces for the import and export of data, the WfMS has to perform data conversions as well. The management of data integrity is shifted from the application level towards the level of WfMS.

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 WfMS 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 WfMS and the access model of the invoked applications fit together. This may lead to a redundant maintenance of access privileges in both systems.

Information generated with respect to the monitoring and control functions of the WfMS show a medium level of detail within the “stones” scenario. This data is suited for controlling purposes as well as a feedback engineering of process weaknesses. In order to enable a detailed process analysis the audit trail data has to be linked to the application data as described in section 3.1. In addition to interfaces for the exchange of control- and application data, component designers have to provide control mechanisms at different levels of abstraction that enable e. g. the starting of the entire application, the calling of a single module as well as the start of a single transaction.

3.3 Sand

The finest granularity of invoked applications can be observed, if a WfMS controls elementary functions of the invoked applications. In this case the control flow of the applications is no longer part of the process logic. This way the WfMS can be used to bypass the original “hard-coded” control flow within the application system. A precondition for this is the accessibility of the invoked application at an elementary level, e. g. through Remote Function Calls. The integration of existing applications, however, is just one alternative in the "sand“-scenario. Another alternative is the use of a WfMS as a CASE-tool for the design of process-oriented applications. The development of activity specific application components around a central workflow engine leads to the design of a workflow-specific application. Some WfMS provide the user with functions for the development of small applications such as form generators and script languages. In many cases the functionality of these tools is sufficient for the design of an entire workflow application. Within such *integrated workflow applications* the WfMS takes control of the entire data management, because not only workflow relevant data but also application data are exchanged via the interface between the workflow engine and the invoked/integrated applications. Data consistency and integrity have to be maintained by the WfMS entirely. Thus, the relevant data should be stored in a central repository in order to minimize redundancies and other integrity problems. Regarding the actor's permissions the scenario "sand" is less sensitive than the scenario "stones". Here the WfMS has a central control over the access privileges of the users, the access control can be embedded in the role concept of the WfMS.

The importance of activity coordination increases with the increasing complexity of the workflow models. However, the significance of application coordination decreases with every external application that is replaced by a workflow-specific integrated application. The coordination of actors is of increasing importance since the assignment of actors and activities is part of the staff resolution process and maintained entirely within the WfMS. In order to assign process managers to segments of the workflow process proprietary functions of the WfMS have to be employed. With respect to monitoring and controlling the amount of logged audit trail data increases with the complexity of the workflow models. On the one hand, this enables a very detailed monitoring of running workflow instances, thus enhancing the ability to answer to customer and supplier inquiries. With respect to a controlling based on historical workflow data the low degree of abstraction leads to a very large amount of data. Therefore, tools such as data mining systems or process information systems (cf. [8]) should be used in order to analyze the audit trail data.

4 Summary and Outlook

Table 3 summarizes which of the single coordination mechanisms of WfMS are affected by the different levels of application granularity. With a finer granularity the intensity of coordination by the WfMS increases. While at the "rock"-level the linking of entire application systems is the primary focus of the workflow application, at the "sand"-level the WfMS performs an intense coordination at all levels outlined. The more coordination mechanisms are transferred from the application systems and manual actors toward the WfMS, the higher the efficiency benefits resulting from the use of workflow technology are. Therefore, the size of the invoked applications not only determines the process logic of the workflow application but also determines the economical results for the specific enterprise. While the decreasing size of applications increases the coordination intensity this goes along with an increasing complexity of the workflow model and an increasing effort for the implementation of a workflow application. As a result, there cannot be a generally accepted recommendation for the size of the invoked applications.

The linking of "rocks" as well as the coordination of "sand" can be useful in different scenarios. The "stones"-level seems to be more of a problem, because the actor’s access rights have to be maintained within the WfMS and the invoked applications. This can lead either to a more detailed modeling of the workflow processes (migration towards the “sand”-scenario) or to a coarse application granularity (migration towards the “rocks”-scenario). Also, a mixture of the scenarios outlined is feasible. Existing larger application systems can be linked using the “rocks”-scenario, ERP-software can be integrated at the “stones”-level, while new applications can be designed using the WfMS as a CASE-Tool.

Table 3. Coordination aspects at different levels of application granularity

	Application level (Rocks)	**Module level (Stones)**	**Function level (Sand)**
Coordination of applications	Linking of entire applications	Linking of single functions / modules	Linking of elementary functions
Coordination of activities	Few activities, low complexity of workflow models	Medium number of activities, medium complexity of workflow models	Many activities, highly complex workflow models
Coordination of actors	Assignment of process managers	Assignment of process managers and actors	Assignment of actors
Coordination of data	Exchange of control data. Exchange of references to application data	Exchange of control data. Exchange of references to application data	Exchange of control and application data
Management and control	Monitoring and controlling at a high level of abstraction (management data)	Monitoring and controlling at a medium level of abstraction (middle management data)	Detailed monitoring data, tool support necessary for controlling (data mining)
Data consistency	To be ensured by the application system	To be ensured across system borders	To be ensured by the WfMS

References

1. zur Muehlen, M.: Evaluation of Workflow Management Systems Using Meta Models. In: Sprague, R. (Ed.): Proceedings of the 32nd Hawai`i International Conference on System Sciences (HICSS '99). IEEE, Los Alamitos 1999.
2. Leymann, F. and Roller, D.: Workflow-based Applications. In: IBM Systems Journal, 36 (1997) 1, pp. 102-123.
3. Becker, J.; v. Uthmann, Ch.; zur Muehlen, M.; Rosemann, M.: Identifying the Workflow Potential of Business Processes. In: Sprague, R. (Ed.): Proceedings of the 32nd Hawai`i International Conference on System Sciences (HICSS '99). Springer, Berlin et al. 1999.
4. Theuvsen, L.: Business Reengineering. In: zfbf, 48 (1996) 1, pp. 65-82. (in German)
5. Fisher, L.: Excellence in Practice – Innovation and Excellence in Workflow and Imaging. Lighthouse Point 1997.
6. McLellan, M.: Workflow Metrics – One of the great benefits of Workflow. In: Österle, H. et al. (Eds.): Praxis des Workflow Management. Vieweg, Braunschweig 1996, pp. 301-318.
7. Workflow Management Coalition: Terminology & Glossary. Document Number WFMC-TC-1011. Document Status-Issue 2.0. Brussels 1996.
8. Rosemann, Michael; Denecke, Thomas; Püttmann, Markus: Konzeption und prototypische Realisierung eines Informationssystems für das Prozeßmonitoring und -controlling. Paper Nr. 49, Department of Information Systems, University of Muenster, 1996. (in German)