Chapter 5

Workflow Technologies

Cite edilecek: [Senkul et al., 2002]

Commercial products

- Staffware
- MS MAPI
- IBM flowmark
- Workflo
- Inconcert
- COSA

Research Projects

- Meteor
- Mobile
- Exotica
- WASA
- WIDE

In simplist terms, a workflow is the movement of documents and tasks through a business process. A Workflow System provides for the automation of a business process, in whole or part, during which documents, information, or tasks are passed from one participant to another for acting, according to a set of rules.

Another definition says, workflows are activities involving the coordinated execution of multiple tasks performed by different processing entities, mostly in distributed heterogeneous environments which are very common in the enterprises of even moderate complexity. These activities could be manual or automated, possibly being already-existing legacy programs.

Starting with early 1990s, workflow systems have been an active research and development area with several research prototypes and commercial products in the market. Meanwhile, when the workflow market started to grow, other market segments started to include some of the workflow capabilities. Enterprise Resource Planning (ERP) started to increasingly support workflow capabilities. Most leading ERP systems (e.g., SAP ERP Solutions [SAP], BaanERP [Baan], and PeopleSoft [Peoplesoft]) offer a workflow component. In fact, as predicted in [Sheth et. al., 1999], currently, workflow process management functions and technology are absorbed by other technologies. Although there are stand-alone workflow management systems on which workflow applications are built, the trend is to have workflow capability in critical enterprise application systems such as Enterprise Resource Planning (ERP) and supply-chain management, and e-commerce solutions.

There are many business models used in electronic commerce (EC) like e-shop, e-procurement, e-mall, electronic marketplace, virtual communities, value chain service providers, value chain integrators, collaboration platforms, and information brokerage [Timmers, 1998]. In all of these models the business processes can be modeled as a set of steps that are ordered according to the control and data flow dependencies among them. This corresponds to a workflow process, where the coordination, control and communication of activities are automated, although the activities themselves can either be automated or performed by humans.

New technology integration standards such as XML Schema, SOAP, and J2EE enable the convergence of legacy infrastructures toward process-oriented enterprise computing. On the other side, emerging protocols such as ebXML, RosettaNet, and BizTalk support the process-level collaboration among business partners. To support enterprise business processes in electronic commerce applications, workflow systems should have certain features that are of critical importance [Muth et al., 1998]:

- The mass-business characteristics of EC workflows requires a high-throughput workflow execution engine. Thus, load distribution across multiple workflow servers is necessary to ensure this kind of scalability. The EC workflow systems must also quickly adapt to network changes due to failed sites or due to load balancing.
- The EC workflows should easily adapt to different and changing requirements of the customers. So a workflow model is more likely to be a template that is dynamically enriched by introducing additional activities along with their control and data flow, and also possibly skipping the parts of the pre-specified workflow template. Also there could be changes in EC process execution flow triggered by collaborative decision points, or context-sensitive information updates or other internal or external events which necessitate dynamic modification of the workflow instance (e.g., cancelation of an order by the customer).
- Electronic commerce processes should be ubiquitous. To achieve this they should be able to run in environments with scarce resources, and they should also have an open architecture. That is the functionality of a workflow system should be tailorable to the needs and available resources of a customer and the system should run on the Internet and should be based on an open and interoperable infrastructure.
- Frequent failures and unavailability of EC workflow servers would immediately weaken the market position of the merchant. This requires efficient replication of workflow servers.

In this chapter we first introduce the standard bodies for workflow systems and briefly describe fundamental features of the workflow management systems. Then in the following two sections, we summarize several research prototypes and a few of the significant commercial workflow products. Finally, Section 5.5 concludes the chapter.

5.1 Standard Bodies for Workflow Systems

There have been founded a number of standards bodies to work on workflow systems and related issues to bring up some standards. We mention three of them here, the most famous one being the Workflow Management Coalition. A general list of Standard Bodies and Technical Comittees can be found at [DSP].

The Workflow Management Coalition [WfMC], founded in August 1993, is a non-profit, international organisation of workflow vendors, users, analysts and university/research groups. Consisting of over 285 members spread around the world, the Coalition has quickly become the primary standards body for this rapidly expanding software market. The Coalition aims to promote and develop the use of workflow through the establishment of standards for software terminology, interoperability and connectivity between workflow products.

The initial work of the Coalition focused on publishing the Reference Model and Glossary, defining a common architecture and terminology for the industry. A major milestone was achieved with the publication of the first versions of the Workflow API (WAPI) specification, covering the Workflow Client Application Interface, and the Workflow Interoperability specification. The Audit Data specification was added in 1997, being followed by the Process Definition Import/Export specification. Further work includes the completion of a common object model with object bindings for IDL and OLE, interoperability extensions for security, and additional interoperability models.

The charter of the Workflow And Reengineering International Association [WARIA] is founded to identify and clarify issues that are common to users of workflow, electronic commerce and those who are in the process of reengineering their organizations. The association facilitates opportunities for members to discuss and share their experiences freely. Established in 1992, WARIA's mission is to make sense of what's happening at the intersection of Business Process Management, Workflow, Knowledge Management and Electronic Commerce and reach clarity through sharing experiences, product evaluations, networking between users and vendors, education and training.

The Business Process Management Initiative [BPMI.org], founded in August 2000, is a non-profit corporation that empowers companies of all sizes, across all industries, to develop and operate business processes that span multiple applications and business partners, behind the firewall and over the Internet. The Initiative's mission is to promote and develop the use of Business Process Management (BPM) through the establishment of standards for process design, deployment, execution, maintenance, and optimization.

[BPMI.org] develops open specifications, such as the Business Process Modeling Language (BPML), and the Business Process Query Language (BPQL), that will enable the standards-based management of e-Business processes with forthcoming Business Process Management Systems (BPMS), in much the same way SQL enabled the standards-based management of business data with off-the-shelf Database Management Systems (DBMS).

5.2 Workflow Management Systems

A workflow process is a collection of processing steps (also termed as tasks or activities) organized to accomplish some business processes. In addition to the collection of tasks, a workflow defines the order of task invocation or condition(s) under which tasks must be invoked (i.e. control-flow) and data-flow between these tasks. This definition may also express constrains and conditions such as when the activities should be executed, a specification of who can or should perform each activity, and which tools and programs are needed during the activity execution [Jablonski and Bussler, 1996]. Management of workflows deals with the automated coordination, control and communication of work as required to satisfy workflow processes.

A Workflow Management System (WFMS) is a software providing support for the necessary services of workflow definition and process creation, workflow enactment, administration and monitoring of workflow processes. It allow organizations to define and control the various

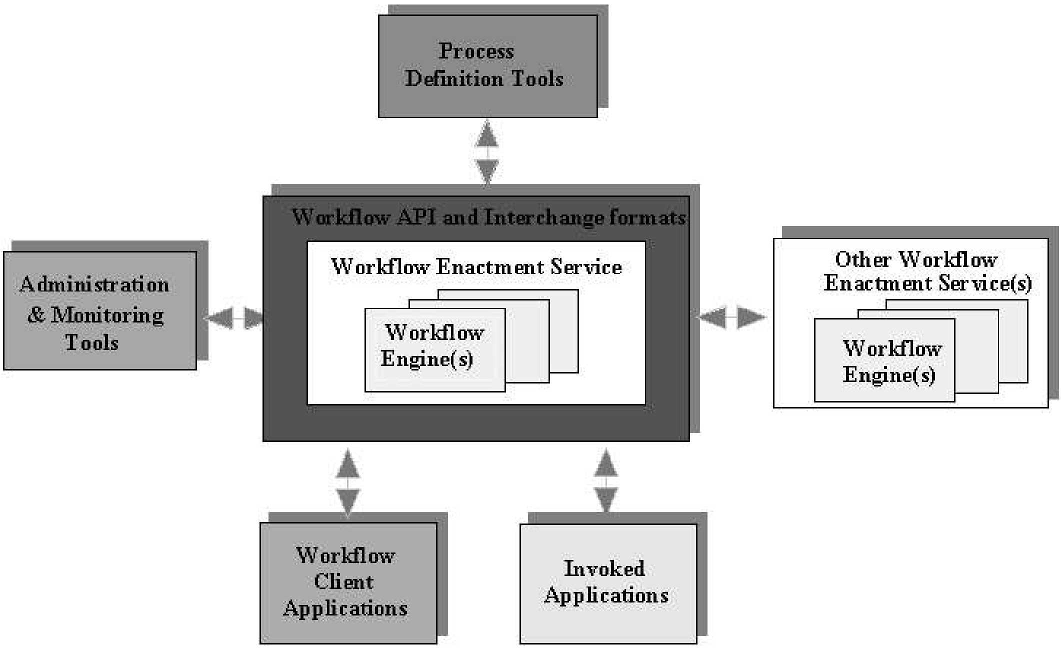

Process Definition Tools
Workflow API and Interchange formats
Workflow Enactment Service
Workflow Engine(s)
Administration & Monitoring Tools
Other Workflow Enactment Service(s)
Workflow Engine(s)
Workflow Client Applications
Invoked Applications

Figure 5.1: Workflow Reference Model - Components & Interfaces

activities associated with a business process. A workflow application developer generally relies on these tools for the specification of the workflow process and the data it manipulates. A workflow process is principally based on a formalized workflow model that is used to capture data and control-flow between workflow tasks.

Workflow management systems enable building up the corporate knowledge. The business experience and knowledge is extracted from worker‘s heads into the process definitions. Process information that may have formerly been scattered among various staff members gets combined, documented, and ready to be reevaluated. For many organizations, the introduction of workflow management tool shoulds better be considered as an opportunity to improve both the underlying business process and the existing organizational structure [DiCaterino et al., 1997].

A comprehensive overview of workflow management systems is presented in [Georgakopoulos et al.,1995].

The Workflow Management Coalition [WfMC] is the primary standard body for workflow systems. The standardization work of WfMC is centered around the workflow reference model. The reference model specifies a framework for workflow systems, identifying their characteristics, functions and interfaces as demonstrated in Figure 5.1.

The major components of a workflow system are as follows:

- Workflow Enactment Service: 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 centralised 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.

A workflow engine is responsible for part (or all) of the runtime control environment within an enactment. A software service or "engine" that provides the run time execution environment for a workflow instance. Typically such software provides facilities to handle:

  - interpretation of the process definition
  - control of process instances - creation, activation, suspension, termination, etc
  - navigation between process activities, which may involve sequential or parallel operations, deadline scheduling, interpretation of workflow relevant data, etc
  - sign-on and sign-off of specific participants
  - identification of workitems for user attention and an interface to support user interactions
  - maintenance of workflow control data and workflow relevant data, passing workflow relevant data to/from applications or users
  - an interface to invoke external applications and link any workflow relevant data
  - supervisory actions for control, administration and audit purposes.

  A workflow engine can control the execution of a set of process, or sub-process, instances with a defined scope - determined by the range of object types, and their attributes, which it can interpret within the process definition(s). In an enactment service consisting of multiple workflow engines, there is a partitioning of process execution across the constituent engines.

