Business Process Flow Management and its Application in the Telecommunications Management Network

HP OpenPM is an open, enterprise-capable, object-oriented business process flow management system that manages business activities supporting complex enterprise processes in a distributed heterogeneous computing environment. It is a middleware service that represents a substantial evolution from traditional workflow technologies.

by Ming-Chien Shan, James W. Davis, Weimin Du, and Qiming Chen

Business process reengineering is emerging as one of the crucial business strategies of the 1990s. Business process reengineering is the fundamental rethinking and reimplementation of business processes to achieve never-before-possible levels of quality, cost, throughput, and service. This is especially significant in an era of workforce downsizing and greater demands for shortened time to market and faster customer response. The need for business process reengineering is pervasive. Organizations are currently engaging in business process reengineering in many domains, including financial services, telecom services, healthcare services, customer order fulfillment, manufacturing procedure automation, and electronic commerce.

While business process reengineering provides a business management concept, business process flow management (BPFM) software—or more accurately, *middleware*—provides the enabling technologies for business process reengineering to support flexible solutions for the management of enterprise-wide operations, including:

- Process flow control, automation, and monitoring
- Resource allocation, authorization, and authentication
- Task initialization and data exchange
- End-to-end communication and security.

BPFM is more than just a technology. It offers an overall environment and approach to unifying, automating, and measuring business processes. In addition, BPFM is not a technology supporting only business process reengineering. It can be used to manage existing nonautomated legacy processes—what is often called "paving the cow paths."

Business Process Flow Management System

At the enterprise level, the process to be managed can be very complex, spanning several organizations with multiple steps being performed in parallel. In such cases, a BPFM system can act as the superstructure that ties together disparate systems whose business purposes are interconnected.

A BPFM system provides procedural automation of a business process by managing the sequence of process activities and the invocation of appropriate human, instrument, or computer resources associated with various activity steps. It involves the high-level specification of flows, and provides the operational glue and environment support for managing and automating the flows, recovering from failures, and enforcing consistency. A BPFM system also enforces various administrative policies associated with resources and work.

The structure and flow of a business process managed by a BPFM system can be preplanned or ad hoc. In the case of a BPFM system managing the process of providing telecommunications service, the flow of the process is ad hoc and depends on the services required by a customer. However, certain aspects of the process will be preplanned and deliberately structured. For instance, regardless of the individual services required by a customer, the process always originates in the sales department and is always ends in the billing department.

Typically, a BPFM system:

- Provides a method for defining and managing the flow of a business process.
- Supports the definition of resources and their attributes.
- Assigns resources to work.
- Determines which next steps will be executed within a business process and when they will be executed.

- Ensures that the business process flow continues until proper termination.
- Notifies resources about pending work.
- Enforces administrative policies such as access control.
- Tracks execution and supports user inquiry of status.
- Provides history information in the form of an audit trail for completed business processes.
- Collects statistical data for process and resource bottleneck analysis, flow optimization, and automatic workload balancing.

HP OpenPM

HP OpenPM is an open, enterprise-capable, object-oriented BPFM system developed at HP Laboratories to manage business activities supporting complex enterprise processes in a distributed heterogeneous computing environment. It is a middleware service that represents a substantial evolution from traditional workflow technologies.

Given the trend towards open systems and standards, a BPFM 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 CORBA (the Object Management Group's Common Object Request Broker Architecture), the OSF's DCE (the Open Software Foundation's Distributed Computing Environment), HP OpenView, and ISO OSI (International Standards Organization Open Systems Interconnection) X.400 technologies are expected to play an important role in the development of BPFM systems. HP OpenPM provides a generic framework and a complete set of services for business process flow management using the above-mentioned standard technologies, with emphasis on performance, availability, scalability, and system robustness.

Basically, HP OpenPM provides:

- An open system adhering to the CORBA communications infrastructure and providing a WfMC (Workflow Management Coalition) standard interface.
- High performance as a result of optimized database access and commitment.
- Effective management with an HP OpenView-based system management environment.
- A comprehensive solution for business reengineering including an extensive set of products.

The overall architecture of an HP OpenPM system is depicted in Fig. 1. The core is the HP OpenPM engine, which supports five interfaces for business process definition, business process execution, business process monitoring, resource and 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 controlled via the process execution interface. Status information of each process instance and load information of the entire system can be queried via the process monitoring interface. The resource and policy management interface is used to allocate, at run time, execution resources to a task, according to the policies defined by the organization (including authorization and authentication) and the availability of the resources. Interaction with the 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 intray) is the task of the business object management interface.

