BLED 2004 Proceedings

BLED Proceedings

December 2004

Integrating Existing Enterprise Systems With Workflow

Patrick Rushe
Cork Institute of Technology

Jeanne Stynes
Cork Institute of Technology

17th Bled eCommerce Conference

eGlobal

Bled, Slovenia, June 21 - 23, 2004

Integrating Existing Enterprise Systems With Workflow[1]

Patrick Rushe, Jeanne Stynes

Cork Institute of Technology, Ireland
prushe@cit.ie, jstynes@cit.ie

Abstract

Reducing costs and reducing time to market are two major keys to survival in the software market. Workflow reduces costs and time dramatically where applications involve the passage of work between recipients in order to meet certain business objectives. New projects in this area often use workflow technology. However, workflow's applicability is often overlooked where developers are working on maintaining or upgrading existing systems. This paper discusses the work involved in integrating an existing system with a workflow management system, and examines the benefits of incorporating workflow into existing systems.

1 Introduction

Workflow is a composite of several technologies and concepts, interwoven together to form a body of knowledge. Workflow Management involves controlling, monitoring, optimizing and supporting business processes (Aalst, 1998). A workflow system is a workflow management system (WfMS) grouped with domain specific applications, databases, resources and data, all of which cooperate to automate business processes (WfMC, 1995).

The term Business Process Modelling (BPM) has been used to incorporate all activities relating to the transformation of knowledge about business systems into models that describe the processes performed by organisations (Scholz-Reiter and Stickel 1996). A resulting model can be converted into a process definition, a representation of the business process in a form which supports automated manipulation, such as future modelling, or enactment by a workflow management system. A process instance is a single enactment of a process (WfMC, 1999).

[1] Research funded by Enterprise Ireland Innovation Partnership Programme

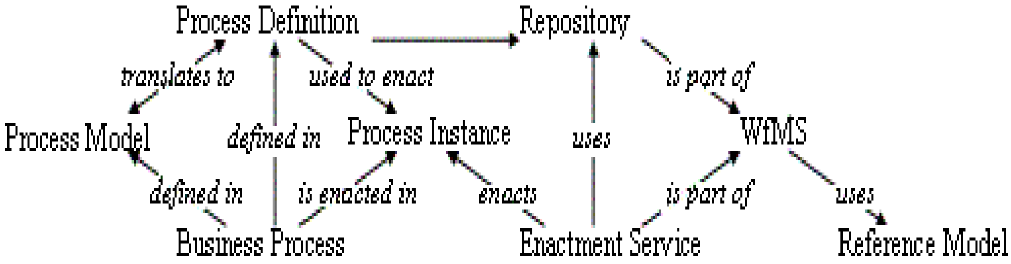

Figure 1: *Ontology Of Workflow*

An organisation forms the context in which process instances will be enacted, and is modelled separately from the process definition. The organisational model is a map of individuals, groups, supervisors, departments etc. Unlike a process definition, the organisation model may be dynamically changed during the execution of a process instance. For example, the model would have to be adjusted to reflect the promotion of a staff member during the enactment of a process.

A WfMS has two critical components: an enactment service to support the execution of process instances, and a repository service to support the storage, access, deletion and update of process definitions. One of the reasons standardisation is so slow in workflow is the infinite complexity of a process enactment. Figure 1 presents the ontology of workflow.

Even though few WfMSs comply fully with the Workflow Management Coalition's reference model (WfMC, 1995) and its related standards, the model serves as a point of reference both to WfMS implementers and academics, playing a role similar to the OSI model in networking. The WfMC reference model defines the four interfaces that a WfMS should implement, namely, a process definition interface allowing management of process definitions, a client and invoked application interface allowing communication with applications that use and are used by the WfMS, an interoperability interface allowing the passage of process instances between separate workflow engines, and an administration interface allowing communication with administrative applications.

Using workflow technology in the development of software systems, where the passage of work between participants is automated can reduce time-to-market and cost. However, there is a tendency to overlook the benefits of applying workflow technology where projects are being upgraded or are undergoing maintenance. The business processes automated by these systems are becoming more important, subject to frequent change, becoming more complex, and increasing in number (Dong and Chen, 2004). Since workflow is specifically designed to cater for this, integration of workflow at any stage of a project can be beneficial.

The key to successful application of workflow is a clear understanding of both the WfMS and the existing system. This paper examines the work involved, and benefits of integrating workflow with existing applications.

2 Integrating Workflow

Integrating an existing system with WfMS involves a considerable amount of work but this can be accomplished in parallel with development. The whole application does not have to be ported across to the workflow system at one time.

