Design of A Business Process Analyzer

Li-jie Jin, Fabio Casati, Aad van Moorsel
Software Technology Laboratory
HP Laboratories Palo Alto
HPL-2002-145
May 15th , 2002*

business process analyzer, process probe, business process analysis

Business process management technologies are being adopted by more and more companies to improve the efficiency of both their internal processes and e-services offered to customers. In order to maintain a certain level of service quality, business processes need to be executed within a well-managed environment and with assistance of a suite of well designed tools. In this paper, we present the design of a business process logic and performance analyzer. This business process analyzer enables process developers and managers to monitor and to analyze business processes in a similar way to how a hardware logic analyzer enables hardware developers to monitor and to analyze their hardware logic design. We describe desirable functions of such a process analyzer and various process probing technologies. We then present an architecture and implementation of the business process analyzer. At the end, we discuss how this analyzer works with a business process management system to improve design and run-time performance of business processes.

* Internal Accession Date Only Approved for External Publication
To be published in the International Symposium on Information Systems and Engineering, 14-18 July 2002, San Diego, California

Design of A Business Process Analyzer

Li-jie Jin, Fabio Casati, Aad van Moorsel
Hewlett-Packard Labs
1501 Page Mill Road, MS 1142
Palo Alto, California 94304

Abstract

Business process management technologies are being adopted by more and more companies to improve the efficiency of both their internal processes and e-services offered to customers. In order to maintain a certain level of service quality, business processes need to be executed within a well-managed environment and with assistance of a suite of well designed tools. In this paper, we present the design of a business process logic and performance analyzer. This business process analyzer enables process developers and managers to monitor and to analyze business processes in a similar way to how a hardware logic analyzer enables hardware developers to monitor and to analyze their hardware logic design. We describe desirable functions of such a process analyzer and various process probing technologies. We then present an architecture and implementation of the business process analyzer. At the end, we discuss how this analyzer works with a business process management system to improve design and run-time performance of business processes.

Keywords: Business process analyzer, process probe, business process analysis

1. Introduction and motivations

Business processes usually represent business logic of business activities of an enterprise or business activities among multiple trading partners. Correct design and effective execution of these business processes is critical to success of these enterprises. Process design, automation, and management technologies are being increasingly used in both traditional and newly-formed, Interned-based enterprises in order to improve quality and efficiency of their administrative and production processes, to manage e-commerce transactions, and to rapidly and reliably deliver services to business partners, customers, and employees. Software applications that support the execution of business processes are typically called Business Process Management Systems.

To design, deploy and execution business processes correctly and effectively, process developers need various process managing and measuring tools that can help them get more information about process properties and process execution, and then control process service quality. There are three methods to get behavior knowledge of business processes. The first one is through process simulation [JCS02]. A process simulator is a valuable assistant tool. Newly designed processes can be tested with various execution environment assumptions before they are deployed. Business process reengineering action can be verified with a simulator before the business process is actually changed. The second method is through mining process execution log data [BCDS01]. Business process engines such as HP Process Manager have an embedded audit system that logs plenty of business process execution information at different abstraction levels based on the process engine configuration. This log information include engine status, process definitions, service configuration, time stamps of start and end point of each activity in a process instance, data of work items that are sent to and received from resource worklists, to name a few. The third method is runtime process probing and analysis through a process analyzer.

Hardware developers often use a logical analyzer to locate and fix subtle bugs and performance bottlenecks that occur when booting up complex system and integrating software with custom hardware. A high quality logic analyzer allows hardware developers to obtain real-time digital, analog and software behavior information with very complicated and high-speed hardware systems. It is desirable that business process developers can also get a similar analyzer to help their process designing and controlling. However, there are some new challenges on designing such a logic analyzer for business processes. This paper introduces a business process performance and logic analyzer that is able to generate testing workload and to collect process execution information in run time by connecting itself to a target process instance, or the process engine through a group of probes.

The focuses of this paper are on the concepts, architecture, and implementation of the business process analyzer (BPA). In section 2, we present the background and concepts of the BPA. In section 3, we introduce the architecture of the BPA. We discuss design of various types of process probes, probe mechanism and probe generation methods in section 4. We then discuss how to use BPA to set up a test bed for mission critical business process analysis in section 5. Finally, we conclude our works in section 6.

