The Workflow Reference Model 10 Years On

David Hollingsworth, Fujitsu Services, United Kingdom
Chair, Technical Committee, WfMC.

INTRODUCTION

Last year saw the 10th anniversary of the Workflow Reference Model. This short paper reassesses the relevance of the Model in the current context of Business Process Management. It discusses the principles behind the Model, its strengths and weakness and examines how it remains relevant to the industry today. It concludes by introducing a number of considerations required to establish a "BPM Reference Model" and discusses how the various overlapping standards in this space may be categorised.

THE WORKFLOW REFERENCE MODEL

It is not the purpose of this article to re-examine the Reference Model in detail[1]. Its significant aspects can be summarised into the following three categories, each building incrementally on the preceding:

(i) A common vocabulary for describing the business process and various aspects of the supporting technologies to facilitate automation.[2] This provided the essential foundation for the subsequent detailed discussion on how a workflow system could be architected in a general sense.

(ii) A functional description of the necessary key software components in a workflow management system and how they would interact. This was developed in a "technology neutral" manner, to allow the model to be independent of any particular product architecture and implementation technology.

(iii) The definition, in functional [or abstract] terms, of the interface between various key software components that would facilitate exchange of information in a standardised way, thus enabling interoperability between different products. Five such interfaces were identified and became the foundation for the WfMC standardisation programme.

An important principle was that the Reference Model focussed specifically on workflow management technology and standards. It deliberately did not attempt to define standards in other, related areas, in which other industry bodies were working; these were seen as complementary.

Over time the WfMC became actively involved in the incorporation of its workflow standards into the OMG object architecture and standards[3]; sub-

[1] The Workflow Reference Model - referenceTC-1003, accessible at www.wfmc.org

[2] The WfMC Terminology and Glossary - reference TC-1011

[3] See OMG Workflow Management Facility [also known as jFlow]

sequent work also identified how various security standards could be deployed to enhance workflow interoperability [4]

THE FIVE INTERFACES

Each interface was initially specified as a business level statement of objective, that is to say what the interface was intended to achieve in business terms and why a standardised approach was desirable. This was subsequently followed by a detailed, but abstract specification of how the interface operated and finally (for most interfaces) a "binding" specification covering the implementation of the interface in a particular technology.

Interface 1 was developed to support the exchange of process definition data between BPR tools, workflow systems and process definition repositories, enabling users to select the most appropriate tool for different aspects of the business process lifecycle. It was specified as a Process Definition Meta-Model, defining the process objects, their attributes & relationships, and a textual grammar for expressing the process definition structure and information content. This was subsequently re-expressed as an XML document definition [XPDL][5]. Further proposals include specification for event handling and inter process messaging.

Interface 2 was developed to facilitate client application integration with different workflow systems, in particular to support the principle of [client] application portability and reuse with different workflow management systems. It was specified as a series of Workflow APIs [WAPI] to allow the control of process, activity and worklist handling functions. These were originally defined in "C" and subsequently re-expressed in IDL [as part of the OMG workflow management facility] and OLE. A set of "C" APIs for manipulating process definition objects and attributes was also defined[6].

Interface 3 was scoped to provide a common framework for 3rd parties to integrate other industry applications & services, including specific support of agent interfaces to provide a common framework for access to legacy applications. It was developed as set of five basic API calls, defined within the WAPI document to support a common mechanism for connection, disconnection and calling to a variety of agents or other third party software environments.Interface 4 was developed to facilitate process automation across multiple heterogeneous implementation environments. It comprises an interchange protocol covering five basic operations, specified in abstract terms (initially it was defined in IDL) and with separate concrete bindings. The initial version was defined as a MIME body part for use with email; subsequent versions have been specified in XML (Wf-XML). Ongoing work has lead to version 2 of Wf-XML, layered over SOAP and ASAP.[7]

The purpose of Interface 5 is to allow consistent audit and administration of workflow cases across systems, through the specification of a common

[4] See TC-1019 Workflow Security Considerations

[5] See WfMC-TC-1025

[6] Published as WfMC TC–1010, later consolidated into WAPI specification TC-1009

[7] See ASAP/Wf-XML 2.0 Cookbook by Keith Swenson, in this publication & WfMC TC-1023

model for audit data, including event identification, formats & recording. As such it was originally specified in abstract terms, although a set of common APIs for access to audit data was subsequently developed. [8] Recent work is aimed at expressing the audit data structure as a set of XML structures.

Although conceived as five individual interfaces, the separation is apparent only when viewed in the context of the stated business objective. In reality there is significant commonality of function between the various "interfaces"; for example the triggering of the initiation of a process execution is fundamentally the same action whether it is done client side (i/f 2) or server side (i/f 4). The evolution of the WAPI [API] specification started with client application interactions but expanded to include a full repertoire of API calls. Similarly, Wf-XML was developed initially for server-server interaction but has also been used successfully for client-server interactions[9].

A more useful and fundamental distinction is perhaps to take a view of each interface from the perspective of process ownership and administration control. In particular, interfaces 2 and 3 may be considered to be "tightly bound" to the local workflow management system and reflect a local view of resource management—interface 2 handling interaction with human resource and interface 3 interaction with automata resource. This has two significant consequences.

