The Workflow Management Coalition

A Common Object Model

Discussion Paper

Document Number - WfMC-TC-1022

Document Status - 2nd Draft for Discussion

January 98

Submitted by ICL Enterprises

Send comments to wmc_tc@ossi.com
or
d.c.hollingsworth@x400.icl.co.uk

Workflow Management Coalition
2 Crown Walk
Winchester
Hampshire SO22 5XE
United Kingdom
Tel: +44 1962 873401
Fax: +44 1962 868111
Email: wfmc@wfmc.org
Web Site: http://www.wfmc.org

Synopsis

This document describes the current status of the WfMC standardisation program and identifies the current work on object related standards which includes interface bindings for both OLE Automation and OMG/IDL objects. The scope of the current Reference Model is discussed, plus possible extensions to support a lower granularity component architecture, internal to the workflow enactment service, as a complement to existing work focused on the service functions provided at the boundary of a workflow enactment service. This is postulated on the basis of defining a common underlying object model which can be mapped to the two important component architectures emerging in the industry, the OMG object architecture and services and the ActiveX/DCOM architecture.

This document is a discussion paper only and does not represent WfMC views or constitute an agreed positioning with respect to any workflow object related standards.

Contents

1. INTRODUCTION

1.1 Background

The Workflow Management Coalition is a non profit organisation with the objectives of advancing the opportunities for the exploitation of workflow technology through the development of common terminology and standards.

The WfMC has pursued a policy of
1. concentrating its effort on standards relating strictly to workflow
2. "adopting" existing standards in related areas which complement the core workflow standards within its overall architecture, or, where such standards do not exist, working with other appropriate standards bodies to develop them
3. working on the basis of strict vendor neutrality by accommodating, where necessary, alternative implementation approaches within its core architecture framework

The WfMC has published a number of specifications including:
- The Reference Model - which provided the architectural framework for the initial program of standardisation, focusing on 5 interfaces
- Glossary & Terminology - providing a basis of common terminology and concepts
- WAPI Specification - Workflow APIs for access to common workflow services from client applications (covering interfaces 2, 3 and API functions within 1)
- Interoperability Specification - abstract specification and MIME bindings for process interoperability between workflow domains (covering interface 4)
- Audit Data Specification - a common model for workflow audit data and data formats for its recording

Work is well advanced on the specification for Process Definition Interchange (Interface 1), providing for the exchange of process definition data between business process modelling tools and workflow management systems.

1.2 Workflow and Object Technology

The Reference Model identified object based technology as one of the common target implementation models for workflow systems, although object technology was not mature enough to serve as the primary focus for the WfMC standardisation program at its inception. Hence the early work (on WAPI) concentrated on the definition of the interface functions and their specification as API calls in "C" as the key language binding. The interoperability specification was subsequently developed using IDL for the abstract specification and concrete bindings based on MIME, for use via Internet mail.

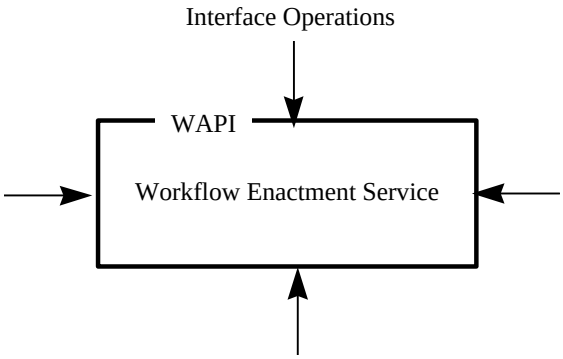
During 1995/96 the OMG expressed an interest in workflow standardisation and a program of liaison was initiated between the two organisations.

In parallel the WfMC commenced work on object orientated versions of several of its interfaces (2 and 4) based upon the use of IDL and OLE automation bindings as alternatives to the existing C and MIME specifications (these are currently at working draft status). During 1997 a number of WfMC members submitted a proposal to the OMG based upon the standards work of the WFMC. It is intended that alternative versions of the current WfMC

specifications will be published during 1998 providing for object based variants of the interfaces (OMG/IDL and OLE/COM).

2. CURRENT ARCHITECTURE

The initial basis of standardisation was to specify the workflow enactment service as a “black box” - effectively a large granularity object with various exposed interfaces through which other software objects can request services. This approach hides a great deal of the internal complexity of the workflow service which is difficult to standardise (- for example, the enactment service may be monolithic or composed of various distributed components, but this is masked from the interface functions).



A standardised API model (WAPI) is provided for communication between software applications and the workflow enactment service. The API model is essentially (and intentionally) independent of any underlying component distribution mechanism, since many different construction paradigms are used by workflow system vendors. The mechanism for underlying communication between components is not specified by the Coalition; however an underlying RPC service is a typical common approach, although some vendors provide interaction between various system components via email or via shared document/object stores, or other approaches. The WfMC WAPI specification assumes that vendors will provide appropriate stubs or code hooks to support client application access to the workflow enactment service from distributed platforms.