2.1 Obtain A WFMS

The first step to integration with workflow technology is to obtain a workflow management system. There are three ways of accomplishing this, namely build, buy or adopt. As always, performance, reliability, cost and scalability will all be factors.

Though rarely considered, building a lightweight WfMS can be a viable option when integrating with an existing product as its interfaces can be tailored to suit the client application, especially in the area of participant management. This can also be a low cost option, as the developers need only implement those features of the WfMS that support the existing application's business process. For example, if the application does not involve parallel processes, it is not necessary to build a WfMS capable of handling them. However, note that designing a new WfMS generically can facilitate its reuse in other projects. Technical support is not an issue as those who build the WfMS can provide in house expertise.

Buying a WfMS is rarely cheap. Installation and licensing fees can be excessive, especially where the WfMS is only going to be used for one project. Also, developers will have to spend some time learning how to use and interface with the system. As commercial systems need to provide a wide array of functionality, they are often very complex systems.

An alternative is to adopt an "open source" system such as JPBM, OpenSymphony, OpenFlow and others. These systems are free, and are built by internet communities. While developers generally recognise the quality of open source products, customers are often reluctant to accept this software due to concerns over quality. Also, most open source projects do not have free technical support; in some cases, it is very expensive to obtain.

2.2 Process Modelling

The ultimate goal of process modelling, which is the next step in integrating with workflow, is to develop a process definition that clearly specifies the business process automated by the existing application. A process definition consists of cases, transitions, states, resources:

- A case is “a product in progress” that is traversing an instance of a process.
- A transition represents an activity; a logical unit of work that will result on the case being moved from one state to another.
- A state (*workflow state*) is a particular point along the process which is the cumulative effect of the sequence of transitions fired so far.
- A resource is a means of completing a transition, or a group of transitions. It may be a person, a machine or an organizational group.
- A process is a network of transitions that allow a case to be completed.

Fig. 2 is a model of a conference room reservation workflow that might form part of a facility management system. The circles represent states, and the squares represent transitions. An arrow from a state leading to a transition indicates that should a case be in that particular state, the transition can fire. An arrow leading from a transition to a state indicates that after the firing of a transition, the case will move to that state.

Fig. 2 is a Petri net, which Van Der Aalst proposes as a suitable media for the expression of business process models (Aalst, 1998), and a subgroup of Petri Nets called workflow-nets or WF-nets is defined for this purpose. Time WF-nets (TWF-nets) are an extension of WF-nets that add the capability to model time constraints (Ling and Schmidt, 2000). Like all graphical models, Petri nets are easy to understand. Since Petri nets use a formal semantics, ambiguities, uncertainties, contradictions are avoided. Their expressiveness means that selective, parallel, iterative and sequential routing can be catered for. Moreover, a formal mathematical basis allows for the application of existing analytical techniques.

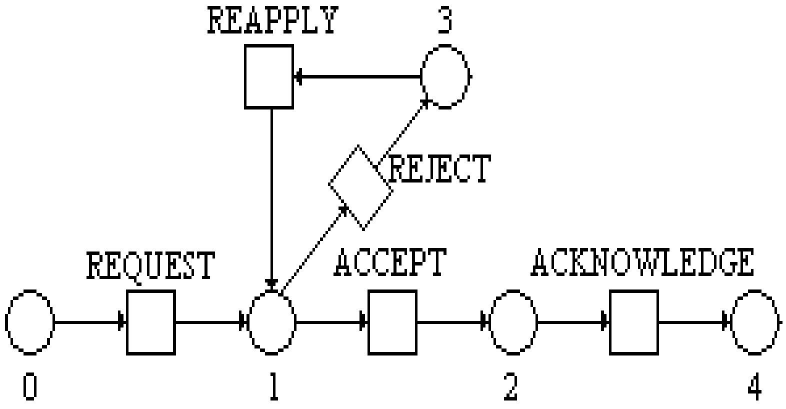

***Figure 2**: Petri Net Of Conference Room Booking*

It can be difficult to identify the atomic activities in an existing system and to pick the correct level of abstraction for the model. If one delves too deep, the model becomes unmanageable and overly complex. For example, an insurance claim process definition could have a transition called "Fill in claim form" which could be decomposed into "Fill in surname", "Fill in first name", "Fill in address", etc. Whether the latter level of abstraction suits the application is questionable if all the details are filled in on one form in the existing application. Hence, the atomic transitions will have to correlate with the atomic activities of the existing application.

