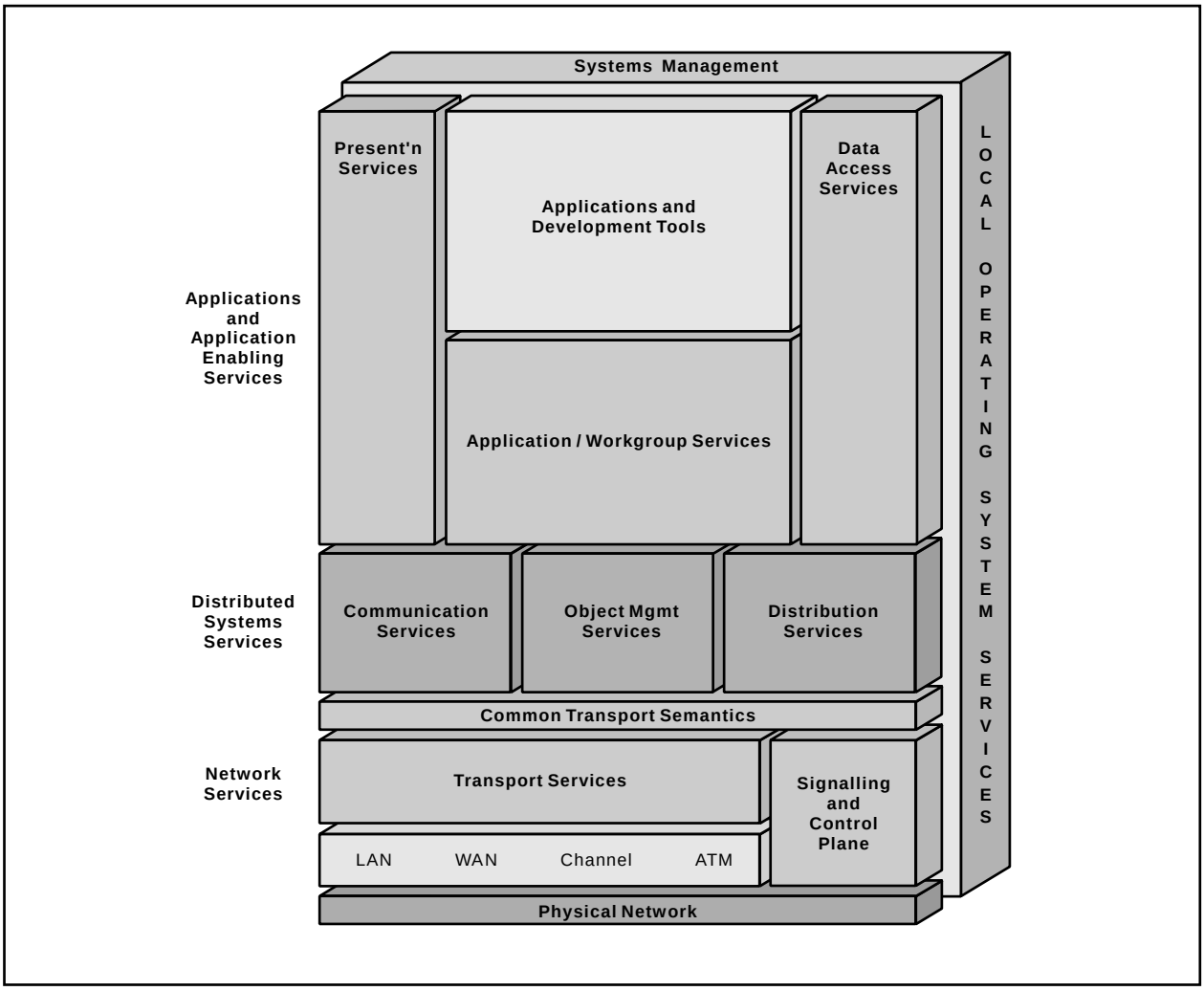
Open Blueprint

Workflow Resource Manager



GC23-3935-01

Open Blueprint

Workflow Resource Manager

GC23-3935-01

About This Paper

Open, distributed computing of all forms, including client/server and network computing, is the model that is driving the rapid evolution of information technology today. The Open Blueprint structure is IBM's industry-leading architectural framework for distributed computing in a multivendor, heterogeneous environment. This paper describes the Workflow resource manager component of the Open Blueprint and its relationships with other Open Blueprint components.

The Open Blueprint structure continues to accommodate advances in technology and incorporate emerging standards and protocols as information technology needs and capabilities evolve. For example, the structure now incorporates digital library, object-oriented and mobile technologies, and support for internet-enabled applications. Thus, this document is a snapshot at a particular point in time. The Open Blueprint structure will continue to evolve as new technologies emerge.

This paper is one in a series of papers available in the *Open Blueprint Technical Reference Library* collection, SBOF-8702 (hardcopy) or SK2T-2478 (CD-ROM). The intent of this technical library is to provide detailed information about each Open Blueprint component. The authors of these papers are the developers and designers directly responsible for the components, so you might observe differences in style, scope, and format between this paper and others.

Readers who are less familiar with a particular component can refer to the referenced materials to gain basic background knowledge not included in the papers. For a general technical overview of the Open Blueprint, see the *Open Blueprint Technical Overview*, GC23-3808.

Who Should Read This Paper

This paper is intended for audiences requiring technical detail about the Workflow Resource Manager in the Open Blueprint. These include:

- Customers who are planning technology or architecture investments
- Software vendors who are developing products to interoperate with other products that support the Open Blueprint
- Consultants and service providers who offer integration services to customers

© **Copyright International Business Machines Corporation 1995, 1996. All rights reserved.**
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Figures

© Copyright IBM Corp. 1995, 1996

Summary of Changes

This revision includes:

- A classification scheme for types of workflow applications
- A description of the relationship between the Collaboration resource manager and the Workflow resource manager
- A description of the relationship between the Web Browser resource manager and the Workflow resource manager
- Additional information about the relationship between object technology and workflow technology

© Copyright IBM Corp. 1995, 1996

Workflow Resource Manager

The focus of this paper is the Open Blueprint Workflow resource manager and the support it provides for business reengineering.

Workflow Management

Workflow management provides a set of functions that allows customers to define, validate, execute, manage, and reassess their business processes in an automated, yet dynamic way (White and Fisher, 1994). These business processes can be any line-of-business activities—for example, contracting for a purchase, the processing of an insurance claim, managing a lawsuit from start to finish, the steps in application development, or an enterprise-wide management function (such as installing a software update).

Workflow management is defined as “a proactive computer system that manages the flow of work among participants, according to a procedure consisting of a number of tasks. It coordinates user and system participants, together with the appropriate data resources, which may be accessible directly by the system or offline, to achieve defined objectives by set deadlines. The coordination involves passing tasks from participant to participant in correct sequence, ensuring that all fulfill these required contributions, taking default actions when necessary” (OVUM Corporation, 1991).

The term *workflow* is used heavily within the industry, however it is associated with different meanings. These meanings are best explained using Figure 1 on page 4, as originally proposed by BIS Strategic Decision in 1995. The figure categorizes workflow according to two different aspects: the leverage of the business process on the success of the corporation and the degree of repetition of the associated business processes. Each quadrant shown in the figure represents a particular category of workflow. The Workflow resource manager focuses on highly repetitive workflow and consequently addresses the quadrants' production and administration workflow. For a description of the support focused on collaborative and ad hoc workflow, see *Open Blueprint Collaboration Resource Manager*.