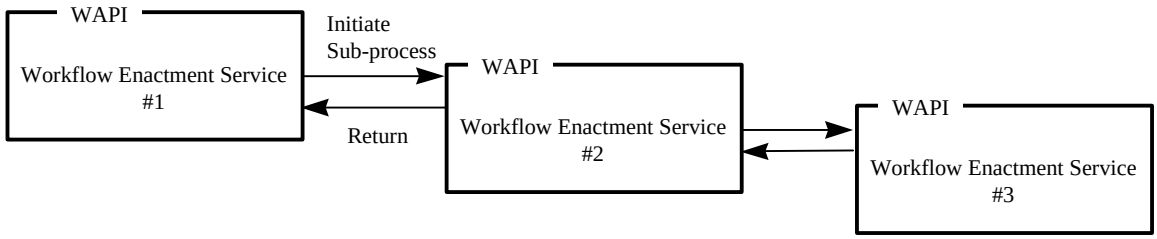
2.1 Current Interoperability Models

To support interactions between different workflow systems various levels of interoperability model were defined between workflow enactment services - in particular to allow the enactment of a single business process across several different workflow systems in different departments or organisations.

2.1.1 Hierarchic / Chained Sub Processes

Current standards are based upon the “simplest” level of interoperability - hierarchic or chained sub-processes. In this scenario the workflow services are modelled as separate, “loosely coupled” domains which each offer a defined range of capability to other workflow services. Essentially this provides the ability to enact a local sub-process whose process definition is local to the domain, but which is known [by external co-ordination] to other

domains such that they can invoke the sub-process remotely from within a “superior” process running within their own domain. On completion of the sub-process, control is either:
(i) returned to the superior process in the initiating domain (hierarchic model) or
(ii) the process is terminated, or control further transferred to another sub-process in a different domain (chained model)

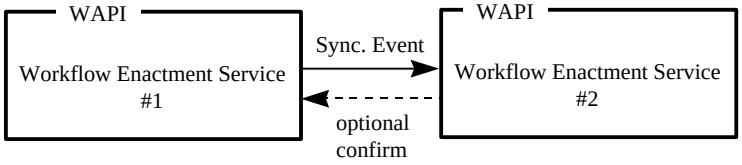


This approach greatly simplifies the co-ordination required between the workflow services. Each domain is responsible for its own user administration (including organisational model, where appropriate), audit data, internal process and activity naming, security[1], recovery, etc. Thus there is no general requirement for common control data to be maintained between domains. However, the corollary of this is that the two (or more) domains do not provide a single homogeneous service at their boundary, so that a business process must be structured into the different sub-processes supported within the various domains and the appropriate process definitions and any workflow relevant data organised and co-ordinated. The current interface specifications, including object bindings, are based upon this model.

This style of operation is well suited to certain electronic commerce applications where workflow interactions between different organisations occur on a pre-planned basis with each organisation responsible for service provision within their own domain with a relatively opaque interface between them. However, there are other styles of interaction where more flexibility and dynamism (for example in establishing binding relationships) are desirable.

2.1.2 Parallel Synchronised Processes

The parallel synchronised model provides support for synchronisation between parallel executing processes (or process threads) as an additional common service. In this style of operation processes run independently (for example, within different organisations) but co-ordination points are (pre-) defined within the processes at which synchronisation with one (or more) other process(es) is required.



Work within JSA has been progressing towards this model of operation. It is intended that WfMC will add support for the event and synchronisation standards during 1998.

[1] Proposals have been made for interoperability extensions to transfer certain common security related data between domains - see Workflow Security Considerations - White Paper.

As with the hierarchic sub-process model, this style of interworking can be achieved with relatively limited co-ordination between the various workflow services. It requires additional information to be added to the specifications on thread addressing (where parallel threads are supported within a process) and explicit facilities for achieving synchronisation (e.g. event handling for process suspension and rendezvous / resume for synchronisation). However, other aspects of administration can be delegated separately to each workflow domain, as with the hierarchic sub process scenario.

2.2 WAPI Object Bindings

The current WAPI specifications (for interfaces 1, 2 & 3) are defined (primarily against a C language environment) in terms of data structures and functions; operations are synchronous (although various asynchronous operations can be so initiated). List based operations are widely supported using a filter on list selection and individual fetch operations to retrieve information items. Handles are frequently referenced as a pointer to data structures to indicate context. The interoperability specification (interface 4) includes protocol exchanges conveyed through MIME; many of the underlying API functions are a common subset of the process control operations defined within WAPI (interface 2).

