Workflow Management Standards & Interoperability

Workflow Management Coalition

and

Keith D Swenson
Fujitsu OSSI
kswenson@ossi.com

Introduction

Workflow Management (WfM) is evolving quickly and expolited increasingly by businesses in a variety of industries. Its primary characteristic is the automation of processes involving combinations of human and machine-based activities, particularly those involving interaction with IT applications and tools. Although its most prevalent use is within the office environment in staff intensive operations such as insurance, banking, legal and general administration, etc., it is also applicable to some classes of industrial and manufacturing applications.

Many software vendors have WfM products available today and there is a continual introduction of more products into the market. The availability of a wide range of products within the market has allowed individual product vendors to focus on particular functional capabilities and users have adopted particular products to meet specific application needs. However, there are, as yet, no standards defined to enable different WfM products to work together, which is resulting in incompatible "islands" of process automation.

The Workflow Management Coalition (WfMC) is a grouping of companies who have joined together to address the above situation. Most WfM products have some common characteristics, enabling them potentially to achieve a level of interoperability through the use of common standards for various functions. The WfMC was established to identify these functional areas and develop appropriate specifications for implementation in workflow products. It is intended that such specifications will enable interoperability between heterogeneous workflow products and improved integration of workflow applications with other IT services such as electronic mail and document management, thereby improving the opportunities for the effective use of workflow technology within the IT market, to the benefit of both vendors and users of such technology.

Workflow Systems Overview

Despite a tremendous variety in WfM systems available, they all exhibit certain common characteristics, which provide a basis for developing integration and interoperability capability between different products. All WfM systems may be characterized as providing support in three functional areas:

- Build-time functions, concerned with defining, and possibly modelling, the workflow process and its constituent activities
- Run-time control functions concerned with managing the workflow processes in an operational environment and sequencing the

various activities to be handled as part of each process

- Run-time interactions with human users and IT application tools for processing the various activity steps

These components exchange information in a variety of ways, which leads the WfMC to standardize 5 main interfaces:

1. specifications for process definition data and its interchange
2. interfaces to support interoperability between different workflow systems
3. interfaces to support interaction with a variety of IT application types
4. interfaces to support interaction with user interface desktop functions
5. interfaces to provide system monitoring and metric functions to facilitate the management of composite workflow application environments

Product Implementation Model

The **Process Definition Tool** is used to create the process description in a computer processable form. This may be based on a formal process definition language, an object relationship model, or in simpler systems, a script or a set of routing commands to transfer information between participating users. The definition tool may be supplied as part of a specific workflow product or may be part of a business process analysis product, which has other components to handle analysis or modeling of business operations. In the latter case there must be a compatible interchange format to transfer the process definitions to and from the run-time workflow software.

The **Workflow Enactment Software** interprets the process description and controls the instantiation of processes and sequencing of activities, adding work items to the user work lists and invoking application tools as necessary. This is done through one or more co-operating workflow management engines, which manage(s) the execution of individual instances of the various processes. The workflow enactment service maintains internal control data either centralized or distributed across a set of workflow engines; this workflow control data includes the internal state information associated with the various process and activity instances under execution and may also include checkpointing and recovery/restart information used by the workflow engines to co-ordinate and recover from failure conditions.

The workflow engines also include some form of application tool invocation capability to activate applications necessary to execute particular activities. The generality of such mechanisms may vary greatly, with some simple systems only offering support of a single fixed tool such as a form or document editor, whereas others may provide methods for the invocation of a wider range of tools, both local and remote to the Workflow engine.

Where process navigation decisions, or other control operations within the workflow engine, are based on data generated or updated by workflow application programs, such data is accessible to the workflow engine and termed **workflow relevant data** (also known as "case data"); this is the only type of application data accessible to the workflow engine. **Workflow application data** is manipulated directly (and only) by the invoked applications, although the workflow engines may be responsible for transferring such data between applications (if necessary), as different applications are invoked at different activity points within the workflow process.

Where user interactions are necessary within the process execution, the workflow engine(s) place items on to **worklists** for attention by the worklist handler, which manages the interactions with the workflow participants. The worklist handler is a software component which manages the interaction between workflow participants and the workflow enactment service. It is responsible for progressing work requiring user attention and interacts with the workflow enactment software via the worklist.

