HP Workflow Research: Past, Present, and Future

Ming-Chien Shan, Jim Davis,
Weimin Du, Ying Huang
Software Technology Laboratory
HPL-97-105
August, 1997

workflow,
business object,
legacy application
integration,
business process
reengineering

Workflow research at HP Labs has evolved through three stages during the past four years. Our effort started with the enhancement of an existing HP workflow product – WorkManager. Then, based on customers' feedback, a new workflow prototype was developed to support the requirements of enterprise complex business process management. This formed the base for HP's second generation workflow product – AdminFlow. We are now exploring the third generation of workflow system that supports the Internet computing paradigm.

To be published in and presented at the *North Atlantic Treaty Organization –Advanced Study Institute Workshop*, Istanbul, Turkey, August 20, 1997.

HP Workflow Research: Past, Present, and Future

Ming-Chien Shan
Jim Davis
Weimin Du
Ying Huang

Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304
{*shan, davis, du, huang*}*@hplabs.hp.com*

Abstract

Workflow research at HP Labs has evolved through three stages during the past four years. Our effort started with the enhancement of an existing HP workflow product – WorkManager. Then, based on customers' feedback, a new workflow prototype was developed to support the requirements of enterprise complex business process management. This formed the base for HPs second generation workflow product – AdminFlow. We are now exploring the third generation of workflow system that supports the Internet computing paradigm.

1 Introduction

The emergence of a truely global economics has immersed all businesses into an intensively competitive environment moving with accelerating rates of changes. Gradual improvements in productivity and enhancements in quality are no longer enough to maintain market leadership. The fast delivery of new products/services and the rapid modification of existing applications are key survival factors. These requirements have forced enterprises to look for new solutions, and workflow has emerged as one of the crucial technologies to meet these needs.

As the research arm for HP, we at HP Labs, have engaged in workflow research since 1993. Our research effort has moved through three stages: advanced feature enhancement for HPs first generation workflow products; the development of a new workflow prototype to support the operation of enterprise level, mission critical business processes and the development of integrated business applications; and currently the investigation of the requirements for a workflow system supporting Internet-based business operations.

2 Stage 1: Advanced features enhancement

In the early 90's, HP rolled out several products to support collaborative computing, including WorkManager[1] and SynerVision. Research activities at HP Labs were focused on

how to consolidate these products into one product, i.e., how to design the model and function mappings between them. We also designed advanced features such as nested process, distributed process, and efficient rule engine to enhance WorkManager.

Like many other products on today's market, these products had their root in office automation and focused on document management capability with minimal support for task routing functions. These first generation workflow systems were intended mainly to automate document or forms flows among human workers in an organization.

However, business process operations at the enterprise level are quite different from these in many crucial respects. First, there is a difference in scale and performance. The enterprise business processes may span many organizations within an enterprise and even across enterprises. Second, these first generation workflow systems typically used electronic mail for delivering tasks to human workers. Little support for automatic operation execution, monitoring workflows, enforcing consistency, or recovering from failure was provided. Third, the actual work supporting a step within an enterprise business process may be performed not only by humans, but also by computer software, or machines (e.g., instruments or robots). Hence, facility for acquiring, coordinating, invoking, and monitoring these resources are necessary.

As a matter of fact, the sales of these first generation workflow products stumbled in 1996 [2], due to the lack of features supporting enterprise-level business operations.

3 Stage 2: Enterprise Business Process Management System

Through comprehensive discussion with HPs world-wide customer partners engaged in business process re-engineering, we collected a solid set of enterprise business process management requirements and started the **OpenPM** project[7, 10] in 1994 to design a new workflow system to meet these demands. The OpenPM prototype was completed in 1996 and forms the base for the HP **AdminFlow** product to be released in 1997. Because of these enterprise business operational features, AdminFlow won the "Best Collaborative Computing" product award at the 97 Electronic Messaging Association annual conference.

3.1 The requirements

We highlight some key requirements for OpenPM prototype below:

- Performance.

  The OpenPM engine needs to achieve a very high throughput. It should be able to provide a minimum dispatching capability of 12,000 activities per hour in steady state.

