

Integration of BPM Systems

Chaoying Ma, Andrea Caldera, Miltos Petridis,
Liz Bacon and Gill Windall
*CMS, Greenwich University, Old Royal naval College, Park Row, London
United Kingdom*

1. Introduction

New technologies have emerged to support the global economy where for instance suppliers, manufactures and retailers are working together in order to minimise the cost and maximise efficiency. One of the technologies that has become a buzz word for many businesses is business process management or BPM. A business process comprises activities and tasks, the resources required to perform each task, and the business rules linking these activities and tasks. The tasks may be performed by human and/or machine actors. Workflow provides a way of describing the order of execution and the dependent relationships between the constituting activities of short or long running processes. Workflow allows businesses to capture not only the information but also the processes that transform the information - the process asset (Koulopoulos, T. M., 1995). Applications which involve automated, human-centric and collaborative processes across organisations are inherently different from one organisation to another. Even within the same organisation but over time, applications are adapted as ongoing change to the business processes is seen as the norm in today's dynamic business environment. The major difference lies in the specifics of business processes which are changing rapidly in order to match the way in which businesses operate. In this chapter we introduce and discuss Business Process Management (BPM) with a focus on the integration of heterogeneous BPM systems across multiple organisations. We identify the problems and the main challenges not only with regards to technologies but also in the social and cultural context. We also discuss the issues that have arisen in our bid to find the solutions.

2. Background

2.1 The need for integration at different stages

There has been an increasing demand from businesses in different geographical locations to be able set up and share processes such as a number of supplier-chain processes required by many major companies. eLearning and the concept of a virtual university has also become a popular topic for consideration today and it is this example that will be used to explain the need for integration at different stages of BPM. A Networked Virtual University is formed by a number of participating universities of different countries to provide a coherent set of courses. The idea is that students from all over the world would be able to register to study courses. Academics from these universities would need to work together through a shared

process such as exam paper setting, coursework marking and so on. One of the main challenges in setting up and managing such processes is to cater for the needs of the individuals in the different organisations involved. One would end up having to use a good array of tools and platforms just to follow a shared business process such as coursework marking initiated by another university. Most of the tools currently available do not recognise the fact that users of different organisations involved in a shared process are often using a different set of tools for modelling, designing and interacting with their processes. For instance, in a Networked Virtual University (NVU), where several universities partner to provide a number of coherent study programmes through a combination of online and traditional means, a unit coordinator of a programme responsible for setting an exam paper would sometimes be required to work with an external examiner for the purpose of quality control. This would require the creation of a cross-domain business process that automatically coordinates the activities carried out by the internal and external parties, monitors the events as activities complete, notifies and/or alerts the interested parties by sending reminders and/or taking escalation actions. Suppose that each institution had a BPM (Business Process Management) system to start with, it is unlikely that they could create and then interact with a system using tools familiar to all parties. If there is a dominant party (i.e. whose business objectives will be satisfied by finishing the process), it's more likely that their BPM system would be used but the other parties will have to adapt to a "foreign" practice, if this is possible, e.g. through a web application interface.

When organisations are working on workflows that cross their organisational boundaries they are likely to need to collaborate at three stages (Fig. 1)

The complexity of each stage is significantly increased by the involvement of multiple participants. The sections below examine each of these stages in more detail.

Stage 1 - Understand and model the workflows

In order to come to a shared understanding of the workflows the participating organisations need to create a model that is understood and agreed by all participants. This will normally involve the use of some diagrammatic modelling notation created using a modelling tool. BPMN is popular as a modelling notation but not every organisation uses it. Some may use simpler generic models such as UML Activity Diagram or alternative BPM modelling notations such as Event-driven process chains (Van der Aalst 1999). Even if all the participants use the same modelling notation they may not use the same modelling tools which gives rise to the need to exchange models between tools.

Stage 2 - Create an executable representation of the workflows for use by a workflow engine
For a workflow model to be automated it needs to be converted into an executable form. Some modelling tools make this very easy whereas with others there is a need to carry out a translation. If the workflows are to be executed in several workflow engines belonging to different participants then there may be the problem of translating the model into several different executable formats suitable for the variety of engines.

Stage 3 - Interact with running workflow instances

When workflows are automated by a workflow engine there is obviously a need for people at the various collaborating organisations to interact with them. This can be the most problematic step. Different organisations may interact with workflows in different ways. For instance one organisation may use a push approach where tasks requiring action are presented to the user in a in-tray or via email whereas another organisation may use a pull approach where the user occasionally checks to see if anything requires their attention. Tasks will be carried out using different applications. For instance in a networked

University one partner may record marks using a spreadsheet whereas another may use a database. Organisations may or may not have their own workflow engines.

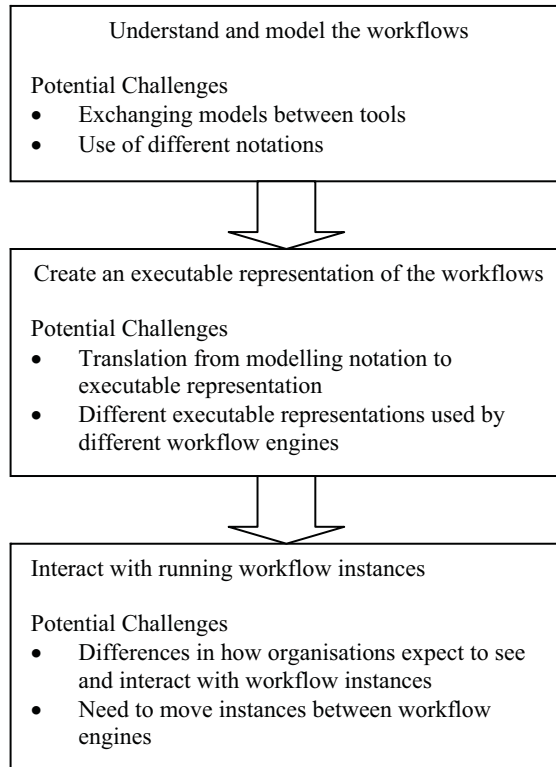


Fig. 1. Potential challenges at each stage of workflow integration

Stage 1 - Understand and model the workflows

Stage 2 - Create an executable representation of the workflows for use by a workflow engine

Stage 3 - Interact with running workflow instances

2.2 Motivation and problem statement

In the UK, higher education institutions often form partnerships to provide the same courses to students residing in the various institutions. In recent years, collaboration with overseas universities/colleges has grown rapidly and in many institutions, workflow automation is now common place. Problems arise when the partner institutions share a process in the collaboration but the partners in the collaboration may change from time to time, such that one partner may decide to leave or a new one may join. For instance a workflow for marking coursework may involve the students submitting the coursework online, the tutors mark them locally at each institution. The marked courseworks are then fed to a moderation procedure in order to ensure the quality of the marking. A lead institution will set up the workflow that all institutions will follow. A number of ways to set up and follow through such a process are possible. One is to ask everyone to follow the process created and maintained by the lead institution. A good reason for doing so is that

the objectives of the workflow are defined by the lead institution, and are imposed as requirements to the partner institutions. Another way to set up a shared process may require the use of more than one type of BPM system run by the different institutions. In this case, a shared process has to be agreed by all involved, and the setting up becomes more complex. The former would normally require a Web-based interface for the remote partners. In both cases problems can arise when the staff and students use the system because they are required to deal with an array of unfamiliar interfaces due to the fact that the shared process may bring with it various interfaces and indeed different practices. The research challenge lies in how to find a way to alleviate the users of the BPM systems at all the participating institutions from the burden of having to learn and deal with new interfaces and practices. The less the users have to learn new things unnecessarily the more productive they become. It is also a challenge to the IT department in each of the institutions if setting these processes up requires a lot of work such as project planning, design and programming. In conclusion, a cross-domain business process often comprises activities and/or tasks to be carried out by people and/or system functions residing in different organisations. The main concern about the integration of BPM systems is how to get the workflow engines from the different organisations to work together towards shared business objectives. BPM system integration, however, often stops short once these engines can work together technically. Cultural factors, ROI and user proficiency issues are often not considered. The latter however is vital to the success of such integration. In this research we assume that each organisation defines the business objectives at a high level in terms of use cases and user interactions. They also use their preferred BPM systems and tools for process modelling, design and execution. The employees of these organisations have preferences for the tools they use and different proficiency levels in using them. Culturally speaking, people prefer to use the tools they are already familiar with, and are generally reluctant to adapt to different tools; it may not always be possible to do so in some cases. Furthermore, organisations would generally want to preserve their investment of the various tools that they acquired over time. How can we make integration of information and process from the different organisations easier, i.e. at a higher level of abstraction rather than through lower level programming? This research aims to survey the existing work, the current technologies and related standards, and explore the possible solutions.