In the first place the process definition is localised to the point of process enactment through the expression of the resource assignments (e.g. participants and applications).

Secondly the Reference Model could make the simplifying assumption that specification of messaging between a WFMS and participants need not be contained in detail within the process definition. It becomes a function of the WFMS locally to organise the most appropriate form of interaction with the participants via local *Worklists* (web access, email, etc), according to the defined (within the process definition) *Activity* or P*rocedure*.

[This is also why, in XPDL, *Participant* and *Application* assignments are invalid within *Activity* definitions of type *Subflow*—since the resource assignments are only known to the remote process invocation environment or "factory". Furthermore, audit data relating to participant and application can only be collected and interpreted locally to the WFMS "factory" environment since this owns the resource assignment.]

There are important consequences when considering the wider context of interaction between processes and participants through public infrastructure such as the Worldwide Web. This whole area is further discussed later in this paper when considering internal and external properties of the process and its execution factory in the context of the current view of a Business Process Management (BPM) System.

---

[8] See WfMC TC-1015 [Audit Data specification] and TC-1022 [Access APIs]

[9] See B2B Interoperability through Presentation Layer Integration, Alan Rickayzen – Workflow Handbook 2003

STRENGTHS AND WEAKNESS

The original workflow reference model was based upon a number of fundamental premises that are worth reflection. To the author's mind these have been the enduring strengths of the Reference Model.

Abstraction of the business process architecture

The model attempted to construct an abstract view of the business process in terms of its core characteristics, separated from the technologies that could be used to deliver its functionality in a real world situation.

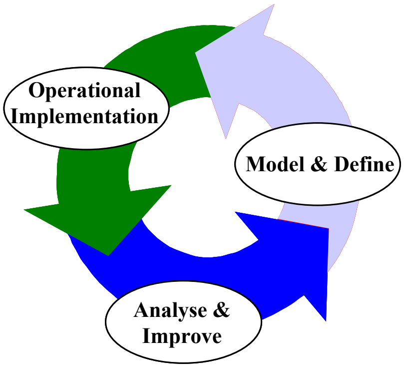
The separation of abstract and concrete views similarly flowed down into the specifications of its interfaces, which were defined first in abstract terms and then concrete "bindings" of each abstract interface to the specific interface technologies that could be used to deliver it.

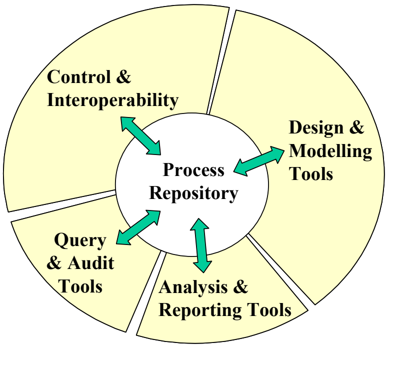
Over the years, the value of this approach has become apparent. As technology has evolved a range of different interface specifications have been defined—the so-called concrete bindings—each appropriate to the technologies of the time, but adhering to the same overarching abstract model of interface operation.

The initial interface bindings reflected a relatively low level programming view of the interfaces around a workflow system—these were based on APIs typically expressed in "C". As the value of higher level, more productive interface technologies became accepted bindings were added in the form of IDL and CORBA (for the OMG), MIME email (for process interoperability). The most widely used interfaces were further redefined using Web Services and XML (XPDL and Wf-XML).

A Lifecycle View of the Business Process

During the 90s there was the culture of continuous improvement and TQM. This century we have seen Six Sigma and the concept of the "agile" business. The Reference Model was founded on the principle that ongoing change to the business process would be the norm. It provided a lifecycle view of three broad phases, but attempted to ensure that all were "joined up"—in the sense that each contributes to the overall consistency of view and uses a common model for the representation of the business process.





Lifecycle Model—Business Phases and BPMS Components

In the original model, change was supported by the concept of a process repository, with the ability to incorporate a set of modeling and business process definition tools around it (using Interface 1). Audit and analysis tools operating on a common audit data specification (Interface 5) supported the feedback loop to allow improvements into the process definition.

In this architecture, the process repository is central, with interfaces specified for tools and software to interact with it throughout the lifecycle.

One sensible refinement to the original model is to separate query and audit functionality (required for records and operational management of individual cases) from analysis and reporting tools (required essentially for statistical analysis and input to modeling and process improvement).

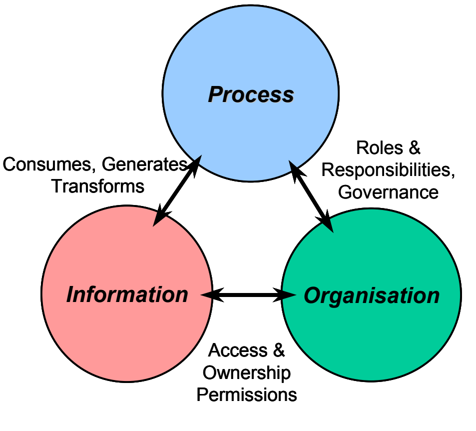
It is also clear that more emphasis is required on the decomposition of processes into fragments and their consolidation in various ways to support more dynamic operational business processes. This stems from the vast increase in co-operating e-businesses brought about through the Worldwide Web [which was still in its infancy when the original model was developed].