- Scalability.

  The OpenPM engine should be scalable to support both small installations with few and simple precesses as well as vast installations with many complex processes. A design goal was that on a HP Enterprise Business Server model H50, an OpenPM engine should support up to 1,000,000 active process instances. System capabilities should scale with the computational power available.

- Distribution.

  The OpenPM system should support configurations with multiple engine running on separate machines with remote sub-process execution capability.

- Reliability.

  The OpenPM engine should support 24x7 (8760 - 4) up-time, providing the underlying infrastructure, including database, backup strategy, network technology, and hardware, also supports 24x7 up-time.

- Flexibility.

  The OpenPM system should provide flexibilities in application interaction, database usage, and process enactment (e.g., ad-hoc routing, dynamic resource and priority assignment, transactional recovery and compensation, and mixed synchronous and asynchronous transports).

- Usability.

  The OpenPM system should provide open integration with third party tools for process design, and legacy system/application integration.

3.2 OpenPM overview

The OpenPM prototype was designed as an open, enterprise capable, object-oriented workflow system to manage business activities supporting complex enterprise processes in a distributed heterogeneous computing environment. This middleware service that represents a substantial evolution from first generation workflow technologies.

Given the trend towards open systems and standards, a workflow system must coexist with and take advantage of standards-based commercial products for network communication, legacy application invocation, and system monitoring. In particular, the OMG's ORB, Microsoft DCOM, OSF's DCE, HP OpenView, and OSI X.400 technologies are expected to play an important role in the development of workflow systems. OpenPM provides a generic framework and a complete set of services for business process flow management, utilizing the above-mentioned standard technologies, and emphasizing performance, scalability, availability, and system robustness.

Basically, OpenPM provides:

- an open system adhering to CORBA-based communication infrastructure and providing W*f*MC standard interface,
- high performance due to optimized database access and commitment,
- effective management due to OpenView-based system management environment,
- a total solution for business re-engineering including a complete set of business application development tools.

The major research areas for OpenPM are:

- process flow model and language to support both computerizable and human activities,
- flow analysis and optimization,
- business rules and constraint management,
- failure and exception handling, including compensation activity management and distributed transaction manager coordination,
- resource assignment management to define and enforce the access constraints and organizational (e.g., role resolution) policies governing the assignment of resources to activities,
- process flow monitoring and dynamic change management,
- data consistency between OpenPM engine (internal) database and external information systems,
- high availability support,
- overall architecture of distributed process flow management system, including ORB, DCOM, DCE and E-mail technology deployment, and
- business application development toolkits.

3.2.1 OpenPM system

The overall architecture of OpenPM system is depicted in Figure 1. The core is the OpenPM engine, which supports five interfaces for *business process defining, business process execution, business process monitoring, resource & policy management,* and *business object management.*

A business process is specified via the process definition interface. An instance of the business process can be started, stopped, or intervened via the process execution interface. Status information for each process instance and configuration and load information for the entire system can be queried via the process monitoring interface. The resource and policy management interface is used at run-time to allocate execution resources to a task, according to the policies defined by the organization (including authorization and authentication) and the availability of the resources. The interaction with external world (e.g., the invocation of an application, the control of an instrument, or the delivery of a work order to a person's E-mail in-tray) is performed via the business object management interface.

3.2.2 OpenPM process model

A *business process* describes the sequencing, timing, dependency, data, physical agent allocation, business rule and organization policy enforcement requirements of business activities needed to enact work.

In OpenPM, a business process is represented as a directed graph comprising a set of nodes connected by arcs. There are two kinds of nodes: *work nodes* and *rule nodes*, as well as two kinds of arcs: *forward arcs* and *reset arcs.* A work node has at most one inward arc and one or more outward arcs. A rule node can have any number of inward and outward arcs.