Object interfaces for both sets of functions are under development for both OLE/COM and OMG/IDL environments. (Due to various differences in the approach to interface operations some differences between OLE/COM and IDL are inevitable. There is also an ongoing question of the extent to which other operations available for objects in each environment should be subsumed into the basic underlying object technology and equivalent functions removed from WAPI).

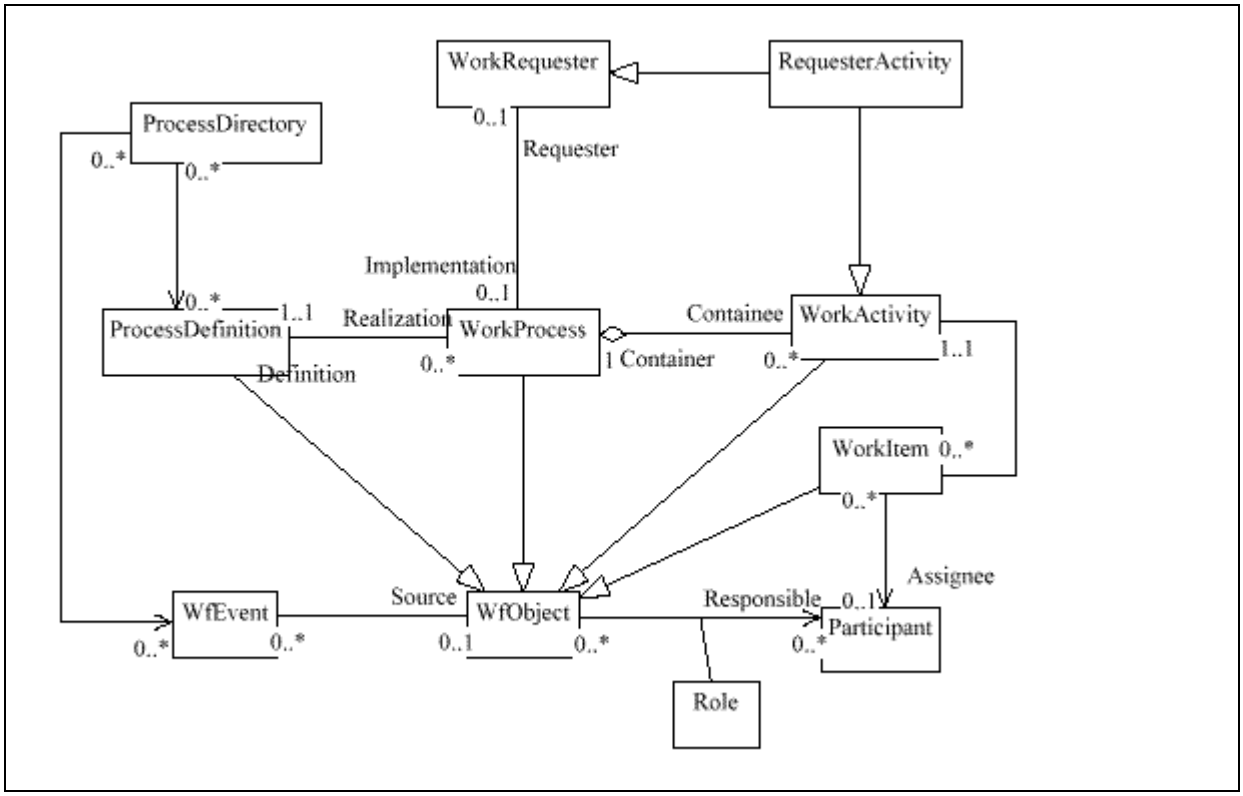
2.2.1 OLE / COM

An OLE automation interface consists of object classes, each with properties and methods, with a class hierarchy supporting navigation. The OLE automation binding for WAPI2 has been derived (essentially) by defining an OLE automation class and property for each WAPI data structure and defining a method on the appropriate object class. WAPI query handle is replaced by the use of OLE collection type objects and a filter object is defined to replace the WAPI Filter datatype, for use as a parameter to methods on collection type objects. Collection objects in OLE support a count parameter.

2.2.2 OMG/IDL

The IDL binding of the WfMC standards defines an object model that combines the Application Client Interface (IF 2), Interoperability Interface (IF 4) and the Auditing Data specification (IF 5); it also addresses the area of Invoked Applications, where applications are assumed to be Business Objects as defined by the emerging OMG Business Object Facility standard.

The following diagram shows the core interfaces defined by the jFlow specification (using UML notation); the interfaces are described in more detail below. Note that this is work in progress; a final version will be available in March 1998.



WfObject defines the attributes and operations common to all workflow entities. Each significant status change of a WfObject triggers a **WfEvent** to be published; WfEvents are distributed using the CORBA Event Service. The WfObject interface provides operations to access the History (i.e., the set of WfEvents produced by the object) of a WfObject. A WfObject is associated with one or more workflow **Participants**; the association is qualified by the **Role** of the Participant. **ProcessDefinition** represents a workflow process model and serves as a Factory for instances of that model.