The original model identified various ways in which process fragments could interact—hierarchic subprocess, parallel synchronised processes, etc and did develop runtime models for binding them in execution terms. However, it did not attempt to develop anything beyond a primitive choreography capability in the area of process definition support for interactions between process fragments. The needs in this area are considered later in the article.

Information and its relationship to process and organization

Process, information and organization are inexorably linked; one can approach an architectural model from any of the three dimensions but for coherence all three must fit together. Process-based architectures tend to emphasise process as the dominant dimension; processes consume, generate or transform information, behaving in accordance with a set of corporate governance rules. By contrast, information based architectures emphasis the information dimension, viewing processes as operations that are triggered as a result of information change.

The following diagram shows the relationship between the three viewpoints.



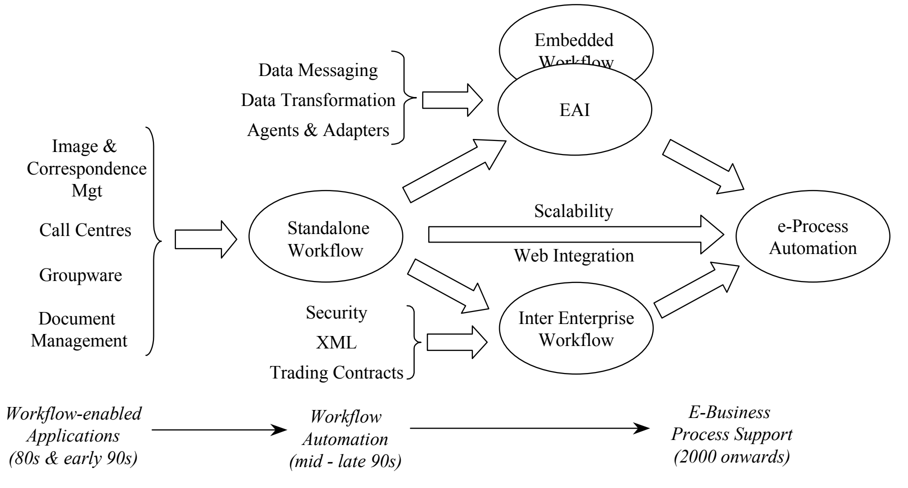
The Reference Model does embrace all three dimensions but takes a relatively simplistic view of the information dimension. It recognises three different data classes—workflow control, workflow relevant and application data, but can be validly criticised as weak in the area of information marshalling within a process. In retrospect, activity attributes could have been defined identifying incoming and outgoing information flows associated with the activity, which would have aided generality. Equivalent provisions are provided for information at process level[10].

Some provisions for data co-ordination and recovery were identified within the original model; the assumption was made that either two phase commitment mechanisms would be in place and/or transaction compensation would be invoked. Many products support either or both, but specific standards to assist their specification within a process definition were not developed. Instead a simple concept of exception transitions was developed to allow specific failure handling or compensation activities to be user defined following an exception event.

WHAT IS BPM?

Several other articles in this handbook discuss this question, so only a brief overview is provided here. It has been a subject of some debate whether there is any practical difference between workflow management and business process management. Certainly many of the concepts are the same and, where there are differences, these tend to be in points of detail, or through different emphasis.

The following diagram appeared in the 2001 edition of the Workflow Handbook to illustrate the evolution of what is now typically called BPM [in the original article it was described as e-Process Automation].



Three main technologies have been converging—workflow, EAI and Web, each however, coming from a rather different perspective.

[10] XPDL defines data at process level as *Formal Parameters* [of types in, out, inout] or, for internal data, as *Data Fields* [also referred to as Workflow Relevant data]

Traditionally workflow has placed more emphasis on organisation structure and associated roles and responsibilities. Business process models typically start from an organisational perspective with views of accountability and responsibility attributes and the roles and responsibilities associated with processing work activities. Work resources thus tend to embrace both human and machine.

The typical EAI approach has placed more emphasis on the engineering and automation aspects—sophisticated agents and transactional qualities. Process models typically start from a work perspective—data flows or transactional definitions—and focus on fully automated tasks without human involvement.

The Web has brought a new infrastructure base built around web services protocols, XML structured information content and massive potential scalability.

However, one of the most fundamental characteristics required in the BPM world is the ability to support the flexible management of dynamic business change. From the process perspective this can be thought of as compressing the time around lifecycle model. From a different perspective it can be thought of as requiring more adaptive technology to be deployed within process execution. Several characteristics may be noted and are discussed later in the paper:

- Late bindings—to introduce flexibility to the run time environment
- Rules engines—to facilitate complex expression evaluation, independently of the core process specification
- Adaptive processes—to facilitate dynamic change during execution

Of course these characteristics were not specifically excludes from earlier architectural thinking in the workflow space, it is rather that they become more significant when dealing with BPM.

A BPM REFERENCE MODEL

So if BPM and workflow are seen as essentially the same, albeit with some differences of emphasis, how relevant is the original workflow reference model to the BPM world?