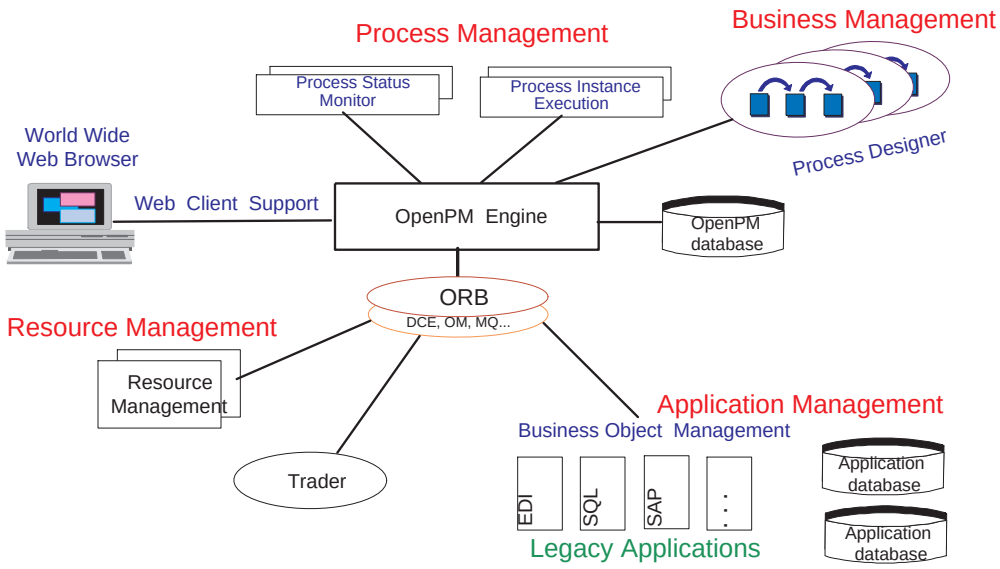

Figure 1: OpenPM system architecture

Work Nodes represent activities to be performed external to the OpenPM engine. These activities include authorization, resource allocation, the execution of business objects, and the provision of input data for and output data from the business objects. Rule Nodes represent processing internal to the OpenPM engine. This processing includes decisions as to which nodes should execute next, the generation or reception of events, and simple data manipulation.

A work node is a place holder for a *process activity* that is a logical representation of a piece of work contributing toward the accomplishment of a process. A process activity is mapped to the invocation of an operation on *business objects* during the execution of the process. In a sense, a *process activity* represents the reusable aspects of a business activity, while *work node* is used to specify additional non-reusable aspects specific to this process.

Each *process activity* may represent a manual operation by a human or a computerizable task to execute legacy applications, access databases, control instrumentation, sense events in the external world, or even effect physical changes.

A process activity definition includes a *forward activity* with optional *compensation activity, cancel activity, resource management activity*, timeout/deadline information, and input/output data.

Rule nodes are used to specify process flows that are more complex than a simple sequence. A rule language is used to program the rule node decision. When executed, a rule node determines which outward arc(s) to fire, based on the status passed along the inward arcs, the time at which each inward arcs are fired, and the process relevant data associated with the process instance.

Rule nodes are also used to support events. A rule node can raise events when conditions defined by the rules are met, and an event can activate rule nodes that have subscribed to the event.

Forward arcs represent the normal execution flow of process activities and form a directed acyclic graph. Successful completion of a node at the source end of a forward arc triggers the starting of the node at the destination end of the forward arc.

Reset arcs are used to support repetitions or explore alternatives in a business process. Reset arcs differ from forward arcs in that they reach backwards in the process graph.

Rule nodes are executed each time any inward arc fires. Work nodes have states of *initial* or *fired*. When the inward arc is fired on a work node in the initial state, the work node changes to fired state and performs its associated activity. When the inward arc is fired on a work node in the fired state, nothing is done.

A reset arc, together with the forward arcs between its destination and source, form loops. When traversed, a reset arc causes all nodes within these loops to be reset. Resetting a fired work node changes its state to *initial state* so that the node can be re-executed. Resetting an active work node cancels the current execution of the corresponding process activity and changes its state to *initial state*.

