zur Muehlen, Michael; Allen, Rob: Embedded vs. Autonomous Workflow - Putting Paradigms into Perspective. In: Layna Fischer (Ed.). Excellence in Practice Volume IV: Innovation and Excellence in Workflow and Knowledge Management. Future Strategies, Lighthouse Point, FL 2000, pp. 49-58

Embedded vs. Autonomous Workflow – Putting Paradigms into Perspective

Michael zur Muehlen
University of Muenster
Department of Information Systems
Steinfurter Str. 109
48149 Muenster, Germany
ismizu@wi.uni-muenster.de

Rob Allen
Open Image Systems, Inc.
1000 Great West Road
Brentford
Middlesex TW8 9HH, UK
rob_allen@sns.ca

1 INTRODUCTION

This paper distinguishes, at a high level, the differences between the architectures of workflow management systems. Its main function is to clarify the segmentation between autonomous and embedded workflow deployments.

Workflow is the automation of a business process, in whole or in part, during which documents, information, or tasks are passed from one participant to another for action, according to a software representation of the process logic, the workflow model.

A workflow management system defines, creates and manages the execution of workflows, through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants, and, where required, invoke the use of information technology (IT) tools and applications.

A workflow engine provides the run-time execution of business processes. Engines can be embedded within other applications or they can be deployed as independent applications inter-operating with other applications. Some engines can be deployed in either mode. It is claimed that some application components, which deal with rules or exceptions, are in fact workflow engines, whilst they do not display most features that identify common workflow features. With the advent of new requirements for workflow engines to inter-operate for tasks such as Supply Chain Management, it is important for the market to be able to distinguish between inaccessible rules-based application components, and workflow engines, be they embedded or not.

2 CLASSIFICATION OF WORKFLOW APPLICATIONS

Autonomous Workflow

An autonomous workflow management system is functional without any additional application software, with the exception of database management systems and message queuing middleware. For the deployment of an autonomous workflow solution, application systems that are external to the workflow management system are invoked at runtime and workflow relevant data is passed between the workflow participants.

Autonomous workflow management systems are separate pieces of application software that provide the workflow functionality. They normally have their own user interfaces and will access data from other applications. They are usually installed to support a variety of different applications.

The modeling of autonomous workflow applications requires the specification of interface information for the invoked applications, relevant data structures and involved participants, and thus can become a complex challenge

Embedded Workflow

An embedded workflow management system is only functional if it is employed with the surrounding (embedding) system - for instance, an Enterprise Resource Planning (ERP) system. The workflow functionality of embedded workflow management systems is exhibited by the surrounding software system. Common examples include ERP systems, payment and settlement systems. The workflow components are used to control the sequence of the application's functions, to manage queues and to assist with exception processing.

It is valuable for users to be able to differentiate between rules based sectors of an application that are normally activated by database triggers, and workflow engine-based components, that usually allow for a more complex specification of processes.

The former is normally written by the application authors and only operates within their application and only supports relevant functions. The later is normally an interchangeable component, that is, the same engine will work in many applications. Normally, these engines provide more functional interfaces that are generally standards based.

Embedded systems are available in two distinct categories. Whereas workflow-based solutions are not functional without the built-in workflow functionality, workflow enabled systems leave it to the discretion of the implementer, if the built-in workflow component is used in a given context.

3 CLASSIFICATION OF EMBEDDED WORKFLOW SOLUTIONS

Embedded workflow solutions can be differentiated according to their level of accessibility, ranging from proprietary solutions that are exclusively used within their surrounding system, to standardized solutions that implement interfaces that are defined by external standardization organizations such as the Workflow Management Coalition or the Object Management Group. In detail, the different classes of embedded workflow management systems are[1]:

Proprietary Solution

The workflow component supports the built-in application functionality of the application exclusively. It is not possible to invoke external application systems through the workflow engine at runtime.

Semi-Open Solution

The workflow component offers proprietary interfaces for the integration of external systems, such as office applications, as client applications. This way, a business process that is executed in part with the help of additional software packages can be automated using the built-in workflow functionality of an integrated system.

Open Solution

The workflow component offers interfaces for the integration of external systems as server applications. Open solutions are accessible from the outside and display interfaces for the manipulation of workflow execution, such as the starting, stopping, suspending etc. of workflow instances. If the integrated

[1] Cf. Becker et al. (1998).

system does not offer facilities for the integration of Internet services, one common application of these interfaces is the invocation of workflows through an external source, such as a web browser.