2. Backgrounds

The BPA design intends to be neutral to any business process management system. We use HP Process Manager (HPPM) [HPPM] as the business process execution environment in this paper for convenience. Business processes that are running on HPPM and HPPM engine itself are the probing and analysis targets of the BPA.

The HPPM comprises three types of major components: Process Engine, Work List Server and Resource Resolver. The Process Engine controls execution of a process. It steps through definition of a process to determine the sequence in which activities are performed. The Resource Resolver executes resource rules and assigns activities to the selected resources for execution. The Work List Server is a client management component, which acts as a work queue from which humans and applications can retrieve work items to execute. The Work List Server picks up work items from worklist queues and enables their access to client applications. When a client notifies the Work List Server that it has completed a work item, the Worklist Server puts this work item into a Send Queue, from which the Process Engine will pull out completed work items. The Work List Server guarantees that clients can only access worklists that are assigned to them by a HPPM administrator. The Resource Resolver is the role management component. It receives role description rules from the Process Engine, read in the rule definition files. By running those rules, the Resource Resolver resolves an address of the role that will carry out the activity and return this address to the Process Engine.

In HPPM, a business process is described by a directed graph that has several kinds of nodes: *work nodes* represent the invocation of activities (also called services), assigned for execution to a human worker or an automated resource. *Route nodes* are decision points that route the execution flow among nodes based on associated routing rules. Event nodes denote points in the process where an event is notified to or requested from other processes. Start node denotes the entry point of a process. Complete nodes denote termination points. Arcs in a graph denote execution dependencies among nodes: when a work node execution is completed, the output arc is "fired", and the node connected to this arc is activated. Arcs associated with output of route nodes are instead fired based on evaluation result of routing rules. Every work node is associated with a resource description, service, that defines the logic for selecting resource (or resource group) to execute the activity defined by the work node. Service also includes process data items passing to the resource upon invocation and receiving from the resource upon completion of the work. Several work nodes can be associated with one service description.

3. Concept of A Business Process Analyzer

The primary goal of designing a Business Process Analyzer is to enable active interaction between business processes and the business process measurement, management and control system. The major functions of a BPA may include: browsing deployed process definitions; configuring probes; measurement performance of a business process, an activity or the process engine; generating workload for performance and logic testing purpose; running simulation of a process; process emulation through creating a mirror process of a running process but replacing all associated services with a group of BPA probes. Figure 1 gives out the concept of how does a BPA connect to a business process.