ProcessDirectory is a locator for all Process Definitions within a Business System Domain. **WorkProcess** represents an instance of a workflow model; it provides operations to control process execution (start, terminate) and to access the Process Relevant Data associated with the process. A WorkProcess contains a set of **WorkActivities** that represent the process steps as defined in the underlying process model; this interface is primarily used for process monitoring and provides information on the status of a process step. A WorkActivity can be implemented by another WorkProcess; this case is handled by a **RequesterActivity** which provides a 'callback' interface that is used by the sub-process to report significant changes (e.g., completion) to the parent process. The implementation of a WorkActivity can be a WorkProcess that has no WorkActivities; this can be used to 'plug-in' applications that support the WorkProcess interface only. Each WorkActivity is associated with a set of **WorkItems** which represent the assignment of a particular workflow Participant to a process step.

2.3 Extended Interoperability (Peer-Peer)

The peer interoperability scenario models a workflow domain as embracing multiple distributed workflow components such that process enactment can be supported seamlessly across these components. In effect this requires support for a common component

construction architecture which offers flexibility in deployment as a basis for building a common solution inside the enactment service (i.e. an architecture internal to the “black box”).

This was not an early standardisation target for the WfMC due to the different construction paradigms within the industry and the relative immaturity of component construction technology. However, more recently interest has been expressed within the industry in the potential for adoption of such a common architecture to exploit industry standard components such as those provided by the OMG common object services and the ActiveX COM/DCOM model.

3. OPPORTUNITIES FOR A COMMON OBJECT MODEL

As part of its general work on object orientation, the WfMC has been reviewing the scope of its Reference Architecture, to facilitate more effective use of common object services and improved component integration in implementation scenarios.

3.1 Common Component Services

As the requirements grow for the WFMC to address common infrastructure issues such as security, recovery, naming and directories, administration, etc. (which is a continued theme from large user focused organisations such as BFG), there is increasing overlap with the functionality provided by distributed object technologies. The WfMC would not wish to duplicate industry work in these areas but rather seek to exploit it as a component of its own architecture and recommend a standards set for usage within the workflow arena.

Particular areas which can be of benefit include the following (non-exhaustive):

dynamic binding - currently the WfMC interoperability specifications are based on static pre-defined relationships between different workflow environments. The ability to establish dynamic bindings with location service between different workflow components or between different workflow services is an important area of flexibility which will become increasingly important as workflow applications become more prevalent in the electronic commerce marketplace.

recovery and transactioning - this is becoming important for certain customers / vendors (i) within a workflow service and (ii) in co-ordination between a workflow service and conventional TP applications (typically legacy). Transaction semantics alone are normally inadequate for complex workflow systems - which also require support of other recovery functions such as compensation or alternative transactions - but are none-the-less an important underlying technology for the construction of certain classes of workflow solution. Industry standard transaction models are now supported within both OMG (OTS) and DCOM (MTS) which can provide underlying transaction support for use in workflow solutions.

security - the WfMC is embarking on extensions to its interoperability specification to incorporate various security services over MIME. However, where other standard underlying communications services are employed security provisions are often an integral part of such

services. This provides the opportunity for WfMC to adopt existing security models such as those provided within DCOM security provider and OMG security services.

administration - this can be a complex area with most workflow services requiring to maintain directories of user names, roles, applications, an organisation model, process definitions, etc. Much of this administration is potentially common with other underlying platform or network related functions, with only certain properties and functions being inherently workflow based. This is a further area where commonality with underlying distributed architectures can both reduce the complexity of the workflow software and improve the integration with other related distributed services such as document management, email, etc.

3.2 Internet-centric Workflow

The WfMC has recently published a White Paper "Workflow & Internet, Catalysts for Radical Change" [Martin Ader]. This identifies how the convergence of workflow and Internet technologies can be complementary in establishing a framework for the control of business process within an open, electronic trading environment. Much of the standardisation required to support this has already been achieved, although there are certain areas where a more distributed workflow control model can bring additional value to this style of operation. *[See comments from the WfMC meeting, Windsor 1997.]*

1. Distribution of work items via a "push" interface to facilitate scaleable operation in very large public networks. Associated with this is the potential visibility of worklists as network addressable objects in their own right.

2. The ability to transfer a business process (for example, to a service provider within an open network domain) as a work object in its own right, as an alternative to an individual activity or workitem. The current sub-process model constrains this by a pre-partitioning of the business process model into sub-processes aligned typically to organisational boundaries. A dynamic form of binding a (sub-)process to a service provider organisation is desirable to provide this flexibility. The implication is that some form of standardised representation of the operational business process instance is required, which could be dynamically transferred during process enactment *[a non-trivial problem; note that XML may also have a role to play in this area]*.