Within a workflow system there are a number of supervisory functions which are normally provided; these are typically supported on the basis of supervisory privilege to a particular workstation or user(s). These functions may enable supervisors to alter work allocation rules, to identify participants for specific organizational roles within a process, to track alerts for missed deadlines or other forms of event, to trace the history of a particular process instance, to inquire about work throughput or other statistics, etc. Where distributed workflow engines are used there may need to be specific commands to transfer such control operations or (partial) responses between different workflow engines to provide a single administrative interface.

In a concrete product implementation the structure of implementation may be realized in a variety of different ways; among the main alternatives considered are:

- centralised or distributed workflow enactment service
- worklist handler location(s) and distribution mechanism

The approach taken by the Coalition is to define a boundary around the workflow enactment service, which exhibits various standard functional attributes accessible via a set of common APIs. The internal mechanisms by which the enactment service delivers this capability are not defined and may include one or more homogenous workflow engines, communicating in a variety of ways.

To support interworking between different products, interfaces are defined for specific co-operative functions between different enactment services so that a composite multi-vendor workflow application may execute parts of a particular process on different enactment services (each comprising one or more specific vendors workflow engines). This is considered a more realistic approach (except perhaps in the long term) than attempting to standardize the internal interfaces

There are various possible product implementations of this worklist interaction model depending upon the nature of the product implementation and, in particular, on the type of infrastructure used to support the distribution of worklist handling. The four example scenarios are as follows:

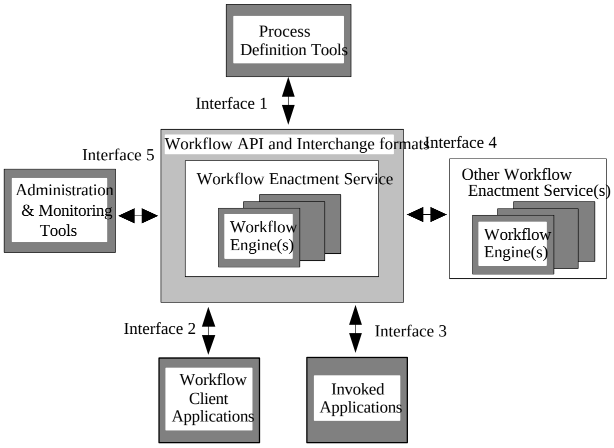
- Host based Model - the client worklist handler application is host based and communications with the worklist via a local interface at the workflow engine. In this case the user interface function may be driven via a terminal or a remote workstation.
- Shared filestore model - the worklist handler application is implemented as a client function and communication is via a shared filestore, which lies on the boundary between host and client platform environments and is accessible to both.
- Electronic mail model - communication is via electronic mail, which supports the distribution of work items to individual participants for local processing. In this scenario the worklist would normally lie at the client.
- Procedure Call or Message Passing model - communication is via procedure call, or other message passing mechanism. In this scenario the worklist may be physically located on the workflow engine or at the worklist handler according to the particular implementation characteristics.

In each case it is feasible to construct a common API, which supports worklist handler access to the worklist and workflow engine functions, but which is located behind a specific worklist access function appropriate to the product implementation style.

The Five Interfaces

The above figure illustrates the major components and interfaces within the workflow architecture.

The **workflow enactment service** provides the run-time environment in which process instantiation and activation occurs, utilizing one or more workflow management engines, responsible for interpreting and activating part, or all, of the process definition and interacting with the external resources necessary to process the various activities. A **workflow engine** is responsible for part (or all) of the runtime control environment within an enactment service.


Process Definition Tools
Interface 1
Workflow API and Interchange formats
Interface 4
Interface 5
Administration & Monitoring Tools
Workflow Enactment Service
Workflow Engine(s)
Other Workflow Enactment Service(s)
Workflow Engine(s)
Interface 2
Interface 3
Workflow Client Applications
Invoked Applications

The Workflow Application Programming Interface (WAPI) may be regarded as a set of API calls and interchange functions supported by a workflow enactment service at its boundary for interaction with other resources and applications. Although this architecture refers to 5 "interfaces" within WAPI, a number of the functions within each of these interfaces are common (for example process status calls may be issued from the client application interface or the administration interface). The WAPI is thus being defined as a common core of API calls /interchange formats with specific extensions where necessary to cater individually for each of the five functional areas.