The models must then be mapped to a process definition language (PDL), or to some format that the WfMS can understand. PDLs are an XML meta-language for the expression of process definitions. Many PDLs exist, among them XPDL and BPML. It is worth noting that while proponents of certain PDLs claim that they offer the required tool for workflow definition expression, opponents claim that they are lacking in expressiveness, are largely proprietary and lack properly defined semantics(Aalst, 2003).

2.3 Modelling The Organisation

Another aspect of the existing system that must be modelled is the organisation within which the various business processes exist (Klarmann, 2001). Generally, this will be modelled in the existing system already. For example, most enterprise systems will include information about different users and their privileges. This information must be made accessible to the WfMS. When integrating with an existing enterprise system, there are two approaches to making this possible: either provide an interface that allows the WfMS to access the information via the existing application, or offload organisational information into a separate component that is used by both the WfMS and the existing application. The former is fast, the latter allows for the separate component to be generic, facilitating re-use.

The WfMC has published a model specifically for modelling an organisation where the model will be used in a WfMS. The model is based on only four participants, namely humans, roles, organisation units, and resources. The model also takes into account the idea of substitutes, that is participants that can assume the rights of others, and the idea of supervisors, who can assume the position of subordinates.

2.4 Moving Flow Control

After modelling the business process that the existing application implements, it is necessary to remove the flow control from the existing application. Before workflow, the business process was expressed in the code of the existing application. Since it is now expressed in the process definition, it is no longer needed in the code. Removing the flow control can be a tedious process; however it is not necessary to remove it all in one go. Primary business processes can be migrated immediately but supporting processes such as adding a member of staff need not be migrated for some time, if at all. In the later case, no changes are made to the existing application. It is still possible to add a member of staff; the application just does not use the workflow engine to facilitate the process.

Generally, the flow control for a process instance can be recognised in the code of the existing application as a form of status management. For example, a particular insurance claim has a status of “pending approval”. The option to approve the claim is provided by the system to the investigator based on the value of the status variable. This is hard coded into the application. This type of logic must be replaced with calls to the workflow engine. The basic flow using the workflow system would be:

1. Claim investigator logs in to existing application.
2. Existing application calls workflow system and checks what work is available for investigator.
3. Workflow system returns that he can approve the insurance claim, or reject it. This is passed back through the existing application to the investigator.

When, for example, the investigator clicks “Approve”, the WfMS is notified that an “Approve” transition has fired; it then updates the process instance, moving its state to “Approved”.

An activity is generally a composite of several smaller activities, each conducted by individual components. Since each component has to call the next to pass control, components end up being tightly coupled. A benefit of workflow is that it handles flow control so that components no longer need to be linked together. The WfMS will instruct each one to carry out its small activity in sequence, without the components interacting directly with each other. A major benefit of using workflow, and developing components that are workflow driven, is that the components are generally more lightly coupled than traditional components. This is important in reuse, as lightly coupled components have few dependencies thereby reducing the number of components that must be included in any new project where the component is used. This effectively reduces the time-to-market and cost of development, as less components have to be built or adjusted for use.

3 Case Study: Real Time/FM

This section documents an instance where an enterprise application was successfully integrated with a WfMS. The application is Real Time/FM developed by Spokesoft Cork. Real Time/FM is a J2EE facility management product. The original version imposes a generic facility management business process on its customers as a pre-requisite for its use. To provide a service that is more in line with existing procedures within an organisation, a decision was taken to incorporate a WfMS that allows the business process of the application to be specialized for each customer.

To facilitate easy integration, a J2EE WfMS was required. Open source projects lacked technical support and commercial products were excessively expensive. A new WfMS was built specifically for the project. The implementation of the WfMS followed the WfMC Reference Model and associated standards as closely as was practical.

A walk-through of the original Real Time/FM was performed to establish its business process. The information was expressed in a Petri net and then translated into a process definition language, based on XPDL (WFMC, 2002). This is now the default process definition on installation, specialized upon request from customers. Specialisation involves changing the process model to suit a particular customer.

Organisational functionality was already part of Real Time/FM. The decision was made to remove this functionality and transfer it to an independent component to facilitate the component's reuse. An organisation manager component was created, responsible for maintaining and updating the organisational model. Additional functionality was provided, allowing the relationships between individuals and process instances to be mapped. While the back end functionality for organization management was changed, the GUI at the front end was not. A user of the new system would not be aware that Real Time/FM is no longer in control of organisation management.

The next step was to integrate the WfMS with the existing application. A staggered approach to this integration was adopted, allowing the development of user functionality to remain the priority. At first, Real Time/FM was passively integrated with the WfMS. This means that Real Time/FM functions as normal while using the WfMS to duplicate its behavior. The options made available by Real Time/FM and those derived from the business process definition by the WfMS were compared to insure they matched each other. When the business process had been validated, Real Time/FM over time relinquished flow control to the WfMS.