These requirements are broadly in line with elements of the peer-peer interoperability model described within the *WfMC Reference Model* document and discussed earlier in this paper. Ideally the above requirements would be met by a general abstraction of work units ranging from a business process to an individual work item.

[It is, of course, important that any solution should not constrain the choice of workflow technology within a service domain so these aspects of operation should be independent of the use of particular component technology by various service providers.]

4. APPROACH TO A COMMON OBJECT MODEL

4.1 General Policy

It is not the aim of the WfMC to abandon its general approach to workflow standards in favour of one based solely on the use of object technology but rather to recognise the increasing importance of the latter and to facilitate its effective adoption within the workflow arena. Furthermore, in more complex interoperability scenarios there is a requirement for considerably more common state data and common underlying services and it is likely that such complex scenarios can only be efficiently supported through the increasing adoption of common component technologies to support reuse.

In keeping with its aims of vendor neutrality the WfMC identified the possibility of specifying a common object based model for workflow which can be mapped into several underlying object technologies for implementation, using appropriate variants for different object environments. Such an approach will become increasingly important when more complex interoperability scenarios are required to be supported. In such cases the ability to construct a single unified workflow service from a common component architecture embracing many distributed services and common methods becomes very attractive.

4.2 Alternative Technologies

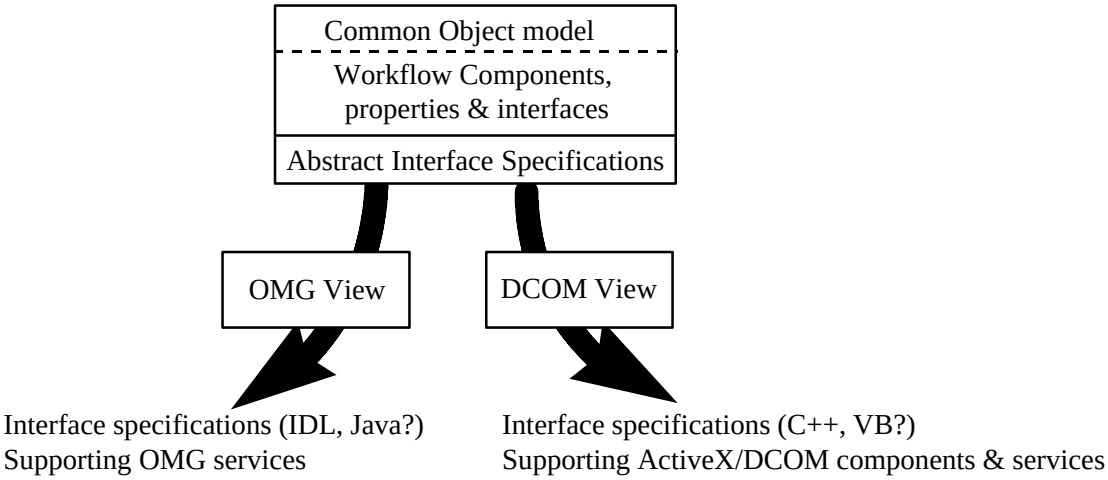
There are currently two important object based component architectures emerging within the market:

1. The OMG object model and related common object services linked via the CORBA interoperability model and underlying RPC communications
2. The ActiveX/COM/DCOM object model and component architecture which from an initial focus on document centric operation is broadening into a wider object oriented application construction environment.

The WfMC has followed the approach of offering both IDL and OLE Automation variants of its WAPI specifications; both mappings are provided at working draft status in the WAPI V2 specifications.

There is now scope to go somewhat further and embrace a general object model within the core architecture which could be mapped to both OMG and DCOM component technologies.

As noted earlier, in addition to the “simple” WAPI interface to a monolithic workflow service, it becomes possible to architect a number of important component boundaries “internal” to the workflow service and in relation to other related common services - to enable the workflow system to be distributed as a series of components for flexible deployment and operation. It is for the WfMC to decide how far it should move in this direction, but this discussion paper assumes that some extension to its Reference Model is desirable.



Within this approach it would still be intended to preserve a common external interface to a workflow service (WAPI), which would be expressed in whatever concrete language bindings are appropriate, supporting their use in OMG, DCOM and non-object based environments. The extensions to the architecture dealing with wider components within the workflow service would be scoped in terms of general workflow service components, their properties and interactions, independently of the underlying implementation environment. It is considered likely that such “tightly coupled” workflow components will be realised typically within a homogeneous component architecture.

4.3 Interoperability and gateways

Standards for interworking between different workflow domains (“loosely coupled”) will continue since for many organisations different workflow technologies will be deployed and will be perfectly adequate to support distributed process operations between the organisations. This area will require some further work to develop appropriate common interoperability mappings over the important communications services, which will include both email and RPC-based services and Internet enabling.