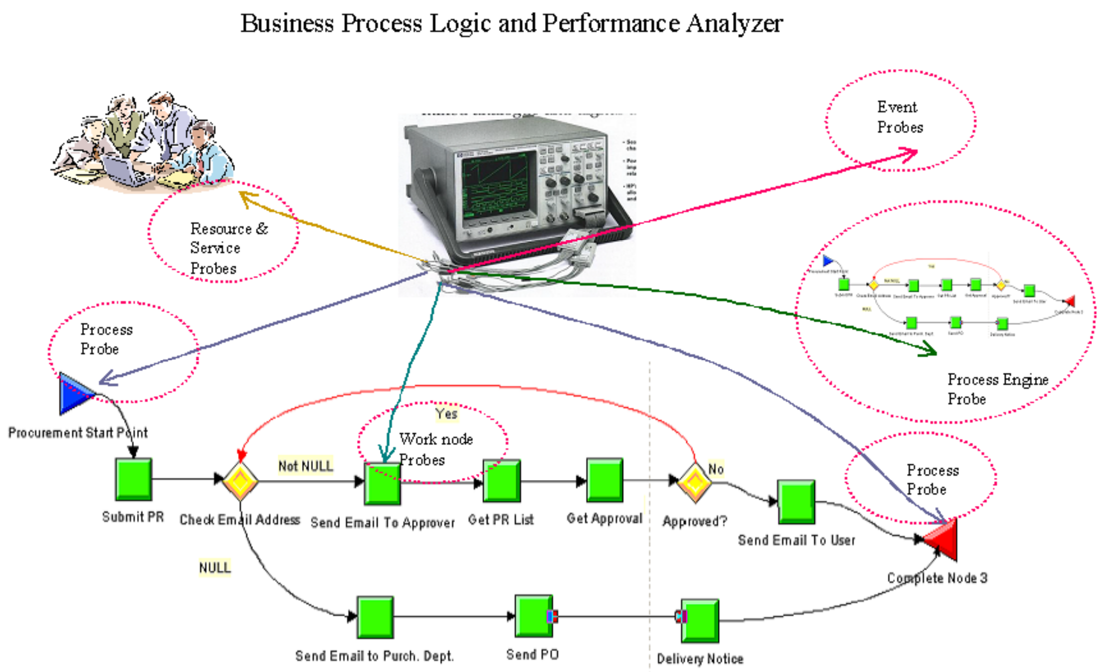


Figure 1 Concept of a Business Logic and Performance Analyzer

The BPA consists of three major parts: the Operational Interface, the body of process information generation and processing (BPA body) and a collection of probes. The Operational Interface allows user to operate BPA. The probes collect information from process instances and process engines. Some of the probes can also generate workload to the process engine. Some other probes may be able to behave like a service on a worknode. The BPA body analyzes the target business process through processing data that come from probes. Users can configure the BPA to do various processes measuring and analysis through the BPA Operational Interface. In the following we detail these three components.

3.1 BPA Probes

Probes are designed to collect timing information about processes and Process Engine in real-time according to users' configuration. Based on those data, the BPA can help users to deeply understand the dynamic behavior of a process they designed and mutual impacts between Process Engine and multiple running process instances.

There are various types of probes. For instances, A worknode probe is an automatic client of Process Manager that can talk to a process instance as a resource through a Work List Server. A configured Worknode Probe works in two different modes: *listening mode* and *interactive mode*. In *listening mode*, a probe can only read variable values of work items and performance data from the pre-defined performance parameter fields. In *interactive mode*, a probe can not only read variable value, but also generate output for he associated activity. Through worknode probes, BPA can collect performance information like incoming request rate to a service that is associated with a worknode, service response delay and service throughput.

A Process Probe is actually a special business process that is designed for performance information generation, recording and reporting. Users deploy the process probe to a Process Engine just like any other process. However, this probe process can only be initialized by the BPA. Activities that are included in the probe process can only be executed by a worknode probe. Event Probes inspects event activity through access event publish and event subscribe node. Since the event publish and subscribe have globe wide impact, an event probe usually connects to event nodes that are imbedded into the process probe. A Service Probe is a virtual probe that integrates information of a particular service from some other worknode probes. This service that the service probe is inspecting may work on various activities that belong to different processes simultaneously. Service probes concentrate on resource performance and behaviors.

Users define a BPA Connection between the BPA and its target. The BPA Connection is implemented by a group of BPA probes. The description of a BPA Connection explains number of probes, their types and positions on the target process instance. To define such a BPA Connection, users need to understand the business process that is under investigating. In HPPM, a process definition is described with both XML in text file and graphic file. By analyzing the process definition, users make their decision of probing on the whole process, or some activities of this process. With probing targets in mind, users can use BPA to configure probes from a generic probe template. The configuration information includes probe type, probing location, e.g. activity name; associated service; Process Engine name; a list of interested variables in the work items.

3.2 BPA Body

After BPA gets process information from its probes, it begins process analysis procedure with its BPA body. The business process performance metrics that the BPA is interested in are much similar to those of other process analysis technologies [JCS02] [GCDS01]. For example, average duration of a process instance, Process Engine throughput, financial costs at process level, average queue length of involved services, resource utilization. The BPA can also inspect resource behavior through information collected from its Service Probes. Those behavior parameters include resource average occpation time, average waiting time, average idle time, et al. The BPA aims at metrics that can only get from run time and can help dynamic performance adjustment in real-time. For example, the BPA can measure the current Process Engine load level that is defined in [JCSS01]. With the current Process Engine load level, the BPA can help the process administrator understand how busy the Process Engine is, what kind of service level each running process instance has. The BPA is also able to foresee duration of a process instance based on known system and process runtime information through the imbedded process simulator.