- Process Definition Tools: A process definition tool should be able to express the data and control flow within a process as well as the activities and the roles involved. The WfMC has identified a set of six primitives with which it is possible to describe flows and hence construct a workflow specification. With these primitives it is possible to model any workflow that is likely to occur. These primitives are: *Sequential Routing, AND-split, AND-join, OR-split, OR-join* and *Repeatable Task.*

  **** bunu atip, 'process modeling tools' diye ayri bir section yapalim. ****

  Sequential Routing expresses the fact that the workflow activities are executed in sequence. For example, in an e-marketplace workflow the customer needs to be registered first before he starts filling his shopping basket. *AND-split* shows a case where a single thread of control splits into two or more threads in order to execute activities in parallel. *AND-join* allows two or more parallel executing activities to converge into a single common thread of control. *OR-split* makes it possible for a single thread on control to make a decision upon which branch to take when encountered with multiple branches to workflow process activities. As an example, an e-marketplace workflow system can allow its customers to pay by check or credit card and one of the routes is chosen according to customer preference. *OR-join* also allows two or more workflow process activities physically connect or converge to single activity like *AND-join.* However, in contrast there is no synchronization of threads of control of the involved activities.

  Buraya ornek lazim.
  ****

- Administration and Monitoring tools: Within a workflow system there are a number of supervisory functions which are normally provided. These functions may enable supervisors to alter work allocation rules, to identify participants for specific organisational 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 enquire about work throughput or other statistics, etc.

- Workflow Client Applications: In the workflow model interaction occurs between the client application and the workflow engine through a well defined interface embracing the concept of a worklist - the queue of work items assigned to a particular user (or, possibly, group of common users) by the workflow engine. At the simplest level the worklist is accessible to the workflow engine for the purposes of assigning work items and to the worklist handler for the purpose of retrieving work items for presentation to the user for processing.

- Invoked Applications: A Workflow Management System should be able to invoke applications. The invoked application may be local to the workflow engine, co-resident on the same platform or located on a separate, network accessible heterogeneous platform; the process definition should contain sufficient application type and addressing information (specific to the needs of the workflow engine) to invoke the application.

The WfMC specifed the five APIs that surround the workflow engine (Figure 5.1). These APIs provide a standard means of communication between workflow engines and clients including other workflow components such as process definition and monitoring tools.

Major benefits of using workflow systems can be summarized as follows [WfMC]:

- *Improved efficiency:* Automation of business processes results in the elimination of many unnecessary steps.

- *Better process control:* Business processes management is impoved by standardizing working methods and the availability of audit trails.

- *Improved customer service:* Consistency in the processes leads to greater predictability in levels of response to customers.

- *Flexibility:* One of the greatest challenges of business process management is the flexibility [Klein et al., 2000]. Software control over processes enables their re-design in line with changing business needs.

- *Business process improvement:* Focusing on business processes leads to their streamlining and simplification.

5.2.1 Workflow Categories

Popular classification distinguishes three types of workflows:

- *Ad hoc Workflows:* Workflows are controlled by users at run time. Users can react to situations not considered at build-time.

- *Administrative Workflows:* Predictable and repeatable workflows described in simple description languages. Activities are mainly performed by humans.

- *Production Workflows:* Predictable and repeatable complex workflows which are predefined completely and in great detail using complex information structures and involve application programs and automatic activities.

5.2.2 Basic Functionality of Production Workflows

There is a large number of applications which can route a document from one person to another. However, these applications are not "Production Workflows". A "Production Workflow" product must provide the following basic functionality [**?**]:

1. Graphical Workflow Design: A means of graphically creating workflow process maps that define the flow of work and the tasks which must be performed from start to finish.
2. Ability to specify data and control flow: The ability to embed complex business logic in the workflow definition without the need for programming or scripting.
3. Exception Handling: The ability to handle exceptions which are omnipresent in every organization.
4. Monitoring: The ability to monitor the status of workflow incidents. Ideally, this ability should be available to each workflow participant for incidents they have participated in, and to a centralized workflow administrator for all workflow incidents.
5. Measuring: The ability to generate workflow statistical reports so business managers can measure the time and cost of workflow process, and can then modify them based upon their cost-effectiveness and timeliness.
6. Pro-Active: The ability to move workflow forward on a pro-active basis. The workflow solution must inform users of new tasks.
7. Database Connectivity: Every workflow process either uses information from databases to enable users to make decisions, or feeds information into databases. In many cases, they do both. Thus, a workflow solution must provide seamless database connectivity.
8. Document Attachments: Documents are an essential part of business processes. A workflow solution must, therefore, provide an effective means of attaching documents to the workflow, which are then used to support the business process.

5.2.3 Process Modeling Techniques

Workflow process is inextricably linked with the notion of business process. Business processes are the main locus of organizational work. In the late 1990s, Workflows evolved to support the need to explicitly manage and sequence work processes. Work steps occur in a sequence from strategy decisions to sales, product or service development, delivery, and billing.

Workflow systems allows isolation of business process definitions from the execution environment aspects. In fact, this is the ideal time for organizations to consider business process reengineering. [Kobileus, 1997] suggests that an organization optimize a process with any of the three goals in mind: “minimizing process time, maximizing value-added process content, or maximizing flexibility at the initial point of customer contact”.

Characteristics of a Process

A process viewed from various perspectives depending on the kind of information required. Usually this of the type, what work is going to be done, who and how is it going to be done, when will it be done, who will take the decision.

The most common views are the functional, informational, behavioural and the organisational views [Bal, ????]. A clarification of these terms is important:

- *The functional view*, represents what activity or element of the process is being performed. It represents the act or activity that is being done by the actors or the employees.
- *The Behavioural view*, It relates to when the process is being performed, and how it is being done. The activity or process as a whole could be going through a feedback loop or a iterative process, etc.

- *The Organisational view*, represents who is performing the process. The mechanism through which there is interaction or transfer of content.
- *The informational view*, represents the information details or entities that are being manipulated by the process, these can be data, products entity details. The information view considers both the data involved and their relationships.

An analysis of the modelling techniques needs to keep in mind the process characteristics and the objectives of the modelling.

Process Modelling Methods

A variety of methodologies are used for process modelling. The methodologies listed below are the most frequently used techniques, they are also representative of the various concepts that have been developed in the field of process modelling.

- *Structured System Analysis and Design Method (SSADM)* adopts a precriptive approach to information systems development in that it specifies in advance the modules, stages and tasks which have to be carried out, the deliverables to be produced and furthermore the techniques used to produce the deliverables. It adopts the Waterfall model of systems development, where each phase has to be completed and signed off before subsequent phases can begin.
- *Structured Analysis and Design Technique (SADT)* There exists two types of SADT models. An Activity model is oriented toward the decomposition of activities whereas a Data model is oriented toward the decomposition of data. Each type of model contains both activities and data; the difference lies in the primary focus of the decomposition.
- *Information Definition Exchange Format (IDEF)* was developed by the US Air Forces ICAM (Integrated Computer Aided Manufacturing) programme. Several variants of the IDEF methodology were defined for specific purposes.
  - *IDEF0* is one of the most widely known tool functional modelling. [IDEF0] is a top-down hierarchical method which provides a structured description of functions and processes in manufacturing. An IDEF0 model consists of an ordered collection of diagrams, text and glossary, all cross-referenced to each other.
  - *IDEF1* was designed to analyse and determine the information resource management needs and requirements. IDEF1x extends IDEF1 to provide better graphical representation, enhanced semantic richness, and simplified development procedures
  - *IDEF2* describes the dynamic aspects of a system: the resources to be used, the path an entity can take, and the status of the resources.
  - *IDEF3* is different from the others as it captures the description of what a system actually does. There are two basic components of the IDEF3 description language: (1) the process flow description and (2) object state transition network description. The two components are cross-referenced to build the diagrams.
- *Data Flow Diagram (DFD)* shows the flow of the data through a system, and the work or processing performed on the data as it moves through the system.
- *Petri Nets* are a graphical and mathematical tool that can be used to represent procedures, processes, machines and organisations. They can be used to describe, analyse and study various business processes [Aalst and Hee, 1996]. Petri nets use tokens to reflect the dynamic nature of a process.

- *Role Activity Diagram (RAD)* focuses on the people aspect of a process in relation to the organisation by defining the roles, their component activities and their interactions.

- *State Transition Diagram (STD)* models a machine that has a number of states (hence the term finite state machine). The machine receives events from the outside world, and each event can cause the machine to transition from one state to another.

- *Unified Modeling Language (UML)* has become the de facto standard for software development. It has four diagrams for process modeling, and supports variants of statecharts. Its activity diagrams are inspired by Petri nets.

5.3 Workflow Research Prototypes

One of the first workflow research projects is the ConTract project [Wachter and Reuter, 1992] The focus of this project has been to extend transaction-oriented run-time mechanism for fault tolerant workflow execution in a distributed environment. The ConTract model tries to provide the formal basis for defining and controlling long lived, complex computations, just like transactions control short computations. The basic idea of the ConTract model is to build large applications from short ACID (Atomicity, Consistency, Isolation and Durability) transactions and to provide an application independent system service, which exercises control over them. As a main contribution, ConTract provides the computation as a whole with reliability and correctness properties: A large distributed application is divided into a set of related processing steps that have appropriate consistency and reliability qualities for their execution.

METUFlow [Dogac et al., 1998d] is a fully distributed workflow management system that uses a block structured process specification language called MFDL. A process tree is generated from a process definition, and it is used for guard generation for each task in the process definition. The guards are used by task handlers to control the activation of the user tasks at run time. Distributed execution of METUFlow is based on CORBA.

Another distributed workflow management system is METEOR (Managing End to End Organization) [Sheth and Kochut, 1998]. Its workflow execution model is driven by inter-task dependency rules that are expressed in a specifically designed script language. METEOR allows workflow definition at two levels of abstraction using two different languages: the Workflow Specification Language, and the Task Specification Language. Process definitions are saved in an intermediate format that is used for automatic code generation at run time. The runtime code generators output code for task managers and task invocation, data object access routines and recovery mechanism. The generated code includes all the inter-task dependencies required by the definition of the process, and it is based on CORBA and Web environment for distributed execution.

In [Miller et al., 1998b], the use of Web technology for workflow is presented with METEOR2 Web-based workflow management system (WebWork). WebWork is said to be web-based rather than web-enabled since both interfaces and communication/ distribution infrastructures are built using Web technology. Data flow is realized through exchanging HTML pages and CGI is the main communication mechanism with servers.

There is also some previous work on realizing a workflow system with the use of agents. DartFlow workflow management system [Cai et al., 1997] uses Web-browser embedded Java applets as its front end and transportable agents as the backbone. A transportable agent is a program that migrates machine to machine in a heterogeneous network. In DartFlow, each business process can be handled by an agent. Agent Tcl system is used to implement transportable agents. Since agents in DartFlow do not use a standard communication language, its usage is limited to those who make use of Agent Tcl system.

WISE system [Alonso et al., 1999] describes an infrastructure for business to business electronic commerce. This infrastructure includes an "internet workflow engine" acting as the underlying distributed operating system controlling the execution of business processes, a process modeling tool for defining and monitoring the processes, a catalog tool for virtual enterprise services. In WISE, virtual business processes are constructed by using the services offered by different companies as building blocks. The WWW catalog provided uses Java Applet/Servlet technology to allow companies in the trading community to advertise and to see the semantics of the services provided by other companies.

CrossFlow [Grefen et al., 2000], [Hoffner et al., 2001] aims at providing high-level support for workflows in dynamically formed virtual organizations. Virtual organizations are formed when some activities of a company is outsourced to external service providers. High level support is obtained by abstracting services and offering advanced cooperation support. Virtual organizations are dynamically formed by contract-based match-making between service providers and consumers.