© Copyright IBM Corp. 1995, 1996

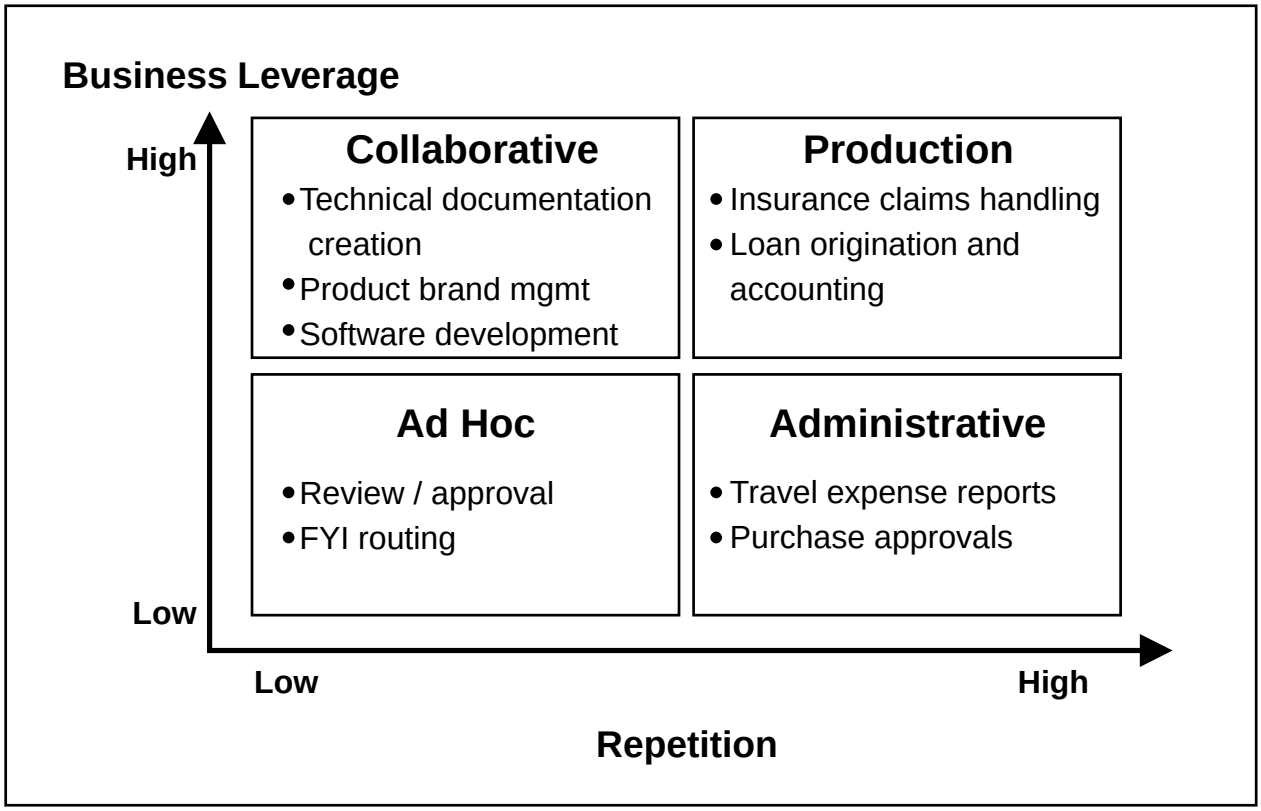


Figure 1. Categories of Workflow

Workflow and Application Structures

In many companies, valuable process definitions are buried deep within the logic of application programs. Changing a business process means changing the application programs, which can be a time-consuming and expensive programming task. With workflow management, programs provide service to processes. The application consists of a set of small programs or other processes; the sequence of program execution is defined as a process to the Workflow resource manager. The process describes:

- The flow of control from one program to the next
- The data to be passed along the process
- The programs and manual procedures to be invoked
- The people to perform the work

At execution time, for each instance of the process, the Workflow resource manager navigates through the process, moves the work to the right person, executes the programs, provides status information, keeps process execution history, and provides recovery and restart. All this takes place in a distributed heterogeneous environment. The Workflow resource manager becomes an application-operating system; applications following this structure are called *workflow-based applications*. Workflow-based applications are not typically created by revamping existing applications. Rather, they are the result of business reengineering, "the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical, contemporary measures of performance, such as cost, quality, service, and speed" (Hammer and Champy, 1994). Business reengineering is typically performed by exploiting business modeling tools.

The result of business reengineering is a well-documented process, which can then be implemented and executed using a workflow management system. The programs invoked by the Workflow resource manager can be new programs or existing applications.

Workflow Based Applications

More than two decades ago, applications were highly dependent upon how data was organized on a disk. Changes in data organization required changes in all affected programs. To overcome this data dependency of applications, database management systems were built. When database management systems are used, applications become more flexible and less vulnerable to changes at the data-organization level (Elmasri and Navathe, 1989).

Similarly, many contemporary applications do not allow enterprises to easily change their processes. Large application systems contain the code that directly represents these processes so that changing the processes requires changing the corresponding application systems. While the algorithmic knowledge of how to perform a particular step of a business process does belong in the proper application, the knowledge of sequencing these steps and distributing them to the responsible organizational units makes the application vulnerable to changes in underlying business processes, that is, applications are *flow dependent*.

By exploiting workflow technology, the flow dependency of applications is removed, because the knowledge of dispersing work items is extracted from the code. As a result, applications become more flexible and less vulnerable to changes in the business processes supported by the applications.