3. Existing work and standards of BPM

3.1 Review of BPM Technologies and standards

Business Process Management covers the complete life cycle of process design, modelling, deployment, execution, management, monitoring, optimisation, error prevention etc. in its attempt to automate a sequence of system and human activities required to complete a business process e.g. registering a student to study for a course at a university. Its aim is to enhance and make existing processes more efficient and to design new processes where appropriate. In the early days software tended to be fairly inflexible however this is improving slowly and the industry is now adopting standards to support the different parts of the lifecycle including the human side of the workflow process.

BPM starts with modelling the business domain, capturing workflows and activities in order to analyse and optimise them. This requires an analyst to check for unnecessary manual steps, processes that can be carried out in parallel, establishing responsibilities, removing duplicate effort e.g. entry of data into dual systems etc. The models are typically analysed

and implemented through a BPM suite's designer, typically a graphical development tool which puts human tasks, system functions, and business rules together to create an executable solution. This is then deployed into a BPM engine for execution which may trigger real-time notifications of activities, workflow progress and alerts e.g. if a process has failed for some reason.

Numerous tools and standards from various bodies such as W3C (W3C 2009) and OASIS (Oasis 1993) have been developed to support this process and in this section we will provide an overview of the current technologies and standards used in Business Process Management.

3.2 Service Oriented Architectures (SOA)

It would not be appropriate to comment on BPM without also talking about SOA (Service Oriented Architectures) due to the close coupling between the two and its dominance in industry today. Service oriented architectures have been around for a long time however, when referring to them these days, they imply the implementation of systems using web services technology. A web service is a standard approach to making a reusable component (a piece of software functionality) available and accessible across the web and can be thought of as a repeatable business task such as checking a credit balance, determining if a product is available or booking a holiday. Web services are typically the way in which a business process is implemented. BPM is about providing a workflow layer to orchestrate the web services. It provides the context to SOA essentially managing the dynamic execution of services and allows business users to interact with them as appropriate.

SOA can be thought of as an architectural style which formally separates services (the business functionality) from the consumers (other business systems). Separation is achieved through a service contract between the consumer and producer of the service. This contract should address issues such as availability, version control, security, performance etc. Having said this many web services are freely available over the internet but use of them is risky without a service level agreement as they may not exist in future however, this may not be an issue if similar alternate web services are available for use. In addition to a service contract there must be a way for providers to publish service contracts and for consumers to locate service contracts. These typically occur through standards such as the Universal Description, Discovery and Integration (UDDI 1993) which is an XML (XML 2003) based markup language from W3C that enables businesses to publish details of services available on the internet. The Web Services Description Language (WSDL 2007) provides a way of describing web services in an XML format. Note that WSDL tells you how to interact with the web service but says nothing about how it actually works behind the interface. The standard for communication is via SOAP (Simple Object Access Protocol) (SOAP 2007) which is a specification for exchanging information in web services. These standards are not described in detail here as information about them is commonly available so the reader is referred elsewhere for further information. The important issue to understand about SOA in this context, is that it separates the contract from the implementation of that contract thus producing an architecture which is loosely coupled resulting in easily reconfigurable systems, which can adapt to changes in business processes easily.

There has been a convergence in recent times towards integrating various approaches such as SOA with SaaS (Software as a Service) (Bennett et al., 2000) and the Web with much talk about Web Oriented Architectures (WOA) [ref]. This approach extends SOA to web-based applications in order allow businesses to open up relevant parts of their IT systems to customers, vendors etc. as appropriate. This has now become a necessity in order to address

competitive advantage. WOA (Hinchcliffe 2006) is often considered to be a light-weight version of SOA using RESTful Web services, open APIs and integration approaches such as mashups.

In order to manage the lifecycle of business processes in an SOA architecture, software is needed that will enable you to, for example: expose services without the need for programming, compose services from other services, deploy services on any platform (hardware and operating system), maintain security and usage policies, orchestrate services i.e. centrally coordinate the invocation of multiple web services, automatically generate the WSDL; provide a graphical design tool, a distributable runtime engine and service monitoring capabilities, have the ability to graphically design transformations to and from non-XML formats. These are all typical functions provided by SOA middleware along with a runtime environment which should include e.g. event detection, service hosting, intelligent routing, message transformation processing, security capabilities, synchronous and asynchronous message delivery. Often these functions will be divided into several products. An enterprise service bus (ESB) is typically at the core of a SOA tool providing an event-driven, standards based messaging engine.

3.3 BPEL and associated standards

Individual services must be composed into a sequence of steps i.e. the workflow, failure and exceptional cases associated with each service must be dealt with. Note that the latter may require one or more service activities to be 'undone' and control passed back to the user. Workflow design involves analysing existing or planned business processes to understand the different stages of these processes; representing the process being designed in a workflow design notation and converting the final design into an executable program. This could be done by writing in a language such as Java or C# however the standard that has emerged to do this is WS-BPEL (Web Services Business Process Execution Language), known as BPEL (2006) for short and is an OASIS standard for specifying how web services interact. BPEL is a XML-based language which Business analysts can use to specify the orchestration of the web services for execution by BPEL. The BPEL process produced is itself a web service, and exposes itself via a WSDL description. A BPEL engine can then execute the process description. Fig. 2 shows a possible BPEL representation of the exam setting case study. Each human task has to be wrapped in web services and invoked as a web service.

BPEL has three basic components to it

- The programming logic i.e. the BPEL execution code. This contains commands to e.g. invoke a web service, reply to a message received, execute activities in sequence or in parallel in addition to standard programming constructs of loops, selection and variable assignment.
- Data types which are defined using XSD (XML Schema 2006) which is W3C standard for describing the structure of an XML Schema Document.
- Input/Output (I/O) which is achieved through the WSDL (Web Services Description Language) standard.

WS-BPEL has a number of other associated standards, too many to list here but key standards or emerging standards are:

- WS-CDL (Choreography Description Language) (WS-CDL 2007). This is currently a candidate standard for W3C and is an XML-based language that describes the semantics of peer-to-peer collaborations of Web Services across organisations. It is

generally used to provide common rules about how companies will participate within a collaboration across multiple organisations.