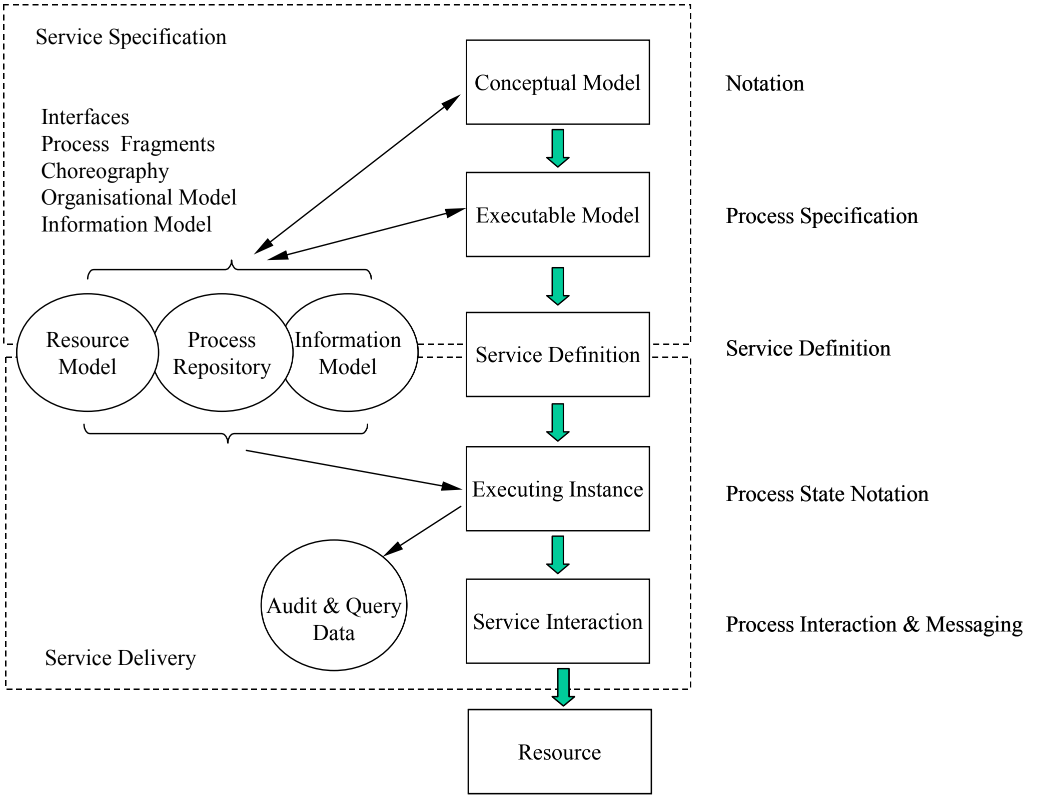
Clearly several of the original principles are still entirely valid:

(i) The lifecycle oriented view is, if anything, enhanced, due to the increasing rate of desired business change.

(ii) The abstraction of the business process from the implementation technology remains an important goal, because organizations tend to think differently, using different skills and methodologies, in the two domains. Although current thinking has placed great emphasis on web services architecture as the natural technology choice, it should not be allowed to constrain the development of the business process architecture. One current concern is that some of the thinking in the process space is being constrained by the specific view of web services as the implementation environment; this raises the danger of ignoring the organizational and human aspects of the business process, in favour of a resource model wholly based on web services.

(iii) The development of a functional component model of BPM, to identify those points where product interoperability is required, is still a sensible approach to categorise and develop the various standards that should apply.

The following diagram illustrates such a BPM component model, derived from a traditional view of the development and subsequent execution of a e business process.



The top half of the diagram illustrates the derivation of a process model along with its service delivery characteristics. The lower half illustrates the enactment of the process in a service delivery environment.

In some respects the above can be though of as a restatement of the workflow reference model into a sequential flow from process conceptualization through to realization as a series of service interactions with either process execution or human resources.

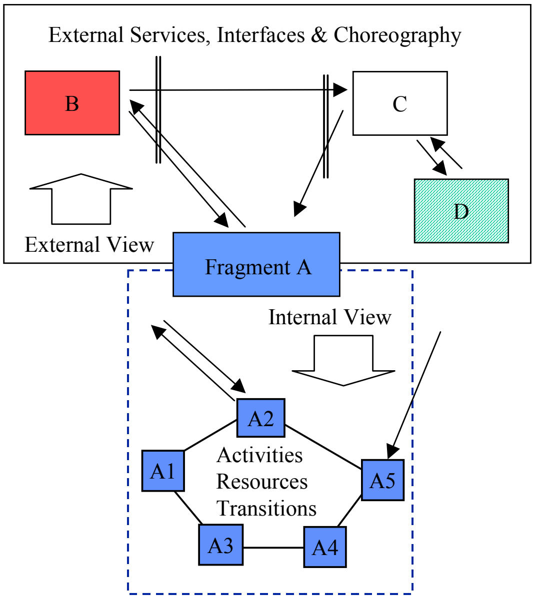
However, when looking at the components there are several areas of refinement that deserve to be considered in some detail within such a model.

Conceptual Model

Looking firstly at the conceptual model, this is concerned with the formulation of the business process in terms of business component objects and their interactions. Increasingly this phase needs to focus on the position of the process within an end-to-end process delivery chain, involving interfaces with existing or planned processes within other organizational domains. These may be relationships that are either up-stream (e.g. customers), downstream (e.g. suppliers or subcontract service organizations) or lateral (for example, to support internal governance or external compliance relationships).