The actual integration of Real-Time/FM with the WfMS involved removing business logic from Real-Time/FM and replacing it with simple calls to the WfMS. Before integration, business logic was generally expressed in complicated code that used a state variable to make decisions. After integration, the code just involves a simple call to the WfMS passing relevant information, for example the person who is logged on. The WfMS then uses this information, and applies it to the process model to make the decision.

On completion of the integration of Real Time/FM with the workflow engine, the original Real Time/FM and the new version were stress tested. A significant difference in performance was not present, but the WfMS did increase the footprint of the application in memory as the process definition had to be cached in order to reduce access time.

The GUI provided to the user when the system is operating using the default business process is the exact same as before integration with the WfMS took place. However, as specialisation of the business process occurs, the GUI dynamically changes to represent the new business process. The actual development of a GUI for a system that incorporates WfMS functionality is dramatically reduced. This is because the GUI code does not have to incorporate business logic to make decisions on what a user can do. Instead, the GUI can simply call the WfMS and it returns a list of tasks that a particular user is allowed to undertake. This list can be represented to the user as a simple menu. Hence, development and maintenance of the GUI for the product was simplified dramatically.

4 Conclusion

This paper presented the ontology of workflow and a summary of the steps necessary to integrate this technology with existing enterprise systems. We documented how an enterprise application, Real time/FM, was successfully integrated with a WfMS.

The new Real Time/FM demonstrated that the use of workflow facilitates reuse, thus reducing time-to-market and costs. The product is now more configurable as the supported business process can be dynamically adjusted without affecting the application code. One beneficial side-effect of using workflow technology was the discovery of the requirement for a far more complex organisational model than was originally provided. This was incorporated in the new component designed for organisational management. Even when workflow technology can not be fully integrated, insights can be gained that enable designers to develop more refined and better structured applications.

Distributed technologies such as the Common Object Request Broker Architecture (OMG, 1999), and Java Remote Method Invocation (Sun, 1998) greatly enhance the viability of WfMSs. They allow workflow systems to be spread over disparate locations, easing performance problems. They simplify communication between clients, invoked applications and WfMSs. They support distributed workflows: multiple WfMSs interoperating to execute a single process instance. Using the J2EE or .NET platforms, WfMS implementers can offer communication over any number of protocols.

References

Aalst, W.M.P. van der, (2003): Don’t go with the flow: Web services composition standards exposed, IEEE Intelligent Systems, Vol. 18, No. 1, pp. 72 - 76.

Aalst, W.M.P. van der, (1998): The application of Petri-nets to Workflow Management, Journal of Circuits, Systems and Computers, Vol. 8, No. 1, pp. 21 - 66.

Dong, M., Chen, F.F. (2004): Petri net-based workflow modelling and analysis of the integrated manufacturing business process, International Journal of Advanced Manufacturing Technology, Volume Online, Springer-Verlag, London.

Klarmann, J., (2001): Using conceptual graphs for organisational modelling in workflow management systems, “In: Proceedings Professionelles Wissensmanagement ”, Baden-Baden, March 2001

Ling, S., Schmidt, H. (2000): Time Petri nets for Workflow Modelling and Analyses, IEEE International Conference on Systems, Man and Cybernetics. October 2000 Vol. 5. pp. 8-10

Object Management Group (1999): The Common Object Request Broker: Architecture and Specification, Minor revision 2.3.1, OMG TC Document formal/99-10-07

Scholz-Reiter, B. and Stickel, E. (1996): “Business Process Modelling”, Springer, Berlin, 1996, pp. 313-332

Sun (1998): Java Remote Method Invocation Specification Retrieved September 18, 2003, at http://java.sun.com/j2se/1.4.2/docs/guide/rmi/spec/rmiTOC.html

Tsalgatidou, A. & Junginger, S. (1995): Modelling in the Re-engineering Process. ACM SIGOIS Bulletin: Special Issue on Business Process Reengineering, Vol. 16, No. 1, August 1995, pp. 17-24.

Workflow Management Coalition (1998): Interface 1: Process Definition Interface Organisational Model, Retrieved April 14, 2003, from http://www.wfmc.org/standards/docs/if19807o.pdf

Workflow Management Coalition (1995): WfMC Reference Model. Retrieved September 10, 2003, from http://www.wfmc.org/standards/docs/tc003v11.pdf

Workflow Management Coalition (2002): XML Process Definition Language Specification Retrieved September 10, 2003, from http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf