Business Process Simulation with HP Process Manager

Li-Jie Jin, Fabio Casati, Ming-Chien Shan
Software Technology Laboratory
HP Laboratories Palo Alto
HPL-2001-285
November 7th , 2001*

E-mail: {ljjin, casati, shan}@hpl.hp.com

workflow, simulation

Business processes present business logic of enterprises and or web services. In current Internet based dynamic business environment, business logic and resource allocation may have to change more frequently than before to match various business conditions in time. Business process deployment is a very costly procedure. It is desirable that a business process reengineering action could be verified before it is deployed. In this paper, we present an innovative business process simulation system that can use historical execution information of business processes when it is available as reference to set up simulation conditions. Integrated with a process modeling facility, this simulation system allows business processes to be simulated before they are deployed.

* Internal Accession Date Only Approved for External Publication

Business Process Simulation with HP Process Manager

Li-Jie Jin, Fabio Casati, and Ming-Chien Shan
Hewlett-Packard
1501 Page Mill road, 1-U4
Palo Alto, CA, 94304
Email: {ljjin,casati,shan}@hpl.hp.com

Abstract - **Business processes present business logic of enterprises and or web services. In current Internet based dynamic business environment, business logic and resource allocation may have to change more frequently than before to match various business conditions in time. Business process deployment is a very costly procedure. It is desirable that a business process reengineering action could be verified before it is deployed. In this paper, we present an innovative business process simulation system that can use historical execution information of business processes when it is available as reference to set up simulation conditions. Integrated with a process modeling facility, this simulation system allows business processes to be simulated before they are deployed.**
***Keywords* – workflow, simulation.**

I. INTRODUCTION

Business Processes Management Systems (BPMS) are software applications that support the definition, execution, and management of business processes. BPMSs are being increasingly used both in traditional and in the newly formed, internet-based enterprises, to support administrative and production processes, execute e-commerce transactions, and monitor business operations. In fact, BPMSs typically allow companies to reduce costs as well as improve the speed and, in general, the quality of process executions.

One of the main features of BPMS tools is the availability of a process modeling facility, that enables process designers to describe the many aspects involved in a process execution, such as tasks, execution flows, data flows, resources, constraints, and exceptions. In addition, BPMSs also provide support for process modification and versioning.

While designing processes with such tools is very easy, designing "good" processes is extremely difficult [Casati00, Leymann01]. This is due to several reasons:

- Processes are designed by a process modeler, that interviews (business and IT) people in an organization (the domain experts) in order to discover and understand their actual or desired processes. This is a difficult endeavor, and problems often arise due to lack of complete information, lack of communication, lack of understanding, etc.
- A business process has many different facets (such as the ones listed above), that need to be perfectly orchestrated in order to obtain optimal results.
- It is difficult to predict the actual workload of the process, and therefore it is difficult to define process aspects (such as assignment to resources) that are affected by workload considerations.
- Although different processes are often designed independently and may be conceptually unrelated, they do interact in several ways. For example, they may share some (human or automated) resources, invoke the same services, and run on top of the same BPMS (thereby sharing the system resources). For these reasons, executions of a process may impact (and be impacted by) executions of other processes.

Taking into account all of the above issues is very challenging. Therefore, process designers would greatly benefit from a process simulation environment that allows them to understand, measure, and evaluate the impact resulting from the introduction of new processes, from modifications to existing ones, or from the reallocation of resources.

Traditional process simulation environment are fairly simple, and simulate based on user-defined parameters such as assumed transaction coming rates, the expected subtask execution time and the expected outcome of branch condition evaluations [iGrafx]. While this can be useful for an approximate analysis, it is by no means sufficient to get a complete understanding of the potential impact caused by the new (or modified) process. In particular, existing approaches have the following drawbacks:

- They do not take into account the load on resources and services (possibly caused by executions of different processes). The performance or resources and services are likely to drop as the number of process executions increase.
- They do not take into account the load on the BPMS and the limited system capacity to execute processes.
- They limit the impact analysis to the process being simulated, while other processes are impacted too. These effects need to be carefully

scrutinized as well, with the same attention devoted to the process being designed.