This has lead to the view of the overall process as a combination of process "fragments" which can be recombined in various ways to deliver new or modified business capability. This has become important to support business agility and follows from the increasingly integrated business relationships between trading organizations. Business processes are already extensively intermeshed between such organizations—although this may not always be supported by formal automation. Business process outsourcing operations also require this capability to support efficient reuse.

Such process fragments can be seen to have both an "internal" and "external" view.



The internal view defines the actual or intended internal behavior of the process fragment—it includes not just the activities and transitions between them but, also [significantly] the internal resources required to support enactment of the process. It will also identify the boundaries of the fragment in terms of interactions with other process fragments or outside objects.

The external view defines the behaviour of the fragment as a "black box", seen from the outside and addressed through its interfaces. This view sees the process fragment very much as a source and sink of messages or events of different types. (Note that not all participating process fragments may be directly visible at the interfaces of another. In the diagram fragment D is opaque to both A and B—hidden behind C.)

Some form of choreography is required to identify the valid sequences of messages and responses across and between the participating process fragments. The choreography requires each process fragment to exhibit a set of prescribed external behaviours in response to such message sequences. The external behaviour is clearly derived from the internal process behaviour but represents only the subset that is chosen by the process owner to be made externally visible. Those familiar with Wf-XML, or interface 4 of the WfMC Reference Model, will recognise that this provides a consistent message scheme for developing such message choreography—based on a set of defined inter-process operations.

In summary, the process definition for a fragment can be seen to fall into two halves, each dealing with the different properties required for internal and external behavior. At the conceptual model stage not all these properties will be known; in particular the detailed internal behaviors associated with resources are unlikely to be fully defined.

The principal modeling and definition tools required at this stage are likely to focus on graphical design and interface definition[11].

Executable Model

To turn the conceptual model into an executable model requires a more detailed specification of the process in machine processable form, including not only its detailed internal structure but also its interfaces and internal resource usage. (There is, of course, often merit in delaying this binding to resources until execution time and most BPM products support the specification of resources in terms of roles or interface definitions, which are mapped to real resources during execution.)

It has proved relatively difficult to develop a fully standardised framework model to support this function within the BPM model. Not only do different BPM / workflow products have different internal representation structures but a large amount of local detail needs to be defined to develop the complete internal representation of the executable model. XPDL attempts to step around this problem through the use of extended attributes for bi-lateral agreement. In practice many BPM products have process design tools that are closely coupled to the execution capabilities of the BPMS. Generating a common approach across multiple product environments may depend more on agreement on a common meta-methodology for modeling the business process.

The Service Definition

The service definition is required to instantiate the executable model(s) into operational process instances. Where local resources are used within the operational process this is essential a local matter for the BPMS. However, where access to external process fragments is required, or incoming access from other external processes is offered, the service definition must provide the necessary addressing and resource identification information.

In a web services environment, a set of supporting [non-BPM specific] standards already exists for defining such services, resource access points, ac-

[11] Hence this is closer to a BPMN or BPEL abstract view of the process [fragment] than an XPDL view.

cess permissions and basic message types, etc. In other implementation environments equivalent schemes are required.

Service Interactions and Choreography

These represent the actual, run time exchanges between the resources associated with execution of particular process fragments.

Again, a distinction is drawn between interactions internal to the process fragment and those between fragments. Those internal to a service (for example invocation of local application resource or allocation of work to a local participant) will be regulated by the local BPMS.

Those interactions between process fragments will result in protocol exchanges. These interactions are typically layered into process and context data semantics carried over transport semantics. For example, Wf-XML has an upper, process-oriented exchange layer (e.g. generic operations such as start, stop, or query process instance) carried over SOAP or HTTP as the transport layer. Wf-XML also allows context data that is payload specific to the process instance to be incorporated within the interaction.

To coordinate the external service interactions some form of choreography is required. This will identify what permitted set of process operations and context data exchange is possible between the executing fragments and how this set of operations should be sequenced under various circumstances arising within the end to end process. (Conceptually at least, such choreography could also involve interaction with a human resource that is implementing a minimal process fragment such as a single task, as well as interactions with services.)

In one sense the choreography could be likened to a very high level process definition that links together process fragments by providing a set of high level business rules and identifiers for the locations of the resource implementing each process fragment.

Resources

The earlier diagram identifies a resource model as one of the required components on the boundary between process specification and delivery. Again, this has separate characteristics when looking at internal and external perspectives of a process fragment.

The internal view is broadly similar to that described in the Workflow Reference Model—a model of the resources to be applied to enactment of the process fragment, permitting the binding of appropriate resource(s) to activities according to a set of rules (for example expressed as roles or responsibilities) during the process definition stage. These need to embrace both human and system resource and be flexible enough to permit (controlled) resource substitution—late binding of resource to task.

A special case arises where two or more process fragments are executed within a single, common resource domain. This is, in effect, internal distribution of a larger process fragment operating under a single resource management system. (This can be approached in one of two ways. An inline block allows the execution of the fragment as an extension of the current execution thread within the existing data and resource space. A subprocess call allows the execution of the fragment as a separate execution thread, but can specify the use of the same common, resource model.)