Standardized Solution

The workflow component offers standardized interfaces for the integration of external applications, the interoperability with other workflow enactment services and the client control of the workflow engine. Examples for these standardized interfaces are the WfMC Workflow Application Programming Interface (WAPI)[2] or the OMG workflow facility[3].

4 MEASURING THE BENEFITS OF A WORKFLOW APPLICATION

A workflow management system determines the flow of work according to pre-defined business process definitions. It manages the resources (i. e. applications, data, people) required to meet goals and provides monitoring facilities and control capabilities.

Normally, this yields significant savings of idle time, search-related activities and supply chain transportation delays. In combination with document management systems (DMS), the elimination of paper-based procedures such as copying, manual archiving and retrieval as well as in-house distribution is often the most important economic argument for the introduction of a workflow management system.

All simple business decisions can be automated, such as the assignment of a task to either a customer representative or his manager, depending on the value of the customer request. If the decision can be expressed formally, it can be automatically evaluated by a workflow engine.

Information relating to business operations are available instantly so that mangers can have a much closer knowledge of what is going on in the business, and have the opportunity to react faster. Process-oriented monitoring and controlling capabilities enhance the process transparency, help identifying potential problems in the process design and foster early-warning mechanisms for potentially overdue work items.

Modern component object architectures together with better-defined API's have led to workflow engines that can be commonly used to inter-operate with other application systems. The WfMC Reference Model, published in 1997, outlines five functional interfaces between a workflow management system and external application systems. By now, many workflow engines have facilities to inter-operate with a number of other processes. This exceptionally useful functionality is fundamental to differentiating workflow engines, both embedded and autonomous.

5 PUTTING WORKFLOW INTO PERSPECTIVE

In the following section, the difference between embedded and autonomous workflow management systems is analyzed from a variety of perspectives that may influence the decision for or against one of the two solutions. It should be noted, however, that our intention is not to state a recommendation for a specific type of system, but instead to present those aspects that distinguish the two system types.

[2] Cf. WfMC (1999).

[3] Cf. OMG (1999).

Since the selection of a workflow management system has to take a multitude of aspects into account, such as the characteristics of the process to be supported, the existing organizational and technological infrastructure as well as economic and strategic goals of the workflow project, the decision for or against a specific system can only be based on a thorough investigation of all relevant fields, e. g. through a differentiated scoring model. First, the single criteria are valued using binary values (e. g. yes=1, no=0) or numerical scores (e. g. very good=5, good=4, average=3, below average=2, poor=1). The valuations are then multiplied with weights for every criterion. The summary of the resulting numbers represents the score of the respective system.

It should be noted, that the typical scoring models for the evaluation of different software systems are based on a number of (often neglected) assumptions. One (hidden) assumption within this approach is, that criteria can compensate each other, e.g. a low score for "usability" can be compensated by a high score for "supported database platforms." This has to be taken into account when selecting criteria and evaluating the results of a scoring approach.

Furthermore, a numerical score for a system has no specific dimension and is hard to interpret (in what way do two systems, that are valued 94.5 to 87.3 differ?). Therefore, while we try to evaluate the advantages and disadvantages of the respective architectural approaches, a workflow project manager needs consider a multitude of other factors before deciding on a specific system.

The following perspectives can be differentiated, as these roles are typical for a workflow project. The user perspective takes into account the view of the end user that works with the system on a daily basis. Aspects such as ease of learning and usability are most important for this role.

The process designer typically is a business analyst that models the business process at an abstract level (typically using a business process modeling tool) and leads reorganization efforts prior to the introduction of a workflow management system. From this perspective the transformation of process into workflow models is of interest.

The workflow developer is a system analyst that implements the process model delivered by the process designer in a given workflow development environment. This role deals with more of the technical aspects of a workflow management system, such as the interface standards supported or the expressiveness of the built-in modeling language.

The administrator is responsible for the maintenance of the workflow application at run-time and is mainly interested in system specifications such as maintainability, scalability, security, recovery mechanisms etc. Finally, the enterprise perspective takes into account the strategic and economic goals of the company as a whole. For each perspective, the relevant attributes are analyzed and a short characterization of the specific properties of embedded and autonomous workflow solutions is given.

User Perspective

How does the user perceive the different types of workflow? With embedded workflow the presentation, the user's view of the application is usually identical to other parts of the application. Indeed, users may not even recognize that they are using workflow technology. Autonomous workflow technology usually presents itself as a separate application. This can lead to additional training requirements, but normally, autonomous products are available for additional functionality. If this is not necessary, most autonomous workflow systems can be integrated into existing office communication platforms or employee portals; however, this requires additional implementation effort.