In Panta Rhei Workflow System [Eder et. al, 1998], the user interface is completely integrated in a Web browser. The system supports workflow transactions, exception handling and allows for definition of activities, roles and organization structures. There are sophisticated mechanisms to handle time dependent aspects of workflows.

In [Muth et al., 1998], an approach towards the specification, verification, and distributed execution of workflows based on state and activity charts is presented. The workflow definition extracted from the state and activity charts are used for execution of processes in a distributed manner.

5.3.1 MARIFlow: A Maritime Industry Workflow System

Conventional workflow systems focuse on improving the efficiency of business processes within an organization. Due to the different nature of inter-organizational workflows, conventional workflow technology can not be directly applied to the processes crossing enterprise boundaries. In this section, an inter-organizational research workflow system involving many organizations is discussed.

The MARIFlow system is developed in scope of the [MARIFlow] project for automating and monitoring the flow of control and data over the Internet among different organizations, thereby creating a platform necessary to describe high-level processes involving several organizations and companies.

Materials used in ship building or repairs need to be certified by a classification society. The certification process involves organizations such as the steel companies, ship yards, classification societies and legal/insurance/government establishments. In current practice, the material is checked while it is at the production plant and the "quality data" is delivered to the classification society. If the quality data fulfills the requirements, a paper certificate is issued and delivered to the production plant as well as to the customer. Once issued, the certificate follows the material to the main shipyard or to one of the subcontractors from where it will eventually be added to the ship's documentation file. The certificate is checked at every production stage as well as at the ship's handover and at each survey during ship's life cycle. Circulation of large quantities of paper documents among different organizations is a slow, expensive, tedious, error-prone and very limiting process which in some cases, can hinder the ability to improve the service quality.

In the MARIFlow system, the workflow process is defined through a graphical user interface which is then mapped to a textual language called FlowDL. It is a block structured language that encapsulates the six primitives defined by the Workflow Management Coalition [WfMC]. FlowDL additionally allows specifying the source of the documents, their control

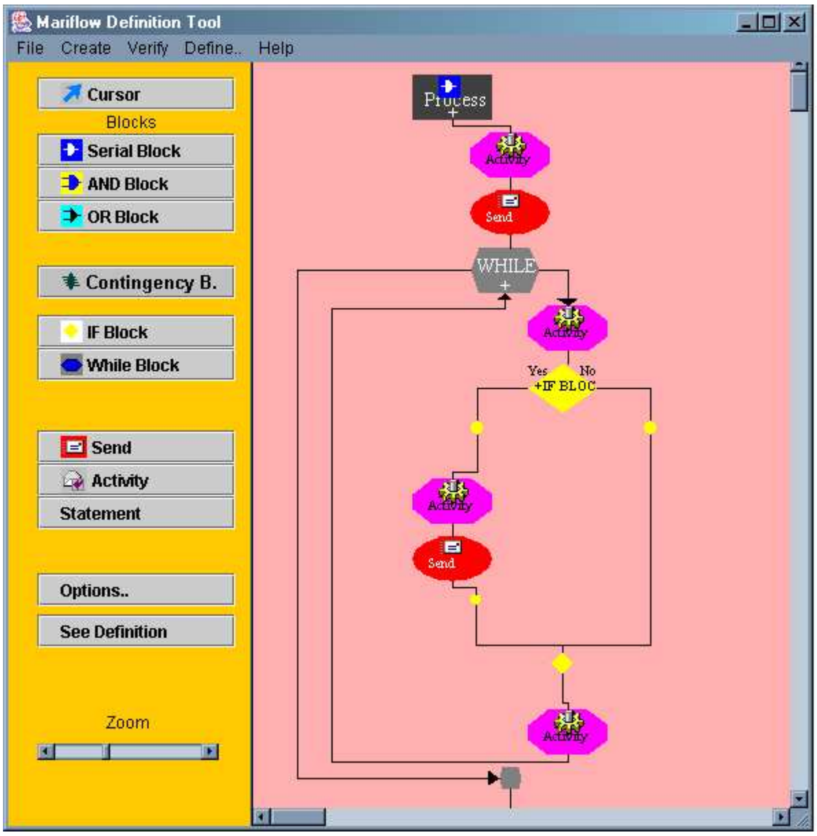


Figure 5.2: The Graphical User Interface for Process Definition in MARIFlow

flow and the activities that make use of these documents.

A MARIFlow process is executed through cooperating agents, called Mariflow Cooperating Agents (MARCA). Each participating site has its own MARCA. MARCAs handle the activities at their site, provide for coordination with other MARCAs in the system by routing the documents in electronic form according to the process description, keeping track of process information, and providing for the security and authentication of documents as well as comprehensive monitoring facilities.

The responsibilities of the MARCAs are as follows:

- A MARCA receives messages through a persistent queue and evaluates them to decide what specific action to take.
- It persistently stores the documents it receives. It should be noted that the organizations may be reluctant to grant access inside the corporate firewall. In such cases, MARCA passes these documents to an in-house system by properly acknowledging the in-house system on further processing that may be necessary on the documents. The MARCA is also responsible for getting the documents from the in-house system and forwarding them to the related agents as specified in the process definition.
- Process related information also needs to be stored persistently for recovery and monitoring purposes. MARCAs store this information to a database through the JDBC interface.
- There is a single MARCA at each site that handles all the activities of all workflows related with that site. A new MARCA need to be generated when a new site is introduced

for the first time to participate in workflow processes.

5.3.1.1 Workflow Definition Language: FlowDL

In MARIFlow, an inter-organization workflow process is defined graphically as shown in Figure 5.2. This GUI tool allows the workflow designer to specify domains, tasks and process information which are then used in building the process definition graphically. The resulting process definition is mapped to the textual FlowDL language.

FlowDL contains seven types of blocks, namely, serial, and_parallel, or_parallel, xor_parallel, contingency, conditional and iterative blocks. These block types encapsulate the workflow primitives defined by [WfMC]. In FlowDL, a workflow process is defined as a collection of blocks, tasks and other subprocesses, as well as some explicit declarations to specify Internet domain addresses, sources of documents, and process specific information to be used for monitoring document flow.

Figure 5.3 presents an example workflow definition reflecting a business scenario for a certification process in maritime industry, and Figure 5.4 illustrates the process tree generated from this example process definition. The process tree consists of nodes representing processes, blocks and tasks, and is used only during compilation time, execution being completely distributed. Each activity in the process tree receives/sends messages from/to other activities through its parent block. At compile time the guards are generated and stored locally with the related objects. A detailed formal description of obtaining guard expressions from a given workflow specification is given in [Dogac et al., 1998e].

The advantages brought by the FlowDL process definition language can be summarized as follows:

1. As noted in [NSF], most workflow specification languages are unstructured and/or rule based. Unstructured specification languages make debugging/testing of complex workflow difficult and rule based languages become inefficient when they are used for specification of large and complex workflow processes. This is due to the large number of rules and overhead associated with rule invocation and management. FlowDL avoids these disadvantages through its block-structured nature.
2. A block structured language confines the intertask dependencies to a well formed structure which in turn proves extremely helpful in generating the guards of activities for distributed scheduling of a workflow.
3. A block clearly defines not only the data and control dependencies among tasks but also presents a welldefined recovery semantics, i.e., when a block fails (aborts), the tasks that are to be compensated and the order in which they are to be compensated are already provided by the block semantics.

5.3.1.2 An Overview of the Mariflow Architecture

Figure 5.5 shows an overview of the general architecture of the MARIFlow system. Each organization may have in-house applications inside a firewall protected from unauthorized accesses. MARCAs live outside the firewall, and send a documents and process related information inside the firewall as an e-mail attachments.

Inside the firewall, there is the User MARCA Interfacing Application (UMIA) and the Application Server. The Application Server reads the incoming mails and informs the UMIA accordingly. In order to invoke external applications automatically within the MARIFlow System, a mapping should be provided from the activities defined in the workflow definition

```
PROCESS MARIFlow ();

ACTIVITY archive (IN document arch_document);
ACTIVITY convert_qlty_data(IN document quality_data
          OUT document convted_q_d);
ACTIVITY convert_certificate(IN document certificate
          OUT document convted_cert);
ACTIVITY issue_certificate (IN document conv_quality_data
          IN document product_spec OUT document certificate);
ACTIVITY delete_from_archive();
ACTIVITY cancel_converted_data();
ACTIVITY cancel_certificate();

DOMAIN_DEFINITION     {
        salzgitter_ag.de   szag;
        germanlloyd.org    gl;
        balance_bremen.de  bal;
        isisanisi.com      isisan;
}
struct process_info   {
        string  order_no;
        string  material_no;
        string  customer_name;
        string  supplier_name;
        string  class_society_name;
        string  certificate_number;
}
DEFINE_PROCESS MARIFlow ()
{
  szag SENDS (quality_data) TO bal GENERATES (order_no
       material_no customer_name supplier_name
       class_society_name);
  AND_PARALLEL {
     SERIAL {
         START convert_qlty_data (IN quality_data OUT
             convted_q_d) AT bal COMPENSATED BY
             cancel_converted_data();
         bal SENDS (convted_q_d) TO gl AND isisan;
         START archive (IN convted_q_d) AT isisan NON_VITAL
         COMPENSATED BY delete_from_archive();
     }
     isisan SENDS  (prod_spec) TO gl;
  }
  START issue_certificate (IN convted_q_d IN prod_spec OUT
      certificate) AT gl GENERATES (certificate_number)
      COMPENSATED BY cancel_certificate();
  gl SENDS (certificate) TO bal;
  START convert_certificate (IN certificate OUT convted_cert)
      AT bal;
  bal SENDS (convted_cert) TO szag AND isisan;
  AND_PARALLEL {
      START archive (IN convted_cert) AT szag NON_VITAL
      COMPENSATED BY delete_from_archive();
      START archive (IN convted_cert) AT isisan NON_VITAL
      COMPENSATED BY delete_from_archive();
  }
}
```