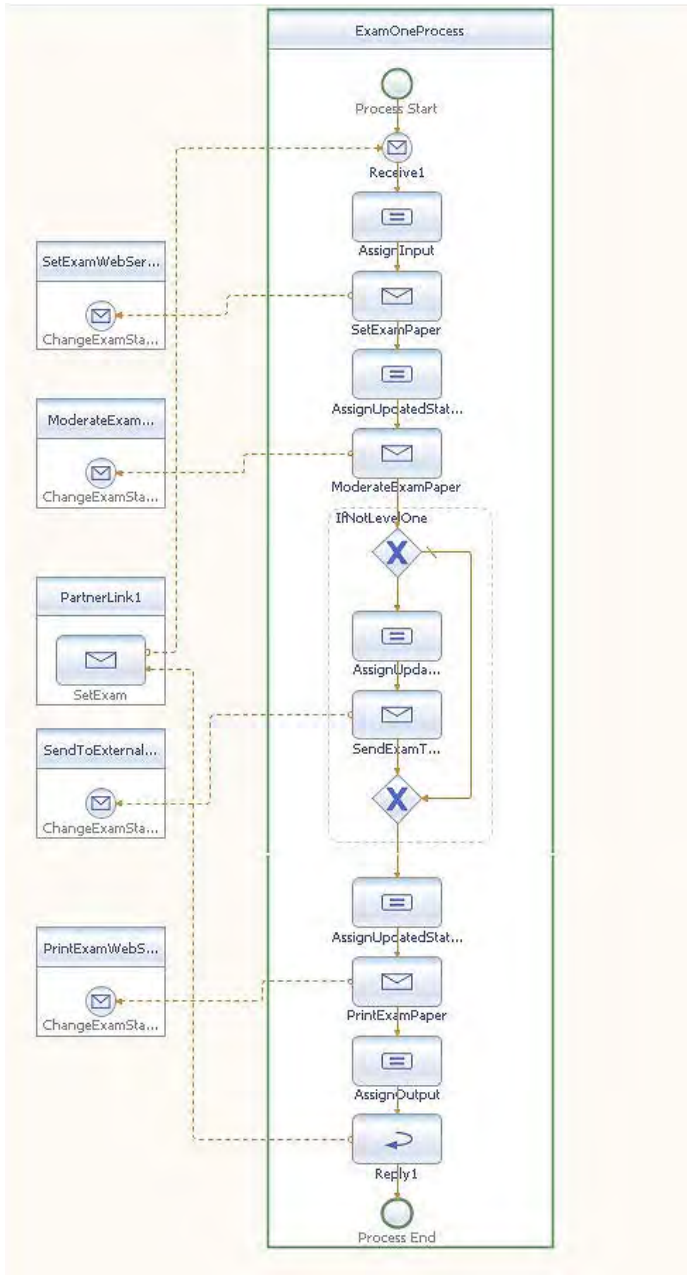


Fig. 2. A BPEL representation of an exam setting process

- WS-Coordination is an OASIS standard (WS-Coordination 2009) and is an extensible framework for coordinating activities and providing protocols that coordinate the actions of distributed applications. For example it might be used in several applications spanning multiple organisations that need to reach an agreement on the outcome of a distributed transaction.
- BPEL4People is the WS-BPEL Extension for People standard by OASIS [BPEL4People 2005]. The WS-BPEL standard only deals with web services, it does not address the need for human interaction in workflow which many real-world business processes require in order to complete. The BPEL4People standard is an extension of WS-BPEL to include interaction with humans.

There are many other standards, particularly from OASIS which cover areas such as transactions, security, trust etc. however their details are beyond the scope of this chapter and the reader is referred to the standards bodies websites for further details.

3.4 Business Process Modelling Notation (BPMN)

BPMN is a business process language and is an OMG (BPMN 2006) standard. It essentially provides a graphical notation to help visualise WS-BPEL code and is designed to be understood by business users and technical developers alike, thus trying to help bridge the communication divide between the two that is so often talked about. The standard specifies a BPD (Business Process Diagram) which is based on a flowcharting technique and contains the following key elements:

- Flow Objects – these include Events (something that happens), Activities (a process that is done) and Gateways which show how paths merge and fork.
- Connecting Objects – include sequence Flow (which show the order in which activities will be performed), message flows (which show the messages that flow across organisational boundaries), and associations (these associate an Artifact or text to a Flow Object).
- Swimlanes are used to help partition and organise activities. A “Pool” represents a participant in a process and a “Lane” is used to organise activities within a pool.
- Artifacts (Artefacts) are essentially additional information you might need to show on the diagram. There are three types of artifacts: a Data Object which could, for example, be the data associated with a message e.g. an invoice, Groups which are often used to highlight particular logical areas of the diagram e.g. the parts of a diagram associated with the registration of a student at University; and Text Annotations which allow additional detail to be provided on the diagram where appropriate.

It should be noted that BPMN is significantly different to other common standards for providing views on a system such as UML. Whilst they both provide a graphical notation of business processes in some form, UML takes an object-oriented view of the system and is not easily understandable by business users whereas BPMN takes a process-oriented view of the system and is more intuitively understood without formal training in the notation. The two approaches are not in competition, they serve different purposes, providing different views of the system processes and are complementary.

3.5 XML Process Definition Language (XPDL)

XPDL is a Workflow Management Coalition (XPDL 2005) standard and defines how Business Process Definitions can be exchanged between different workflow products and

engines. It is designed to enable the complete reconstruction of a workflow diagram including all the semantics, the X and Y coordinates of the diagram elements and how the nodes are linked etc. XPDL is currently the best format for exchanging BPNM diagrams. Note that BPEL does not contain graphical information its focus is purely on the execution of the processes hence the application to BPNM not BPEL.

3.6 Business Process Definition Metamodel (BPDM)

BPDM is a standard from the OMG (BPDM 2008) which extends BPMN and BPEL to support the exchange of business processes definitions between tools and execution environments. It is similar to XPDL for process interchange but offers some additional features to try and provide a common metamodel to unify many existing notations to business process definition notations. It also aims integrate the approach with UML and other industry standards to provide a consistent and complete approach to lifecycle development. OMG aims to reconcile BPMN and BPDM into a consistent language.

3.7 Wf-XML

Wf-XML is a BPM standard (Wf-XML 2006) developed by the Workflow Management Coalition and is an extension to the OASIS Asynchronous Service Access Protocol (ASAP 2005) which is itself an extension to SOAP. ASAP provides support for starting and monitoring services in other workflow engines that might take a long time to complete. Wf-XML extends this functionality providing additional web service operations to send and retrieve process definitions of a service thus providing a standard way for design tools and execution engines to communicate. Wf-XML 2.0 is defined using WSDL and is therefore provided as a web service.

3.8 BPM engines

There are many BPM engines on the market e.g. eClarus software (eClarus 2009) and Singularity's Business Process Management (BPM) Suite [Singularity 2009]. Most, if not all, use the standards described above as core in their delivery. The standards for BPM execution, management, information interchange etc. are however constantly evolving and being regularly updated. For example, the proposed standards for open services grid architecture (OGSA2006) from the Open Grid Forum are based on existing standards although not a standard itself at the time of writing.

BPM engines can be stand alone products or embedded into other products such as Oracle's Business Process Management solution (Oracle 2009). It should be noted that BPM tools are not often a solution in themselves; they are generally only part of a solution. Many other tools and techniques are needed to help solve the complexity of managing workflow which often requires human input and the need to address issues across organisational boundaries. For example, Business Analysis tools (BPA), Business Activity monitoring (BAM), Business rules engines (BRE) and Business process management suites (BPMS) which according to Gartner will be among the fastest growing software markets in 2011 (Gartner 2009).

3.9 Review of related research

Jung et al., (2006) propose an approach to BPM integration which they refer to as "multi-phase process composition". The essence of this approach is that organisations have private workflows which are linked at various points into a collaborative workflow. This approach

can be implemented in various ways. Jung et al. have developed a prototype where each partner organisation has its own internal workflow or BPM engine running its private processes. In addition there is a BPM engine that runs a shared process which co-ordinates the collaborative workflow. XPDL is used to describe the private workflows and BPEL for the shared collaborative workflow. Communication between the engines is via Wf-XML.

There are several advantages to the approach proposed by Jung et al. Organisations are able to use their familiar workflow engines and retain control of their private workflows. If there are three collaborating partners then each only has to link into the shared BPM engine rather than link to each of the other partners' engines. As the number of partners grows (as may be the case in a virtual enterprise) this provides a significant advantage in terms of maintainability. A disadvantage of the approach is that it is quite complex to set up. Also, in order to see the status of a workflow it may be necessary to interact with several workflow engines.

Meng et al., (2006) describes Dynamic Workflow Model (DWM) and an implementation Dynaflow that is based on DWM. It tries to solve problems in virtual enterprises which are similar to those in virtual university: namely how to model and manage inter-organisation workflow for businesses that need to be more agile and flexible, and to maximise the use of their existing resources. DWM provides support for creating and running dynamic workflows across organisational boundaries. DWM extends the WfMC's WPDL with new modelling constructs such as connectors, and events, triggers and rules. It also encapsulates activities definitions and allows web service requests to be part of the activity specification. DynaFlow then makes use of Event-Triggers-Rules (ETR) server to trigger business rules during an enactment of a process thus rules are enforced or the process model is modified at run-time. Their approach is based on web services enhanced with asynchronous events. It combines workflow engine and middleware services to form an enterprise infrastructure. The main drawback of the work is that each organisation will have to install the same set of servers including the event server, the ETR server, and the Workflow server and so on, in order to manage inter-organisational processes.

3.10 Some related work in progress

Most current approaches to the computer assisted management of business activities focus on the automation of tasks in a way that the computer systems assist and direct the business processes according to the predefined business process definition. Work done by the Artificial Intelligence (AI) group at the University of Greenwich (Kapetanakis 2009) investigates the use of AI for the intelligent monitoring of business activities. The approach proposed is inspired by the way human managers interact with and manipulate processes in an agile way to deal with unforeseen circumstances and with the uncertainty stemming from the limitations that most systems have in capturing every detail of business workflows, especially at the interface level between systems and human actors.

Modern enterprise systems are able to separate the definition of workflow based business processes from the software implementing the operation of these workflows, offering much more flexibility and agility than was possible in older systems. This allows enterprise computer systems to monitor and control business processes and workflows within an organisation. Additionally, this allows for the agile change of workflows to adapt to the changing business needs of an organisation.