Associated with each business process, there is a *process data template* to be defined by the business process designer. The process data template is used by users to provide initial data for the creation of process instances. Based on the process data template and *read/write lists* of activities defined in a business process, at run-time, OpenPM will generate a *case packet* for each process instance to facilitate the data passing between activities and OpenPM engine.

An example of the process definition for SONET configuration management process is shown in Figure 3 in section 3.4.

3.2.3 OpenPM process execution

Figure 2 shows a simplified version of the component structure of OpenPM engine, which coordinates the overall execution flow of business processes. It functions as a highly reliable, log-based state machine. OpenPM engine interfaces with external environments through a uniform CORBA-based transport interface, independent of the actual physical dispatch of the requests.

The OpenPM engine launches business process instances in response to user requests. For each instance, OpenPM engine steps through the nodes according to the order specified in its business process definition. For work nodes, OpenPM engine will execute the associated process (forward) activity. For rule nodes, OpenPM engine will evaluate the rules and perform the rule actions when the rule conditions are met.

Each node transition will be durably logged to facilitate forward rolling of incompleted business processes at system restart time in the event of system failure or support activity compensation process in the case of business activity failure. In addition, OpenPM allows flexible specification of compensation scopes and actions [5] (e.g., compensation activity or cancel activity) to support various application needs.

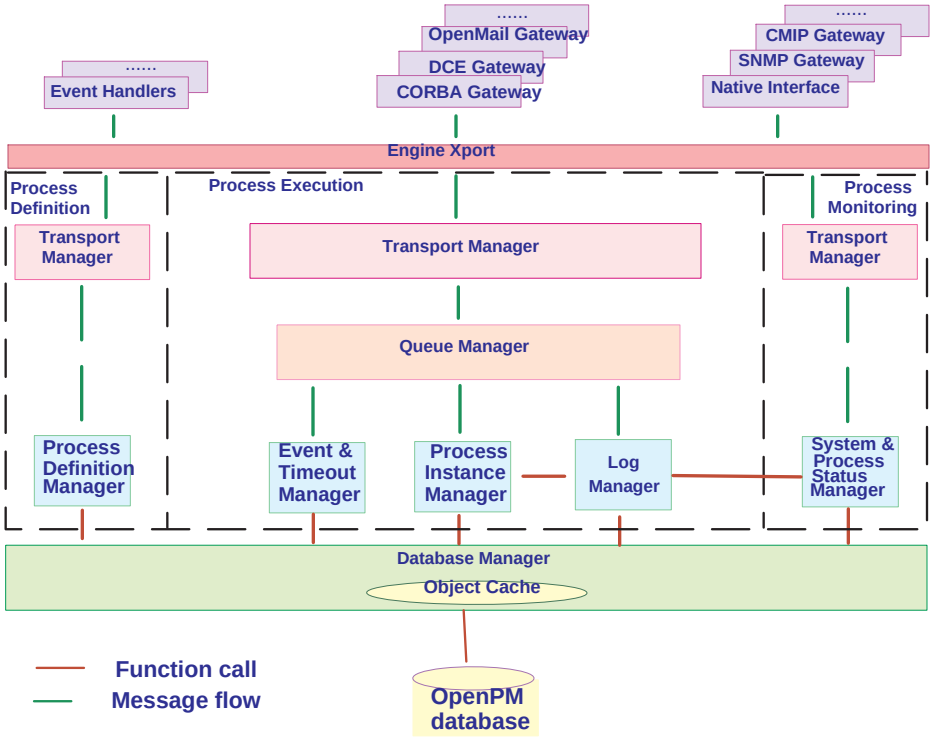

Figure 2: OpenPM component structure

In OpenPM, different versions of similar business processes are supported by the engine under the concept of *process group*. Users can designate a particular version as the default one, which will be used when no specific version is requested at business process instance creation time.

3.2.4 OpenPM business object

The OpenPM engine interacts with business activities supported by various kinds of implementations encountered in real life. These range from manual handling by human to automated application execution by computer. An infrastructure is needed to enable the effective management and invocation of these business activities.

Distributed object technologies have become the primary infrastructure for enterprise-scale distributed computing. Among them, the OMG CORBA technology supports interoperability for application integration.