The external view is essentially one of externally accessible services or other resources, each associated with a particular process orientated service delivery capability. These accessible external services may be predefined within a resource directory or may be subject to dynamic discovery at fragment execution time. (In the latter case there is a presumption that process orientated service characteristics can be incorporated into the service description to allow discovery through mechanisms such as UDDI.)

WHAT BPM STANDARDS ARE REQUIRED?

At the heart of any BPM reference architecture lie the methodology and standards for representing the business process.

The discussion above postulates the need for the business process to be considered at two levels:

(i) a lower level, internal view of each process fragment that is similar to the traditional workflow process model

(ii) a higher level view concentrating on modeling the overall process flow, linking re-usable process fragments together via a choreography. This is a view of the external behaviors of the process fragments, the executing resource locations and the dynamics of the interactions

Each of these aspects is developed further in the following sections. One interesting consideration is the extent to which a common model can be applied to both cases (as for example the BPEL concrete and abstract models).

The Internal Process Definition

There is already a number of overlapping public standards in this space, which all provide a means of representing process flow, events and decision points, and the classification of various process context data associated with executing processes. Some standards also provide a means of specifying the work resources associated with the process work items, or activities.[12]

The purpose of these tools is to enable the integration of different process design products with different execution products or to allow simple migration of existing process definitions to a different design / execution product.

A particular problem has been that different vendor products in the BPM space tend to use different process design paradigms. Many classes of real world problems can be represented in different ways in different tools, particularly the more complex aspects such as handling exceptions and various error conditions.

This is particularly true in the representation of process flow. Typically this involves conditional logic and a number of alternative routes (navigation paths) through the process, which may need to support a mixture of sequen-

[12] Process models in Manufacturing may also take a view of physical resources required to deliver work items - machines, raw material, work in progress, etc

tial and parallel activities. This logic may be defined in quite different ways within different process definition methodologies

Transition Based Representation

This is typically derived from Petri Net methodology; a process is represented graphically as a network of activity *nodes* and explicit *transitions* between them. *Edges* connect nodes to transitions (*input arcs*) or transitions to nodes (*output arcs*). Parallelism within a process is supported by transitions with multiple output arcs (a *split* into multiple execution threads transferring to different activities) or with multiple input arcs (a *join* of several execution threads into one). Alternative routes between activity nodes are evaluated by reference to *conditions* associated with the transitions. Although arbitrary complexity can be supported, multiple transitional expressions involving complex conditional evaluations can become cumbersome to represent in a machine-processable form.

A significant practical distinction is whether the transition logic allows [backward] transition to "earlier" [preceding] activities, allowing cycling through a set of nodes, or constrains transition flow to acyclic paths only.

Block Structured Decomposition

Any single node in a process model may be decomposed to a lower level of underlying process (a paradigm based upon the hierarchic sub-process model). In this approach parallelism is constrained to operate only within the context of a single level of decomposition (i.e. parallel threads cannot transcend block boundaries). A product based upon this approach cannot cope with an arbitrary complexity of split and join constructs—for example an unbalanced split where one path continues beyond the context of the current block.

Activity Pre- & Post-conditions

In this approach no explicit transitions between activities are declared. The process is defined as a set of activities each having entry (*pre*-) and exit (*post*-) *conditions*; parallelism is implicit and when pre-conditions are met the activity is initiated, independently of the status of other activities within the process. To provide sequential operation, pre-conditions may relate to the completion of a particular prior activity (and by extension to multiple prior activities, providing an "*and-join*" capability). Post-conditions may be used to control looping within an activity.

Role Activity Diagrams

RADs focus on defining the process through the actions that are taken within abstract roles associated with the organization, and the interactions between roles. In this way the process flow is modeled as the combined actions associated with the cooperating roles and their interactions. Graphical representation of the roles and interactions is provided with swim lanes used to represent the behaviour within a role. Modeling of data and documents is not normally handled.

The problem for the systems integrator is that it is not easy to transfer process information between design tools and/or workflow control software based upon the different design paradigms. A very large amount of work has been undertaken by both industry and academia in attempting to define a common representation of a business process which may be translated between these different paradigms, and, by extension, the different process modeling and definition tools associated with the paradigm[13].

Early work by the WfMC indicated the scale of the task to achieve anywhere near 100 percent practical transfer of process definitions across the diversity of design approaches in use. More recent analysis of workflow patterns by Wil Van der Alst and others[14] reinforces this view. The recent work on BPMN represents an interesting approach to this problem from a different perspective. By encouraging adoption of a common modeling notation to express the core components of process structure in a standard manner, it reduces some of the variety that constrains the production of a common machine interpretable process definition.

Once the core process structure has been modeled to a common notation it becomes more straightforward to specify the additional properties associated with resource assignments, data structures and so on.

One desirable extension from earlier work is the ability to specify additional internal interfaces for the integration of new component technology. The original reference model identified such interfaces to a directory server (for evaluating Organisational Model relationships) and process repository.

Rules evaluation was considered very much to be part of the core BPM functionality. However, current technology is producing flexible rules processing components that can be interfaced to BPM engines to enhance rules evaluation capability. This has significance in supporting complex evaluations for transition behaviour, resource allocation and, potentially, other functions. Adoption of rules engines into this space also supports more dynamic change by modification of the rules base independently of the core process.