Case Based Reasoning (CBR) (Kolodner 1993) has been proposed as a natural approach to the recall, reuse and adaptation of workflows and knowledge associated with their

structure. Minor et al., [Minor 2007] proposed a CBR approach to the reuse and adaptation of agile workflows based on a graph representation of workflows and structural similarity measures. The definition of similarity measures for structured representations of cases in CBR has been proposed (Bunke 1994) and applied to many real life applications requiring reuse of domain knowledge associated with rich structure based cases (Mileman 2002; Wolf 2008).

A key issue associated with the monitoring and control of workflows is that these are very often adapted and overridden to deal with unanticipated problems and changes in the operating environment. This is particularly the case in the aspects of workflows that directly interact with human roles. Most business process management systems have override options allowing managers to bypass or adapt workflows to deal with operational problems and priorities. Additionally, workflows are liable to change as the business requirements change and in many case workflows involving processes from different parts of an organisation, or between collaborating organisations can “tangle”, requiring the need for synchronisation and mutual adaptation to allow for compatible synergy.

The flexibility and adaptability of workflows provides challenges in the effective monitoring of a business process. Typically, workflow management systems provide outputs in terms of event logs of actions occurring during the execution of a workflow. These could refer to an action (such as a sign-off action or uploading a document), or a communication (such as a transaction initiation or email being initiated and sent). The challenge in monitoring workflows using event information is that even where the workflow structure is well defined and understood, the trace of events/actions does not usually contain the context behind any decisions that caused these events/actions to occur. Additionally, there are often a lot of contextual information and communications that are not captured by the system. For example, some actions can be performed manually and informal communications/meetings between workflow workers may not be captured by the system. Knowledge of the workflow structure and orchestration of workflows does not necessarily define uniquely the choreography and operation of the workflows. The effective monitoring of workflows is therefore required to deal with uncertainty stemming from these issues (Kapetanakis 2009).

The overall exam moderation workflow process is formally defined and constrained by the system operation as seen in Fig. 3. There are also some limited facilities for manual override by the system administrator. However, the overall process in conjunction with the actions and communications audit trail do not uniquely explain the exact cause of individual actions and cannot predict reliably what the next event/action will be and when this is likely to occur. Most of the uncertainty stems from the problem that a significant part of the workflow occurs in isolation from the system. The system does not capture all of the contextual knowledge associated with workflows. A lot of the communications between workflow stakeholders can occur outside the system e.g. direct emails, physical meetings and phone calls adding to the uncertainty associated with past or anticipated events and the clear definition of the current state.

Discussions with workflow monitoring managers showed that patterns of events indicated, but did not define uniquely, the current context and state of a workflow. Managers were able to guess from looking at the workflow events and communications audit what the context and current state of a workflow was and point to possible problems. Most problems occur due to human misunderstanding of the current state and confusion with roles and responsibilities and usually result in the workflow stalling. Managers will then try to restart

the process by adding comments to the system, or initiate new actions and communications. However, this depends on managers realising that such a problem has occurred.

A typical problem series of event could be one where a stakeholder has missed reading an email requiring an action. In that case, the workflow would stall until a manager or another stakeholder spots the problem and produces a manual action (such as sending an email) to get the workflow moving again. For example, using our assessment scenario, a module coordinator upload notification may have been missed by a moderator who would then not read the new version and either approve or try to amend by a new upload as s/he needs to do. In that case, the coordinator may take no further action and other stakeholders will not act expecting an action from the moderator to occur.

The CBR Workflow Monitoring System

The aim of the CBR Workflow Intelligent Monitoring System (CBR-WIMS) is to provide an automatic monitoring system that will notify managers and stakeholders of potential problems with the workflow and provide advice on actions that can remedy a perceived problem.

The monitoring system is designed to work based on experience of past event/action temporal sequences and the associated contextual knowledge and classification in a Case-Based Reasoning system. Similarity measures allow the retrieval of close matches and their associated workflow knowledge. This allows the classification of a sequence as a particular type of problem that needs to be reported to the monitoring system. Additionally, it is intended that any associated knowledge or plan of action can be retrieved, adapted and reused in terms of a recommendation for remedial action on the workflow.

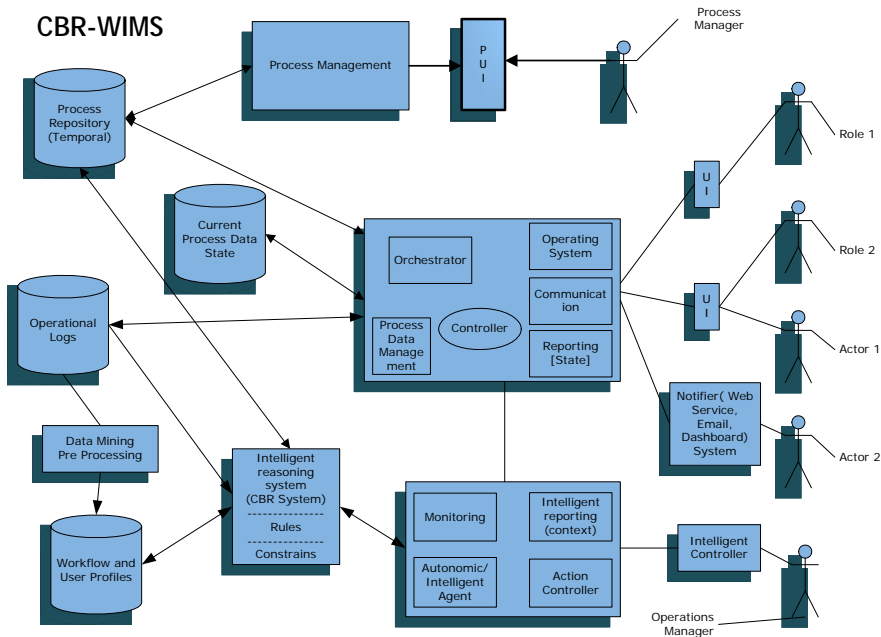


Fig. 3. The Intelligent Workflow Management System Architecture

The CBR monitoring system uses similarity measures based on a linear graph representation of temporal events in a workflow normalized by experience from past behaviour on individual user workflow participation patterns (Kapetanakis 2009)

The Architecture of the Workflow Intelligent Monitoring System

CBR-WIMS is an Intelligent Workflow Monitoring System incorporating a CBR component. The role of the system is to assist the transparent management of workflows in a business process and to orchestrate, choreograph, operate, monitor and adapt the workflows to meet changing business processes and unanticipated operational problems and inconsistencies. Fig. 11 shows the overall architecture and components of CBR-WIMS. The system allows process managers to create, modify and adapt workflows to suit the changing business needs, and/or to allow for variations related to special business requirements. Workflow descriptions are stored in a temporal repository and can be used for looking up past business processes and to provide historical context for past event logs of operations.

The main part of the system controls the operation of the workflows. It responds to actions of various actors to the system and communicates messages about the operation of the system to them. The control system has a workflow orchestrator component that looks up the current workflow definition and orchestrates responses by invoking specific Web Services. The control component also manages and updates the data stored and current state of the workflow operation and provides an event audit log of the key events and actions that occur within the operation of the workflow.

The workflow monitoring and intervention controller monitors, reports, and proposes possible remedial actions to the workflow operation manager. The monitoring system uses a CBR system to retrieve past useful experience about workflow problems occurred in the past by retrieving similar sequences of events/actions in the events log for a given workflow (or workflow part) compared to the current state and recent sequence of events/actions in the operation of the workflow. If a fault or possible problem pattern is detected, this is reported to the workflow operations manager together with the retrieved similar cases and associated recorded experience of any known remedy/course of action.

In the CBR system, workflow execution traces are represented as an event log. So, a workflow event log audit trace is represented as:

(Action1, Actor1, Interval1, Action2, Actor2, Interval2, Action3, Actor3, Interval3)

An example of this would be (intervals are in days):

(CoordUpload, John, 3, ModUpload, Phil, 0, CoordUpload, John, 5)

Similarity metrics between events are defined as:

$$\sigma(e_1, e_2) = \begin{cases} 1, & \text{if } e_1 = e_2 \\ 0 < \sigma < 1 & \text{if } e_1 \text{ is similar to } e_2 \\ 0, & \text{if } e_1 \neq e_2 \end{cases}$$

The overall similarity between two workflow traces (cases) is calculated cumulatively over the minimum common subgraph between the two traces. For each new (unknown) target case, the n nearest neighbours are found using the KNN algorithm. The classification of the nearest neighbours is used to classify the new (unknown) target case (Kapetanakis 2009).