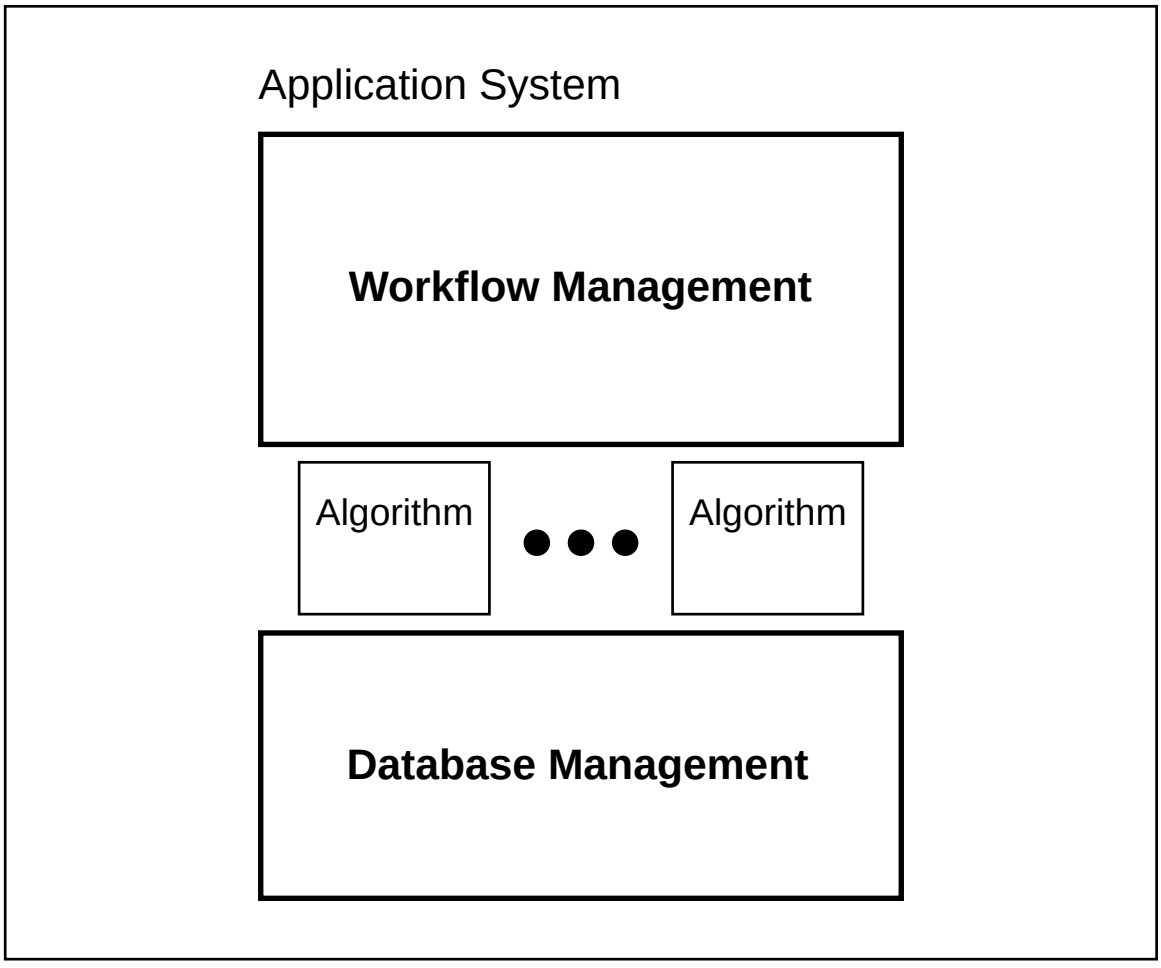


Figure 2. Data and Flow Independence

An application that is flow independent consists of a workflow definition and a set of programs that are algorithmic components of a business process directly supporting its activities. Applications exploiting

workflow are *topology-independent* because the Workflow resource manager does not assume any inherent structure among the programs associated with activities. This allows a new implementation structure in which the application can be a distributed network computing, or client-server application, because the programs can be running on different machines and in heterogeneous environments. The binding of each activity to its supporting program occurs at run time based on the definitions in the workflow. The locations where the various parts of the applications are executed are transparent and can be changed at run time.

Finally, one can expect that the productivity of a corporation's application development will benefit from flow-independent applications. As a result of process model definition, the interfaces between the programs implementing the basic execution steps of the business process are defined. In general, implicit assumptions about data routing, program sequencing, and status are removed. Thus, parallel development becomes more effective.

Application Integration

The Workflow resource manager is a means to couple both new and previously existing applications, even if each is developed independently. These applications can be implemented with procedural or object-oriented technology. The common context often needed for such a coupling is provided by the dataflow facility of the Workflow resource manager providing coherence of the required context information. The result is an application system providing added value to its users when compared to its encompassed independent applications. For example, the correct sequencing of independent applications to achieve a business goal is automatically provided while paying attention to each particular business situation. Idiosyncrasies of the various invocation mechanisms are hidden, and information is automatically passed from one application to another.

Because the context and state of such an integrated application is persistent data, the history of the executed discrete applications, their order, and their local context is tracked. In error situations that require compensation, the system can—based on this persistent data—automatically schedule discrete applications previously defined as *compensation steps* to correct the situation. The result is a degree of partial backward recovery introducing a compensation-based semantic transaction paradigm for integrating applications.

The coupled applications can run under the control of various transaction monitors or as native operating system programs, they can be transactions or nonrecoverable units, and they can run locally or distributed in heterogeneous environments. The Workflow resource manager, a key piece of middleware, ties the applications together.

Three Dimensions of Workflow Definition

To describe enterprise-wide processes, it is necessary to define the actions or activities of the process. In addition, the people or means that perform these activities must be defined as well as the resources that are used. Therefore, the Workflow resource manager supports the definition of processes by the following three independent views, that is, *what*, *who*, and *which*:

- Process view (*what*)
- Organizational view (*who*)
- Infrastructure view (*which*)

The Process View—What Is Performed

A process is described through a number of activities that need to be performed in order to build a product (see Figure 3 below). A product can be, for example, an insurance policy, a credit, or a program.

The activities can be performed by people or resources (that is, computers or robots). The activities are connected by arrows that describe the potential flow through a process. The decision as to which arrow to follow is described by transition logic (logic predicates). In addition, the start and end of a process can be governed by logic predicates. Therefore, the static form of a process (process model) is defined by a directed, weighted, colored graph (Leymann, 1992).