Gateways will also continue to be required where there is no appropriate common underlying communications mechanism available between different workflow services; such gateways are likely to be encapsulated within object adaptors and/or as agent modules integrated behind a local WAPI 2 interface.

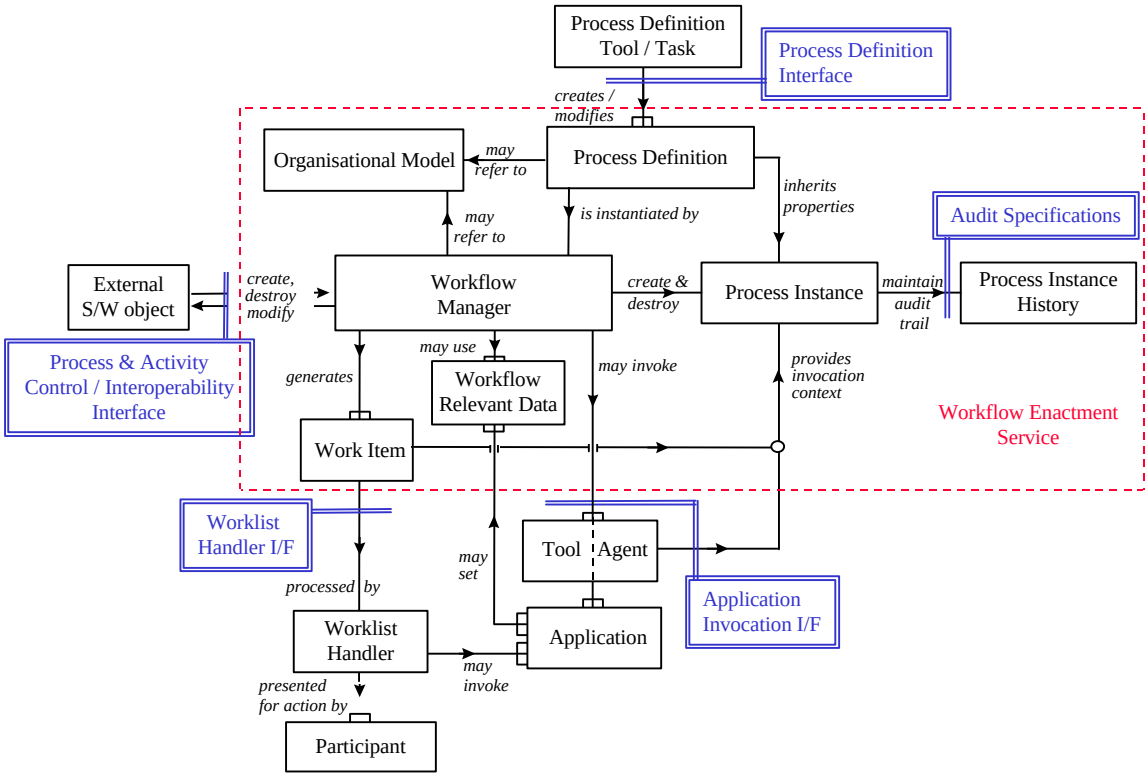
5. ELEMENTS OF A COMMON OBJECT MODEL

The WfMC has already identified aspects of its architecture in OO terms, as part of the work done by members to support the OMG submission; this work is based on the currently defined interoperability model of loose coupling between workflow services and the defined set of interface methods corresponding to the existing WAPI and Interoperability functional specifications *[see Jflow - Workflow Management submission to OMG]*. Several areas of opportunity have been identified to adopt common object services as part of such an architecture and there is also the opportunity to add further interfaces and common methods internal to a workflow domain - although this is not part of the current submission. Much of

this work done to date is essentially independent of the underlying object technology, which is assumed to be CORBA based using IDL bindings.

5.1 An Object View of WAPI

It specifying such an architecture one complexity has been understanding the granularity of objects specified within the model (e.g. at a business object level the workflow system may be regarded as a single object). In current WfMC interface specifications a workflow service is essentially a black box object supporting various interfaces supporting particular methods on lower granularity objects. It has also proved necessary to conceptualise the contents of a process definition as an object set, though these are not currently standardised.



An above representation of a workflow system identifies the major objects and their relationships. This is an oversimplification but shows how the enactment service is treated as a black box entity with specific interface procedures available at its periphery to external software entities. The process and activity control interface may be to an external controlling application (WAPI2) or to another Workflow Manager (Interface 4). The audit interface is essentially an internal specification of audit requirements on a process instance history object.