The BPA architecture details the structure of the BPA body. The major function objects of the BPA body include Display/Setting interface control, process information synthesis, engine load generator, a process simulation engine, process event management and a process data repository. These function objects are connected by a message bus. Figure 2 shows the architecture of the BPA and the message flow among BPA function blocks. The Display/Setting interface control object marshals process investigation configuration from front-end user interface and redirect them to other functional objects. It also creates a group of probes according to the users investigation interests. The details of how to create a group of probes will be described in the next section. The load generator is an object that can initialize instances of a process probe or other performance measuring processes. It also defines workload characteristics of the simulator. The Process Information Synthesis object responses for merging information from each probe and produce process level runtime data of a process instance. It also produces load index of the Process Engine itself. The process level information includes the elapsing time of the process instance, resource response time for executed activities in the process, costs until current, etc. The real-time Process Engine load index will be used to adjust the business process management system capacity. This BPMS capacity is a performance reference that is used by the simulation engine to set up the simulation parameter for the Process Engine. All information from Process Information Synthesis object will be send to a repository and the simulation engine as well.

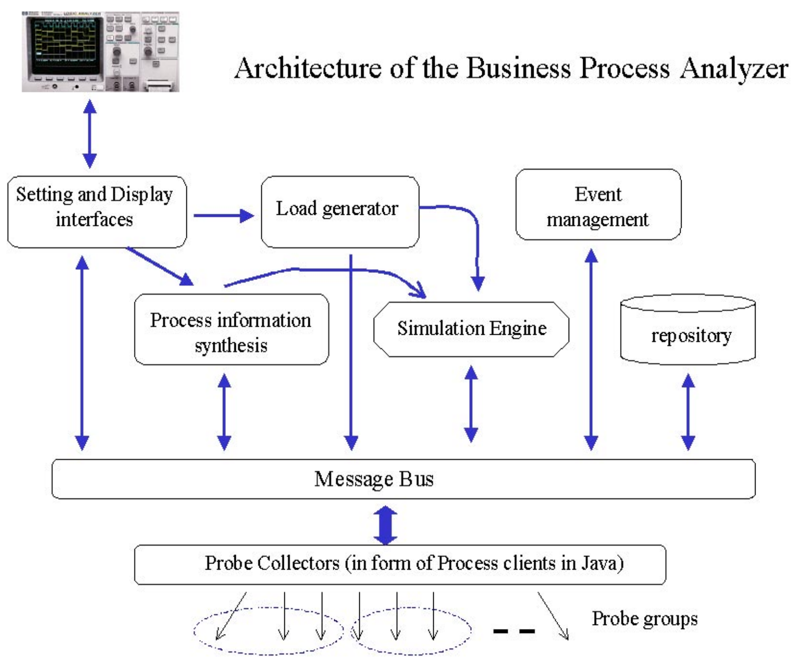


Figure 2 Architecture of the BPA

4. Architecture and Implementation of the BPA Probes

BPA distinguishes itself from other business process analysis approaches with dynamic runtime process probing technology and automatic generation of probes. We begin this section by describing structure of various types of BPA probes. Then, we present the solutions of probe generation.

4.1 Probe structure

The major probe types have been presented in section 2.2. Here, we are going to show the inner structures of those probes. As a software client of a process, a Worknode Probe consists of four components: Process Engine Connection, Worknode Execution Control, Data Parsing module and Message management. Figure 3 shows its inner structure. The Worknode Probe connects to Process Engine with the Process Engine Connection. The Process Engine Connection takes resource account information of the associated service or resource and the address of the Process Engine as parameters to create a HTTP connection between Process Engine and the probe. These parameters come from user process investigation configuration. In some cases, the probe can replace the associated service to execute the worknode. To do so, the worknode probe should get more business logic information from the process designer so that it can behave like a substitute resource of the worknode. After figuring out the proper business behavior of the replaced service, the Worknode Probe execute the worknode with its Worknode Execution Control component by calling a execution method of the worknode object through the Process Engine API. Data Parsing module is the data processing center of a worknode probe. A worknode probe should be aware of the semantic meaning of the work item that is associated with the target work node. Section 4.2 will discuss from where the probe get the semantic information of the work item. According to this information, the Data Parsing module is able to sort out contents of the work item that is going to be handled by the associated service from the data flow of a particular process instance. The Data Parsing module has two ways to get work items: on demand from user through BPA GUI, or whenever a work item is available. The Data Parsing module will stick a timestamp to any data that it read from the worklist. The Message management component is responsible for all data communication between the worknode probe and any other BPA components.