However, one can overplay the significance of incomplete standards in this space. The reality is still the case that most organizations make use of process design and specification tools that are tightly bound to the process execution environment (e.g. products from a single BPM vendor). In some ways it is the need for common standards between organizations that is more important. Against this background, standards in the interoperability and choreography space are considered in the following section.

Choreography & External Process Interactions

Work in this area has been approached from two directions. Firstly, work on process specification standards has been recognizing the need to extend conventional [internal] process models to cope with external [B2B or B2C] process flows. Secondly e-business standards, traditionally focused on simple message exchange between organizations, have been expanding to con-

---

[13] Amongst efforts to define standard representations of business process flow are elements within IDIF, UML, RADs, PIF, PSL, WPDL, XPDL, XLANG, BPML, BPEL4WS

[14] See Process Modeling Notations and Workflow Patterns by Dr. Stephen White in this publication and Workflow Patterns: On the Expressive Power of Workflow Languages, W.M.P. van der Aalst, A.H.M. ter Hofstede

sider structured sequences of messages and the process implications behind such sequences.

At the heart of the debate is the relationship between messages, events and sequences and the requirements for both synchronous and asynchronous[15] relationships between executing process fragments.

There has been some debate about the extent to which the scope of all potential runtime interactions can be pre-defined in a Choreography. One school of thought assumes that all potential process interactions can be so scoped (and hence standard WSDL/SOAP based messaging operations may be adequate for web based interoperability).

The other school of thought believes that this approach will be impossible when dealing with large numbers of organisations and individuals, dynamically interacting through the web. Hence a generic process interoperability protocol such as Wf-XML will be fundamental—in the same way that HTTP has become fundamental as a generic protocol for transporting hypertext.

The scope of the choreography problem depends upon what assumptions one makes about the practical requirements for interactions between process fragments—remembering that the prime requirement is to support processes intermeshed across organizational boundaries.

At the most simplistic level one could assume that the internal structure of a fragment has no interactions with other process fragments whatsoever. In other words, apart from a mechanism for invoking the fragment and returning results on completion, the fragment execution is completely isolated from external factors.

At the other extreme one could envisage fragments whose entire internal structure is visible externally and where multiple interactions with other fragments may occur from anywhere within a fragment. In such cases the global [B2B] choreography could resemble a highly complex Petri-net or Pi-calculus model embracing all interactions crossing organization boundaries.

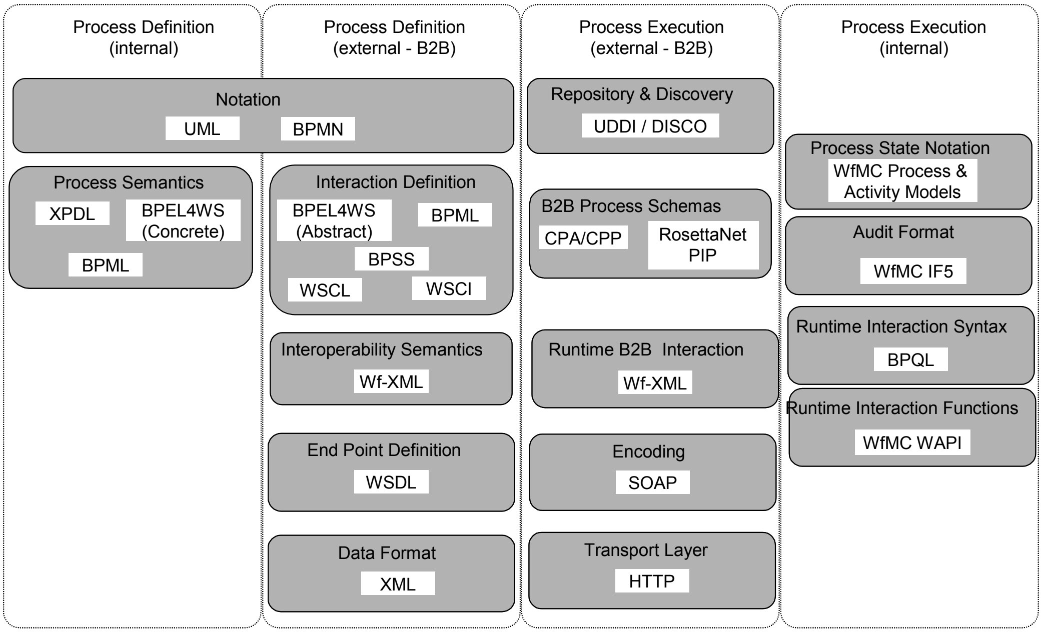
Most real world examples of B2B choreographies will lie somewhere between the two extremes, most likely at the simpler end. The WfMC made a simplifying assumption that most B2B process interactions will follow one of a small number of relatively simple models (hierarchic, chained or parallel synchronized). The WfMC Interface 4 (Wf-XML) defines a framework of process semantics (five process operations) that can be used across these models within a choreography.

Although no formal language has been defined to support choreography using these operations, several are represented by equivalent expressions in BPEL4WS (Invoke, Terminate, Assign) and the missing ones could be added relatively easily (Query, Notify). Future opportunities may arise for consolidation in this respect.