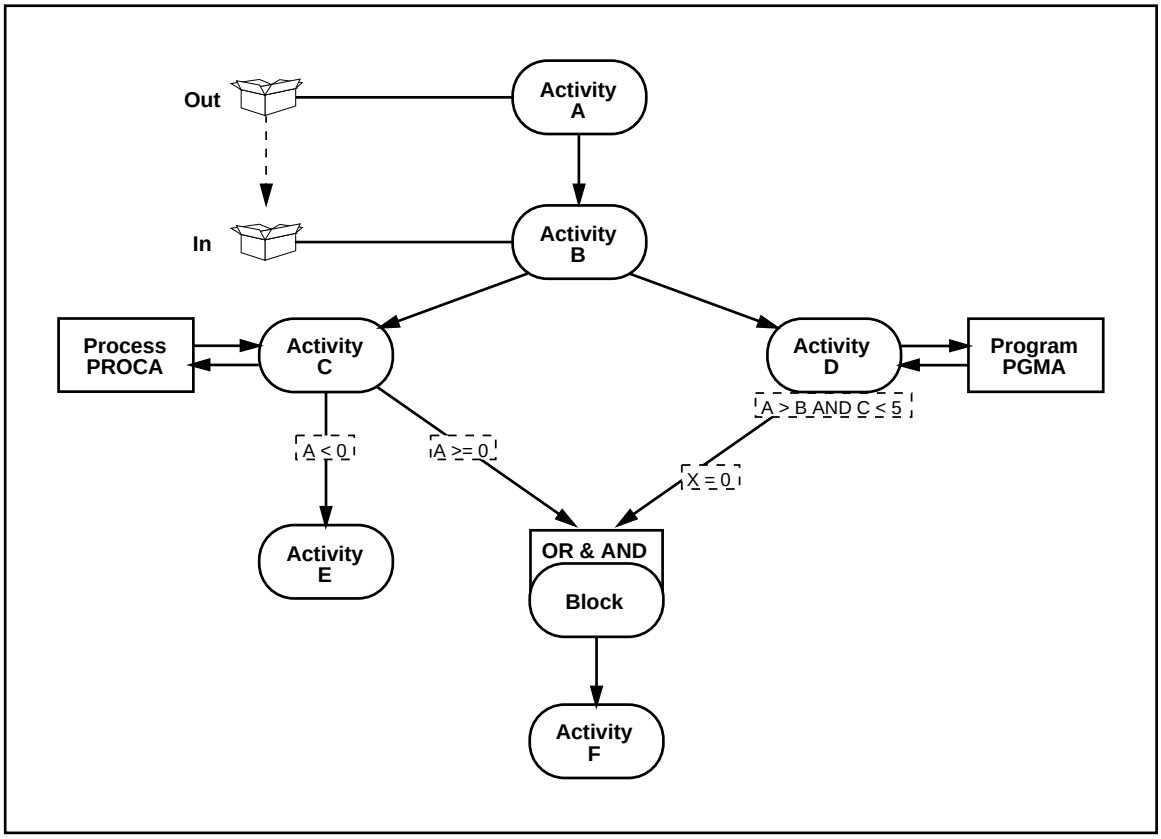


Figure 3. Process Model

A particular process that is performed according to such a process model is called a *process instance* (or process for short). Therefore, it is possible to have many process instances of the same process model and each of them can have different status and shapes at a given point in time. An activity is defined as an action performed at a certain location by a person (or its surrogate). At an enterprise level this is the smallest piece of action described in a process model. However, at the next lower level it is possible to split this activity into even smaller actions, which describe in detail all the actions a person, computer, or robot performs. In this case, the activity can again be defined as a process (that is, a subprocess) that describes the internals of the original activity (Leyman and Roller, 1994).

This procedure is called *refinement* and allows a consistent transition from enterprise processes into software structures or applications, and then into tasks that can be performed by machines and people.

The process view of a process shows the dynamic behavior or control view of a process. To complete the description of a process, the data required by the various activities must be defined.

Input and output information for each activity is defined as *templates*. These information templates are called *input and output containers*. Therefore, activity containers are a view of the information that is handled during a process instance. In other terms, containers are different local views of the total process information resources. In addition, programs that are executed as part of the process can access other data in the usual ways.

The Organizational View—Who Performs

As a part of process modeling, people or resources performing the atomic activities are defined. This organizational view describes the structure of the enterprise (for example, departments or projects) and the people performing the activities as they relate to that structure. The information about the people includes capabilities such as skills and roles they can perform. The organizational information includes rules like escalation and substitution. For each activity, the process modeler defines who should perform this activity (see Figure 4). This is specified in terms of organizational information, which at run time is resolved into a set of people or resources. The association of activities and organizational information is called *staff assignment*; the resolution is called *staff resolution*.

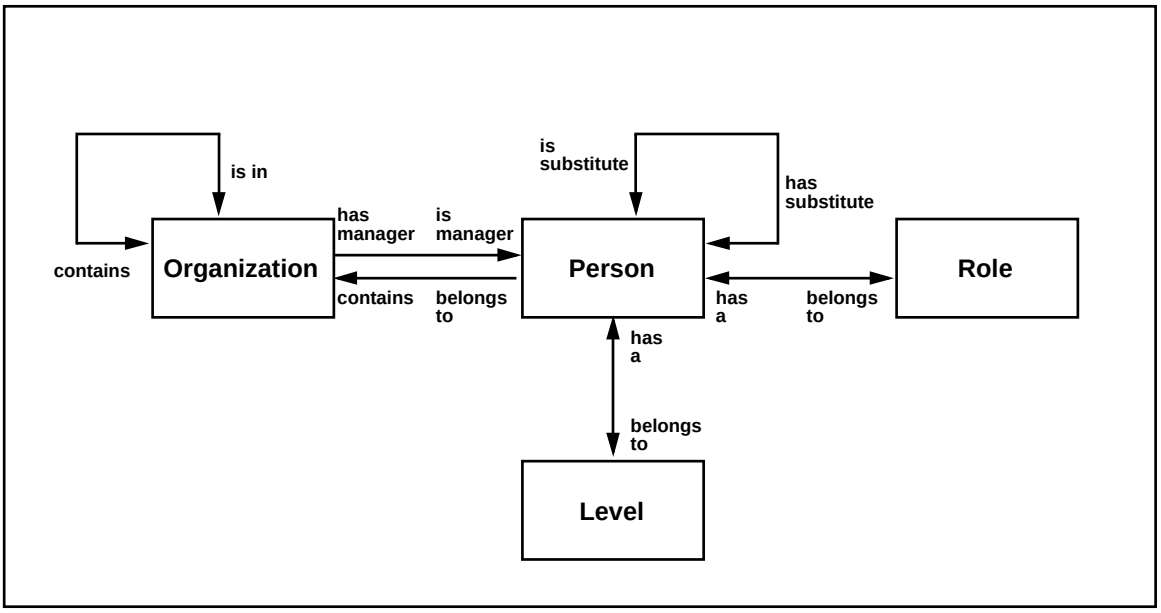


Figure 4. Staff Assignment

The information gathered allows the combination of the skills of individuals through teamwork (*process teams*, Hammer and Champy, 1993); this is a necessary task in a successful process reengineering effort. For example, intelligent use of the staff definition is a means for an enterprise to structure their education plans and to guide resource allocation, organizational adaptations, and hiring plans. Consequently, the organizational structure can be changed to reflect the business processes. This leads to leaner and more efficient organizations (see Figure 5 on page 9). It is also interesting to note that staff assignment can be seen as defining constraints on a process model.

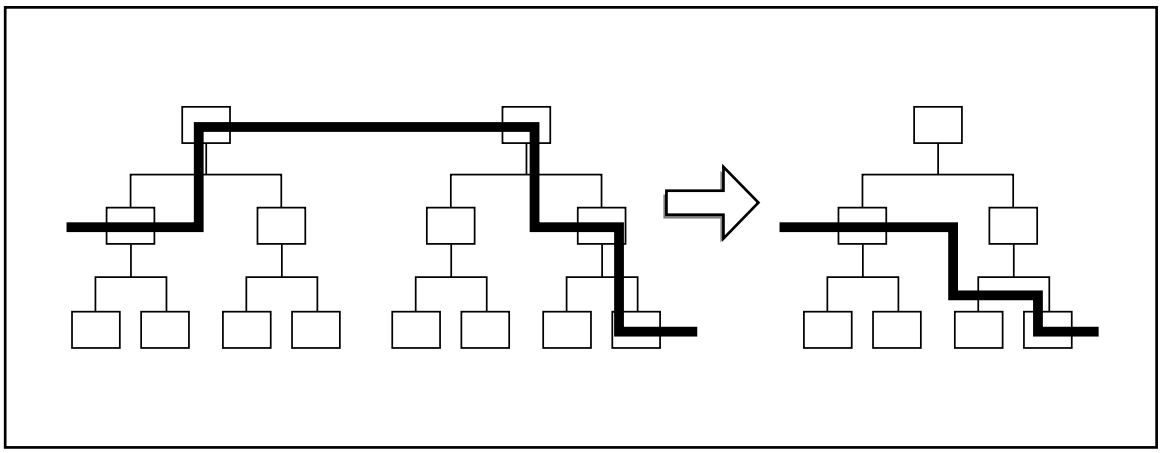


Figure 5. Reengineering Organizations

The Infrastructure View—Which Resource is Used

An activity (what) is performed by a person or on behalf of a person (who). The infrastructure defines *which* IT resources are available to perform an activity—for example, which program is used (that is, with which tool) and which machine runs the program (that is, where the program is executed).