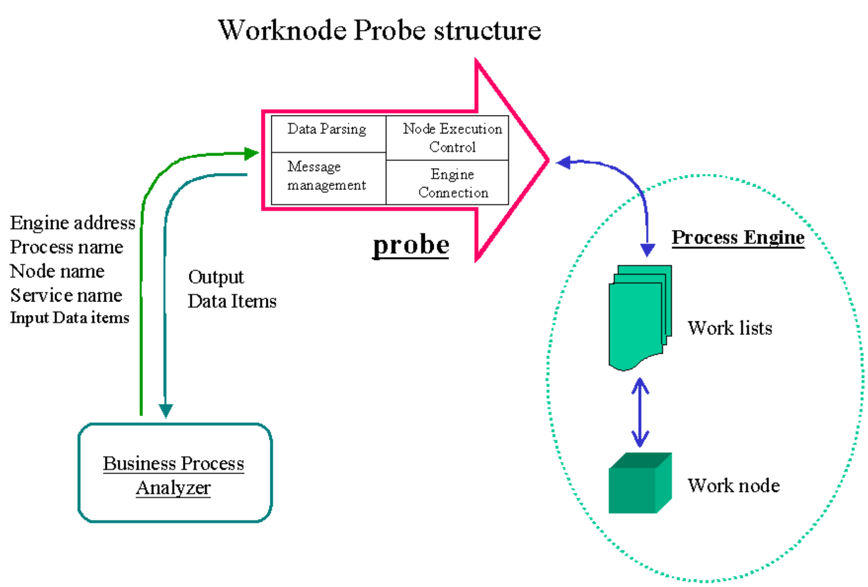
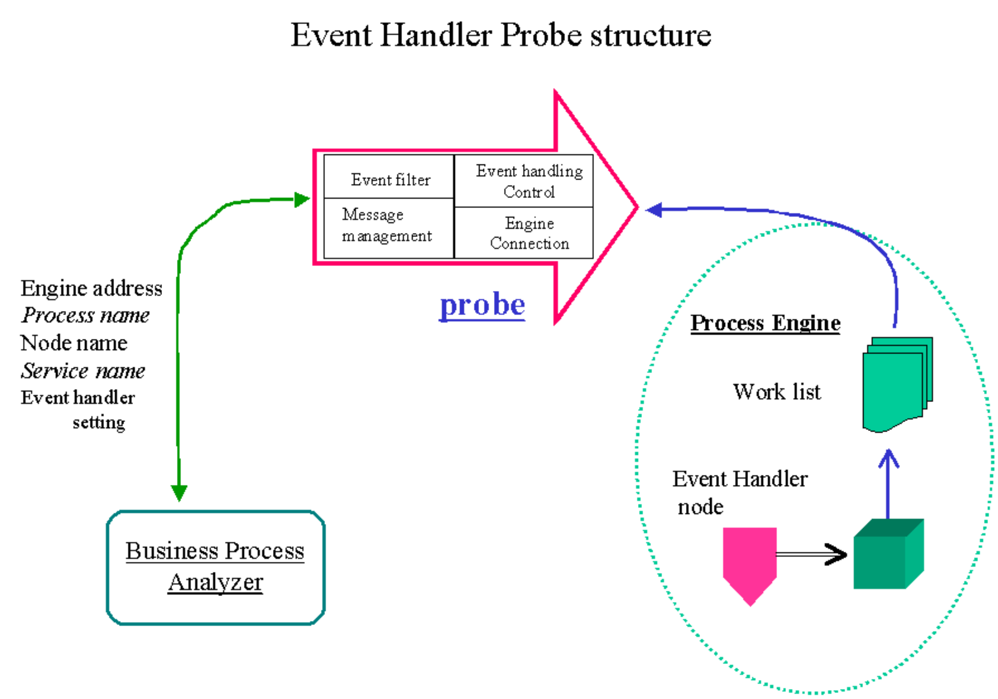