Based on CORBA technology, an OpenPM abstraction, called *business object*, will be built to encapsulate whatever the piece of work each process activity has to accomplish. The wrapping code will provide an IDL interface and the business objects are cataloged in OpenPM business object library.

Business object, as defined by OMG BOMSIG, is a representation of a thing active in the business domain, including its business name and definition, attributes, behavior, and constraints. It provides an uniform way to encapsulate legacy systems and applications, and

a direct mapping, in understandable business terms, between the business model and the possibly sophisticated operational procedures of the business process system.

By representing these process activities in business objects, new business processes can be quickly created by assembling business objects to describe business processes. The business object library will also avoid the repetitive coding to tailor the business activity implementation into each individual business process.

3.2.5 OpenPM resource & policy management

A **resource** is a person, computer process, or machine that can be utilized to accomplish a task. A resource has a name and various attributes defining its characteristics (such as job code, skill set, organization unit, and availability). A **role** is a definition of a set of resources sharing some common characteristics. The introduction of roles provides a level of abstraction from the physical resource that performs the task.

A **policy** is a set of rules that determines how resources are related to tasks within a workflow system. One common use is for task assignment. Policies can be used to specify which resource or role is eligible or available to perform a task. Policies are also used to ensure proper authorization and authentication.

In OpenPM, the mapping between the business activity (i.e., task) specified in a business process and the business object (i.e., resource) to be invoked is resolved by the *Resource Manager* during run-time as part of the execution of the business activity. The Resource Manager also assumes the responsibility of life cycle management (i.e., creation, modification, and deletion) for resources.

OpenPM allows multiple resource managers to be used to resolve a single resource assignment request; each resolves the request at a different level within an organization[4].

At this point, we can characterize the overall OpenPM conceptual model of business process by four independent aspects. The OpenPM model:

- **functionally** defines what a process intends to achieve (via process definition specification),
- **behaviorially** describes how a process activity is conducted (via business object specification),
- **organizationally** determines who performs an activity (via role and policy specification), and
- **informationally** indicates which piece of data is consumed or produced by an activity (via process data template specification).

The separation of these four aspects for a business process maximizes the reusability and extensibility of business process applications since it allows changing one aspect of a business process application without having to change the others.

3.2.6 OpenPM worklist handler & application data handler

Two optional components, i.e., the *Worklist Handler* and the *Application Data Handler*, can be used in the OpenPM environment to facilitate the execution of business processes. Both components are designed to enhance the scalability of OpenPM systems.

The Worklist Handler supports both *engine-push* and *performer-pull* modes to provide more freedom of task assignment. The OpenPM uses *engine-push* to dispatch activities eligible to be performed. It supports high level of efficiency in the engine.

On the other hand, users often want to select from available work in a *performer-pull* mode. It supports:

- process activity performers to control the order in which they work on activities. They are able to see and select among the activities assigned to them,
- better response times and reduces the load on the Engine. Process activity performers interact with a Worklist Handler that may be in the same network vicinity as the process activity performer, and do not need to communicate with the Engine.
- the activity assignments to a group of performers that each carry out the same role. On completion of an activity, the performer claims the next outstanding activity from the common activity pool. This is useful for environments such as travel desks where it does not matter which travel consultant carries out an activity; the group simply shares the same activity pool (they are thus one resource from the Resource Manager's point of view).

In addition, the Worklist Handler supports the concept of *integration on demand*. Based on the task performer's profile, the Worklist Handler determines and launches a specific environment to an activity at run-time, rather than hard-wiring it into the process definitions.

The Application Data Handler supports the separation of *application specific data* and *process relevant data* to reduce the amount of data flow over the network. It also provides the preparation facility on application specific data to remove the burden on database access from activity performers.

3.2.7 OpenPM system and process management

OpenPM uses the Manager/Agent model [9] for system and process management. It provides the following functions:

- System Configuration.

  This function supports the engine configuration setting and engine alert message display,
- Operations Management.

  This function supports the engine startup/shotdown process and component registration,

- Performance Management.