Figure 5.3: An example Process Definition for the Maritime Industry

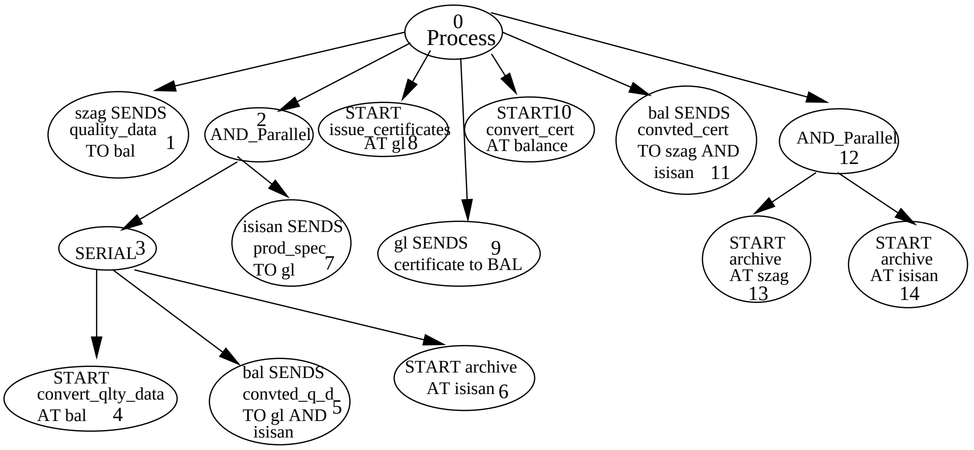


Figure 5.4: Process tree of the Example Process Definition

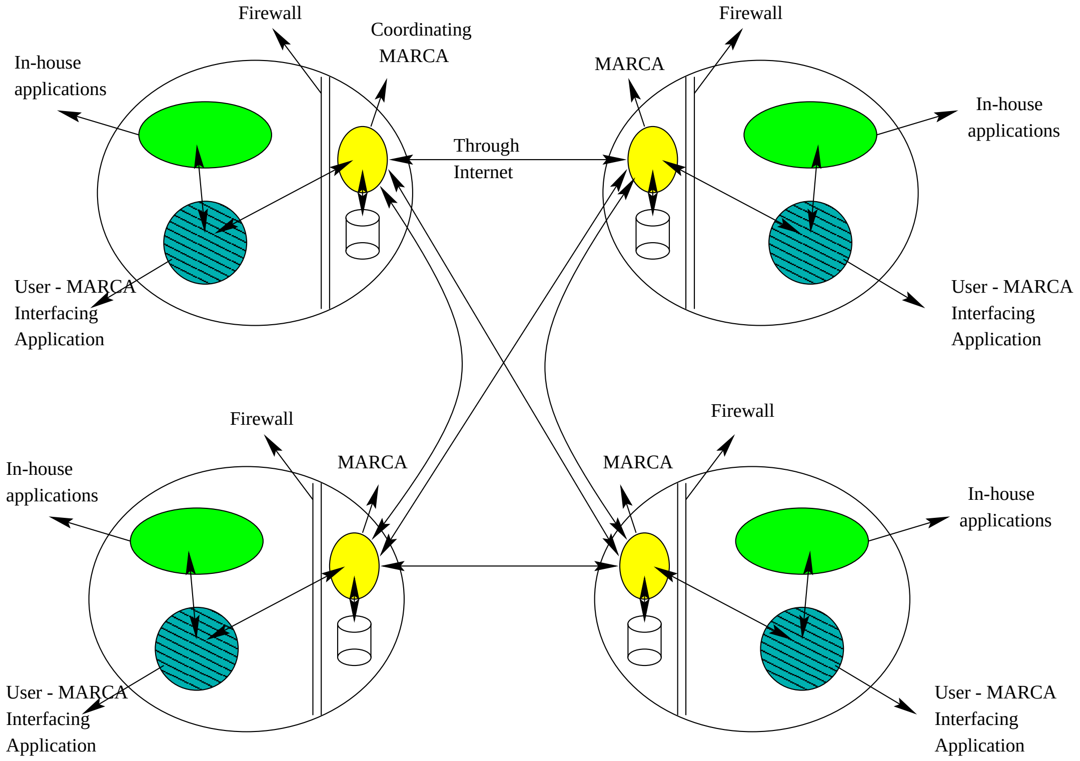
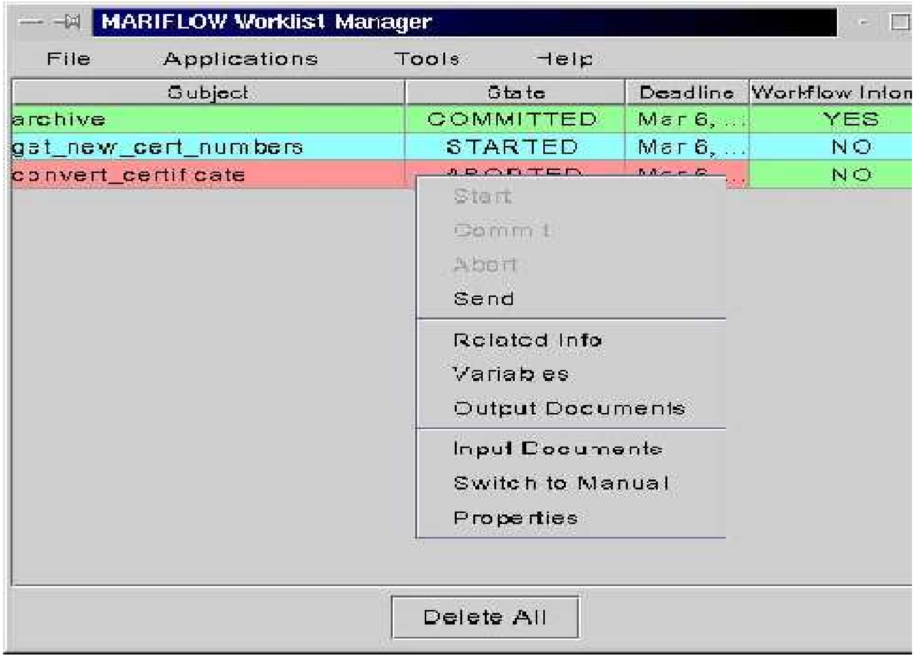


Figure 5.5: An Overview of the Architecture

to the external applications in the company domain. This mapping is achieved through a graphical user interface, which is a part of the UMIA, that also allows the parameters of the application to be specified. If the invocation is to be done manually, i.e., no mapping is defined, then the user invokes the task from the UMIA manually as shown in Figure 5.6.

To provide the global coordination of the Mariflow system, a coordinating MARCA is installed at one of the sites. The functionality provided by the system can be summarized as follows:

- A declarative means to specify the control and document flow over the Internet where



Subject	State	Deadline	Workflow Infon
archive	COMMITTED	Mar 6, ...	YES
get_new_cert_numbers	STARTED	Mar 6, ...	NO
convert_certificate	ABORTED	Mar 6, ...	NO

Figure 5.6: GUI of User MARCA Interfacing Application

it is possible to define the source of data, its control flow and the activities that make use of this data.

- Invoking external applications (which may be inside the company domain) when necessary.
- Authentication and security of documents and the process related information.
- A monitoring mechanism for keeping track of the documents and/or for providing detailed account of the current status of a process instance within the system.
- Measures for failure recovery and exception handling

5.3.1.3 MARIFlow Cooperating Agents: MARCAs

A workflow instance in MARIFlow is executed by cooperating agents called MARCAs. There is only one MARCA at each site participating to the workflow execution and it handles all the activities running at its site. An overview of MARCA architecture is shown in Figure 5.7 and Figure 5.8 shows the technical architecture details.

Activities start, terminate or fail according to some guards and conditions. These guards and conditions are evaluated through messages exchanged among MARCAs. In other words, a MARCA receives the messages sent to it by other MARCAs, evaluates the guards and the conditions of the significant events (start, fail or terminate) of activities under its responsibility and for each of these significant events determines which other MARCAs should be informed.

MARCAs communicate with each other in a Mariflow specific Agent Communication Language, and exchange messages with each other through TCP/IP over the Internet. Network sockets are the basic mechanism of communication among MARCAs.

Before cooperating agents can be used within a workflow process, they need to be downloaded and setup at each site participating in that process. This download operation is