In order to deal with the uncertain and contextual dimension of workflow similarity, the CBR system relies on knowledge discovered from past cases about workflow norms and user profiles created by statistical and data mining pre-processing. The pre-processing component analyses operational logs and attempts to discover knowledge about norms and patterns of operation that can be used in the calculation of the similarity measures for the CBR process. This is particularly important for the monitoring process as any “interesting” or “abnormal” states need to be seen in the context of what has been normal or abnormal behaviour in past event sequence cases.

The Intelligent monitoring part of the CBR-WIMS system has been implemented into the system. Preliminary evaluation has shown that an intelligent workflow monitoring system using and past experience can help workflow managers to monitor complex business processes in an agile way (Kapetanakis 2009).

3.11 Summary

In summary the mainstream BPM solutions have provisions for business process modelling, design, enactment, execution and monitoring but do not address directly the problem described in Section 2.2. It is also clear that the existing approaches are mostly based on the web services technology and allow some level of application integration but fall short in providing satisfactory solutions to the problem. The work in progress is not directly on BPM integration but provides an interesting angle of looking at BAM which could be included into future work. A new approach is required and has been developed. In the following sections we describe a framework for BPM integration.

4. A case study from the Networked Virtual University (NVU)

Before we present the framework for BPM integration, we describe in detail a case study which is used to illustrate why such a framework is needed and how it would help.

4.1 The NVU project

The mENU project was an EC funded project that started in 2002 involving 11 partner institutions from 7 European countries (Hjeltnes & Mikalsen 2003). Its aim was to create a model for a European Networked Virtual University. The model proposed a management structure and quality assurance system spanning the partner institutions. Examples of joint courses and study programmes across institutional and national borders were also developed.

The core concept of mENU is to link universities in a network. Each individual university is able to offer courses from partner universities as part of their programmes. The partner offering the course would carry out some adaptation to make the course useable within their context e.g. by translating material and adjusting the method of assessment.

4.2 A Process for setting exam papers

Workflows are implementations of business processes. Once a business process is modelled, it can be instantiated with whom, what and when: i.e. who are the participants of the process; what is to be carried out in it; and when does it start and finish. In this section we use an exam paper setting process to illustrate a workflow. In Fig. 4 the main objectives of an exam paper setting application are depicted using the UML use case diagram. It shows the main roles played by the different participants (called actors and depicted as stickmen in UML), namely:

- the unit (module) coordinator - responsible for delivering the unit and setting the exam paper in terms of the expected learning outcomes;
- the unit moderator – responsible for approving the paper in terms of the teaching and the learning outcomes;
- the drafter – responsible for approving that the paper meets quality assurance regulations of the programme set by the university;
- the external examiner – responsible for approving that the paper meets quality assurance regulations of the programme set across the universities;
- senior member of staff – responsible for reconciling any unresolved issues by the moderator/external examiner/drafter;
- admin staff – responsible for preparing the paper for final printing

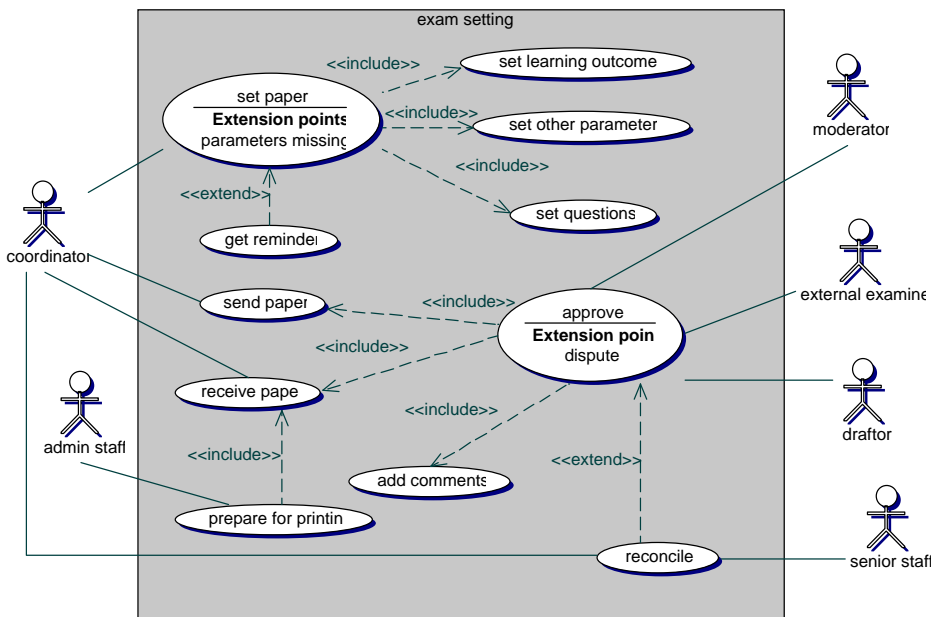


Fig. 4. A UML Use Case Diagram for exam paper setting from the NVU case study

It must be emphasised that some of the roles are played by externals, i.e. actors that reside in another organization. The main activities are:

- setting the papers: including setting the learning outcomes to be examined and other appropriate parameters such as date and time of the exam;
- approving/disapproving the papers;
- adding, sending and receiving comments;
- updating the papers;
- preparation for final printing

Fig. 5 depicts a UML activity diagram which shows the process of setting a paper and the timing when each actor is involved in certain activities. We assume that the templates for the exam papers have been well designed for re-use thus unresolved issues are rare. The

workflow starts when a unit coordinator registers a draft exam paper with the system and enters the corresponding parameters. Note that there is no indication as to how the function may be implemented; for instance, send/receive may be implemented as upload/download in a web-based system. Next the unit moderator is notified and is granted access to the paper. If he/she approves it, the workflow moves to the drafter, otherwise comments are sent to the coordinator. The activities may be repeated several times until the paper is approved by the moderator. Some reconciliation procedure may be needed if the involved parties can't settle some of the issues but this aspect is not depicted in Fig. 5 or the diagram would look much cluttered. Similarly, the workflow moves to the activities to be carried out by the external examiner and the administrative staff until the paper is finalized for printing. The process of exam setting has been simplified as in the real world more than one actor may be assigned to an activity, and the administrative staff may get involved before the external examiner. The simplification should not affect in general the definition of the main problem. There are many other examples such as a coursework marking process which is more complex in the sense different activities have to be synchronised before the process can move a step further.

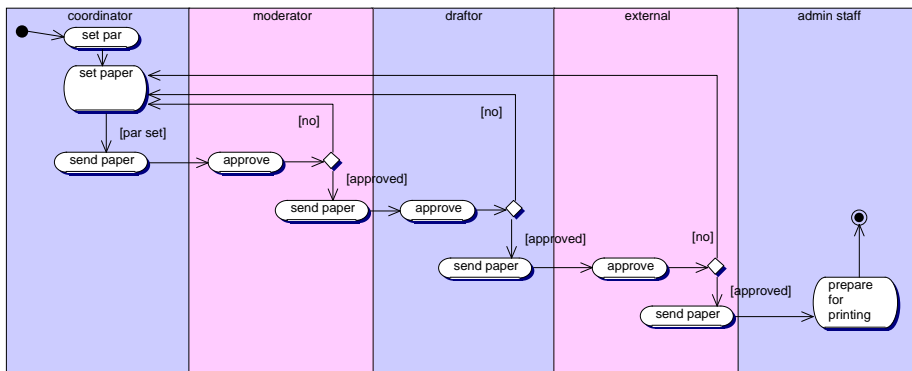


Fig. 5. A UML Activity Diagram for exam paper setting from the NVU case study

In the remaining sections, we present a framework for easy integration of BPM systems using this case study as an example.

5. A framework for BPM integration

As discussed earlier, BPM integration may take place at different stages. The major challenges for it vary by stage. A framework is clearly required and a few exist already (Ma et al., 2006, Meng et al., 2006 and Jung et al., 2006). In contrast to the others, Ma et al., (2006) proposed a portal-based framework that aims to make the integration easier and at the user level with minimum requirements for programming at a lower (i.e. API) level. The advantage of this approach is that it is very flexible no matter how the partnership would change. It supports BPM integration on the fly. One of the disadvantages of the approach is that it relies on an existing portal framework such as the uPortal framework (uPortal 2009). A portal framework that conforms to the WSRP standard (WSRP 2003) would allow a BPM system to be made available to different organisations through a portlet in a standard way.