[15] Asynchronous Service Access Protocol – see XML Cookbook by Keith Swenson in this publication

THE STANDARDS CLASSIFICATION

The potential scope of standards that contribute to the BPM lifecycle model is wide. Several overlapping standards exist and hence one of the key problems for process architects and designers is to understand how the various initiatives relate. The following diagram was produced by the WfMC Technical Committee and provides a simple structure for classifying tools in accordance with the functional model described earlier.



This diagram is based on a standards “stack” aligned with the functional component breakdown from conceptual design tools [at the top] to specific interoperability protocols, formats and encoding [at the bottom]. In the vertical dimension it thus reflects the progression from abstract design and modeling, through to concrete process and message interactions.

In the horizontal dimension the diagram separates the *Process Definition* phase (1st & 2nd columns) and *Process Execution* aspects (3rd & 4th columns). It also separates *internal* and *external* (“B2B”) views of the process—in both definition and execution phases.

Looking briefly at each column in turn:

Internal Process Definition

In column 1 the main components that may need to interact in a standardised way are in the design and modeling domain. Hence the lower layers of the stack are void. The use of standards in this space is primarily focused on the integration of different software tools—for example enabling a process definition tool to pass a process definition to an execution environment. Often software from a single vendor environment will be used within a particular organisation or department for both purposes. Not all aspects of internal process behaviour will need to be standardised or made visible at external boundaries.

External Process Definition

In the *external* space the essential requirement is interoperability. At definition time this covers specification of the permitted business interactions between different process management systems.

In column 2 the upper layers are concerned with standards to support components for E2E process modeling and choreography. Below that are standards defining the semantics of process interoperability (process operations such as start, stop, query, etc)—this corresponds to the abstract service definition for WfMC interface 4 and the different interoperability models. Below that are the standards (based on web services) defining the service endpoints and data formats supporting interactions.

There is some significant overlap in the standards potentially applicable in this area, particularly in the Choreography area.

External Process Execution

Column 3 identifies the standards stack necessary to support process execution, starting with standards for discovery of external interoperability services, progressing through standards for the process schema(s) to support interoperability and then the specific standards to support runtime process interoperability. Currently the only identified protocol to support true process interoperability is Wf-XML.

However, it could be argued that other forms of messaging protocol could be incorporated at this level if they are clearly linked to process semantics. For example, numerous HL7 messages are defined for bilateral interaction in a healthcare environment[16] and are associated with specific trigger conditions which cause their initiation and specific actions [to generate defined outcomes] on their arrival. Many other vertical industries have similar messaging standards.

To the purist this may not be regarded as true process interoperability since there is no process id handle exchanged, to support subsequent process based actions between the systems. This is needed to provide asynchronous connection between the end systems, which then allows querying of the remote process state or actioning dynamic change in the process execution (for example to change context data or state). None the less such messaging will remain widely used in many industries, although simplifications are probably possible using a generic process interoperability protocol carrying industry specific context data payloads.

Internal Process Execution

The standards applicable in this space (column 4) are principally those that provide a common framework to support execution functionality. Not all BPM systems will support all aspects of these standards but they provide a basic level of functionality for achieving common interpretation.

[16] XML based healthcare message standards Health Level Seven, see www.hl7.org

At the highest level is the model of process and activity state notation, which is used to underpin most execution time activities—including API functions, audit data collection and query state (or any other state based interactions). The notation developed allows local extension of the basic WfMC defined states (open/closed/running /terminated etc).

All other standards make use of these state models (for process, and activity within process). The standards defined cover:

- audit data collection associated with various state change events, including internal resource assignments
- process or activity status query
- APIs for consistent access to BPM functions from client applications to query or set process, activity or worklist control data

A subset of this internal information, depending upon the policies of the BPM administrator, may be made visible at the external boundary of the process execution environment. This enables cooperating external systems to have visibility of internal states or audit data in a defined manner. This might typically be through a filter or mapping to a commonly agreed set of generic states.

CONCLUSIONS

In looking at the various components that make up a BPM reference system, much of the previous work of the original workflow reference model lives on. Some of the architectural gaps have become more apparent; some have been filled by other standards. Over time there has been a major shift in the technology used for realization of the abstract interfaces; most of the original architecture is now expressed in XML and as interfaces to web services.

One significant change presented in this paper is in the area of process fragments and the choreography of interactions between such fragments. Although the reference model did introduce the idea of distributed processes (and defined several types of interaction model) it never really tackled the problem of defining a notation for expressing their interaction—the province of the emerging choreography standards.

Other material changes in technology have emerged; recent discussions on the definition of standard interfaces for rules execution represent an interesting area of opportunity for future development.

So what of the plethora of standards currently filling the BPM space? The author expects some degree of rationalization over time.

The correct approach is to recognize what standards are needed where in the architecture, and for what purpose. Then they can be populated through the various industry and de jure standards bodies. Product vendors will adopt them if they add value—and this stems from having a thought through underlying architecture that clearly identifies the value and purpose of each standard.

Perhaps this is the core legacy of the Reference Model. At the very least it has provided a common framework for people to think about Workflow and BPM architecture and ten years of fascinating discussions!