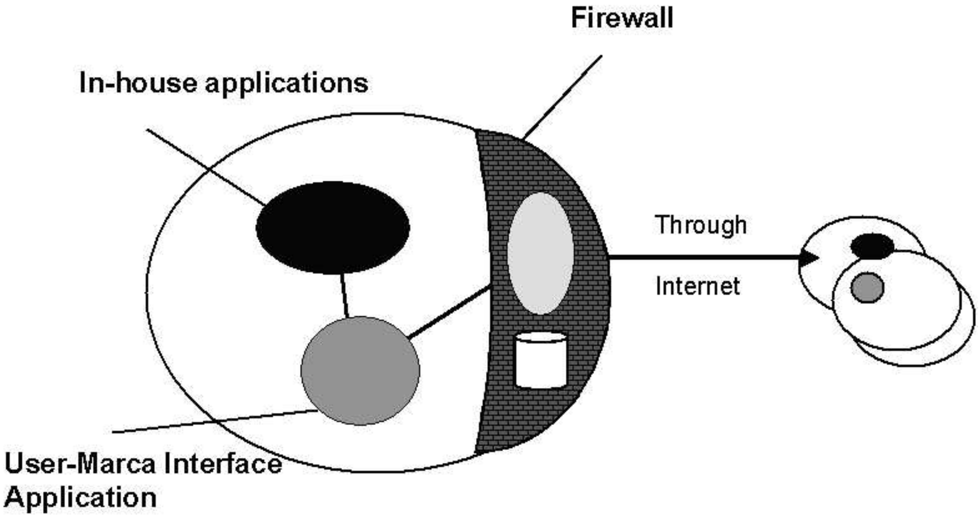


Figure 5.7: Overview of an organization/site with a MARCA

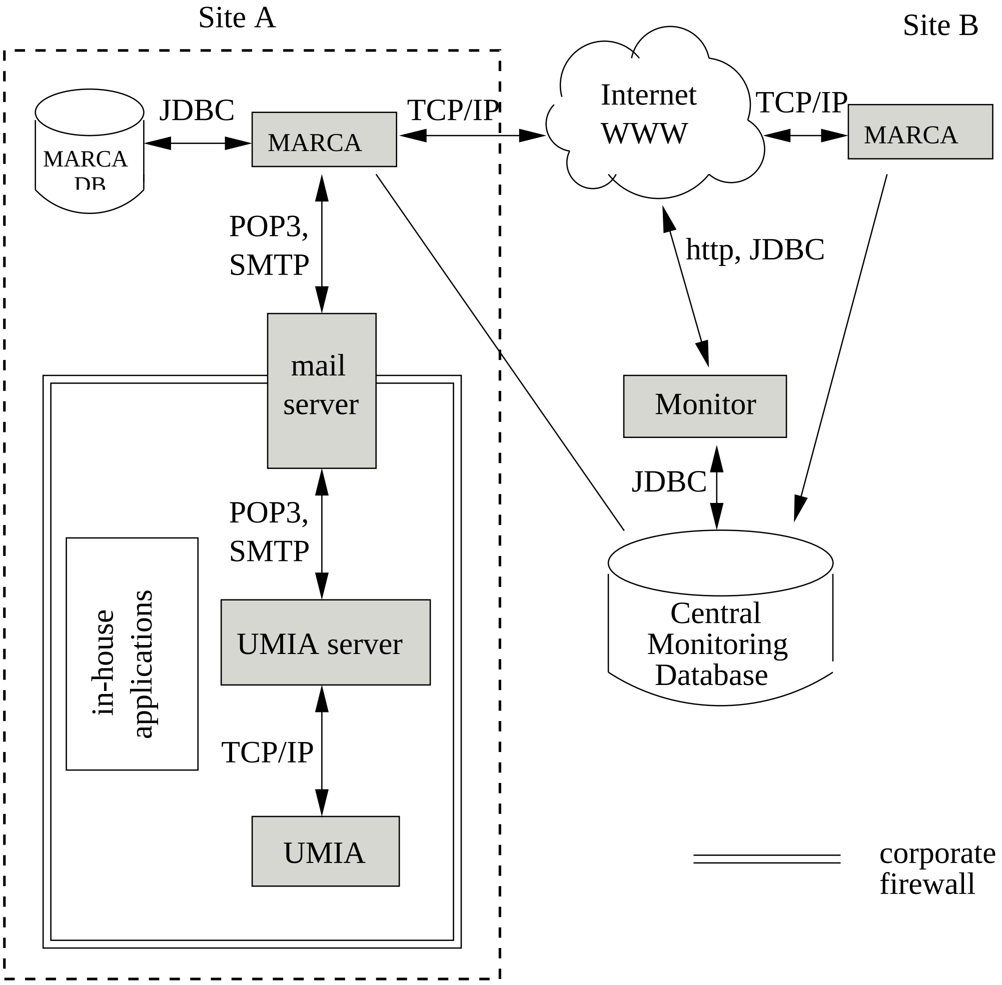
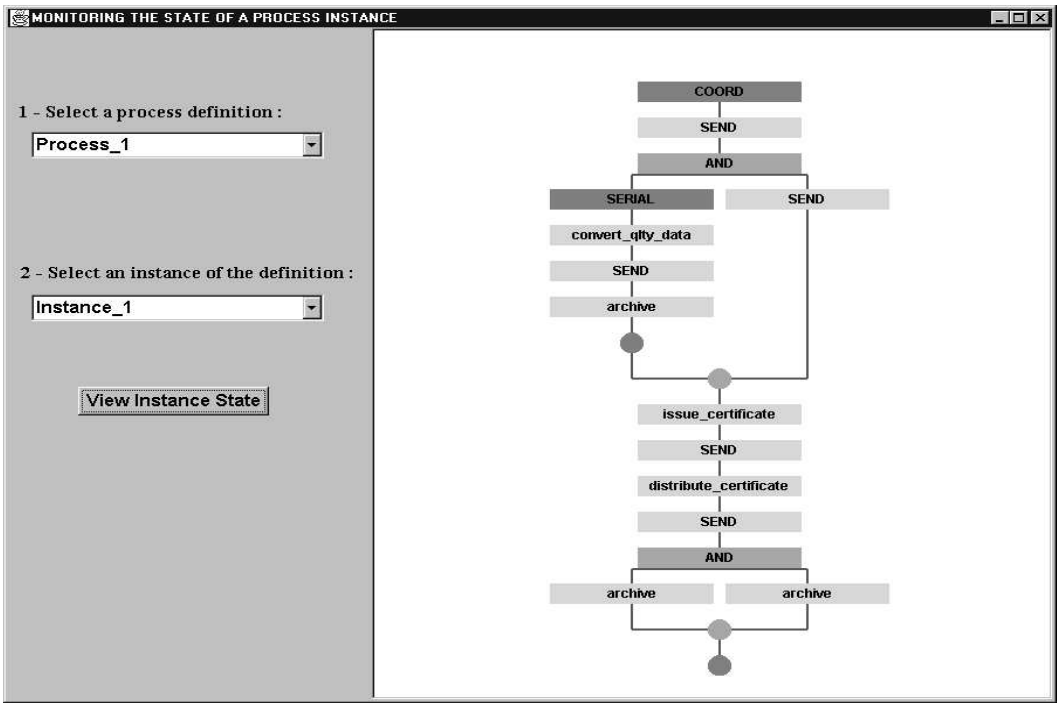


Figure 5.8: Technical details of the site architecture



MONITORING THE STATE OF A PROCESS INSTANCE

1 - Select a process definition : Process_1

2 - Select an instance of the definition : Instance_1

Figure 5.9: Graphical Monitoring Interface of the system

necessary only for *new* sites that participates in the execution of the process.

5.3.1.4 Monitoring and Recovery of the System

Each MARCA stores all the documents and messages it receives persistently in its database. Since Java is used in coding the MARIFlow system, a Java native JDBC interface is used in accessing the database system. The information stored also serves for the purpose of a log, that is, after a crash, a MARCA can be brought back to a consistent state by using this information.

Each MARCA sends a copy of the information it stores to the central database for global monitoring. This site is available to any authorized user on the Internet through a Web interface and may further be replicated for availability purposes.

The monitoring facilities provided by the MARIFlow system are twofold: (a) When an authorized user provides the process identifier (supplied to her by the MARIFlow system when the process starts), s/he can track the flow of that process instance through a graphical interface as shown if Figure 5.9, (b) All the information in the coordinating MARCA's database can be queried via SQL queries.

What happens if a site (or the MARCA running on that site) goes down? In Mariflow, for each MARCA installed there is a background process at that site, called the "rescue process". The rescue process is responsible for monitoring the life time of the agent, and polls the MARCA at specific time intervals through a predetermined socket. A thread of the MARCA listens to this socket and responds to the signals. If the MARCA does not respond to the rescue process polls for a given period of time, the rescue process starts sending signals more frequently. After sending a bunch of signals if the MARCA still does not respond, the rescue process assumes that the MARCA is not functional. There are two possibilities in this case: the MARCA could be blocked or it could be dead. When the rescue process is unable to find the OS process that belongs to this MARCA (i.e., it is dead), it instantiates a new MARCA by the help of the persistent logs related with the state of the agent. Otherwise if the MARCA is blocked, it is necessary to kill the old instance prior to installation of a new instance. Since the logs are persistent it is possible to recover the state of the MARCA killed and hence the site does not suffer from any inconsistencies.

Indeed, for this mechanism to work, it is necessary to make sure that rescue process stays alive. Therefore, just as the rescue process polls to see that the MARCA stays alive, the MARCA also checks to ensure that the rescue process stays alive by polling the rescue process at predefined time intervals. The MARCA is responsible fro reinstantiating the rescue process if it goes down.

5.3.1.5 MARIFlow Security Services

Within the life cycle of a process definition an application in the company domain may prepare a message, then may pass it to local UMIA, which in turn, passes it to the corresponding MARCA. The MARCA sends the message to one or more other MARCAs according to the process definition. The receiver MARCAs pass the messages to their corresponding UMIAs, which pass them to the local applications. Thus, inter and intra company communications basically consists of communication sessions between MARCAs and between a MARCA and its UMIA.

The security considerations of the system can therefore be summarized as follows:

- Confidentiality of data transfer between MARCAs: The contents of a message, including the process description fields, should not be visible to anybody except the sender and the receiver.

- Confidentiality of data transfer between a MARCA and the corresponding UMIA, as above.

- Confidentiality of data transfer for inter/intra-company communications. Here, even the MARCA software, and also the UMIA, is excluded from reading the message. The process description fields, however, are visible, to enable the UMIA and MARCA to decide on how to handle the message.

- Authentication and integrity for all those three levels.

- Signatures for some of the inter/intra-company messages, such as certain types of test and certification documents. Here, a message consists of a document that is passed in order to be stored.

Note again that a message sent inside a company, as part of a MARIFlow process, should not be readable by MARCA software, or by anybody, except by the sender and receiver inside the company. Thus, confidentiality, authentication and integrity exist in the system at two levels: user to user, and UMIA-MARCA-MARCA-UMIA.

5.3.2 A Component-Based Adaptive Workflow System

The Internet and corporate Intranets have become the omnipresent communication infrastructure. Web browsers have become the universal client interface for interactive remote access since virtually every computer in an enterprise has a Web browser software installed. It is thus compelling to have workflow tools on this platfom. Users will need no workflow specific built-in functionalities, but will only rely on generic Web browsers. Equipped with such tools, a user can interact with a workflow system virtually from any computer that is connected to the internet and has a Web browser, which provides user mobility.

With network-transportable applets written in Java or Javascript, the end user can acquire workflow tools from the workflow manager over the internet. It is not necessary to have any software pre-installed on the user machine. This promotes user mobility further, as well as easy maintanance of software: tool components can be upgraded transparently on the server side.

The Internet and the object-oriented technology also enable the decomposition of a workflow engine into a collection of distributed, interoperating workflow component servers which work together to comprimise a distributed workflow engine. In particular, by making the component servers lightweight applets, a ligth-weigth workflow engine can be quickly configured for simple processes that do not require the full function workflow machinery necessary for complex enterprise-scale business processes.

In most workflow systems, even for very simple workflow processes the full scale engine runs thus consuming resources unnecessarily. These systems also suffer from difficulty in load balancing, migration of workflow processes, and efficient replication of servers.

Another very important issue, evolved from past workflow applications experience, is the ability of handling adaptable and dynamic workflows. Can a workflow process react and adapt rapidly to the changes in the run-time environment, or changes in the process execution flow triggered by collaborative decision points, context-sensitive information updates, and other internal or external events? There may be a wide range of events or conditions to which workflows and WFMS may need to adapt to, including reasons like exception handling, load balancing and a change in network confugration.

5.3.2.1 Architecture of the System

In this section, we introduce an adaptable workflow system which runs on the Internet, and adapts itself to the needs of executing workflow processes, and to the changes in the environment or in the network configuration. The architecture of the system is message driven and component based as shown in Figure 5.10. An implementation is available in [Yarimagan, 1999]. This is the architecture used in the [MESChain, 2000] project as annotated in Section 11.1.

The communication infrastructure of the system is based on TCP/IP over the internet. The components of the system are coded as Java classes, and they send and receive XML messages to communicate with each other. In this way, these workflow components can remotely be activated by the workflow domain manager or by another component. The architecture provides high availability by allowing replication of the component server repository and the workflow domain manager.

Workflow processes are defined in XML conforming to the “workflow.dtd” presented in Appendix **??**. The use of XML for process definitions provides very high transportability, thus enabling exchange of process definitions among companies when necessary. In this way, industry-specific sharable standard business process definitions may be developed.

The basic components of the workflow system presented in Figure 5.10 are discussed in the following:

1. *Workflow Domain Manager:* The domain manager is the server of the system. It communicates via XML messages conforming to “message.dtd”, and it can perform any of its functionality in response to authorized service requests through these messages. For example, a catalog agent may send a ”wf.process.run” message to initiate the related process when a customer purchase order is accepted. The domain manager may also, in