  This function supports the display of the current value of engine heath parameters,

- Process Management.

  This function supports the status information for each individual business process instance.

To support these management functions, OpenPM engine maintains a comprehensive log of all events and provides a native interface as well as SNMP/CMIP gateways to facilitate integration with OpenView environments. Various formats and contents of the logged information can be customized to support the specific application needs.

3.2.8 OpenPM security

In today's business environments, security must be implemented enterprise-wide. OpenPM supports the following security features:

- authorization.

  The engine determines from the role information associated in the process definition whether a process activity performer or a system component carries out a particular operation. When a request is made to the engine, the Resource Manager checks the credentials of the requester against the security list. If they are not acceptable the request is rejected.

- authentication and encryption.

  The security service developed by OMG/OMA will provide for authentication and encryption. OpenPM plans to use this security service to prevent eavesdropping and forgery. The OpenPM infrastructure components will be able to identify each other and vouch for the credentials of end-user components.

3.3 OpenPM application development facilities

In the "Industry Trends Scenario: Rethinking the IT Investment Paradigm" paper, dated March 28, 1997, Gartner Group defines strategic enterprise mutation as "the ability to rapidly modify enterprise business processes to meet changing market conditions".

To address this challenge, OpenPM separates the business relevant constructs of business processes, resource assignment, and task activities into independent layers. This allows corporations to develop strategic enterprise mutative solutions that can be modified on each layer independently so that business teams can focus on business processes, resources, and activities while IT teams can focus on the application/data management technologies.

Furthermore, we support the concept of *software factory*, which consists of a set of factories, tools, and methodologies collectively to support the "*Just in time*" business process application development lifecycle.

We developed the following factories, which are used to instantiate the different components as the building blocks for a business process application.

- business process factory.

  It consists of libraries of templates for various processes in different business domains.

- business object factory.

  It consists of libraries of skeletons for business objects to represent a resource, a process, an organization, or a policy. It also provides a way to make use of past investments in database systems, applications and computing platforms.

- common facility factory.

  It consists of libraries of routines providing interface glue to different communication infrastructures.

3.4 An application example

This section will describe the application of OpenPM system in a specific domain, that of SONET Configuration Management. This application was demonstrated at Telecom'95 Expo in Geneva.

The scenario demonstrated consists of the provisioning a new VC4/VC12 paths for customer which includess several different steps: search for a new route; negotiation of the Service Level Agreement (SLA) with the customer; configuration of this new path; and finally update of the SLA for this customer.

Searching for and configurating a new path in SONET are complex processes requiring a lot of interaction with the SONET MIB and network elements. This type of operation generates errors when performed manually by an operator as a set of individual uncorrelated activities.

In the demonstration, such complex operations are handled as business processes and automated by an OpenPM engine in an environment interacting with OpenView DM and Oracle DBMS applications.

Depending upon changing business needs, a customer may add or drop communication paths between certain end-points in his Private Virtual Network (PVN). In OpenPM, these services can be modeled as business processes to be executed by the service provider. Adding a new path may consist of the following activities and decision points:

1. Retrieve the customer's profile from the *customer* database for customer PVN specific information.

2. Locate the closest Add-Drop Multiplexes (ADMs) to the end-points, based on the information stored in the *SONET physical configuration* database.

3. Check whether fiber connections exist between the endpoints and the two end-ADMs.

4. If not, issue a request for an engineer to go onsite and physically connect the endpoints to the end-ADMs. After the establishment of the connection, the process will continue with step 5 and an independent subprocess will be initiated on the side to watch for resource changes.

5. Find valid routes between end-ADMs.

   This requires accessing to the *routing* table in SLA database to determine whether any valid routes exist between the two end-ADMs.

   Either a list of ADMs is returned identifying the ADMs that must be configured in order to realize the route, or "No Route Found" is returned. For a returned list of ADMs, this activity will then use the OpenView DM *Facility agent* to collect the port information stored in MIB and to determine the available ports between the ADMs that are fibered together and can be used to enable the path.