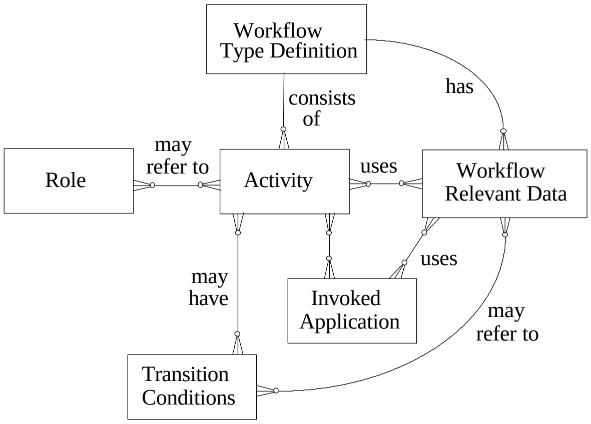
A variety of different tools may be used to analyze, model, describe and document a business process. Where a workflow product provides its own process definition tool, the resultant process definitions will normally be held within the workflow product domain and may, or may not, be accessible via a programming interface for reading and writing information. Where separate products are used for defining and executing the process, the process definitions may be transferred between the products as and when required or may be stored in a separate repository, accessible to both products (and possibly other development tools).

The interface between the modeling and definition tools and the runtime workflow management software is termed the **process definition import/export interface**. The nature of the interface is an interchange format and API calls, which can support the exchange of process definition information over a variety of physical or electronic interchange media. The interface may support the exchange of a complete process definition or a subset - for example a set of process definition changes or the attributes of a particular activity within the process definition.

There are two aspects to the Coalition's work in this area:

1. derivation of a meta-model which can be used to express the objects, their relationships and attributes within a process definition and which can form the basis for a set of interchange formats to exchange this information between products
2. API calls (within the WAPI) between workflow systems or between a workflow system and process definition product, providing a common way to access workflow process definitions. Access may be read, read/write or write only and may manipu-

late the set of standard objects defined within the meta-model or a product-specific set (for example defined in a product type register).



A set of API commands within WAPI is under development to support access to process definition data. It is expected that such specifications will cover a number of functions of the following general types. Commands are expected to be provided which operate on a list, or on individual objects or attributes.

Interface 2 provides general access for workflow enabled applications. The functions provided by interface 2 include session establishment, process starting and termination, actions on process instances, such as state changes, process instance status, worklist retrieval, retrieval / update of workflow relevant or application data, and administrative functions.

Interface 3 provides a standard way for a WfM system to invoke an external application. It is intended to be applicable to application agents and (longer term) applications which have been designed to be "workflow enabled" (i.e. to interact directly with a workflow engine). In the simple case, application invocation is handled locally to a workflow engine, using information within the process definition to identify the nature of the activity, the type of application to be invoked and any data requirements. The invoked application may be local to the workflow engine, co-resident on the same platform or located on a separate, network accessible platform; the process definition contains sufficient application type and addressing information (specific to the needs of the workflow engine) to invoke the application. In this case the conventions for application naming and addressing are local between the process definition and the workflow engine.

A key objective of the coalition is to define standards that will allow workflow systems produced by different vendors to pass work items seamlessly between one another. This is the job of interface 4.

Interface 5 defines a common interface standard for administration and monitoring functions which will allow one vendor's management application to work with another's engine(s). This will provide a common interface which enables several workflow services to share a range of common administration and system monitoring functions.

Interoperability Scenarios

Workflow products are diverse in nature ranging from those used for more ad-hoc routing of tasks or data to those aimed at highly regularized production processes. The work of the Coalition has therefore focused on developing a variety of interoperability scenarios which can operate at a number of levels from simple task passing through to full workflow application interoperability with complete interchange of process definition, workflow relevant data and a common look and feel. In this area it is expected that relatively simple interoperability scenarios will be supported initially, with the more complex situations requiring further work on interoperability definitions. Four possible interoperability models has been identified, covering various (increasing) levels of capability.