When a person starts an assigned activity, the appropriate IT resources are determined and allocated. This includes selecting the proper programs, determining the right invocation mechanism and, if necessary, transferring data from one server to another. Some of these assignments are performed dynamically, which allows balancing of work among people and systems.

Combining all Views: Taking the three views of processes, organizations, and infrastructure, workflow can be represented as the navigation through a three-dimensional space, referring to *what*, *who*, and *which*, see (Figure 6 on page 10). The more clearly a workflow capability can represent these three views, the more able it is to adapt workflow (or enterprise processes represented by workflows) to the requirement of reengineering activities.

When workflow application systems are devised from document or image handling facilities or office activities, very often the three dimensions are not clearly separated. This can result in the need to define and use document status to support navigation within process instances, and because the document flow might not adequately represent the business process—for example, it might be peripheral to the overall business process, the desired adaptability and reengineering might not be achieved.

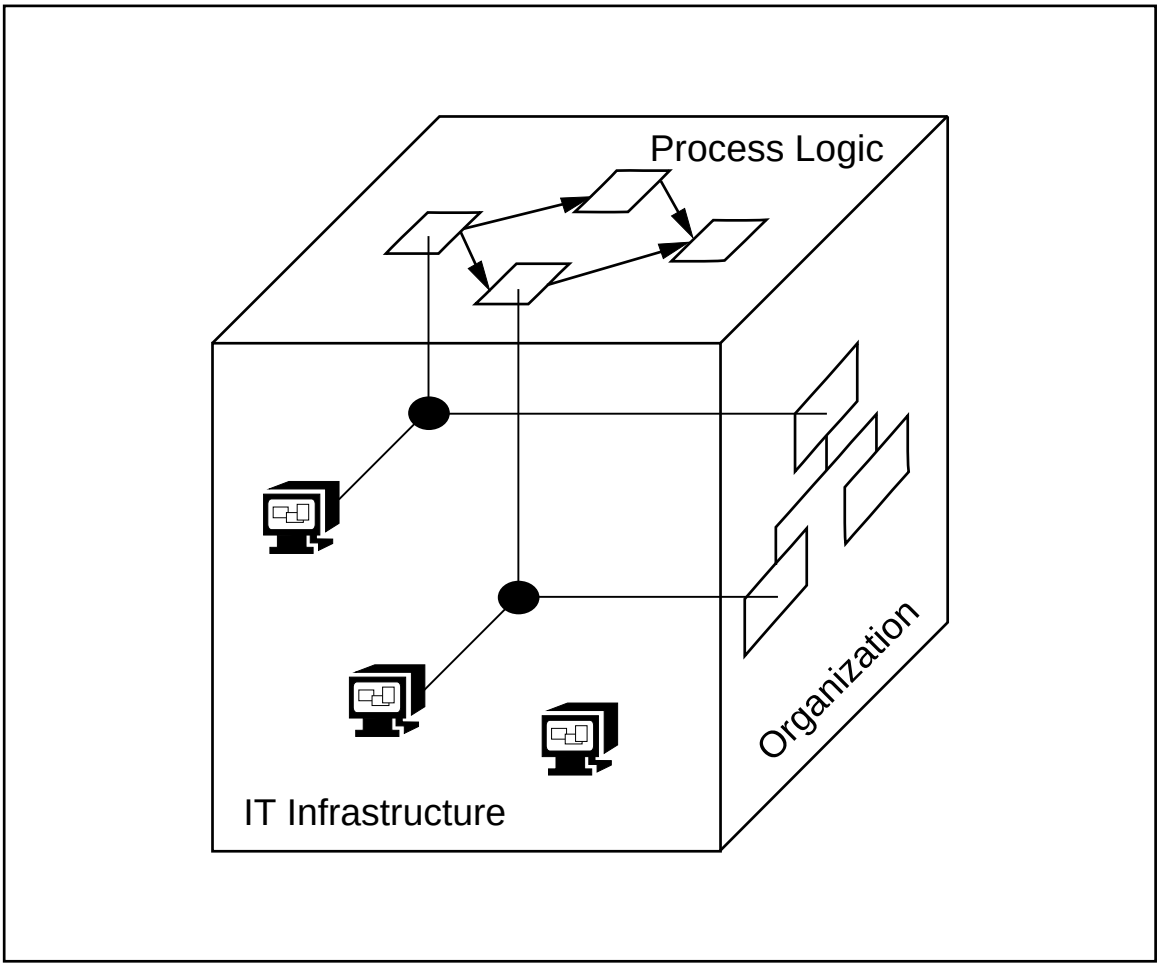


Figure 6. The Workflow Cube

Workflow Resource Manager

The functions of the Workflow resource manager are supported by the following sub-components:

- Build time
  - Workflow definition
  - Animation
  - Simulation and capacity planning
  - Import/Export of process models
- Run time
  - Navigation
  - Work list handling
  - Program invocation
  - Auditing
  - Monitoring
- Administration

For each sub-component there is a client. The build time client uses the Object-Oriented Database resource manager to efficiently store and support the required management of the complex data structures that represent process definitions. The run time client works with an associated runtime server, which connects to the Relational Database resource manager database. The administration client works with the associated run time server to allow an administrator to do system and process administration.

Build Time

Build time supports the definition of process models and the management of organizational information. In addition, it provides the capability to animate and simulate those models and organizational structures, and to import and export them.

Process models are defined with a graphical editor that can be used to place activities on the editor's working area and connect them with arrows to represent control and data flow. It is also possible to define process models and organizational information with the Workflow resource manager's definition language. To support interoperability, this language is used to import and export information between different workflow management systems; it is also used to exchange process information between business modeling tools and the Workflow resource manager.

To verify a workflow model, the model can be visually animated. That is, the modeler navigates through the model and can see the effects of every single navigation step. To predict the total behavior of a workflow management system within an enterprise, it is necessary to simulate the execution of multiple processes. This allows the modeler to detect resource shortages and determine where improvements of the process models are required. Additionally, simulation can use real-time data collected from previous process runs.

Run Time

Run time provides the environment and support for the execution of processes.

Server Functions: The server component is responsible for navigating the process graphs, performing staff resolution, maintaining the work lists of users, invoking programs on behalf of users, and generating audit-trail information. The server does this for multiple instances of multiple processes.

Navigating a process graph and staff assignment is performed by the navigation engine of the runtime server. The engine selects the next set of activities to be performed by evaluating the control flow. For each selected activity, the engine determines who can perform the activity and makes this information persistent in the database to provide forward recovery. The work list function in the server maintains the work lists of all users assigned to that particular server, allowing the user to sign on to the Workflow resource manager at any supported workstation.

Every action of the Workflow resource manager at run time is recorded, if requested, in an audit trail. This data can be processed to obtain the information required to determine the efficiency of the process. A monitoring facility can be used to display the status of every process started by the user.

Work Distribution: If users are assigned to different Workflow resource manager servers, either complete subprocesses or work items are distributed between the servers responsible for supporting the involved users.

In the case of distributed work items, navigation through the process is performed by the Workflow resource manager server where the process originates. If staff resolution determines that a user is serviced by another Workflow resource manager server, the target server is responsible for the work item until it has been completed by the user.

For distributed subprocesses, the level of distribution is the process. The originating server informs the target server about the process to be started and passes the process input container to the target server. The process is managed by the target server until it completes. All relevant information is then passed back to the originating server so that navigation can continue.