6. Check Network Element (NE) capabilities.

   For a ADM in the route, this activity uses the OpenView/DM *NE* agent to access the MIB information to determine whether a VC4 cross-connection can be set up in the ADM between the selected ports of the ADM.

   This activity has to be executed for each ADM in the route.

   Note that, during steps 5 and 6, if any additional resources become available, OpenPM will cancel any currently running activity and start the process over from step 5 in order to consider these newly available resources.

7. Get customer's approval of the selected configuration.

   Once a suitable path is identified, the customer will review the offer, including available date, charges, QoS, etc. Depending upon the business factors (e.g., cheapest service wanted), the customer may request a new search to be initiated, i.e., to loop back to step 5 to find another valid route.

8. Configure the selected route.

   This activity is responsible for setting up the cross-connections in each ADM via the invocation of the OpenView/DM NE agent, and updating the SLA database.

The OpenPM process definition supporting the above-mentioned SONET data path provisioning service is sketched in Figure 3.

4 Stage 3: Internet-based service process management system

Today, the world is undergoing rapid changes to move into the Internet age. The explosion in Internet business activities will raise the demand on business process management technologies to a new level.

Workflow systems, including OpenPM, wwere designed before the advent of the booming usage of Internet and were therefore based on the traditional client/server architecture. Such systems will not serve effectively for future even more dynamic and pervasive Internet-based business operations over various heterogeneous computing platforms. Examples of such operations include world-wide supply chain management, universal telecom service management, global banking service management, and mobile patience care service management.

These applications demand even higher degrees of scalability, flexibility, distribution, robustness, and heterogeneity support on the workflow system. The architecture of OpenPM,

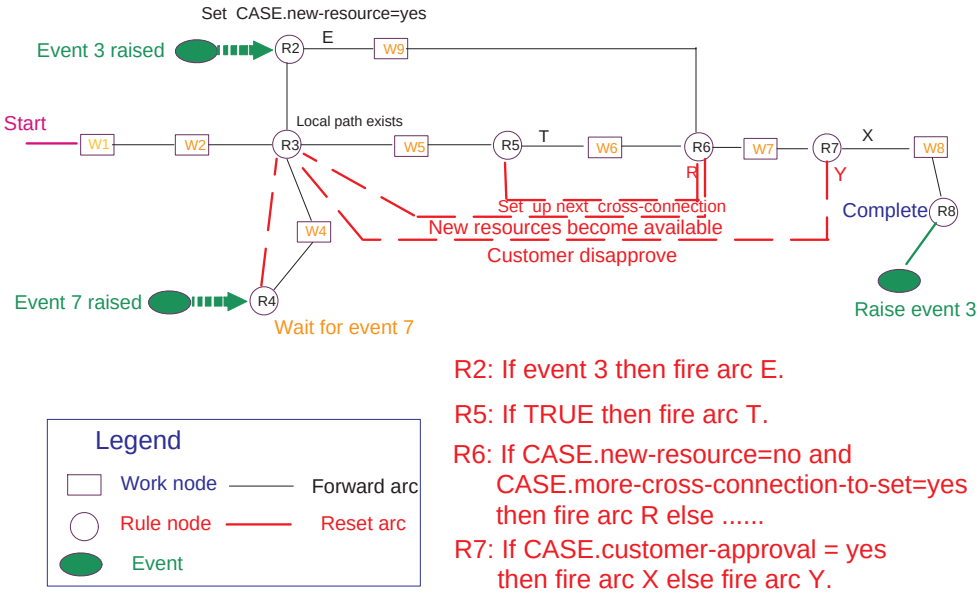

Figure 3: SONET path provisioning OpenPM process

which separates the function, behavior, organization, and information aspects into independent components, provides a sound foundation towards this objective. However, there are new dimensions that need to be addressed.

True Internet applications are atomized - that is, composed of many discrete parts, any combination of which may be collected at run time, called upon to perform certain tasks, and then thrown away until needed again. Under this paradigm, the data – not necessarily the applications – are persistent.