The **Connected Discrete (Chained) Scenario** allows a connection point in one process to connect to another point in another process. It is useful to think to these connection points as being the terminus and starting points of the processes, but for full generality it is presumed that the connection points can be anywhere within the processes. This model supports the transfer of a single item of work (a process instance or activity) between the two workflow environment.

The **Hierarchical Scenario** allows a process executed in a particular workflow domain to be completely encapsulated as a single task within a (superior) process executed in a different workflow domain. A hierarchic relationship exists between the superior process and the encapsulated process, which in effect forms a sub-process of the superior. The hierarchic relationship may be continued across several levels, forming a set of nested sub-processes. Recursion within this scenario may, or may not, be permitted by individual product implementations.

The **Connected Indiscrete (Peer-to-Peer) Scenario** allows a fully mixed environment; for example one process might include activities which may be executed across multiple workflow services, forming a shared domain. In this scenario, the process would progress transparently from task to task, without any specific actions by users or administrators, with interactions between the individual workflow engines taking place as necessary.

The **Parallel Synchronized Scenario** allows two processes to operate essentially independently, possibly across separate enactment services, but requires that synchronization points exist between the two processes. Synchronization requires that once the processes each reach a predefined point within their respective execution sequences, a common event is generated. This type of mechanism may be used to facilitate functions such as process scheduling across parallel execution threads, checkpointing of recovery data or the transfer of workflow relevant data between different process instances.

There are two major aspects to the necessary interoperability:

- the extent to which common interpretation of the process definition (or a subset) is necessary and can be achieved
- runtime support for the interchange of various types of control information and to transfer workflow relevant and/or application data between the different enactment services

Where both enactment services can interpret a common process definition, for example generated from a common build tool, this enables both environments to share a single view of the process definition objects and their attributes. This would include activity, application, organization and role names, navigation conditions, etc. This potentially enables individual workflow engines to transfer execution of activities or sub-processes to heterogeneous workflow engines within the context of a common naming and object model. This approach is particularly applicable to interoperability scenario 3, where several systems are co-operating at peer level, although can also be employed in simpler scenarios.

Summary

The Workflow Management Coalition is moving forward on providing a reference model architecture as a framework for comparing and integrating systems from different vendors. Fujitsu is an active participant in this coalition with the aim that Regatta Technology, Fujitsu’s workflow offering, can interoperate with other systems, regardless of the vendor. This is the goal of the coalition, and one that should be very welcome to end users. Those evaluating workflow systems should ask the vendor whether the system is “open” by supporting these common interfaces and models. Such a system has a far higher chance of being extended and customized in the future.

Acknowledgements

This document was the joint effort of many individuals of the Workflow Management Coalition. Most significantly David Hollingsworth from ICL who served as the chairman of the Reference Model working group for the development of the first version of the reference model, from which most of this information was taken. It could not have been completed without the hard work and dedication of many individuals including Gene Pierce, Dave Shorter, Mike Grabert, Raul Medina-Mora, Robin Maxwell, Jon Pyke, Kurt Ogris and many more people, too many to mention here.

WfMC Contact Information

Workflow Management Coalition
Avenue Marcel Thiry 204
1200 Brussels, Belgium
Tel: +32 2 774 96 33
Fax: +32 2 774 96 90
100113.1555@compuserve.com

Appendix - Terminology

Below is an abbreviated glossary of terms that are being standardized by the WfMC. The WfMC Glossary document, available from the secretariat in Brussels, includes an expanded explanation and the usage of these terms, including synonyms.

AND-Join: When two or more parallel executing activities converge into a single thread of control.

AND-Split: When a single thread of control splits into two of more threads in order to execute activities in parallel.

Application Data: Data that is application specific and not accessible by the workflow management system.

Audit Trail: A historical record of the state transitions of a workflow process instance from start to completion or termination.

Business Process: A kind of process that supports and/or is relevant to business organizational structure and policy for the purpose of achieving business objectives.

Iteration: A workflow process activity cycle involving the repetitive execution of workflow process activity(s) until a condition is met.

Manual Process Activity: The manual process steps that contribute toward the completion of a process.

Manual Process Definition: The component of a process definition that cannot be automated using a workflow management system.

Manual Process Execution: The duration in time when a human participant and/or some non-computer means executes the manual process instance.

Manual Process Instance: Represents an instance of a manual process definition which includes all manual or non-computerized activities of a process instance.

Organizational Role: A synergistic collection of defined attributes, qualifications and/or skills that can be assumed and performed by a workflow participant for the purpose of achieving organizational objectives.

OR_Join: When two or more workflow process activities physically connect or converge to a single activity. In this case there is no synchronization of the threads of control from each of the two or more workflow process activities to the single activity.

OR_Split: When a single thread of control makes a decision upon which branch to take when encountered with multiple branch(es) to workflow process activities.

Parallel Routing: A segment of a workflow process instance where workflow process activity instances are executing in parallel and there are multiple threads of control.

Process: A coordinated (parallel and/or serial) set of process activities that are connected in order to achieve a common goal. A process activity may be a manual process activity and/or a workflow process activity.

Process Activity: A logical step or description of a piece of work that contributes toward the accomplishment of a process. A process activity may be a manual process activity and/or an automate workflow process activity.

Process Activity Instance: An instance of a process activity that is defined as a part of a process instance. Such an instance may be a manual process activity instance and/or a workflow process activity instance.

Process Definition: A computerized representation or model of a process that defines both the manual process and the automatable workflow process.

Process Definition More: The time period when manual process and/or automated (workflow process) descriptions of a process are defined and/or modified electronically using a process definition tool.

Process Execution: The duration in time when manual process and workflow process execution takes place in support of a process.

Process Instance: Represents an instance of a process definition which includes the manual process and the automated (workflow process).

Process Role: A synergistic collection of workflow process activities that can be assumed and performed by a workflow participant for the purpose of achieving process objectives.

Sequential Routing: A segment of a workflow process instance where workflow process activities are executed in sequence.

Sub Process Definition: A process that is called from another process or sub process that includes the manual process and the automated (workflow process) components of the process.

Tool: A workflow application that interfaces to or is invoked by the workflow management system via the workflow application programming interchange/interface.

Transition Condition: Criteria for moving, or state transitioning, from the current workflow process activity to the next workflow process activities in a workflow process.

WAPI: The application programming interface/interchange for client workflow applications and tools in order to be able to interface to the Workflow Enactment System. WAPI is the acronym for Workflow Application Programming Interface/Interchange.

Work Item: Representation of work to be processed in the context of a workflow process activity in a workflow process instance.

Work Item Pool: A space that represents all accessible work items.

Workflow Application: A software program that will either completely or partially support the processing of work items in order to accomplish the objective of workflow process activity instances.

Workflow Enactment Service: A software service that may consist of one or more workflow process engines in order to create, manage, and execute workflow process instances. Client workflow applications/tools interface to this service through the WAPI.

Workflow Interoperability: The ability for two or more workflow engines to communication and interoperate in order to coordinate and execute workflow process instances across those engines.

Workflow Management System: A system that completely defines manages and executes workflow processes through the execution of software whose order of execution is driven by a computer representation of the workflow process logic.

Workflow Participant: A resource which performs partial or in full the work represented by a workflow process activity instance.

Workflow Process: The computerized facilitation of automated component of a process.

Workflow Process Activity: The computer automation of a logical step that contributes toward the completion of a workflow process.

Workflow Process Activity Instance: An instance of a workflow process activity that is defined as part of a workflow process instance.

Workflow Process Control Data: Data that is managed by the Workflow Management System.

Workflow Process Definition: The component of a process definition that can be automated using a workflow management system.

Workflow Process Engine: A software service of "engine" that provides part of all of the run time execution environment for a workflow process instance.

Workflow Process Execution: The duration in time when a workflow process instance is created and managed by a Workflow Management System based on a workflow process definition.

Workflow Process Instance: Represents an instance of a workflow process definition which includes the automated aspects of a process instance.

Workflow process Monitoring: The ability to track workflow process events during workflow process execution.

Workflow Process Relevant Data: Data that is used by a Workflow Management System to determine the state transition of a workflow process instance.

Worklist: A list of work items retrieved from a workflow management system.

Worklist Handler: A software component that manages and formulated a request to the workflow management system in order to obtain a list of work items.