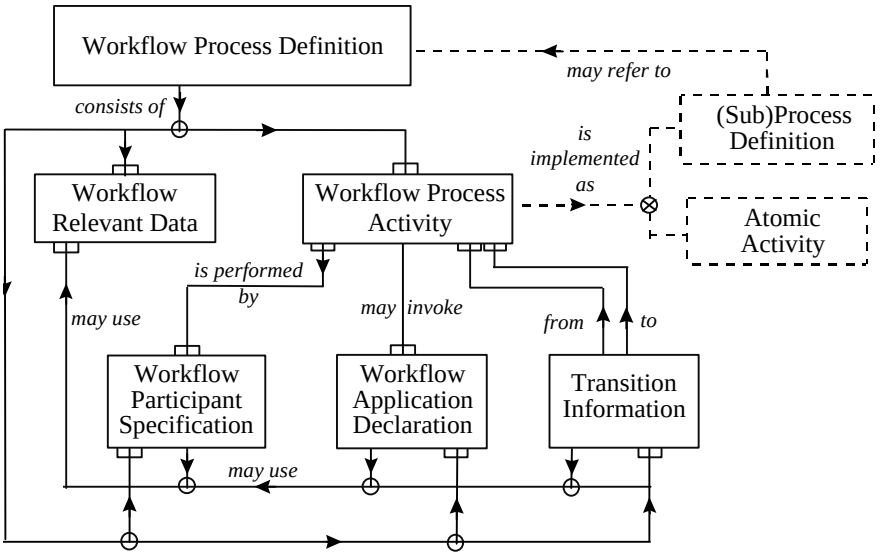
A further distinction could be made between applications invocation directed from the workflow manager (typically out of user context and invoked via a tool agent module) and applications invocation directed from a worklist handler or directly by the participant (and typically invoked in the user context of the specific participant to whom the work item is directed).

The set of interface operations supported are defined in the WAPI specification and are grouped into

- Connection functions(common)
- Process & Activity Control and status operations
- Worklist operations
- Administrative functions
- Application invocation functions
- Process Definition manipulation functions

5.2 The Process Definition Model

The process definition is the persistent object from which is derived the run-time characteristics (in whole or part) of the process instance being enacted. The WfMC has not attempted to standardise an internal representation of the process definition (the local format within which process definition data is held by a workflow manager or process repository) since this is typically vendor or product specific. Instead, work has concentrated on defining a meta-model which identifies the core process definition entities, relationships and attributes plus a neutral interchange representation of the process definition (in a textual grammar form known as WPDL). Additionally, the WAPI interface includes APIs to retrieve, delete and add entities and get/set attributes using the entities and attributes defined in the meta-model.



This simple representation of the meta-model identifies the top level objects and relationships. The (many) attributes are defined within the Interface 1 specification.

Note that the workflow process activity entity includes any required activity “work script” (i.e. identifying the relationship between the activity, work-items, participants and applications); the WfMC has not attempted to standardise a definition of this aspect of workflow behaviour.

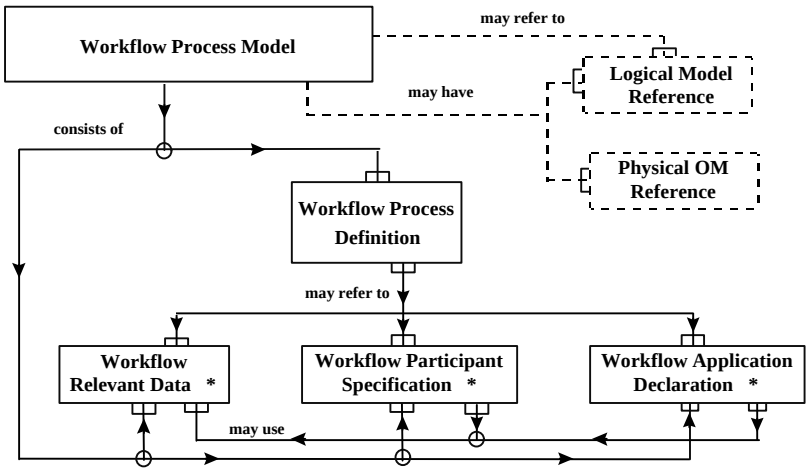
Certain important characteristics of the workflow process activity object may be mentioned:

1. An Activity is the smallest atomic unit of work which has conditional expressions associated with its selection / invocation and between which navigation takes place
2. Resource assignment is at the activity level; multiple resources may be assigned (as a resource set) within an activity or a single resource may be assigned
3. An activity work script relates Applications / Participants and work items. Two scenarios are identified *[although this aspect of workflow operation is not standardised]*:

- Sequential work-items - the work items are created sequentially and all assigned to the same resource set (i.e. a common participant set)
- Parallel work items - the work items are created and assigned to the participant set for parallel processing