Basic Client Function: The client component maintains the local work list, including synchronization with the persistent work list on the server, and executes programs on a client's workstation.

The client component provides application programming interfaces (APIs) such as an API that enables programs to invoke process-relevant operations such as starting or terminating a process, and an API that allows activity implementations to access the data containers associated with each activity. These APIs can be encapsulated in domain- or application-specific wrappers to facilitate integration of the Workflow resource manager services with other application launch mechanisms such as LotusScript Extension Modules (LSXs), and OLE Custom Controls (OCXs).

The Workflow resource manager-supplied run time client exploits the APIs and provides a graphical end-user interface, which allows the user to start, terminate, suspend, and resume processes, to maintain work lists with work items, and to start activities. If the selected activity is a program activity, it retrieves the proper program registration information, determines the data to be made available in data containers, launches the program, and determines the returned information after the program is complete.

The run time client also takes advantage of the Workflow resource manager's service brokers to provide for better application integration. Service brokers allow connections to an invoked application to be maintained so that a subsequent invocation of the application is performed with less overhead. Service brokers support, for example, the Collaboration resource manager and the VisualAge SmallTalk environment.

Network Computing Support: The Workflow resource manager allows access to its run time services through the World Wide Web (WWW). These services are made available through HTML documents, which can be presented by the Web Browser resource manager.

Using the Web Browser resource manager:

- Users can list the process models published on the Web, and they can instantiate any of these models.
- Process instances can be managed. They can, for example, be listed, suspended, and resumed.
- Users can manipulate work items. In particular, work items can be listed and started.

The following figure illustrates how workflows are supported on the Web. To access Workflow resource manager services, a web browser must be installed on the user's machine. This client machine then communicates with the Web server, which offers the workflow services, using the Hypertext Transfer Protocol (HTTP). The Web server presents workflow run time objects such as process models or work items, and the applicable functions that are available to the user on Hypertext Markup Language (HTML) documents. The Web server accesses the Workflow resource manager's run time client's proper workflow functions using the Common Gateway Interface (CGI). The Workflow resource manager client, in turn, communicates with the appropriate Workflow resource manager server to manipulate workflow objects.

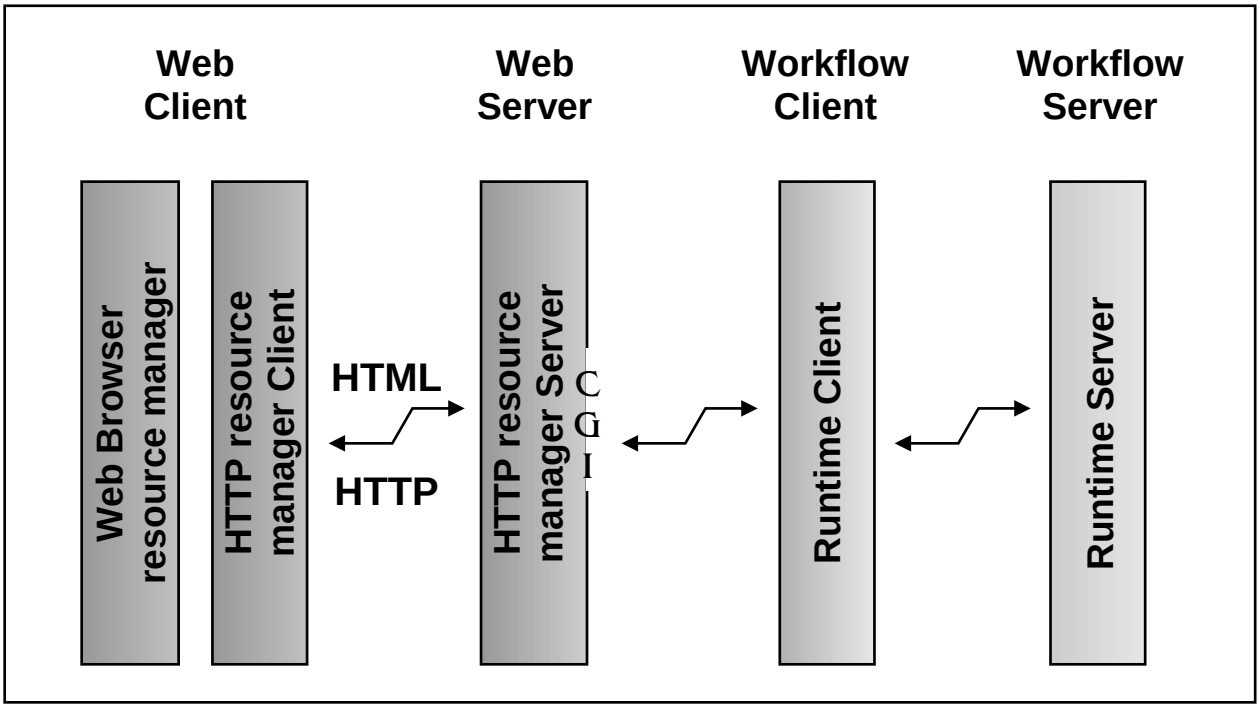


Figure 7. Running Workflows on the Web

Accessing workflow services through the Web has a broad range of applications, for example:

- Companies that advertise on the Web can allow their customers to initiate business processes such as ordering processes and to check the status of such processes, for example, the status of a credit request.
- Employees who are absent from the company's facilities (on business trip or working at home) can check their work lists and perform activities.
- Corporations that exploit Internet technologies for their applications (for example, using HTML documents as their end user interfaces) can integrate Workflow resource manager services into these applications.

Running Workflows from the Collaboration Resource Manager Environment: The features of the Collaboration resource manager can be used to complement workflow applications driven by the Workflow resource manager, allowing work group interaction to be integrated into the structured workflow process.

The Collaboration resource manager client can be used as a front end to the Workflow resource manager. This client presents workflow run time objects such as process models, process instances, and work items to the end user in a Collaboration resource manager environment and supports execution of workflow actions on these objects. Processes can be started, controlled, or monitored, and work items can be manipulated from the Collaboration resource manager's desktop.

The Workflow Management Coalition Workflow Client API (see “Interoperability through Workflow Management Coalition Conformance” on page 17) is used for communication between the Workflow resource manager and the Collaboration resource manager front end. Workflow run time objects are copied into a Collaboration resource manager database as documents and, these documents are kept in synch with the original information maintained by the Workflow resource manager. Workflow actions on the documents are performed using the Collaboration resource manager's agents, which issue requests to the Workflow resource manager. Collaboration resource manager views and forms define the appearance

of the workflow objects in the Workflow resource manager's collaboration resource manager front end. The Collaboration resource manager's application development features can be used to customize this interface.

In addition, Collaboration resource manager applications can be integrated into a workflow process driven by the Workflow resource manager using documents as a bridge between both. For example, a process model document might be linked to a Collaboration resource manager application which, upon completion, invokes the corresponding process. A work item document might be used to start a Collaboration resource manager application which might involve other Collaboration resource manager documents and integrate other Collaboration resource manager users into the processing of the corresponding activity.

The integration offers additional benefits, such as:

- The Collaboration resource manager's support for mobile users can be exploited for disconnected execution of Collaboration resource manager-based activities by checking out the relevant work items and replicating the corresponding documents to a local Collaboration resource manager's desktop.
- The Collaboration resource manager's mail capability might be used for ad hoc routing of work item documents to consult other Collaboration resource manager users.
- Workflow documents can be integrated into a universal in-basket containing to-dos from various sources such as e-mail.

Supporting the Mobile User: Business processes quite often involve mobile users. Mobile users typically use notebooks when performing their work. Their connection to the appropriate Workflow resource manager server is rather sporadic. Typical mobile users include insurance agents and sales people.

Support of the mobile user encompasses the following Workflow resource manager facilities:

- The user can check out work items for later execution. As a result of checking out a work item, the work item is removed from the work lists of all other users who are connected. Whenever the mobile user's machine connects to the server, all completed work item-relevant information is automatically transmitted to the server.
- The user can start process instances. The process instance information is kept on the user's machine, until a connection is made to the server which causes the transmission of the associated data.
- The user can perform parts of the process. This is possible for connected activities that are performed together by the mobile user because the appropriate process model can be transformed into an executable.

Support of the mobile user is achieved by the Workflow resource manager through the exploitation of the Messaging and Queueing resource manager.

Administration

The administration of a Workflow resource manager is divided into system administration and process administration.

System administration deals with managing the basic resources of the Workflow resource manager. This consists of administration of the underlying database, management of the organizational information, process models, distribution of information between multiple Workflow resource manager servers, and the management of audit trails.

Process administration deals with managing the tasks associated with process instances such as querying the status of the Workflow resource manager servers (active process instances, number of work items

assigned), reassigning work items for balancing work load, and monitoring the status of exception conditions.

Associated Support

The three major components of a process model are:

- The flow of control and data between its activities
- The distribution of each activity to its responsible organizational unit
- The linkage of each activity to its supporting program

These programs implement the basic functions of the underlying business process, that is, the basic execution steps for which a further subdivision is not meaningful. Sample criteria for detecting such granules are that they can be reused in other business processes, they are executed in a specific manner (for example, by a single person at a single location without an interrupt), or that they are inherently complex (for example, only one set of tools can be provided that is used on a case-by-case basis by a knowledge worker in a non-foreseeable manner).

When the workflow paradigm is adopted as the fundamental structure, business administrators can be involved from the beginning of the development cycle. A business modeling tool allows them to describe business processes without any data-processing terminology, and in terms relevant to their area of expertise, such as critical success factors, process goals, quality measures, and required resources.

Additionally, some of the information captured by the business modeling tool represents measurable quantities, like expected duration and costs. These quantities are added to the process model and direct the runtime component of the Workflow resource manager to record the required information in the audit log. Based on the audit log, a process performance monitor provides a statistical analysis of the instance's performance. Reports of the actual behavior and state of processes are also produced. The analysis of this information can initiate a new cycle of improvements to the process models that do not meet the corporation's goals.

Similarly, business objects captured by the business modeling tool can be transformed into kernel entities of a data modeling tool. These kernel entities can be refined into local conceptual schemas providing data models for the various business areas supported by the corresponding business processes. Because each local conceptual schema can be perceived as the view of the associated business area onto the global data of the whole corporation, a view-integration tool facilitates the derivation of a global schema. The global schema can be transformed into a relational database schema. Input and output containers of activities can be modeled as views onto the global schema. Based on the known mapping of the global schema onto a relational database, schema code can be generated that will retrieve and store container instances in relational databases.

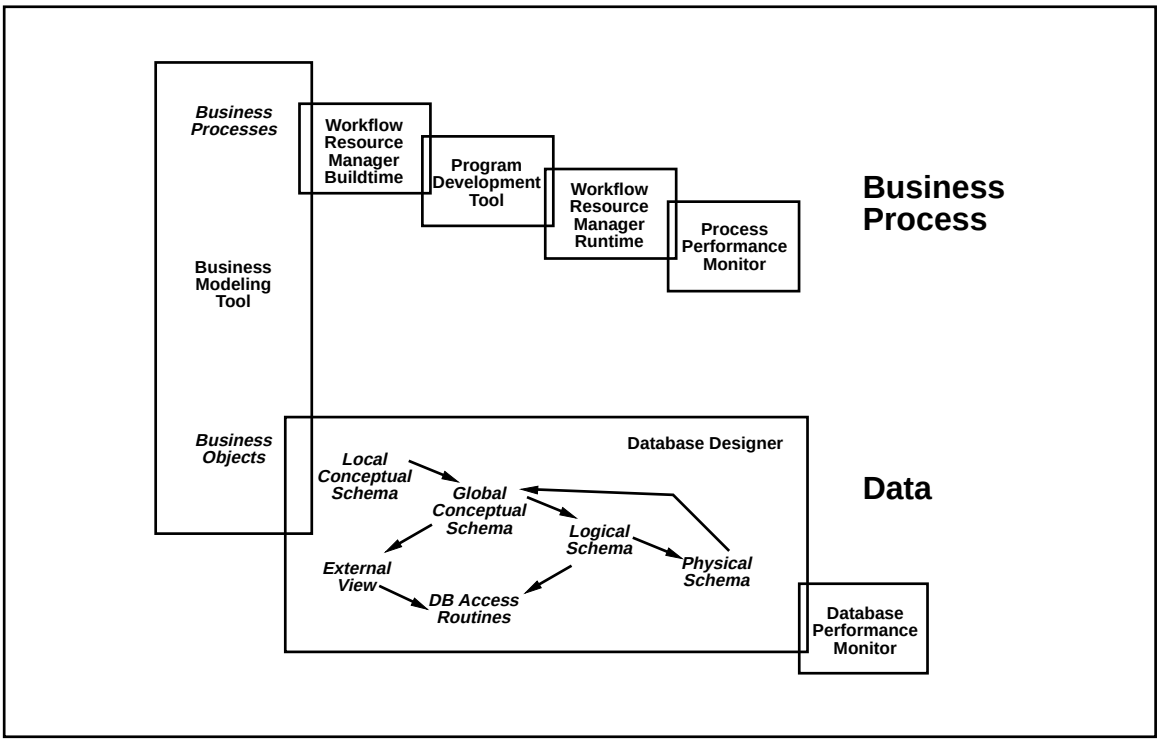


Figure 8. Process-Based Application Development

Verification

The correct and efficient execution of processes is vital for the successful operation of the enterprise. It is imperative that the utmost care is taken to ensure this goal.

An animation tool allows the verification of process logic. This is done by simulating the control flow, the data flow, and the flow between the affected organizational units of a single instance of one particular process model.

If the process model correctly represents the process logic, a simulation tool helps verify the appropriateness of resources. This means that the organization and its structure, and the resources consumed by the various activities are considered to infer, for example, wait times or resource contingencies, based on simulating many concurrent instances of many different process models. A comparison with the quantities specified in the business modeling tool indicate whether more reengineering work has to be done.

When the process model is in production, an ongoing monitoring—that is, the measuring of the various execution parameters, is supported. The process performance monitor will collect and represent the actual and statistical behavior of the processes. The results can indicate the necessity of reengineering work because of deviations from the specified process goals.

Object-Oriented Technology

Enterprises today invest in object-oriented technology to improve the productivity of their programmers and enable even non-data-processing professionals to build applications. As a result, the building of applications will become more and more component-based.

Object-oriented and work flow are two complementary technologies. One can exploit the other in various

ways, resulting in synergies that are equally valuable to both paradigms, but each technology has its own area where it can be applied without the other.

The Workflow resource manager can execute programs written in an object-oriented programming language. The Workflow resource manager's mechanism of starting programs bound with activities supports various invocation mechanisms—for example, starting programs provided as executable (EXE) files, command (CMD) files, or entry points in data link libraries (DLLs), Customer Information Control System (CICS) or Information Management (IMS) transactions, or MQSeries applications. Because the way a program is constructed is irrelevant to these invocation mechanisms, the Workflow resource manager can start the corresponding programs regardless of whether they are written in an object-oriented programming language.

OpenDoc parts representing particular workflow constructs can be provided. For example, with a part handler, work requests can be included in documents and can be taken directly from the document. By having the ability to invoke OpenDoc parts, the Workflow resource manager supports the use of those parts as implementations of activities.

Object-Oriented Application Design and Workflow

If the encapsulation paradigm of object-oriented technology is followed, business processes can be hidden in the object implementations. As a consequence, the object applications can become flow-dependent, and the business processes will not be explicitly described and externalized to a broader community. However, some of the components built for application construction based on object-oriented technology will be *business objects*, that is, objects that are immediate subjects of business processes. Such a business process explicitly describes the rules of how, when, and by whom the services that are provided by the various business objects are exploited. Those services can correspond to activities within business processes. When using object-oriented technology to construct business objects, the business objects can be used directly as implementations of activities within business processes. By supporting the invocation of methods on objects, the Workflow resource manager facilitates the exploitation of services provided by the corresponding business objects as activity implementations. In the case of distributed objects, even the location of these objects is irrelevant.

A business object can communicate in an object-oriented manner with the Workflow resource manager in order to influence the navigation through the business process to which the business object belongs. A workflow framework that provides the corresponding services as objects allows for that.

Interoperability through Workflow Management Coalition Conformance

Workflow management systems that are implemented by different vendors typically share a common set of functions. However, they are delivered with proprietary application programming interfaces and protocols. To allow cooperation of those different implementations both in a single customer location and in a multi-enterprise environment, a common set of APIs and protocols must be defined. With this goal in mind the Workflow Management Coalition (WfMC) was founded. Its members include all major vendors of workflow management systems, users of workflow management systems such as large enterprises, and representatives from the research community.

The WfMC has defined a *Workflow Reference Model* (Workflow Management Coalition, 1994) as a base for defining those application programming interfaces and protocols to achieve the required interoperability. Work is underway to standardize the APIs in the following areas:

Definition and Exchange of Process Models: The purpose of this interface is to allow the definition and exchange of process models. It manifests itself either as a textual language or as a set of API verbs. The

textual process definition language is intended to be used as a means of importing and exporting complete process models. The definitional APIs allow programs to define and modify process models.

Workflow Clients: This interface provides a set of verbs for retrieving and accessing work lists and work items on such work lists.

Application Invocation: Defines standard protocols for the interaction between the workflow management system and programs implementing activities.

Server Interoperation: Defines the protocols used by different workflow management systems to communicate requests for initiating processes and subprocesses, transferring work items, and exchanging status information.

Administration Interface: Defines how actual status information can be obtained from a workflow management system through APIs.

It is the intent of the WfMC to have these APIs approved by public standardization bodies. The WfMC-specified APIs will be supported by the Workflow Resource Manager.

Relationships with Other Open Blueprint Components

The Workflow resource manager provides a broad business process management service. Within the Open Blueprint it supports Systems Management processes. In turn, the Workflow resource manager relies on several components of the Open Blueprint by exploiting their open interfaces.

Collaboration Resource Manager

The Workflow resource manager's run time client function is also delivered by exploiting the Collaboration resource manager. Interactions between the Workflow resource manager and the Collaboration resource manager provide support for structured and unstructured work.

Data Access Services

The Workflow resource manager relies on the Object-Oriented Database resource manager and the Relational Database resource manager for maintaining its persistent information.

Communication Services

The runtime component of the Workflow resource manager exploits the Message and Queuing resource manager for its client/server communication, server-to-server communication, and for invoking activity implementations.

Security

The Workflow resource manager depends on the Identification and Authentication resource manager. As a result, a user has to log on to the system only once to get to all workflow services and all other applications and data for which authorization has been granted. Access to workflow resources is controlled by access control lists which are managed by the Access Control resource manager.

Directory Resource Manager

The naming service of the Directory resource manager is used to provide the Workflow resource manager with the location of resources based on organizational information and the topology of the workflow management system's installation.

Web Browser Resource Manager

The Web Browser resource manager is exploited to provide an alternate front end to the Workflow resource manager's run time environment.

HTTP Resource Manager

The HTTP resource manager provides the interface between the Web client and the Web server (see Figure 7 on page 13) which then provides access to the Workflow resource manager.

Appendix A. Bibliography

Elmasri, R., and Navathe, S.B. *Fundamentals of Database Systems*. Benjamin/Cummings Publ. Comp. Inc., 1989.

Hammer, M., and Champy, J. *Reengineering the Corporation*. New York: Harper Business, 1993.

IBM Corporation. *Open Blueprint Collaboration Resource Manager*. 1995.

Leymann, F. *A meta model to support the modeling and execution of processes*. In Proc. 11th European Meeting on Cybernetics and Systems Research EMCR92. Vienna: *World Scientific* (1992): 287 - 294.

Leymann, F. *Supporting business transactions via partial backward recovery in workflow management systems*. In Proc. BTW'95 Datenbanksysteme in Büro, Technik und Wissenschaft. Dresden. Springer. 1995.

Leymann, F., and Altenhuber, W. *Managing Business Processes as Information Resources*. *IBM Systems Journal* 33(2) (1994): 326 - 348.

Leymann, F., and Roller, D. 1994. *Business Process Management with FlowMark*. In Proc. COMPCON Spring 94. *IEEE Computer Society Press* (1994): 230 - 234.

White, T.E., and Fischer, L. (ed.) *The Workflow Paradigm: New Tools for New Times*. Future Strategy Inc., Book Division. 1994.

Workflow Management Coalition. 1994. *The Workflow Reference Model.* Document Number TC00-1003.

BIS Strategic Decisions, One Long Water Circle, Norwell, MA 02061. *The BIS Guide to Workflow Software*. September 1995.

© Copyright IBM Corp. 1995, 1996

Appendix B. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

CICS
FlowMark
IBM
IBMLink
IMS
MQSeries
Open Blueprint
OS/2
VisualAge

The following terms are trademarks of other companies:

OpenDoc	Apple Computer, Incorporated

© Copyright IBM Corp. 1995, 1996

Appendix C. Communicating Your Comments to IBM

If you especially like or dislike anything about this paper, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply. Feel free to comment on specific error or omissions, accuracy, organization, subject matter, or completeness of this paper.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

- If you prefer to send comments by FAX, use this number:

  United States and Canada: 1-800-227-5088.

- If you prefer to send comments electronically, use one of these ID's:
  - Internet: **USIB2HPD@VNET.IBM.COM**
  - IBM Mail Exchange: **USIB2HPD at IBMMAIL**
  - IBMLink: **CIBMORCF at RALVM13**

Make sure to include the following in your note:

- Title of this paper
- Page number or topic to which your comment applies

© Copyright IBM Corp. 1995, 1996

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC23-3935-01