HP OpenPM Process Model

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

An HP OpenPM process is a directed graph consisting of a set of nodes connected by arcs. Fig. 2 shows an example of the user interface. There are two kinds of nodes—*work nodes* and *rule nodes*—and 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.

Work nodes represent activities to be performed external to the HP OpenPM engine. These activities include authorization, resource allocation, the execution of business objects, and the provision of input data for the business objects and output data from them. Rule nodes represent processing internal to the HP OpenPM engine. This processing includes decisions of what 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*, which is a logical representation of a piece of work contributing towards 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. Each process activity can 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* and optionally, a *compensation activity*, a *cancel activity*, a *resource management activity*, timeout and deadline information, and input and 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 arcs to fire, based on the status

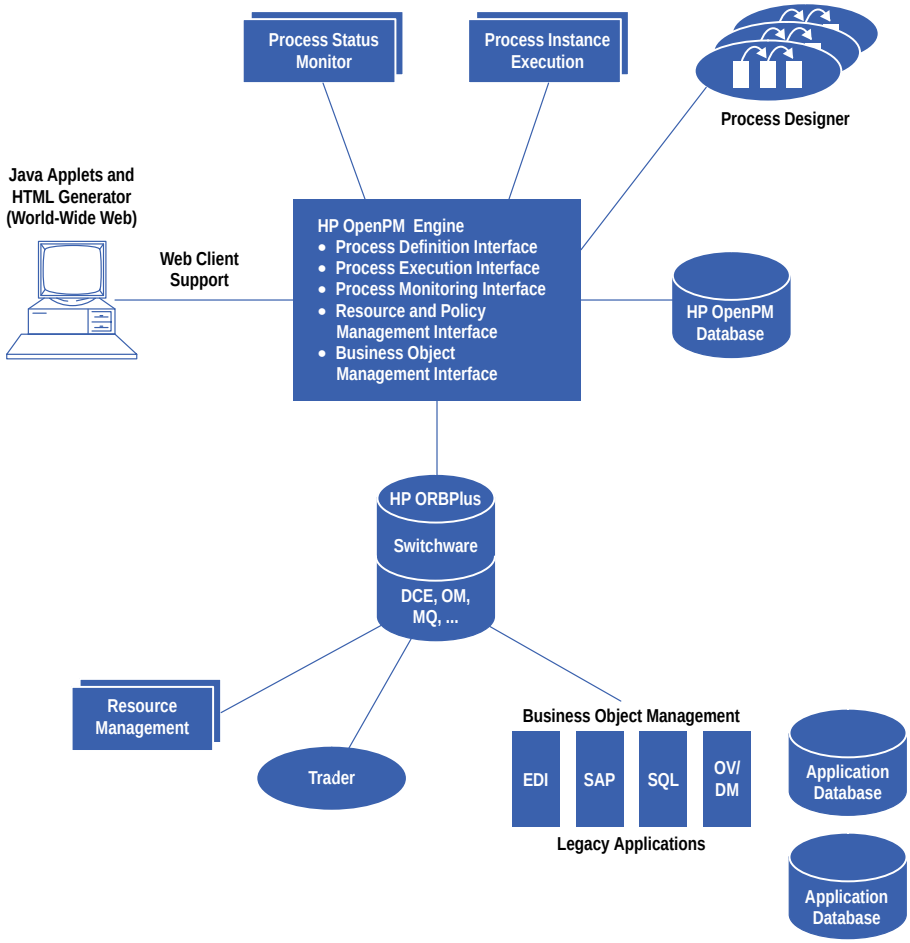

Fig. 1. *Architecture of the HP OpenPM business process flow management middleware.*

passed along the inward arcs, the time at which each inward arc is 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 certain conditions are met as defined by the rules, and an event can activate rule nodes that have subscribed to receive 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 its state to fired 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, forms a loop. When traversed, a reset arc causes all nodes within its loop to be reset. Resetting a fired work node changes its state to initial so that the node can be reexecuted. Resetting an active work node cancels the current execution of the corresponding process activity and change its state to initial.

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

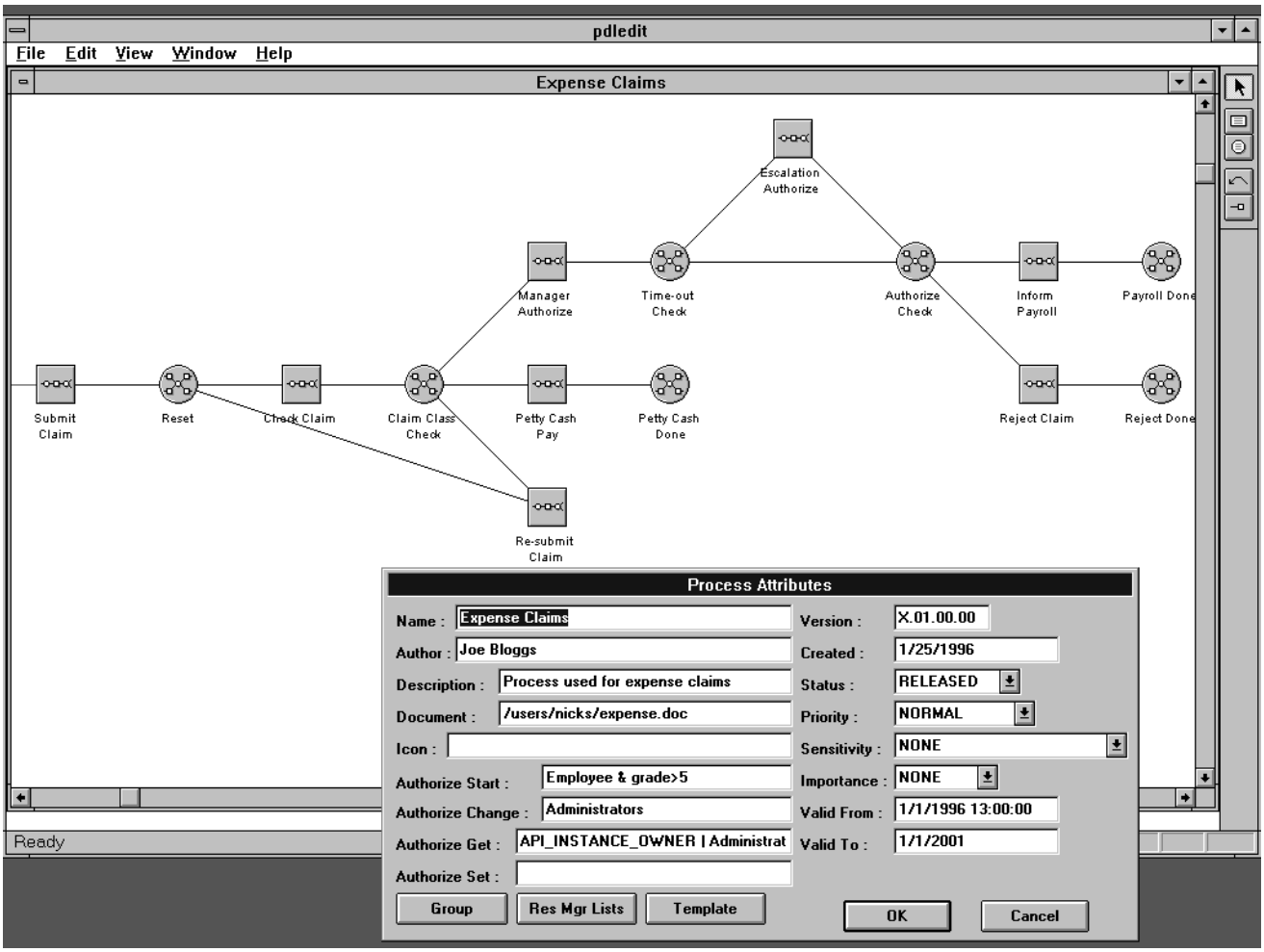

Fig. 2. *An example of the HP OpenPM user interface. An HP OpenPM process is a directed graph. There are two kinds of nodes: work nodes (square) and rule nodes (round).*

HP OpenPM Process Execution

Fig. 3 shows a simplified version of the component structure of the HP OpenPM engine, which coordinates the overall execution flow of business processes. It functions as a highly reliable, log-based state machine. The HP OpenPM engine interfaces with external environments through a uniform CORBA-based transport interface, independent of the actual physical dispatch of the requests.