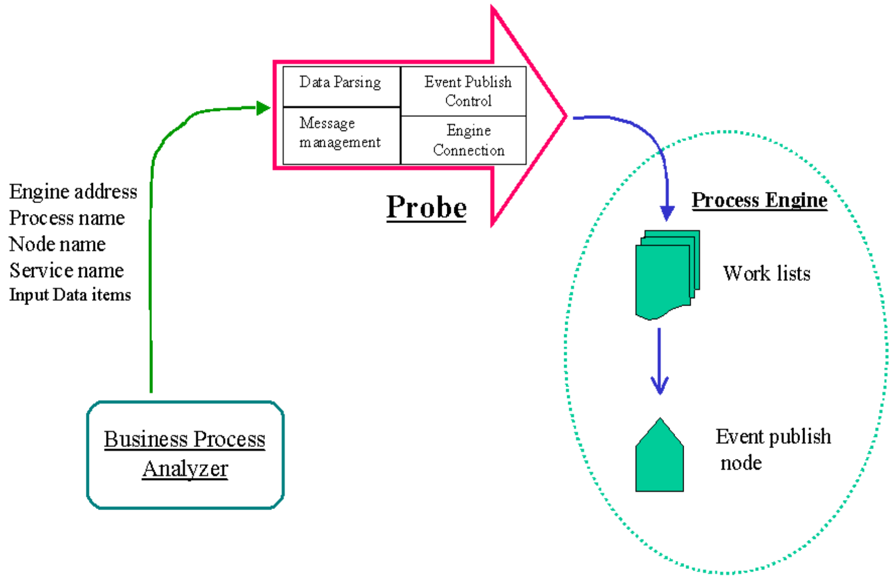
Figure 3. Structure of a Worknode Probe

There are two types of event probes: Event Publish Probe and Event Handler Probe as shown in figure 4. The Event Publish Probe usually only publishes event that is subscribed by process probes. Its structure is similar to the worknode probe except that it has an Event Publish Control component other than a Worknode Execution Control component. An Event Publish Control component can publish an event either for a process testing and/or dynamic debugging purpose, or for a performance measurement need. An Event handler Probe connects to the next conjoint worknode. It has an Event Filter component and Event Handling Control component. The Event Filter decides which event this probe is waiting for according to user configuration. The Event Handling Control component executes the business logic that response the special event.

A Process Probe has two parts: probe client and probe process. A probe client is like a worknode probe except that it doesn't have a Worknode Execution Control but a Process Initial Control component. A probe process is a simple business process that is dedicated to performance measurement. All worknodes of the probe process are executed only by a group of worknode probes. This probe process usually has event publish and handler branch and a worknode loop. Data fields of this process are used to record timestamps and variation of timestamps between connected worknodes. Through initializing an instance of the probe process, the Process Probe can measure response time of the Process Engine it is running on without touch any of other running process instances that may either be mission critical or involve sensitive customer private information. A Process Probe works together with a group of worknode probes to monitor the runtime load level of a Process Engine.



(a)



(b)

Figure 4. Structure of Event Probes. (a) Event Handler Probe, (b) Event Publish Probe.

The Service Probe is different from all other BPA probe types. It is actually a function block of the PIS in the BPA body. A Service Probe is created according to users' interested on a particular service or resource. This probe then will check all worknode probes and the BPA repository to retrieve information such as service time, service duration, service cost, waiting period, and idle period about the service or resource. A Service Probe can only get a snapshot of service performance on probed process instances during probing period.

4.2 Probe generation

When a user investigates a business process with BPA, he or she will need numbers of various type probes. It will be a big headache to use a BPA to test different business processes without assistant from BPA to generate those probes automatically.