However, differentiation in cultural, work practice and user preference supported through the interfaces to the systems are often the key to employee efficiency and productivity rather than standardisation. Standardisation is good at platform, components and service interconnection level, but not always so at the business procedure and user level, especially when collaboration across domain is considered. One important goal of this research is to develop a general framework that would allow organisations to achieve BPM integration in a fast changing environment but minimise the effect on differentiation. In this section, we present the requirements for such a framework, the design goals and an architectural design of the key aspects of the system.

5.1 The requirements and design goals

The main design goals for the framework are:

- to support cross-domain, human centric collaborative business process integration
- to support BPM integration at a higher level of abstraction
- to reduce IT investment through minimising the programming efforts for the integration
- to encourage the use of familiar BMP tools available to each participant of the shared business processes

The main requirements of a general framework for BPM integration are:

- provision for managing the full life cycle of business processes - support the business process life cycle from modelling to execution based on a broad array of industrial standards
- provision for process monitoring - provides notification if KPIs are in question
- provision for BPM integration - support for inter- and intra-domain collaboration and cooperation and task management
- provision for security - provides user identification management and role-based access control
- provision for personalisation - provides role-based access which helps users to focus on information, services and processes most relevant to their job
- provision for customisation - provides flexible web page layout and content organisation so that users have greater control over presentation aspects

Many of these come through leveraging the use of middleware such as an authentication service and an event engine for complex event processing as well as existing BPM engines and business process modelling tools. We developed Process Interceptor and Mapper (PIM) of which the main components and architecture are described in the next section.

5.2 The architecture and main components

Ma et al., (Ma et al., 2007) describe a design pattern for structuring a system that supports cross-domain, human-centric, collaborative business processes with minimum IT investment. A general framework for BPM integration has been developed to address the challenges identified in Section 2 using the design pattern. In the stage one of BPM integration, business users specify the high level business objectives typically in terms of use cases and user interactions by using UML or BPMN modelling tools. Fig. 4 and Fig. 5 show an example specification in UML which outlines the main objectives and workflow in UML. These specifications are typically created by business user together with the system analysts. They can do so with any business modelling tools they prefer as long as the definition can

be passed to a BPM engine for execution in stage two. The interaction with the process will affect its status which are captured in the process instance. The information is intercepted in stage three and will be fed into an IFM (InterFace Mapper). The main purpose of the IFM is to bridge the gaps between the users and the various “foreign” BPM systems the user encounters. This way the approach alleviates the need for heavy IT investment in order to glue the back-end services together to form an integrated system, which includes business and IT planning and programming at the much lower i.e. API level. Fig. 6 shows an architectural view of a system based on the design pattern. The ovals represent the components and the rectangles the views (aka interfaces to the user). The Interceptor component connects directly to the BPM engine and intercepts the running instances of the shared processes before they are passed to the IFM which presents the instance in a pre-defined view to the end users. In order to access the functions of the BPM engine and at the same time stick to the familiar views and steps supported by the preferred tools and user interfaces for monitoring and performing tasks, the user uses the PIM system which produces adapted views that match their preferences.

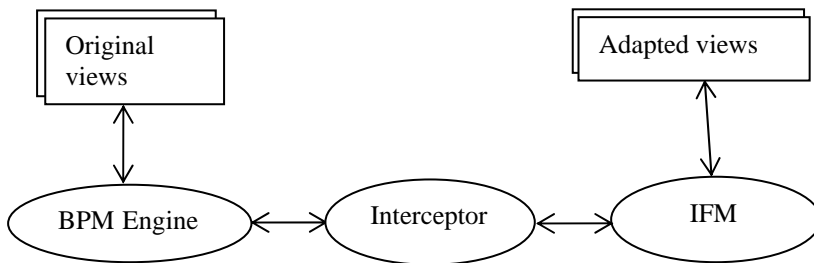


Fig. 6. The architectural design

The mapping of the instance to user adapted views is based on XML technology. In the next section, an implementation of the design is described.

6. An implementation

6.1 General description

A proof of concept implementation based on the design is described in detail in Caldera (2008). An open source Java BPM engine Enhydra Shark (2008) was used and extended for the purpose. Enhydra Shark (ES) supports XPDL as the native language and also allows easy incorporation of a number of database management systems including DB2, MySQL, Oracle and etc. ES comprises a suite of tools: SharkAdmin, SharkWebClient and Together Workflow Editor (TWE). TWE is a graphical editor used for process modelling. TWE can generate XPDL from the graphical process model. The generated XPDL design is then passed to the Enhydra Shark Workflow Engine through SharkAdmin. The same can also be done through the SharkWebClient which in addition supports a Web-based interface. An extension was made to the SharkAdmin to incorporate an Interceptor and an IFM component as described in Section 5.

6.2 Implementing the interceptor

One of the main challenges faced during the implementation was how to intercept the process instances for the IFM component to produce adapted views for the users. Fig. 7

shows a UML class diagram of the implementation which is for holding the detail of a process instance. First of all, a meta-language called procXML was defined to be an interchange format for process instance. procXML contains information about a process instances such as the process definition, activities, statuses of an instantiated processes, its activities and the participants. An XML schema was used to validate procXML files. Java Architecture for XML Binding (JAXB) framework (Ed Ort and Bhakti Mehta 2003) was used to map and bind process instance represented in XML into Java classes, interfaces and objects. Fig. 8 shows how it works. An XML schema is fed into the binding compiler which generates a set of Java classes and interfaces for representing a process instance. Through the JAXB APIs, XML files representing process instances can then be marshalled/unmarshalled to/from Java objects. This way process instances are captured from the Shark Workflow Engine into the XML files.

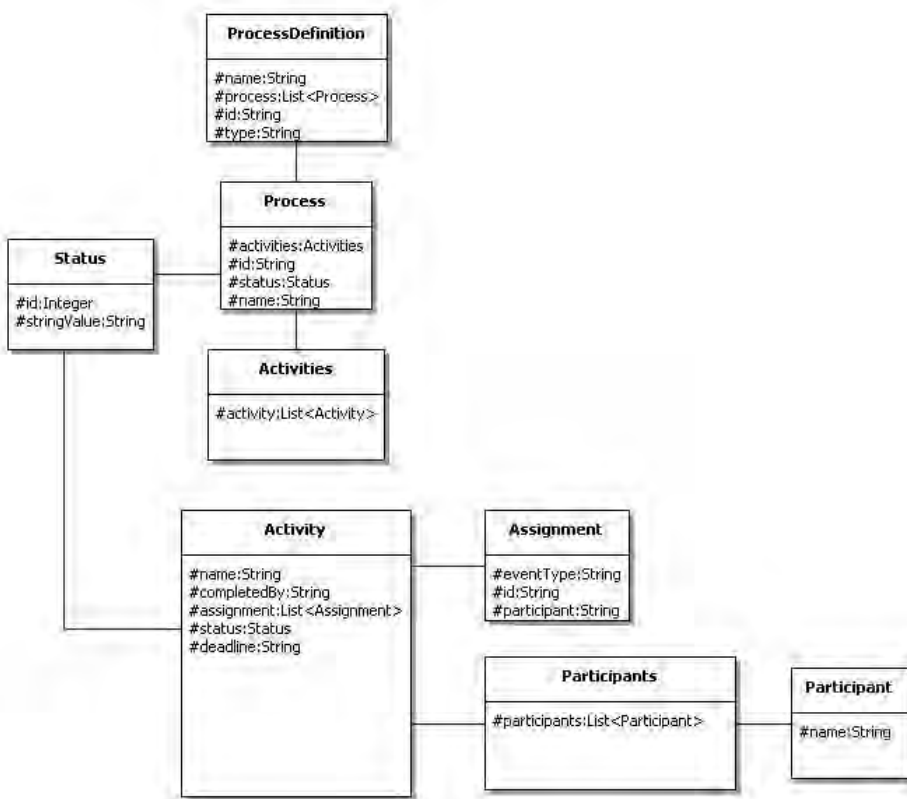


Fig. 7. Classes used by the interceptor to capture the process details

6.3 Implementing the IFM and testing results

The IFM is developed using the XML technology. Process instances captured into the XML files by the Interceptor are transformed according to user preferences using XSLT. Such transformation may occur on the server side or on the client side, and in this case on the