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

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

In HP OpenPM, different versions of similar business processes are supported by the engine under the concept of a *process group*. The user can designate a particular version as the default to be used when no specific version is requested at the time a business process instance is created.

To monitor the progress of running business activities and support system management, the HP OpenPM engine maintains a comprehensive log of all events and provides a native interface as well as SNMP/CMIP gateways to facilitate integration with the HP OpenView environment. The formats and contents of the logged information can be customized to support specific application needs.

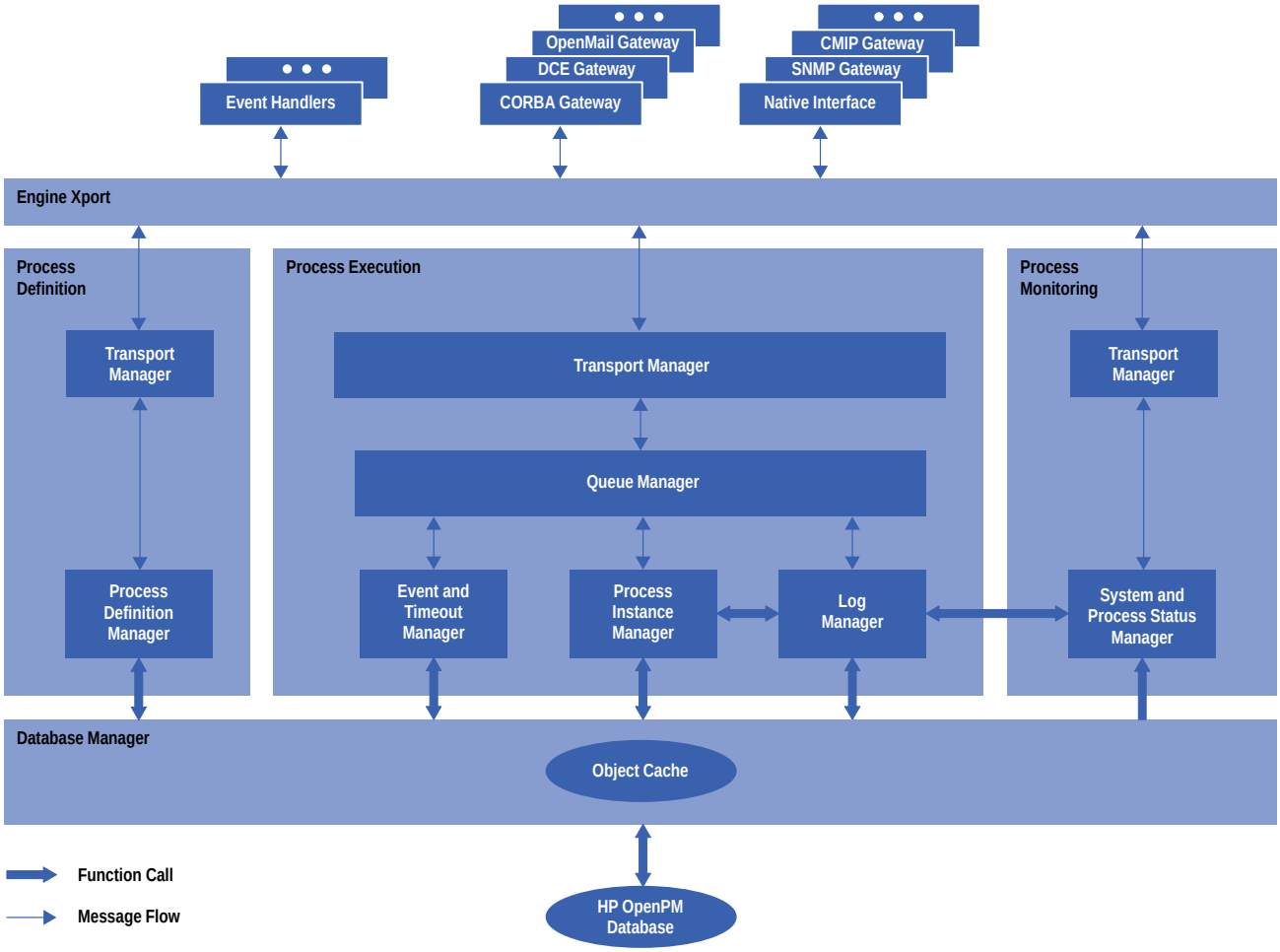

Fig. 3. *Block diagram of the HP OpenPM engine.*

HP OpenPM Business Objects

HP OpenPM has to interact with business activities supported by various implementations encountered in real life. These can range from manual handling by humans to automated processes executed by computers. 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 (Object Management Group) CORBA (Common Object Request Broker Architecture) technology has been developed to support interoperability for application integration.