From the perspective of data availability, users of embedded workflow systems have transparent access to application and workflow data via the same interface, while in autonomous environments, this information may be scattered among several application systems and the workflow system. With regard to the control of workflow within applications, autonomous workflow systems are limited by the accessibility of the invoked application systems, presenting users with monolithic applications, if their functionality cannot be isolated. Embedded workflow applications, on the other hand, can provide control at a very fine level of granularity, down to the level of data field content validation. However, a finer granularity increases the effort for the implementation of the workflow application significantly.

FEATURE	AUTONOMOUS	EMBEDDED
One-Face-to-the-User	Different GUI, if no additional measureas are being taken	Unified GUI
Learning perspective	New, additional application system	Mostly transparent, if built-in in-box is already in use
Data availability	Limited by the invoked application systems	Inherent system function
Control-flow within applications	Medium-coarse granularity	Fine granularity

Table 1: Embedded and autonomous WfMS from the perspective of end-users

Process Designer Perspective

The process designer is faced with the choice of building the business rules from within the embedding application system or through the integration of independent application systems. In the first case the operation is centered on the single application with the opportunity to access external systems; in the latter case the business process automation is built in an autonomous workflow system and can access all individual applications.

Since the process design for organizational reengineering projects is mainly done using a modeling tool, the compatibility of the methods used for BPR-modeling and for workflow modeling is of great interest. In case of an autonomous workflow application, the modeling method in most cases differs to some extent from the standard BPR-methods, such as flowcharts or event-driven process chains.

In some cases a transformation of existing models into the proprietary modeling format of the workflow management system is possible. However, some semantics of the processes may be lost during the transformation process. An embedded solution in many cases uses the same modeling technique also used for the customization of the reference processes of the surrounding system, possibly enhanced to make it suitable for the purpose of workflow modeling.

Of interest for the designer is the use of business reference models, as they have been provided by many ERP and BPR-Tool vendors. Usually the use of reference models within autonomous workflow applications is limited, due to the independent nature of the invoked applications. The integration capabilities of an embedded solution are naturally higher, because the interfaces between the workflow engine and the application logic are defined by the same vendor, whereas autonomous systems have to cope with a multitude of interface standards that add to the complexity of a workflow project.

Automatic data maintenance refers to the mechanisms for the integration of external repositories that are relevant for the design of workflow processes. These repositories can be e.g. the human resource component of an ERM system. If the participants of a workflow are specified separately, the maintenance of consistency between the organization defined in the workflow and the application systems can become a time-consuming and error-prone task. If a workflow system allows for the use of an external repository instead of built-in components, the overall system integrity is easier to maintain.

FEATURE	AUTONOMOUS	EMBEDDED
Modeling Method	Mostly different from application software	Usually homogeneous with embedding system
Reference Models	Difficult to use because of the independent application systems	Can be provided with the embedding system
Integration	Manual integration of application systems, standard interfaces may be supported by default	Integration mechanisms provided by the vendor
Automatic Data Maintenance	Limited to WfMS, integration limited by app.systems	Additional data integration with the embedding system

Table 2: Embedded and autonomous WfMS from the perspective of process designers

Developer Perspective

The workflow developer deals with the actual implementation of the process designer's business process model in the workflow environment. Therefore, design and integration of the workflow model with the surrounding applications play an important role for this type of user.

Authors of multifaceted software products have designed easy-to-use workflow tools, so building rules within the confines of their products tends to be trouble-free. However, interaction with external processes can be challenging. The integration of access control and user rights on both the workflow as well as the application side create additional effort in a heterogeneous environment.

Furthermore, performance optimization and import of existing process models are fostered by the homogeneous development environment of embedded workflow management solutions. The last point should not be valued too high, though. The implementation of a workflow application consists largely of integration tasks, whereas the actual process modeling takes up only a small fraction of the overall implementation time.

FEATURE	AUTONOMOUS	EMBEDDED
Consistency/Integrity	System process integrity and integration management have to be addressed	Many problems are solved due to the integrated nature of the systems
Interfaces	Standardized interfaces are available (CORBA, DDE, OLE etc.), individual programming is necessary for most for large-scale projects	Internal, sometimes proprietary interfaces to the surrounding system, interfaces to external systems are usually available to a lesser degree than in autonomous systems
System Optimization	Rarely possible due to system distribution	Possible due to integrated repositories
Import of BPR models	Interfaces to modeling tools, possible loss of semantict	Customizing of existing models

Table 3: Embedded and autonomous WfMS from the perspective of workflow designers

Administration Perspective

From an administration perspective, the run-time behavior of a workflow management system is the most interesting aspect. Scalability with an increasing number of users and processes, recovery possibilities after system failures, update and release policies as well as overall administrative effort are the determining factors for this type of user.