The structures of BPA worknode probe, event node probes and client part of the process probe are similar. They all can be derived from a probe template. To produce a probe from this probe template, BPA needs to know two things: business process definition and user configuration about interested inspect position in this process.

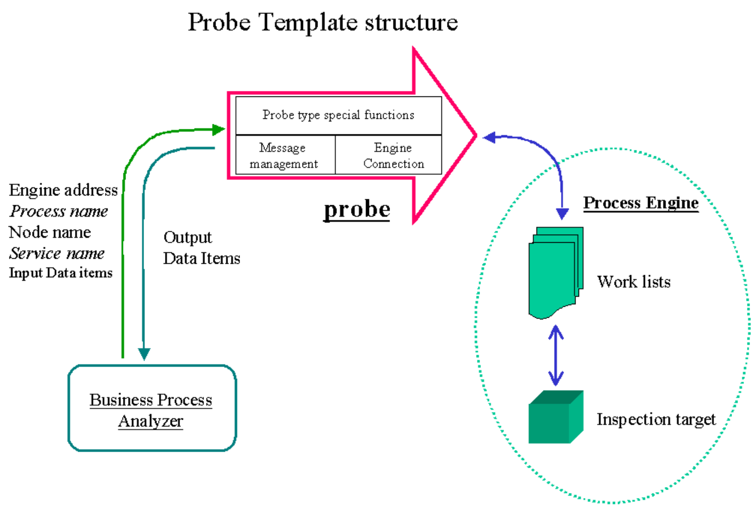


Figure 5. Structure of the Probe Template.

A probe template is shown in figure 5. This template has an Engine Connection component and a Message Management component. To generate a certain type of probe, BPA need to parse the process definition so that it can produce probe type special functions. In HPPM, business process is described with XML. Through parsing process definition file, BPA can get process information like activity types, activity names and associated services, partial relationships among activities, data field definitions and dataflow directions, duration range, cost types and base cost.

An activity is defined in a XML record. If the BPA get the record when parsing a process definition, it will do two things to generate a Worknode Probe from the Probe Template. First, the BPA will add a Worknode Execution Control component to the template. Second, it will go through the "Data-Item-Map" field, find name of each data item from " Alias Service". Then BPA will use this name to parse the service definition to get the idea of how this data item should be handled and set up "Data Parsing" Component and plug into the probe template. BPA puts Worknode Probes of the same process into a probe group. BPA user then can configure the probe group so that only probes that connect to investigation targets are initialized by the BPA.

5. Build a Business Process Emulation Environment With BPA

Business process simulation is very important in process logic design stage. It not only can help process designer verify the process logic but also can get a performance outlook based on a group of assumption data. However, the execution environment the process is going to run will be a very dynamic environment. The load level of the Process Engine keeps change all the time. The capacity of services and resources is also changing time by time because of resource sharing among process instances of this process and/or among process instances of other processes. When a user is interested in performance behavior of a particular process instance, very likely there is barely anything a process simulator can do to help the user. BPA has its own limitation. The worknode probes have the capability to take over a service that associates with the work node. However, it is very hard for a probe to behave entirely like the service that it replaced. It will also generate problems if the replaced service involves some actions that need authorization.

5.1 A Runtime Business Process Test Bed

We present a Business Process Emulation Environment With BPA in this section. The basic idea is to combine the runtime information probing ability and the "real" execution power of a second Process Engine to form a Runtime Business Process Test Bed. In the Runtime Business Process Test Bed, a business process is replicated from the Process Engine it is currently running on through BPA's probes. However, all services that associate with worknodes and event nodes of the process are replaced by probes. The probes on the process that is running on the operational Process Engine are called Primary Probe(s). The probes on the second emulation engine are called Emulation Probe(s). They are the same as those Primary Probes except that these emulation probes not only squeeze performance information from worknodes but also use their Worknode Execution Control components to control the execution of their target process. In the mean time, Primary Probes collect performance data of the operational Process Engine and related services and transfer them to Emulation Probes as emulation parameters.