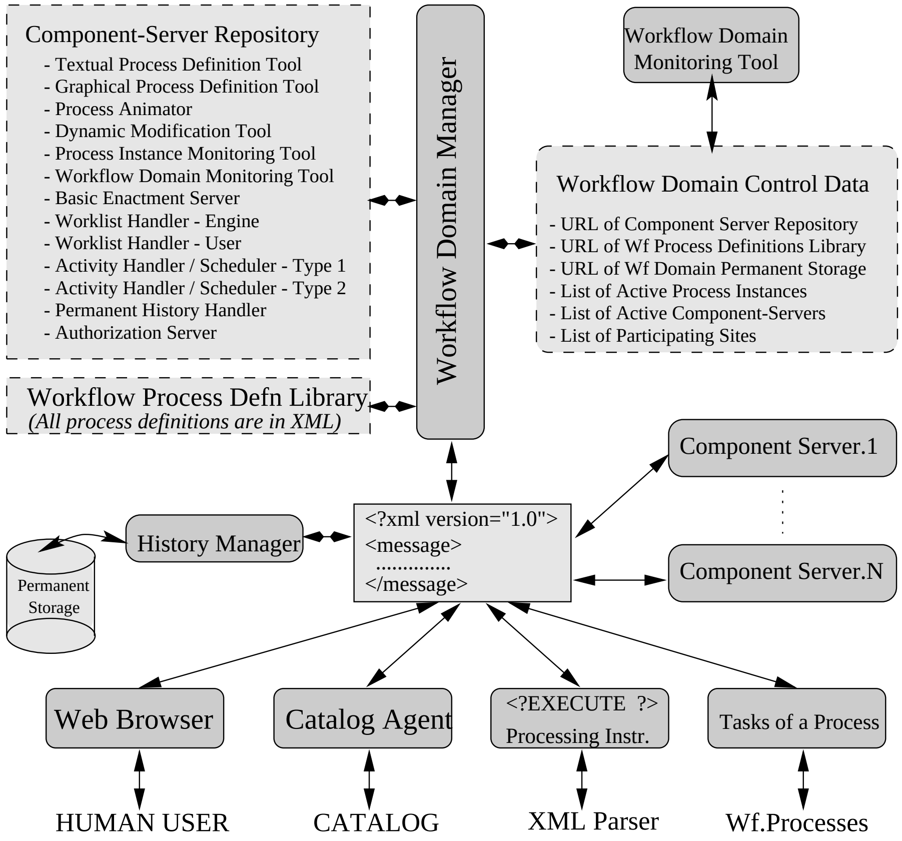

Component-Server Repository
- Textual Process Definition Tool
- Graphical Process Definition Tool
- Process Animator
- Dynamic Modification Tool
- Process Instance Monitoring Tool
- Workflow Domain Monitoring Tool
- Basic Enactment Server
- Worklist Handler - Engine
- Worklist Handler - User
- Activity Handler / Scheduler - Type 1
- Activity Handler / Scheduler - Type 2
- Permanent History Handler
- Authorization Server
Workflow Domain Manager
Workflow Domain Monitoring Tool
Workflow Domain Control Data
- URL of Component Server Repository
- URL of Wf Process Definitions Library
- URL of Wf Domain Permanent Storage
- List of Active Process Instances
- List of Active Component-Servers
- List of Participating Sites
Workflow Process Defn Library
(All process definitions are in XML)
Component Server.1
Component Server.N
History Manager
Permanent Storage
<?xml version="1.0">
<message>
....
</message>
Web Browser
Catalog Agent
<?EXECUTE ?> Processing Instr.
Tasks of a Process
HUMAN USER
CATALOG
XML Parser
Wf.Processes

Figure 5.10: A message-driven, component-based and adaptive workflow system architecture

response to a “wf.query” message, provide the monitoring information to any authorized requester.

Human clients access the domain manager through a Web browser for administrative purposes like defining new processes or monitoring the system. The domain manager downloads appropriate Java applets to the client which then handle subsequent requests of the same client for that particular service which is provided by a component server. The domain manager keeps runtime information such as list of active process instances, active component servers, list of participating sites, etc. for domain monitoring purposes.

2. *Workflow Process Object (WPO):* When an authorized user or component wants to initiate a new instance of a pre-specified workflow process, the Domain Manager creates a new “Workflow Process Object”. The main method of this object is the “Basic Enactment Method” which is activated by the Domain Manager on behalf of the client. The component-based architecture of a WPO and its basic enactment server is conceptually illustrated in Figure 5.11. The WPO contains all the data (such as workflow process definition, workflow relevant data, enactment history of that instance up to the current execution point, etc.) required to complete the execution of the process instance, or to migrate the process instance from one site to another, or to rescue an instance in case of failures. Since a WPO contains its own copy of the workflow process definition and all the run-time information about its own process instance, the dynamic modification of the workflow process definition on instance basis is simply enabled by dynamically modifying the WPO.

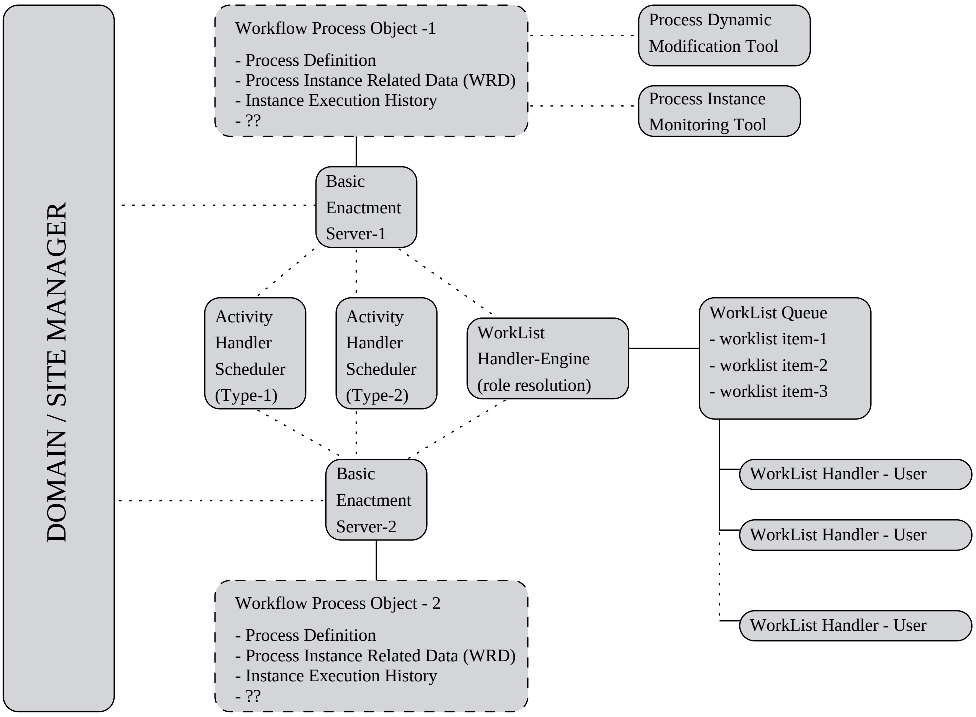

DOMAIN / SITE MANAGER
Workflow Process Object -1
- Process Definition
- Process Instance Related Data (WRD)
- Instance Execution History
- ??
Process Dynamic Modification Tool
Process Instance Monitoring Tool
Basic Enactment Server-1
Activity Handler Scheduler (Type-1)
Activity Handler Scheduler (Type-2)
WorkList Handler-Engine (role resolution)
WorkList Queue
- worklist item-1
- worklist item-2
- worklist item-3
WorkList Handler - User
WorkList Handler - User
WorkList Handler - User
Basic Enactment Server-2
Workflow Process Object - 2
- Process Definition
- Process Instance Related Data (WRD)
- Instance Execution History
- ??

Figure 5.11: A workflow process object and its basic enactment server

The dynamic modification of a WPO may be initiated in two ways: either by a user or by means of a special activity specified in the process definition as explained in the following:

- A user via his Web browser may access the Workflow Domain Manager and download the Dynamic Modification tool (DMt). The user is supposed to provide the identifier of the WPO to be modified. The identifier may also be obtained by the DMt from the List of Active Process Instances contained in the Workflow Domain Control Data via a simple selection process. Once the target WPO is identified, the enactment service of the WPO is paused by sending a proper message to its Basic Enactment Server, and from thereon the DMt takes the control of the WPO. The user modifies the workflow process definition contained in the WPO as s/he wishes. If the modification requires undoing some activities that have been already executed, corresponding compensation activities are included as part of the modification such that when the enactment service resumes, those already executed activities are compensated first. After the user signals explicitly that the modification is completed, DMt passes the control back to the enactment service of the WPO by sending it another message. The user has, at his will, the choice of applying this modification to the other instances of the same workflow process, or to even update the process template stored in the Workflow Process Definitions Library.
- Workflow process definition may contain a special activity, called Workflow Process Modification Activity (WPMA), that when executed automatically invokes the DMt on behalf of a user, so that the user can modify the currently executing WPO. The WPMA is generally used to handle instance-specific run-time differences that can not be determined at the process definition time. Each execution of the WPMA activity results in a separate modification of the WPO. A WPMA

initiated modification may not affect other instances of the same process, nor the template in the Workflow Process Definitions Library.

- Dynamic Workflows which have no pre-specified process definition can be handled with another special activity, called Dynamic Workflow Special Activity (DWSA). The DWSA automatically invokes the DMt on behalf of a user, so that the user can specify the next activity to be executed. A dynamic workflow process definition initially includes only one activity: the DWSA. When this process is initiated, the DWSA invokes the DMt and awaits the user to specify next activity to be executed. When the user specifies the next activity or activities, another DWSA activity is automatically appended such that after the user-specified next activity(s) is executed, the DWSA will be invoked again. The DWSA will not be appended only if the user explicitly states that no more activities are to be entered, in which case the termination of DWSA will indicate the termination of the WPO. In this way a workflow process can interactively be defined on-the-fly by a user, and it is saved in the Workflow Process Definition Library if the user specifies so when terminating DWSA.

WPOs exist only for active process instances. When a process instance terminates, its WPO is destroyed after its history is permanently saved by the History Manager.

3. *Component-Server Repository:* The components of the system are implemented as Java classes and are activated by the domain manager as requested by the executing process instances. The human interaction components like Dynamic Modification tool, on the other hand, are accessed by the authorized users through Java applets.

   The Component-Server Repository includes the following components for human interaction:

   - *Workflow Process Definition Tool:* Authorized users are allowed to define new workflow processes or to delete previously defined process templates through a graphical user interface tool. The process definitions are syntactically verified and permanently stored in the Workflow Process Definition Library in XML conforming to the "workflow.dtd" given in Appendix **??**.
   - *Dynamic Modification Tool:* Authorized users are allowed to modify a particular workflow process instance at run time to respond to external changes that cause variations in the pre-specified process definition. In such a case, the modifications can be applied to executing instances selectively or to all instances of the same workflow process if required. The modifications can also be reflected to the template definitions in the Workflow Process Definition Library if needed.
   - *Process Instance Monitoring Tool:* Users are allowed to trace workflow process instances they have initiated, and extract run-time information about the current execution status of an instance. Collecting and measuring process enactment data are needed to improve subsequent process enactment as well as documenting what process actions actually occurred in what order. This feature provides data for evaluation and optimization of process definitions. An authorized user can monitor any active process instance.
   - *Process Animator:* Graphically simulating the re-enactment of a process is needed to more readily observe process state transitions or to intuitively detect possible process enactment anomalies. Process visualization provides users with graphical views of process templates or instances that can be viewed, navigationally traversed, and interactively edited and animated to convey process statistics and dynamics. Process visualizations enable intuitive analysis and discovery as well

as being a key to user acceptability of the technology. Visualizing and replaying process enactment histories support organizational process drill-downs when execution anomalies are observed.

4. *Workflow Process Definitions Library:* Workflow definitions (i.e., the process templates), organizational role definitions, and participant-role assignments are durably stored into this library. This library supports versioning, and it is maintained by the WFMS library manager. Only workflow process definition tool and dynamic modification tool may insert, delete or update workflow process templates in this library. Accesses by all the other services are read-only, except that, a new process definition may also remotely be inserted into the library through an XML message sent to the domain manager provided that the message contains a valid process definition.

5. History Manager: The History Manager handles the database that stores the information about workflow process instances which have been enacted to completion to provide history related information to its clients (e.g. for data mining purposes). It should be noted that the history of active process instances are stored in the WPO itself.

5.3.2.2 Advantages of the Component-Based Architecture

Advantages of the adaptive, message-driven and component-based architecture can be summarized as follows:

- Each process instance is a Java object that contains all the necessary data and control information as well as its execution history. This feature makes it possible to migrate the process object in the network to provide load balancing. Furthermore, it is possible to dynamically modify the definition of a workflow process on an instance basis at run time.

- Despite having a central control on a process instance basis, the run-time system brings out all the benefits of highly distributed environments. Each instance of a workflow process may execute at a different site.

- The system is highly scalable since the components are activated only when they are actually needed. This componentwise architecture makes it possible to incorporate the functionality and thus the complexity only when it is actually needed at run time. A process instance downloads only the necessary components which results in effective usage of system and network resources. It is also possible to add new components or maintain and upgrade the existing components of the system incrementally without affecting the other parts of the system.

- Components communicate with each other via XML messages over TCP/IP connections. All data exchanges too are realized through XML, providing uniformity, simplicity and a highly open and interoperable architecture.

- The component-based architecture facilitates the replication to a great extend. The Component-Server Repository, Workflow Definition Library, Workflow Domain Control Data and the Workflow Domain Manager may all be replicated for better performance and availability. Each participating site may have its own replication of Workflow Domain Manager as the Site Manager.

- The system is highly dynamic, since no prior installation of any WFMS software is required on the client side. The human interaction components of the system are coded as network-transportable Java applets so that the end user can acquire workflow components from the Workflow Domain Manager over the network. Thus it is not necessary to have the software pre-installed on the user machine. A human user needs only a browser to access the domain manager. This promotes user mobility further as well as easy maintenance of the system components which can be upgraded transparently on the server side.

- Specification of process definitions in XML isolates definition and implementation aspects of a process. Defining in XML, makes process definitions highly transportable and re-usable. Any organization may develop its own processes, in addition to sharing common business processes developed by others without any difficulty. This brings re-usability to the process definitions and provides high productivity.

5.4 Commercial Workflow Systems

There are many commercially available significant workflow products on the market. These products are based on a variety of approaches, and range from simple to very sophisticated. [ODP] maintains a non-exhaustive list of commercially available products.

In this section, we briefly describe a few of the commercial workflow products, namely, Staffware, Ultimus, *ic*X*pert*FLOW and IBM's MQSeries Workflow.

5.4.1 Staffware

[Staffware] is the founder of the independent workflow industry, and also a co-founder of the Workflow Management Coalition. Staffware has become a leading provider of business process automation software which enables organisations to automate and streamline complex business processes. The software has nearly 1.5 million licensed users (in 2002) worldwide in industries including banking, insurance, telecoms, utilities, commerce and government.

Staffware has launched "Staffware 97" in 1997, the first workflow product to provide support for the Workflow Management Coalition Interoperability standards.

Staffware has a portfolio of tools enabling the rapid development of robust workflow applications scaling from departmental to the enterprise level, using both client/server and thin client/Web based architectures. Flexible and easy to use, Staffware automates the execution, co-ordination and control of rules-based procedures with pre-defined objectives no matter how complex the business process. Staffware also provides a full integration tool kit, which allows business processes to be integrated with legacy systems and other applications, such as document management, call centers, and other office applications.

The Graphical Workflow Definer allows business processes to be defined with an easy to use drag-and-drop interface. The Graphical Forms Designer designs and generates forms that the end user faces when the process is run. The Work Queue Manager provides both users and managers with quick on-screen access to outstanding work, permits the control and balancing of workloads, and enables the re-allocation of work items in the absence of participants or an unexpected work overload.

Staffware Global is a unique pure Java based client that provides full participative workflow through a Web browser. It can be used to implement workflow more easily both within and between enterprises allowing organisations to integrate their business partners, customers and suppliers directly in their business processes, streamlining activities and shortening process times dramatically.

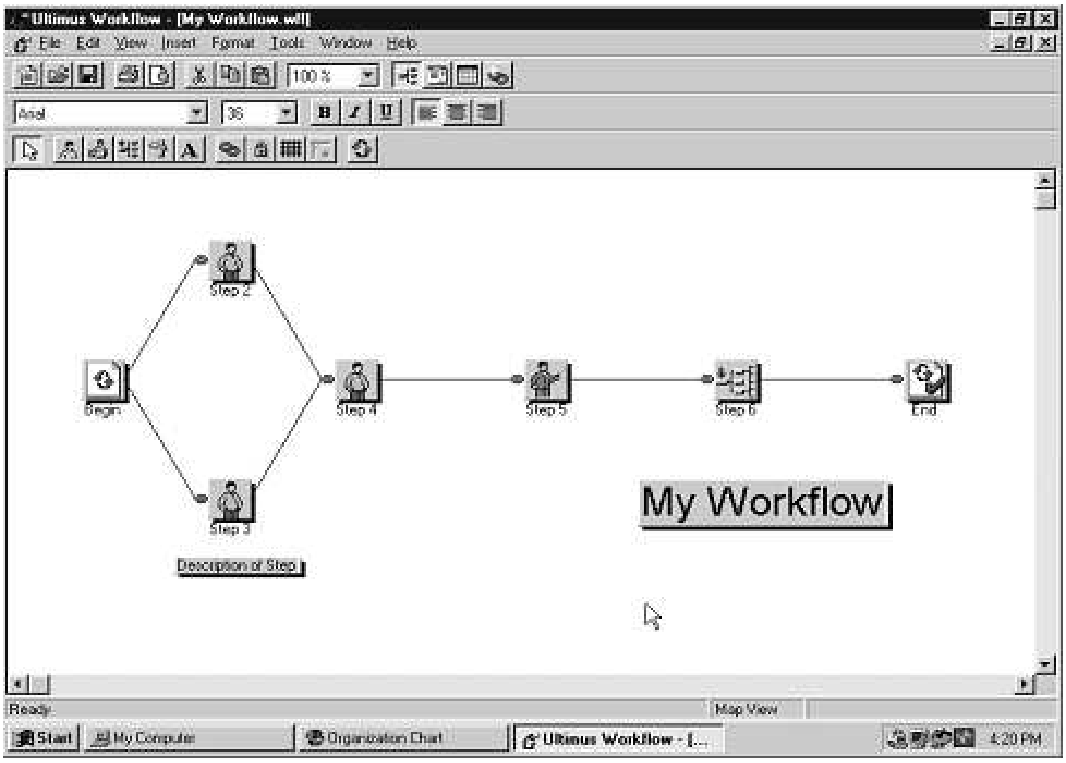


Figure 5.12: Ultimus Workflow Designer

5.4.2 Ultimus

Founded in 1994, [Ultimus] is a pioneer in the development of flexible end-to-end workflow platforms. The Ultimus Workflow System, first released in 1995, is comprised of a set of separate software components, including the Ultimus Workflow Designer.

The Ultimus Workflow Designer, shown in Figure 5.12 is a graphical development environment for defining process flows. Users can build applications by selecting from either preconfigured tasks, or from a custom-built library. Ultimus supports three different type of steps for defining process steps:

- Users steps: These are tasks that are done by people, such as filling out forms.

- Flobot steps: An Ultimus flobot step represents a task that is done by third-party applications. Ultimus provide predefined flobots for integration with Microsoft Word, Microsoft Excel, ODBC-compliant database, and SMTP (Simple Messaging Transfer Protocol) applications.

- Maplet steps: These are tasks performed in other workflow processes. A designer has the ability to break down a complex, multi-departmental process into segments, each of which can be defined by the particular segments. Then when the overall process is defined, these segments can be brought into the map as Maplet steps.

- Linking Steps Together: Steps are tied together with graphical links. The designer draws a line from on step to another in the direction the work is to flow.

- Defining the Properties of the Steps: The details specifying who, what, and when of each step are defined. Key properties include:

- Recipients: In Ultimus tasks can be assigned to workflow participants and job functions, groups, and queues. Business rules can also be defined that indicate who should receive a task based on the specific incident of the workflow.

- Completion and Extension Time: Each step has a completion time. This points that how long it takes to accomplish the task. Based on this timeline, the system automatically determines the status of the step as current, overdue, or urgent.

- Task Rate: Ultimus allows the user to capture the cost of the person doing the work. This cost is calculated by the amount of time the step actually took to complete; when a task is completed a prompt appears asking the user how much time he or she spent on the step. The system keeps track of lapsed time, the reported task time, and the cost. By knowing the cost of each step, the cost of each incident can be calculated. This data can be used to generate process metrics so that the manager can understand the cost of doing business.

- Conditions: Ultimus provides forms to define a company's business logic. The user specifies in an Event Condition Table what should happen when the step is activated, completed, late, rejected, or resubmitted. In addition, users can define specific conditions under which the standard conditions are overridden, such as when the incident is cancelled. The entire conditional rules are defined in the Ultimus Distributed Spreadsheet Model.

The Ultimus Workflow Designer contains an integrated simulation module. This component allows for validation of routing logic, testing of workloads, and verifying the tasks. The designer also contains an organization chart module. It facilitates the graphical representation of organizational structures, allowing process logic to be defined in terms of roles and relationships. This component can also operate a stand-alone application form managing organizational roles and relationships.

5.4.3 *ic*X*pert*FLOW

Established in 1989, Internet Commerce Express Inc. is a leader in business process automation. The company introduced its icXpertSOLUTION Suite in 1999. This integrated e-process management suite combines workflow software with XML and content management technology to expedite business and e-commerce processes. The suite incorporates [icXpertFlow], (previously Keyflow), a workflow/process automation server; icXpertFORMS, an XML-based forms designer; and icXpertVAULT, a transaction content manager.

*ic*X*pert*FLOW is a collaborative workflow authoring solution built on the Microsoft BackOffice -Internet Information Server, Exchange, Windows NT and Windows 2000. Microsoft Exchange is a powerful messaging and collaboration platform. It provides infrastructure needed for a workflow, such as user directories, work transport, security, customizable forms and views, and a database that can be used to store work items and process templates.

The drag-and-drop based graphical workflow designer of the earlier "Keyflow" version of *ic*X*pert*FLOW is shown in Figure 5.13.

*ic*X*pert*FLOW provides the following functionalities:

- Data-Based Rules Logic: Users can determine the direction their flow will take from step to step based on a specific value in application data. Data fields on a form can be automatically populated based on information in the flow and data placed in a form's data field can automatically be mapped to application data in a flow.

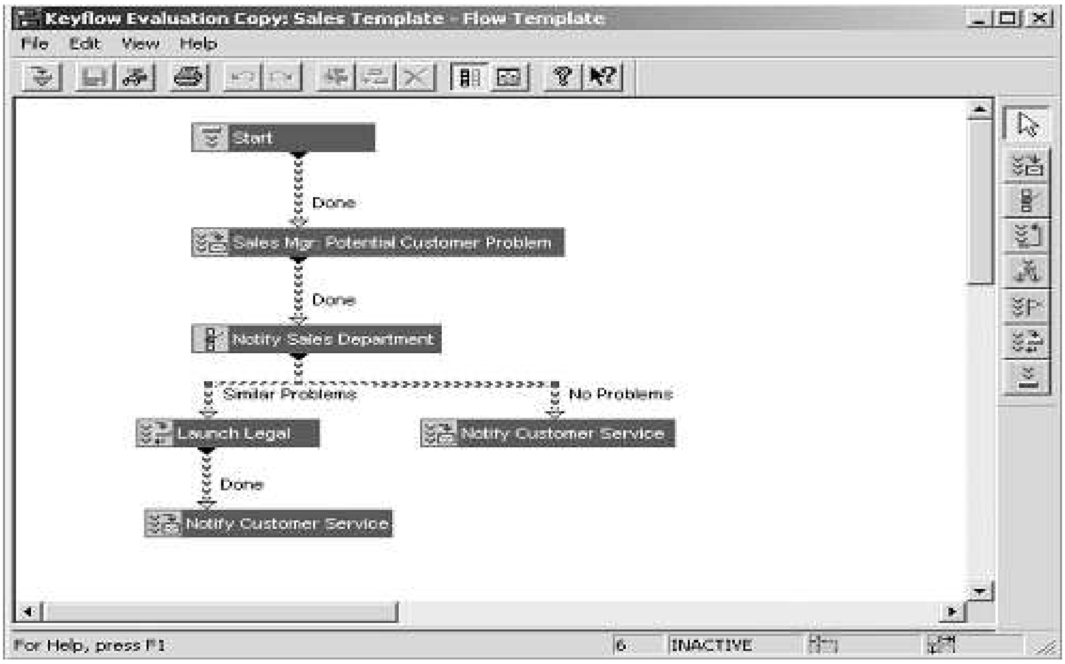


Figure 5.13: Keyflow Workflow Designer

- Business Rules Logic: *icXpert*FLOW's inference engine uses prerequisites to control the execution of steps in a workflow It is possible to define simple and logical prerequisites to control the execution of steps, create sophisticated work processes that run in parallel and specify conditions under which a step's prerequisites are met. Voting prerequisites can also be specified where a step takes place if a predefined number of recipients select a desired response.

- Roles-Based Recipients: It is possible to specify the roles in a workflow such as a specific individual or define a Custom Role (such as "Budget Manager"). It is also possible to specify a role that already exists in Microsoft Exchange.

- Support for attachments: *icXpert*FLOW allows users to forward documents to a list of individuals for review, annotation, revision and approval. It integrates with other document management systems and the Windows NT file system. Messages, links to files or links to Microsoft Outlook messages can be included. It is possible to design and deploy XML "smart forms" that act as the client front-end and which can be accessed using standard Web browsers.

- Web Integration: Users can read and respond to work process messages over the Internet using Microsoft Outlook's Web Access (OWA), URL notification and WebWorkflow with Netscape Navigator and Microsoft Internet Explorer browsers. Participants can see and respond to workflow requests without installing any client software. Users can log into a Web site to see their list of tasks (like a workflow inbox) and the status of a workflow instance.

- Reporting Features: *icXpert*FLOW supports transactions and data logging through ODBC connectivity to Microsoft's Access or SQL Server and Oracle databases. Users can graphically monitor and track the status of active work processes through out the organization, measure the productivity of individuals and/or groups and track the impact of your work process automation application.

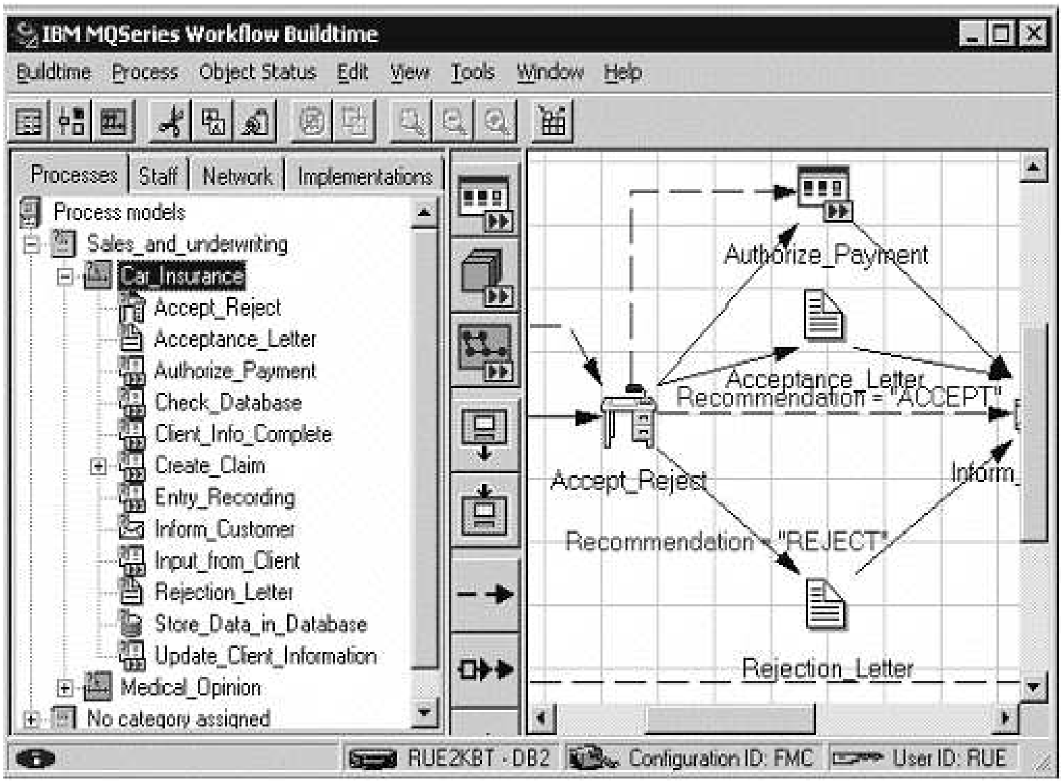


Figure 5.14: The MQSeries Workflow Buildtime

- Flexibility to Change Work Processes: The *icXpert*FLOW Designer allows users to change a running work process. Users can also Consult and Delegate within running work processes, allowing them to forward a copy of a task to one or more people for their opinion before responding to a task.

- Alerts: It is possible to specify when recipients must respond to a task by sending a message (alert) to the Flow Manager and/or recipients when a response is over-due. It is possible to automatically delegate the response to another user after a specified period of time to prevent bottle-necks (in case for example, if a recipient is out of the office).

5.4.4 IBM MQSeries

IBM MQSeries Workflow Version 3.3 [IBM MQ Workflow], formerly Flowmark, is the process engine of IBM's Business Process Management software based on websphere and MQSeries. IBM MQSeries [IBM MQ] enables business applications to exchange information across different operating-system platforms by sending and receiving data as messages. IBM provides an infrastructure of message queuing (MQSeries Messaging) and application integration (MQSeries Integrator) upon which the workflow product is built.

The MQSeries provides application-programming services that enable application programs to communicate with each other using messages and queues. This form of communication is referred to as asynchronous messaging. It provides assured, once-only delivery of messages. MQSeries also provides mechanisms for generating acknowledgements of messages received. The programs that comprise an MQSeries application can be running on different computers, on different operating systems, and at different locations. The applications

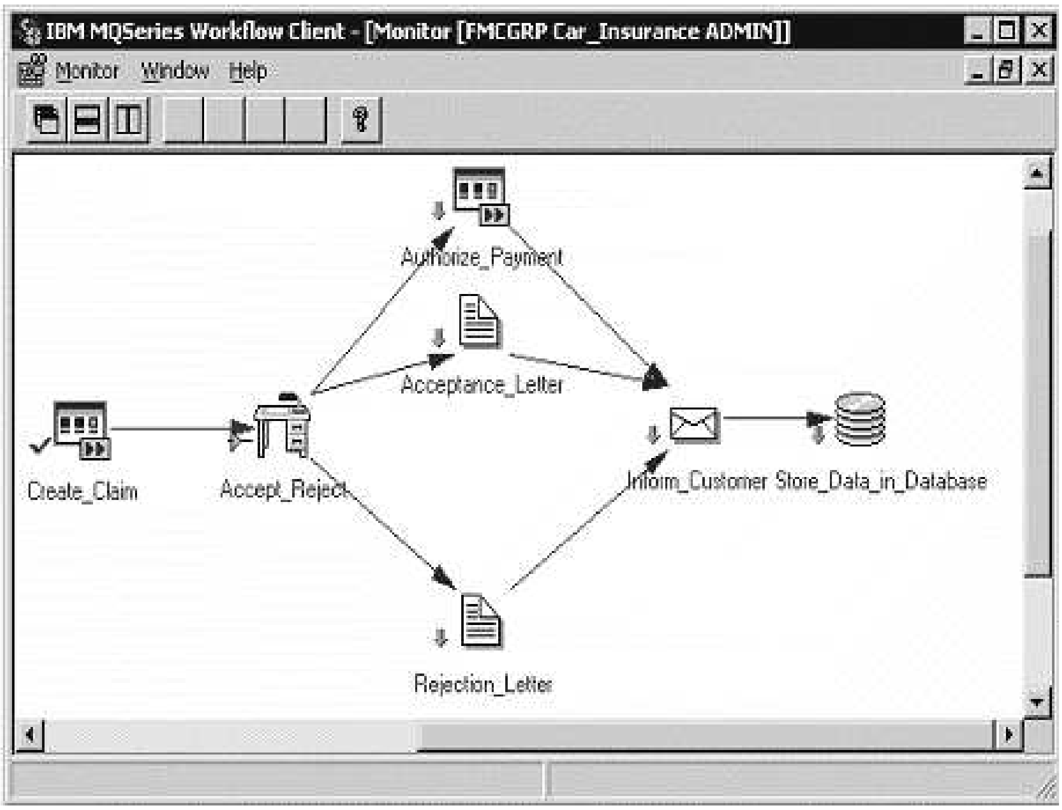


Figure 5.15: The MQSeries Process Monitor

are written using a common programming interface known as the Message Queue Interface (MQI), so that applications developed on one platform can be transferred to another.

The workflow system executes process definitions captured during modeling and ensures that business functions are performed reliably and correctly using the transactional integrity of MQSeries.It executes processes running from subseconds to many months. Its restart and recovery capability makes it fault tolerant.

MQSeries Workflow 3.3 provides a graphical editor to help design the business processes. A process orchestrates interactions with individual services to perform a complex federated business task. Services are defined as activities, or subprocesses, with input and output data, sequential and parallel flows; control and data flows can be different.

The execution of processes is controlled by MQSeries Workflow servers, which maintain process state, determine what's next, log steps performed and interact with users and applications. Servers pass data automatically between resources participating in the process.

Figure 5.14 shows the MQSeries Workflow Buildtime and its basic tools. With the process monitor as shown in Figure 5.15 the status of the activities of the business process can be observed.

5.5 Conclusions

Workflow technology has come a long way from its hype in the mid-1990s to the present. Today, workflows arent just limited to particular processes or subprocesses, but they can span the entire organization. Enabling definition and automation of the business processes, workflow technology offloads a significant amount of overhead managerial work and provides higher productivity. However, human factor issues represent the greatest obstacle to the acceptance of workflow applications in more than 50 percent of the cases, according to [Koulopolous, 1995]. Many employees see the workflow management as a mechanism for removing their decision-making power or as an instrument of downsizing.

Like all the other technologies, the workflow technology too has its highlights and lowlights. It is a perfect tool to systematize and overtly manage formerly hidden processes. It improves the ability of organizations to quickly adapt in response to customer and market changes. Work and worker-attribute data are managed seamlessly in the same system, enabling easy roll-up of data across organization levels from individual employee to organizational-level goals. It allows for automated data tracking of performance, errors, resource utilization, forecasts, and alarm conditions. On the other hand, the current generation of workflow products leaves much to be desired. A number of problems to be worked on are identified by [Aalst, 2001] as follows:

- There is a lack of good standards for workflow management.
- The expressive power of current workflow systems is not sufficient to represent complex processes.
- There is a lack of understanding of how people actually work in a social perspective.
- The current products provide a limited support for process analysis.
- There still are technical problems that remain unsolved, such as the integration of components and long-lived transactions issues.