Embedded workflow applications tend to be regarded as part of the hosting application. Whilst autonomous workflow operations are additional applications to be managed, (deployment, version control, routine back-ups), they are scalable independent of the invoked application systems and thus foster the growth from a smaller pilot implementation to an enterprise-wide solution.

With regard to the upgrade and release policy of vendors, companies that produce embedded solutions tend to be larger than traditional workflow-only companies. This increases the security for future updates of the workflow component. However, if part of the surrounding applications is changed due to the update policy of the company, an existing workflow application may need to be changed as well.

Existing back up and recovery measures for large-scale application systems cover embedded workflow applications as well, whereas the independence of the application systems invoked in autonomous workflow scenarios hinder a rollback after a system failure significantly.

FEATURE	AUTONOMOUS	EMBEDDED
Installation and Maintenance	Additional System	Additional Component

Updates, Releases	Modification of interfaces and workflow models may be necessary, if invoked applications or workflow management system change	Release security, because system vendor is not dependent on workflow component, change of workflow model may be necessary, if embedding functionality changes
Performance	Starting external application systems creates overhead	Optimization possible, if the workflow application uses the embedding system exclusively
Scalability	Possible through the distribution of system components	Only in conjunction with the embedding system
Recovery	Difficult, due to the autonomy of invoked applications	Possible, if used in a homogeneous environment

Table 1: Embedded and autonomous WfMS from the perspective of system administrators

Enterprise Perspective

Managers need to examine whether the workflow functionality is, and always will be, required to manage functionality within the encompassing business activity, or whether it might be required to perform a variety of tasks in the business. Classic examples of where embedded workflow engines perform well as part of a larger system are Enterprise Resource Planning (ERP) and major manufacturing control (MRP). Here the engine manages the rules and events that fall outside pre-determined tolerances.

Where the workflow engines are required to manage heterogeneous environments, and this is more common, autonomous engines are required. Furthermore, if for some reason the replacement of an existing workflow system is desired, an autonomous workflow engine can be exchanged with significantly less effort than an embedded workflow engine. The latter case requires the restructuring of the workflow application and can only be performed, if the functionality of the embedding system is accessible for an “outside” workflow product.

FEATURE	AUTONOMOUS	EMBEDDED
Applicability	Always	At least workflow-enables system is necessary
Domain	Focus on administrative domain, other domains are following slowly.	Determined on the embedding system.
Lock-in Cost	Workflow component can be replaced without affecting the invoked	Workflow component cannot be exchanged in most cases,

	applications	replacement only through an external system which causes a major restructuring of the workflow application
Coverage of special requirements	Systems for domain specific requirements are available	No freedom of choice
Recommended for	Heterogeneous application landscapes	Long term homogeneous environments

6 ARRIVAL OF THE MULTI-LEVEL WORKFLOW

Having established the segmentation between autonomous and embedded workflow engines, the next stage is to describe the environment as to whether the two types may co-exist in a given scenario.

Large organizations are deploying ERP or MRP systems to drive their operations, and, at the same time, are using separate autonomous engines to drive their sales operations or to assist with problematical actions in accounts operations. For some application areas, custom-made workflow solutions are deployed that implement enterprise-specific functionality.

With all these different systems in place, management still needs to keep an overview over the existing business processes, and the overall customer-to-customer process chains should benefit from the increasing use of workflow technology. Therefore, different workflow solutions need to interoperate in order to streamline business operations and reduce media breaks.

Instead of proprietary interfaces that increase with a speed of $n*(n-1)/2$ (for n systems), standardized interfaces reduce the integration effort for different workflow management systems significantly and increase the investment security for workflow users. Interface standards such as the Workflow Management Coalition Interface 4 Specification[4] and the recent Wf-XML[5] standard help users build individual workflow applications without losing the "big picture."

---

[4] Cf. WfMC (1999).

[5] Cf. WfMC (2000).