Based on the emulation parameters, the BPA Load generator starts process instances, thereby providing input data according to the computed value distribution. The emulation Process Engine executes processes according to the process definitions, invoking the Emulation Probes. The Emulation Probes will return data in accordance to the distributions described in the emulation parameters. The services execution time is "compressed". Their duration is recorded into suitable variables. The emulation engine is unaware that these are special emulation processes. It executes them with its usual behavior. In particular, it also logs execution data.

Process execution data are transformed, to replace execution time logged by the engine with the one computed by the Emulation Probes and stored into process data items. Such "emulation" data items are then removed from the logs. Furthermore, Emulation Probe names are mapped back to the actual service names. Then, emulation data can be analyzed with the process analysis techniques. In this way, it is possible to understand the behavior of the newly introduced process, as well as the behavior of the other processes. In addition, it is possible to compare results with the data of the operational systems that describe the behavior of the processes prior to the modifications.

5.2 Benefits of the runtime business process test bed

The advantages to do so are that first this method is applicable with different service composition model and engines. Second, the method allows the simulation (through emulation) of all the aspects of the process model. In contrast, some simulation approaches make several simplifications and do not simulate important characteristics of the process such as events or value-based path selection. Third, process execution behavior can be analyzed with several semantic analysis techniques. In particular, the same techniques that are available for the analysis of operational processes can be used. Analysts can therefore use the tools and the analysis techniques with which they are familiar. The last but not least important point is that besides analyzing the behavior of the newly defined process, the method allows the analysis of the impact on existing processes when they are running. In particular, it is possible to analyze and emphasize the differences between the behavior of the processes before and after the introduction of the process.

6. Summary and Conclusions

The business process analyzer enables process developers and managers to monitor and to analyze businesses. We described desirable functions of such a process analyzer and various process probing technologies. We then presented an architecture and implementation of the business process analyzer. We also discussed how this analyzer works with a business process management system to improve design and run-time performance of business processes.

Our approach that is described in this paper enables user level business process performance monitoring and analysis. It is important that business process owners hold control of performance information while outsourcing the execution of their processes.

7. References

[AP99] M.Aguilar, A.J.G. Pater. Business Process Simulation: A fundamental Step Supporting Process Centered Management. *Proceedings of 1999 conference on winter simulation.* 1999.

[BCDS01] Angela Bonifati, Fabio Casati, Umesh Dayal, and Ming-Chien Shan. Warehousing Workflow Data: Challenges and Opportunities. *Proceedings of VLDB01*, Rome, Italy. September 2001

[CS02] Fabio Casati and Ming-Chien Shan. Semantic Analysis of Business Process Executions. Proceedings of EDBT'02, Prague, Czech Republic. March 2002.

[DHL01] U. Dayal, M. Hsu, R. Ladin. Business Process Coordination: State of the Art, Trends, and Open Issues. Proceedings of the 27th VLDB Conference, Roma, Italy, 2001.

[GCDS01] D. Grigori, F. Casati, U. Dayal, M.C. Shan, Improving Business Process Quality through Exception Understanding, Prediction, and Prevention, *Proceedings of the 27th VLDB Conference*, Roma, Italy, 2001.

[HPPM] Hewlett-Packard. HP Process Manager Process Design Guide. 2001. Available from www.ice.hp.com

[HPUC] Hewlett-Packard, HP utility data center,

http://www.hp.com/solutions1/infrastructure/solutions/utilitydata/index.html

[HR98] Vlatka Hlupic, Stewart Robinson. Business process modelling and analysis using discrete-event simulation. *Proceedings of 1998 conference on winter simulation.* 1998, Washington, D.C., USA.

[JCS02] Li-Jie Jin, Fabio Casati, and Ming-Chien Shan, Business Process Simulation with HP Process Manager, *Proceedings of Collaborative Technologies Symposium.* San Antonio, Texas, USA, 2002.

[JCSS01] LiJie Jin, Fabio Casati, Mehmet Sayal, Ming-Chien Shan. Load Balancing in Distributed Workflow Management System. *Proceedings of SAC2001*, Las Vegas, NV. March 2001.

[MP99] Melao, N., Pidd, M. Business process simulation: an overview. *Proceedings of 4th UK Simulation Society National Conference.* April 1999.