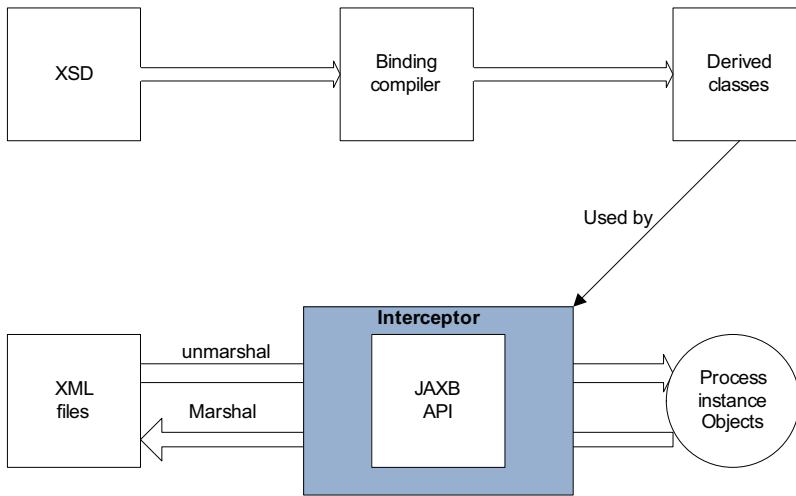


Fig. 8. Implementing the interceptor with JAXB



Fig. 9. The Politecnico Di Milano view of the process

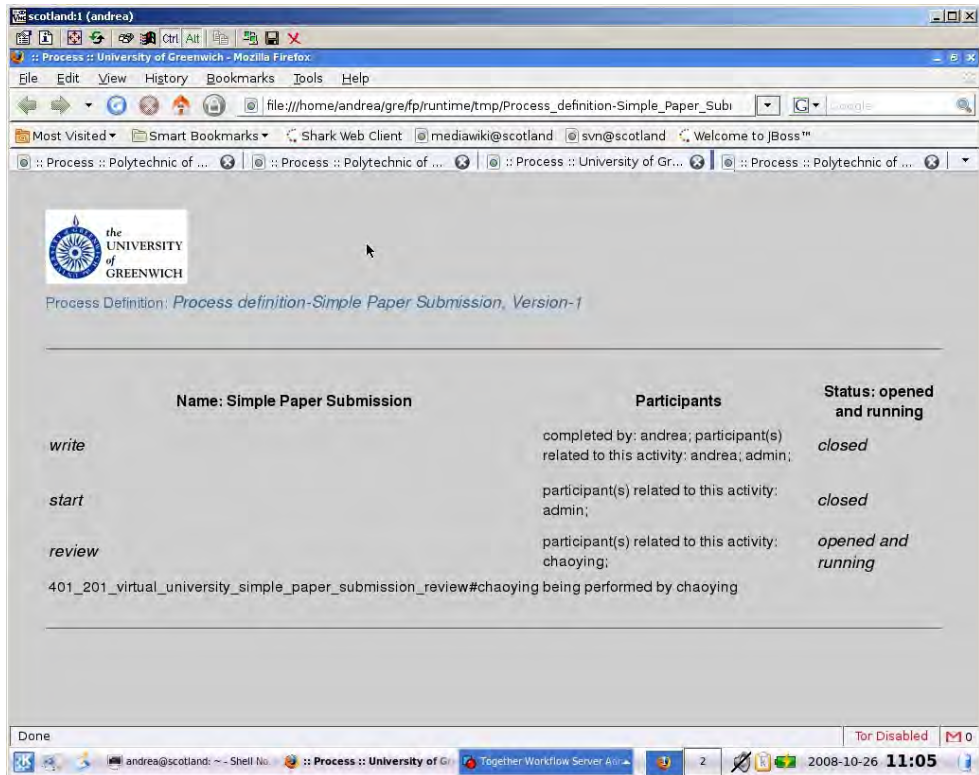


Fig. 10. The UoG view of the shared process

server side, through the Java API for XML processing (JAXP 2008). JAXP comes as a standard component of Java platform, and allows applications to parse, transform, validate and query XML documents using an API that is independent of a particular XML processor implementation. JAXP is used because it allows us to add the IFM as a pluggable layer without introducing dependencies in application code.

To illustrate how the framework may support BPM integration in a cross-domain environment, imagine a scenario in which two institutions work together in an exam paper setting process as described in the case study in Section 4. Note that the process was simplified in the prototype. Suppose the process was defined by the University of Greenwich (UoG) and followed by the Politechnico Di Milano. A member of staff called Andrea started writing and submitting a paper to the system. The paper is to be reviewed by a member of staff at UoG called Chaoying. Fig. 9 and Fig. 10 show two views: one original for the UoG and one adapted at Politechnico Di Milano. One can see that two activities in the process were completed and closed, and the third was still open and running.

As the implementation is only a proof of concept prototype. Several important issues should be addressed in future implementations as discussed in the next section. In addition the PIM component should be a separate entity from the SharkAdmin instead of an extension to it as it currently implemented. This was done to save time for develop GUI in order to interact

with the PIM. Despite this, the current implementation does prove that the framework with PIM as a key system component meets the design goals. In the next section, we discuss some of the main issues encountered in the development of the framework.

7. Discussion and future directions

7.1 Culture and tool issues in workplace

BPM is changing the culture in the workplace. Whilst the scope of BPM can affect everything from role of the business analyst in defining business workflows, to the planning and management of BPM software through to the actual to services executed to implement a BPM workflow, there can be a hidden impact on the user changing the way human-centric business processes are implemented.

Before BPM, humans had a task to do and they were able to do it in their own individualised preferred way. With the advent of BPM, many users can be forced to follow the workflow and algorithm specified by a business analyst. This often doesn't work well as people work and think in different ways. In order to help employees embrace the workflow concepts, there is a view that technology needs to support humans in the way they want to work and not be prescriptive. This means being flexible and adaptable to different needs and ways of working. What the technology needs to do is allow the users to personalise their workflow and define how they want their tasks to be orchestrated. Note that is not always easy to prescribe all processes in advance, some might be ad-hoc and not sufficiently well defined to have a clear start and finish. In these situations it is important that the human remains in control.

There is also a move in the industry towards the integration of workflow with current working practices and tools, so instead of booting up a workflow tool to use, the idea is that the workflow would be integrated with tools the user is using to deliver their normal work e.g. email and mobile devices. The personalisation of workflow and integration with tools is a key direction for the development of this area however there is much work left to do (Schurter 2009). In developing the framework, we attempted to address personalisation and customisation issues through the PIM system and have successfully demonstrated that it is possible for each organisation in participating collaborative human centric processes to adapt the views according to their own definition.

7.2 Evaluation and future improvement

We have described a general framework and demonstrated how it could be used of for integration of cross-domain, human centric and collaborative BPM through use of a case study. With the framework, business users are empowered with the means to specify and create shared processes at a high level with tools such as UML use case, activity diagram, BPMN and/or other graphical modelling tools. They can run the defined processes with their local BPM suites. In order for the process to be shared by their partners from other organisations, we design and implemented a PIM system which can capture runnin process instances and produce user specified views for each of the partners. Although a Java BPM system based on XPDL was used in our implementation, the same design principle should work with any BPM suite no matter which language, e.g. XPDL or BPEL, is used by the engine. The challenge is however that it can be difficult if not impossible to obtain running process instances with many existing BPM packages. The representations of such instance

are vendor specific. The newly released OMG standard BPDM (2008) could be used for standardisation of process instance representation. BPDM was not finalised when our system was developed but it is designed such that it is straightforward to replace ProcXML with a BPDM based solution for intercepting the process instances.

The provision for monitoring in this framework is limited to what are available through the BPM suite used. To incorporate intelligent BAM as discussed in Section 3.10, more work is required. The two approaches are now ready to be integrated more closely in order to address the issues raised in Section 3.10.

As one of the design goals, the framework includes provisions such as an authentication service through leveraging the use of the existing systems or middleware rather than reinventing the wheel. Once the framework is in place, the organisations may define and have their specific views that the various BPM engines generated through the use of XML technology such as XSLT.