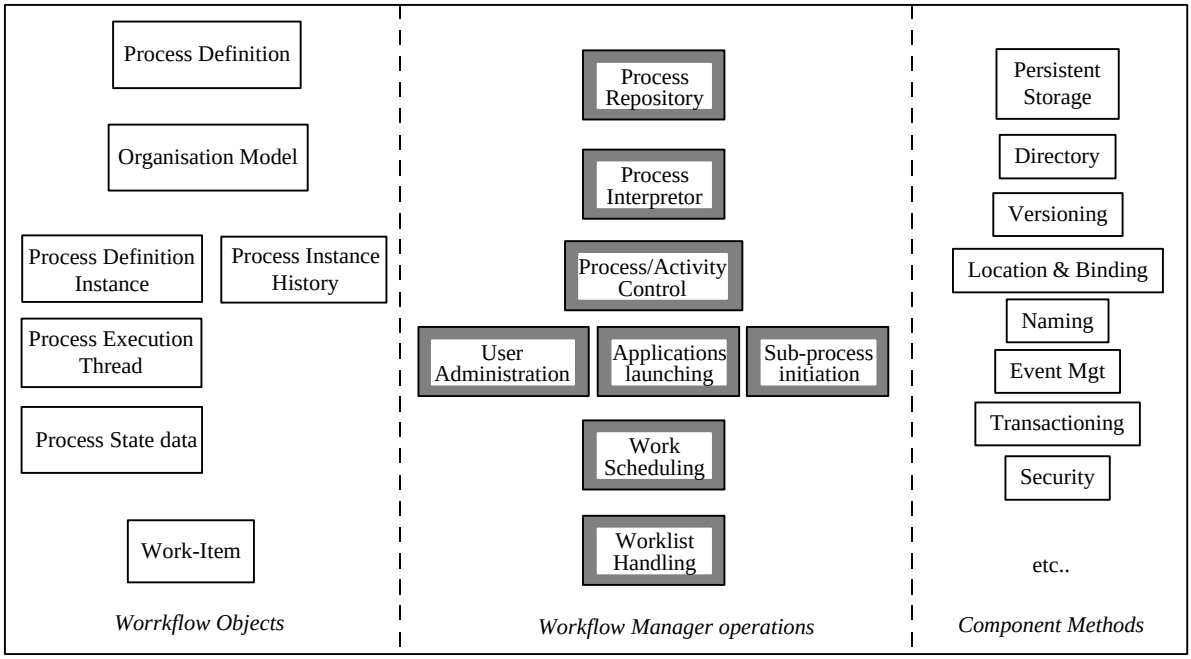
In both cases the activity is not completed until all work items are completed; at this point navigation to the next activity (or activities) may occur

A further level of refinement is defined within a Process Model, essentially a container for objects common to multiple Process Definitions, to avoid repeated specification of their attributes. This has the objects and relationships shown in the diagram following:



5.3 Workflow Manager Component Model

The additional component level objects which may be defined lie essentially inside the large granularity object identified as the workflow manager and have relationships with the process definition instance entities defined at run-time to control the enactment of the process. This model is based upon a decomposition of functions common to a workflow manager, a definition of the interfaces between such functions, and a view of how various standardised component methods may be exploited in the construction of the workflow manager functions.



Various detailed component methods can be adopted to support the internal construction of the workflow enactment service, allowing distribution of the workflow manager (and other elements from the enactment service) and location independence of the components.

In this way a fully distributed workflow enactment service could be constructed to exploit object component technology and support a full, peer-peer style of interoperability in which process enactment could seamlessly operate across different platforms.

Much of the detailed work to be done lies in researching the opportunities for component level objects and refinement of the above model.

6. NEXT STEPS

It is for the WfMC to identify whether, and how, to take forward requirements relating to component technology architecture, and identify other extensions to its framework architecture to support more flexible, dynamic operation in a multiple service provider domain. Since the program of work is potentially large the WfMC will need to consider carefully how it approaches this in co-operation with other industry groups.

1. Ongoing discussions with both the OMG and ActiveX consortium are likely to be required to identify core WfMC sponsored work and the adoption / consolidation of related technology standards from other industry bodies

2. This should facilitate a better understanding of the current and planned industry approaches to standardised component technology offerings.

3. The basic architecture of the WfMC will need extension to clarify:
   - the boundaries of a workflow enactment service,
   - facilities for more dynamic interaction between different services, and

- the extent to which object component technologies can be deployed in standardised way (i) inside a single workflow service constructed of different vendor components, and (ii) as a means of supporting more flexible and dynamic workflow operations between workflow component technologies across multi-organisation operational boundaries.

In approaching this it will be important to maintain a separation between the operational perspective (e.g. organisational and service interfaces and the encapsulation of service functionality behind such interfaces) and the construction perspective (e.g. deployment of individual component technologies to construct individual service elements). This style of separation may already have been considered in related industry consortia groups - as a basis for services other than workflow.

4. It is postulated that this could lead to the definition and adoption of a common underlying object model which is capable of mapping (from a construction perspective) to both OMG and ActiveX component models.