In other words, the Internet computing paradigm requires the support of *dynamic software*. It may require no installation other than the Web-browser frontend, and a library containing a rich mixture of components downloadable from the Internet. All this late-binding and dynamic component assembly implies that the boundaries of future applications will change.

Therefore, we started a new project, called **FlowJet**, to explore the issues around using the **feature on demand** approach to support an open and flexible workflow system for the dynamic and mobile Internet business processes.

We plan to deploy FlowJet over all major computing platforms, ranging from mainframes, server workstations, single user desktop, and to mobile devices to information appliances.

4.1 FlowJet overview

FlowJet is a modularized workflow management system targeted for both Internet/Intranet and traditional business process applications. FlowJet can be used for the following purposes:

- as an enterprise business process management system.

  In this mode, FlowJet runs as a main engine for the management of complex enterprise-wide business processes. This operational mode represents the full-fledged FlowJet server, which is usually executed at data center within an organization. The processes supported at this level are usually quite complex and are divided into subprocesses that will be executed either locally or remotely by other FlowJet full-fledged engine, personal business process manager, or web service manager.

- as a task agent of an FlowJet user.

  In this mode, FlowJet runs as a business object or Worklist Handler to other full-fledged FlowJet servers. The idea is for a particular FlowJet resource to perform tasks that are more complicated than single step activities.

- as a personal business process manager.

  In this mode, FlowJet runs as an independent workflow engine executing personal business processes. Users can define processes that make use of existing services available to them. The service can be a single local application encapsulated as a FlowJet business object, or a complex process to be executed at other full-fledged FlowJet servers.

- as a web service manager.

  In this mode, FlowJet runs as part of a web server. It allows users to model their web activities as workflow processes. This can be viewed as a natural evolution of HTML and JavaScript. With FlowJet, users will be able to model and design their web pages as FlowJet processes. This is more powerful and flexible than HTML and JavaScript, and is also easier to use and maintain.

5 References

1. HP WorkManager Programming Reference, Part No. B2999-90040, April 1994.

2. Workflow - 1996 in Perspective. *Delphi Insight Series Report.* Delphi Consulting Group, Boston, MA, 1997.

3. W. Du, M. Shan, and A. Elmagarmid. "Consistent Execution of Workflow Processes". *HPL Technical Report*, HPL-97-71, May 1997.

4. W. Du, G. Eddy, and M. Shan. "Distributed Resource Management in Workflow Environments", *Proc. of 5th Conf. on Database Systems Advanced Applications*, Melbourne, Australia, April, 1997.

5. Du, W., Davis, J., Shan, M. Flexible Specification of Workflow Compensation Scopes, *Proc. of Int. Conf. on Groupware*, Pheonix, Arizona, 1997.

6. M. Shan, J. Davis, W. Du, and Q. Chen. "Business Process Flow Management and its Application in TMN", *HP Journal*, October, 1996.

7. M. Shan. "OpenPM: An Enterprise Business Process Flow Management System", *ACM SIGMOD Int. Conf. Management of Data, Industrial Session*, Montreal, Canada, CA, June 1996.

8. W. Du, C. Whitney, and M. Shan. "SONET Configuration Management with OpenPM.", *Proc. of 12th Int. Conf. on Data Engineering*, New Orleans, Louisiana, February, 1996.

9. N. Muller. "Focus On OpenView". *CBM Books*, 1995.

10. J. Davis, W. Du, and M. Shan. "OpenPM: An Enterprise Process Management System." *IEEE Computer Bulletin*, June, 1995.

11. W. Du, S. Peterson, and M. Shan. "Enterprise Workflow Architecture." *Proc. of 11th Int. Conf. on Data Engineering*, Taipei, Taiwan, March, 1995.

12. U. Dayal and M. Shan. "Issues in Operation Flow Management for Long-Running Activities." *Data Engineering Bulletin*, Vol. 16, No. 2, June 1993.

13. U. Dayal, H. Garcia-Molina, M. Hsu, B. Kao, and M. Shan. "Third Generation TP Monitors: A Database Challenge." *Proceedings of ACM SIGMOD Int'l Conf. Management of Data*, Washington, DC, May 1993.