Based on CORBA technology, in HP OpenPM an abstraction called a *business object* is built to encapsulate whatever piece of work each process activity has to accomplish. The wrapping code provides an IDL (Interface Definition Language) interface and the business objects are catalogued in the HP OpenPM business object library.

A business object, as defined by the OMG, is a representation of something active in the business domain, including its business name and definition, attributes, behavior, and constraints. It provides a 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 avoids repetitive coding to tailor the business activity implementation to each individual business process.

HP OpenPM Resource and Policy Management

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

In HP OpenPM, the mapping between the business activity (task) specified in a business process and the business object (resource) to be invoked is performed by the *resource manager* during run time as part of the execution of the business activity. HP 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.

HP OpenPM Worklist and Application Data Handlers

Two optional components that can be added into the HP OpenPM environment to facilitate the execution of business processes are the *worklist handler* and the *application data handler* (see Fig. 4). Both components are designed to enhance the scalability of HP OpenPM systems.

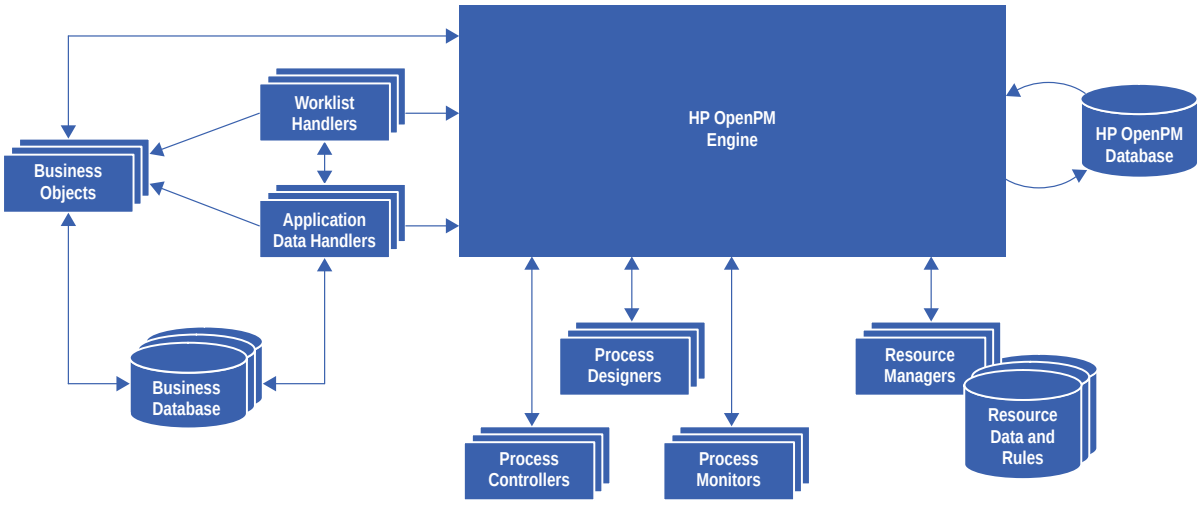

Fig. 4. *HP OpenPM system architecture including optional elements.*

The worklist handler supports both *engine-push* and *client-pull* modes to provide more freedom in task assignment. In addition, the worklist handler can be used to support the concept of *integration on demand*. Based on the task performer's profile, the worklist handler determines and launches a specific environment for 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 for application-specific data to remove the burden of database access from activity performers.

HP OpenPM Security

In today's business environments, security must be implemented enterprise-wide. The security service developed by the OMG provides authentication and encryption for HP OpenPM to prevent eavesdropping and forgery. The HP OpenPM infrastructure components can identify each other and vouch for the credentials of end-user components.

BPFM in the Telecommunications Management Network

The Telecommunications Management Network (TMN) defined by the International Telecommunications Union is changing the way operations support systems and business support systems solutions are being developed. The TMN architecture separates layers of functionality and provides access by elements in any one layer to any element in the layer immediately below. Before the introduction of the TMN model, operations support systems and business support systems solutions were isolated from each other and could not interoperate.

The HP OpenView Distributed Management platform supports the realization of TMN operations support systems and business support systems solutions for the TMN element management layer and network management layer (see ***Article 1*** for a description of the TMN layers). Still needed is a middleware service supporting the service management layer and even the business management layer of the TMN model. This need offers a great opportunity for BPFM added value. The next section presents an example of this support.

At the service management layer, the BPFM process enabling framework is required to be able to:

- Support reengineering and transformation processes for strategic operations support systems and business support systems.
- Integrate existing operational environments to form an enterprise hub for service management and provisioning.
- Deploy new management services as rapidly as possible.

- Monitor and measure processes.
- Tune processes to benefit from experience.
- Automate processes to reduce execution time.

The overall deployment of BPFM technology in the TMN environment is depicted in Fig. 5.

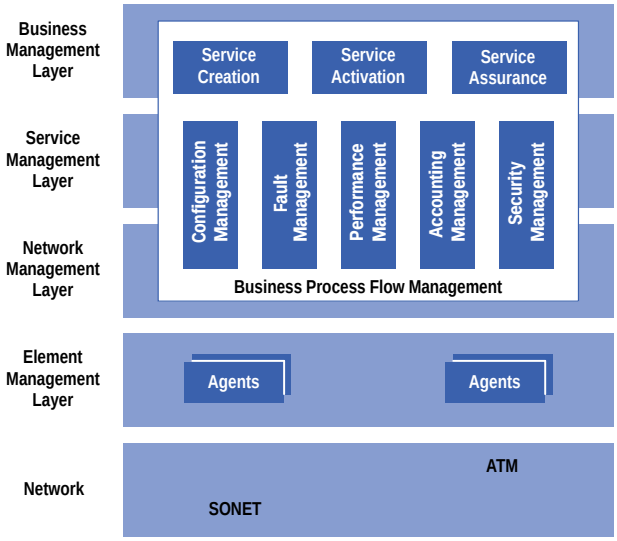

Fig. 5. *Telecommunications Management Network layers, showing management functions provided by business process flow management.*

SONET Configuration Management Prototype

Based on an HP OpenPM system, we built a prototype to demonstrate the application of BPFM technology in the specific domain of SONET (Synchronous Optical Network) configuration management. The prototype was a joint project between HP Laboratories in Bristol, England and Palo Alto, California to demonstrate the middleware technologies required to automate the processes supporting the configuration management of a SONET telecommunications network.

The scenario demonstrated by this prototype consists of the provision of a new VC4/VC12 path for customers. It goes through several different steps for this operation: search for a new route, negotiate the service level agreement (SLA) with the customer, configure the new path, and finally, update the SLA for this customer. The HP OpenPM process definition supporting the process of providing this new SONET data path is sketched in Fig. 6.

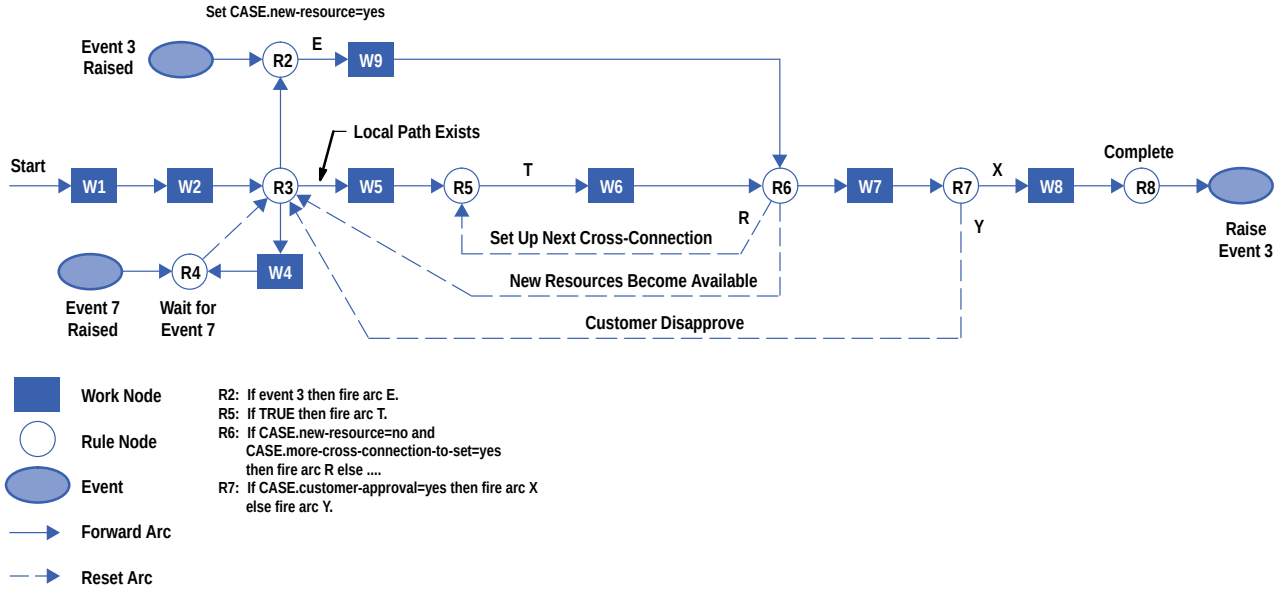

Fig. 6. *HP OpenView process definition for SONET configuration management.*

Searching for and configuring a new path in SONET are complex processes requiring a lot of interaction with the SONET MIB (Management Information Base) and network elements. This type of operation is a source of errors when it is performed manually by an operator as a set of individual, uncorrelated activities.

In the prototype, such complex operations as searching and configuring new paths are handled as business processes and automated by an HP OpenPM engine in an environment interacting with HP OpenView DM and Oracle DBMS applications.

Depending upon the changing business needs, a customer can request to add or drop communication paths between certain endpoints in a *private virtual network* (PVN). In HP 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 multiplexers (ADMs) to the endpoints, 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 continues on to step 5 and an independent subprocess is initiated to watch for resource changes.
5. Find valid routes between end-ADMs. This requires access to the routing table in the SLA database to determine whether any valid routes exist between the two end-ADMs. Either a list of ADMs is returned signifying the ADMs that must be configured to realize the route, or "No Route Found" is returned. For a returned list of ADMs, this activity will then use the HP OpenView DM facility agent to collect port information stored in the MIB 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 an ADM in the route, this activity uses the HP 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. During steps 5 and 6, if any additional resources become available, HP OpenPM cancels any currently running activity and starts the process over from step 5 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, quality of services (QoS), and so on. Depending upon the business factors (e.g., cheapest service wanted), the customer may request that a new search be initiated, that is, 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 by invoking the HP OpenView DM NE agent and updating the SLA database.

Acknowledgments

The authors would like to acknowledge the contributions of several individuals. John Manley and Mike Robinson provided leadership and guidance in the development of an early version of the SONET configuration management prototype for demonstration at Telecom '95. Chris Whitney helped provide the SONET environment simulator. Clemens Pfeiffer spearheaded the initial creation of HP OpenPM and helped gather the momentum needed for the project to survive. Nick Sheard led the product development to commercialize the HP OpenPM research. Chip Vanek helped drive the design of HP OpenPM as a major internal customer.

Bibliography

1. S. Aidarous and T. Plevyak, *Telecommunications Network Management into the 21st Century*, IEEE Press, 1994.
2. W. Du, C. Whitney, and M. Shan, "SONET Configuration Management with HP OpenPM," *Proceedings of the 12th International Conference on Data Engineering*, New Orleans, Louisiana, February 1996.
3. J. Davis, W. Du, E. Kirshenbaum, K. Moore, M. Robinson, M. Shan, and F. Shen, "CORBA Management of Telecommunications Networks," *Proceedings of the Workshop on Distributed Object-Oriented Computing*, Object World Frankfurt '95, October 1995.
4. J. Davis, W. Du, and M. Shan, "HP OpenPM: An Enterprise Process Management System," *IEEE Computer Bulletin*, June 1995.
5, W. Du, S. Peterson, and M. Shan, "Enterprise Workflow Architecture," *Proceedings of the 11th International Conference on Data Engineering*, Taipei, Taiwan, March 1995.
6. U. Dayal and M. Shan, "Issues in Operation Flow Management for Long-Running Activities," *Data Engineering Bulletin*, Vol. 16, no. 2, June 1993.

7. M. Shan, “OpenPm: An Enterprise Business Process Flow Management System,” *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Montreal, Canada, June 1996.

▶ Go to Article 9
▶ Go to Table of Contents
▶ Go to HP Journal Home Page