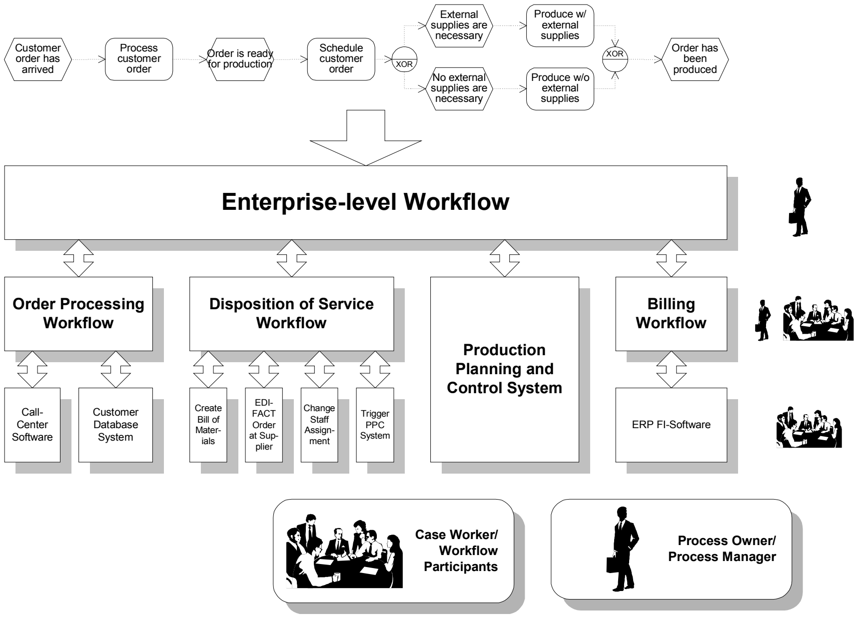

Figure 1: Multi-level workflow scenario (cf. Becker, zur Muehlen (1999))

Figure 1 illustrates the use of workflow technology at different levels of the enterprise. While workflow applications that are either embedded in operational information systems or enhance stand-alone applications have a direct impact on operational tasks, workflow on the department or enterprise level has more of a back-end quality. Still, the use of workflow technology at this level adds value to the company, e.g. through the provision of large-scale audit trail data about workflow execution. This way, process-oriented management information systems can be built on top of an enterprise-wide workflow framework. The benefits of this technology move towards a more managerial level with a larger scope of the application.

7 CONCLUSIONS

Workflow technology is developing rapidly and is increasingly deployed in mission critical applications. Through the integration of workflow technology with emerging technology trends such as enterprise portals, customer relationship management solutions and enterprise-wide controlling solutions, many companies are introducing workflow solutions without the explicit intention of building a workflow application in the first place.

The functionality required of the engine as well as the existing technological infrastructure determines whether an autonomous or embedded engine should be deployed. Both types have advantages and disadvantages in their respective fields; however, non-technical factors need to be taken into account as well when a workflow strategy for a specific enterprise is being developed.

As the requirements for workflow applications become more comprehensive, organizations will be deploying engines of both types. Because these engines need to interoperate, it is essential that they conform with the existing standards (e.g. WfMC) to facilitate the integration of different products. The existing interoperability standards, while originally designed for cross-enterprise processes, may very well be used inside a single enterprise to link different workflow applications across the value chain.

References

Becker, M.; Vogler, P.; Österle, H.: Workflow-Management in integrierter Standardsoftware. Wirtschaftsinformatik 40 (1998) 4, pp. 318-328. [in German]

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

Object Management Group: Workflow Facility. Document Number bom/99-03-01. Framingham 1999.

Workflow Management Coalition: Workflow Standard – Interoperability Abstract Specification. Document Number WFMC-TC-1012. Version 2.0. Winchester 1999.

Workflow Management Coalition: Terminology & Glossary. Version 3.0. Winchester 1999.

Workflow Management Coalition: Workflow Standard – Interoperability Wf-XML-Binding. Document Number WFMC-TC-1023. Version 1.0. Lighthouse Point 2000.

About the Authors

MICHAEL ZUR MUEHLEN

Michael zur Muehlen is working as a lecturer and research assistant at the Department of Information Systems at the University of Muenster, Germany, in the fields of information modeling and workflow management. Mr. zur Muehlen has participated in numerous industrial BPR and workflow projects and has published several articles on the topics of meta modeling, process and workflow management. Together with Prof. Yvonne L. Antonucci, Widener University, USA, Mr. zur Muehlen has received the SAP University Alliance Curriculum Development Award for the establishment of an international curriculum teaching inter-organizational business processes between Germany and the USA. He is a member of the German Computer Society (GI), the Workflow Management Coalition (WfMC) Technical Committee, chairman of the WfMC working group "Resource Model" and acts as the country contact Germany for the WfMC.

ROB ALLEN

Mr. Allen is the director of product marketing at Open Image Systems Inc., acts as the vice chair of AIIM UK and is chairman of the WfMC External Relations Committee.