8. Conclusion

We have designed a general framework for integration of cross-domain, human centric and collaborative BPM system, and implemented the key aspects of it while reusing the existing BPM systems and other standard services as much as possible. We discussed the three different stages of BPM integration along side the issues and main challenges. The main advantage of the framework is that it addresses issues of integration at stage three while most existing work and BPM related standards address issues only at stage one and/or two. The work is still ongoing, and issues as discussed in Section 7 still need to be addressed. It is however a very positive way forward towards BPM integration. Looking to the future, in addition to the issues of working with personalised client devices, with the increasing trend towards more employees working remotely, this provides additional BPM challenges in working both within and across organisations involving issues such as security, firewalls, infrastructure issues, cloud computing and use of SaaS to support the delivery of BPM.

9. References

- ASAP (2005). OASIS Asynchronous Service Access Protocol (ASAP). http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=asap.
- Bennett, Keith; et al. (2000). "Service-based software: the future for flexible software" (PDF). Seventh Asia-Pacific Software Engineering Conference, 2000. APSEC 2000.. Proceedings: Seventh Asia-Pacific Software Engineering Conference, 2000. pp. 214-221. doi:10.1109/APSEC.2000.896702
- BPDL. (2006). The Web Services Business Process Execution Language Version 2.0. May 2006. OASIS. <http://www.oasis-open.org/apps/org/workgroup/wsbpel>
- BPDL4People (2005). OASIS WS-BPDL Extension for People (BPDL4People) http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=bpdl4people
- BPRI. (2005). Business Process Runtime Interface. OMG RFP. Mar 2005. <http://www.omg.org/attachments/pdf/PaulHarmonBPRIarticle.pdf>
- BPDM 2008. Business Process Definition Metamodel. <http://www.omg.org/>
- BPMN. (2006). Business Process Modeling Notation. OMG Specification, Feb, 2006. <http://www.bpmn.org/>

- Bunke, H., Messmer, B.T.(1994): Similarity Measures for Structured Representations. In: Wess, S., Richter, M., Althoff, K.-D. (eds) *Topics in Case-Based Reasoning*. LNCS, vol. 837, pp 106-118, Springer, Heidelberg (1994)
- eClarus (2009). eClarus Business Process Modeler <http://www.eclarus.com/>
- Enhydra Shark (2008). An open source Java workflow engine URL: <http://www.enhydra.org/>, last accessed Nov. 2008
- Hinchcliffe Dion (2006). What Is WOA? It's The Future of Service-Oriented Architecture (SOA) <http://hinchcliffe.org/archive/2008/02/27/16617.aspx>
- Hjeltness T., Mikalsen A. B. (2003), Towards a Model of a European Networked University for e-learning, International Conference on Network Universities and e-Learning May 2003 Valencia, Spain, Available at http://ans.hsh.no/lu/inf/menu/final/publ/thorleif_hjeltnes.pdf accessed 7th November 2009
- Hollar, R. (2004). Will Standards Turn Portal into Commodities? Two approaches that work, *Web Services Journal*, Jan 2004.
URL: <http://webservices.sys-con.com/author/2103Hollar.htm>
- Hollingsworth. D. (2004) The Workflow Reference Model: 10 Years On. *Workflow Handbook 2004*. Edited by Fischer, L. Future Strategies Inc. 2004. Also available for download at www.wfmc.org
- Hopper, G. and Woolf, B. (2004). *Enterprise Integration Patterns*. Addison-Wesley, 2004. ISBN 0321200683.
- Gartner (2009). Business Process Management Suites (BPMS) Will Be Among the Fastest Growing Software Markets through 2011. <http://www.gartner.com/it/page.jsp?id=502645>
- Jung, J., Kim, H., Kang, S-H. (2006). Standards-based approaches to B2B workflow integration. *Computers & Industrial Engineering* 51, Elsevier
- Kapetanakis, S., Petridis, M., Ma, J., Bacon, L.(2009) : Workflow Monitoring and Diagnosis Using Case Based Reasoning on Incomplete Temporal Log Data, *Proceedings of the Workshop on Uncertainty, Knowledge Discovery, and Similarity in Case Based Reasoning UKDS*, in *Workshop proceedings of the 8th International Conference on Case Based Reasoning*, Seattle, USA.
- Kolodner, J. (1993) *Case-Based Reasoning*. Morgan Kaufmann Series in Representation & Reasoning.
- Koulopoulos, T. M. (1995). The Workflow Imperative: In: *Building Real World Business Solutions*. ITP. 1995. ISBN 0442019750
- Ma, C. and et al. (2006) Towards the Design of a Portal Framework for Web Services Integration, *Proc. of ICIIV'06*, pp.163-168, ISBN: 0-7695-2522-9, Guadeloupe, French Caribbean Feb. 19-25, 2006, IEEE Computer Society, Guadeloupe.
- Meng, J. et. al. (2006). DynaFlow: a dynamic inter-organisational workflow management system. *Int. Journal of Business Process Integration and Management*, Vol. 1, No. 2. 2006, pp 101-115, ISSN??
- Mileman, T., Knight, B., Petridis, M., Cowell, D., Ewer, J. (2002): Case-Based Retrieval of 3-D shapes for the design of metal castings in *Journal of Intelligent Manufacturing*, Kluwer. 13(1): 39-45; Feb 2002.

- Minor, M., Tartakovski, A. and Bergmann, R. (2007): Representation and Structure-Based Similarity Assessment for Agile Workflows, in Weber, R., O. and Richter, M., M.(Eds) Case-Based Reasoning Research and Development, *Proceedings of the 7th international conference on Case-Based Reasoning, ICCBR 2007*, Belfast, NI, UK, August 2007, LNAI 4626, pp 224-238, Springer-Verlag.
- Oasis (1993). Organization for the Advancement of Structured Information Standards. <http://www.oasis-open.org/specs/>
- Ed Ort and Bhakti Mehta. (2003) Java Architecture for XML Binding (JAXB), Technical Article, Sun Developer Network, March 2003.
- OGSA. (2006). Open services grid architecture <http://www.ggf.org/documents/GFD.80.pdf>.
- OMG. (1999). Object Management Group <http://www.omg.org/>.
- Oracle. (2009). Oracle's Business Process Management solution, <http://www.oracle.com/us/technologies/bpm/index.htm>
- Schurter T. (2009). BPM State of the Nation. <http://www.bpm.com/bpm-state-of-the-nation-2009.html>.
- Singularity (2009). Business Process Management (BPM) Suite. <http://www.singularity.co.uk/>
- Soap (2007) XML Protocol Working Group. <http://www.w3.org/2000/xp/Group/>
- Sun. (2003). Introduction to JSR 168. July 2003. URL: http://developers.sun.com/prodtech/portalserver/reference/techart/jsr168/pb_whitepaper.pdf
- Silver, B.(2006). The ABCs of BPM. *InfoWorld*. Feb. 2006. URL: http://ww6.infoworld.com/products/print_friendly.jsp?link=/article/06/02/20/75095_08FEbpmmap_1.html
- UDDI (1993). Universal Description, Discovery and Integration. <http://uddi.xml.org/>
- uPortal. (2009) A JA-SIG maintained web site. <http://www.jasig.org/uportal>, last accessed Nov. 2009.
- Van der Aalst., W. (1999). Formalization and verification of event-driven process chains. *Information and Software Technology*, 41(10):639 to 650, 1999
- WS-CDL (2007). Web Services Choreography Working Group. <http://www.w3.org/2002/ws/chor/>
- WS-Coordination (2009). Web Services Coordination <http://docs.oasis-open.org/wstx/wscor/2006/06>
- WF.(2009) Windows Workflow Foundation Overview. <http://msdn2.microsoft.com/en-us/netframework/aa663340.aspx> , last accessed Nov. 2009
- Wf-XML (2006). WfMC Wf-XML <http://www.wfmc.org/wfmc-wf-xml.html> wf-html
- WSRP. (2003). Web Services for Remote Portlets Specification 1.0, wsrp-specification-1.0, OASIS, Sept. 2003. <http://www.oasis-open.org/>
- Wolf, M., Petridis, M.(2008): Measuring Similarity of Software Designs using Graph Matching for CBR, In *workshop proceedings of AISEW 2008 at ECAI 2008*, Patras, Greece (2008)
- WSDL (2007). Web Services Description Working Group. <http://www.w3.org/2002/ws/desc/>.

W3C (2009) World Wide Web Consortium <http://www.w3.org/> Last accessed Nov. 2009
XML (2003). Extensible Markup Language. <http://www.w3.org/XML/>
XML Schema (2006). <http://www.w3.org/XML/Schema>
XPDL. (2005). XML Process Definition Language. WfMC-TC-1025.WfMC. Oct. 2005.
URL: <http://www.wfmc.org/standards/xpdl.htm>