- Simulation parameters are provided by the user, often based on top-of-the-head estimates. However, even a newly designed process typically reuses services and resources that have already been used by other processes. Therefore, we can get an indication of their execution characteristics based on past execution data, instead of asking the user to try to guess the value of these parameters. This approach is even more applicable to process modifications (which are indeed much more frequent than process design from scratch).
- While parameters describing average execution times are important, the behavior of the BPMS and of resources may deviate considerably from these average values. For example, some services may be very effective on weekdays, but very slow on weekends. These deviations need to be taken into accounts in order to avoid processes that have a reasonable average quality but that may still be affected by frequent quality degradations.

In this paper we describe a simulation engine developed for HP Process Manager (HPPM), a workflow product developed by Hewlett-Packard [HPPM]. We first describe the HPPM process model, and then detail the implementation of a system that can simulate HPPM processes.

II. HPPM PROCESS MODEL

The process model of a Business Process Management System outlines essential process definition elements and their relationships. In HPPM, a process is described by a directed graph that has several different kinds of nodes:

- *Work nodes* represent the invocation of activities (also called *services*), assigned for execution to a human or automated *resource*.
- *Route nodes* are decision point that route the execution flow among nodes based on an associated *routing rule*.
- *Event nodes* denote points in the process where an event is notified to or requested from other processes
- *Start nodes* denote the entry point to the processes.
- *Complete nodes* denote termination points.

Arcs in the 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 in output to route nodes are instead fired based on the evaluation of the routing rules.

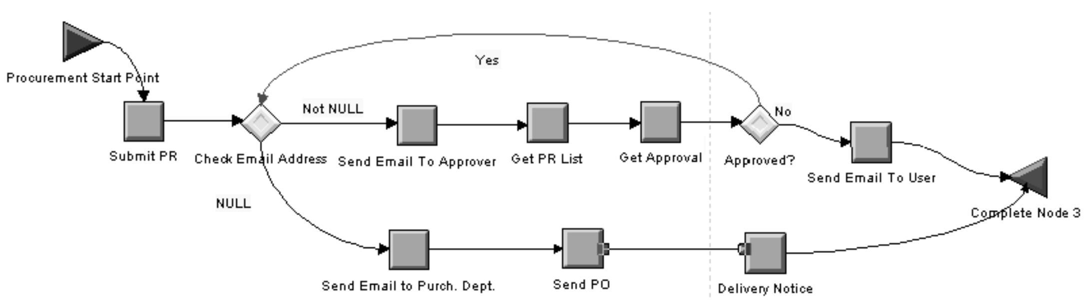

Fig. 1. A sample process modeled in HPPM

Figure 1 shows how a procurement process is presented in HPPM process model. When a purchase request is issued through web based front-end interface, an instance of this procurement process is created to handle this request. In the figure, boxes represent work nodes, diamonds represent route nodes, while triangles model start and complete nodes.

Every work node is associated with a *service (*also called *activity) description*, that defines the logic for selecting a resource (or resource group) to be invoked for executing the work. The service also defines the process data items to be passed to the resource upon invocation and received from the resource upon completion of the work. Several work nodes can be associated with the same service description.

III. SIMULATING HPPM PROCESSES

The modeling and design tools provide useful features for designing HPPM processes and deploying them on op of the HPPM execution engine. However, they do not provide any simulation support. In particular, they are not able to tell process designers and process viewers more information about how the process will run under different kinds of workload distributions. They are also not able to expose what could happen when a process is deployed and executed. For example, it is difficult for a process developer to pin point bottlenecks only from a process flowchart. Those questions can be answered either by deploying and running the process in a process engine with all associated resources and services connected and taking the associated risks of errors or low-quality operations, or by executing the process within a simulation environment.

A process simulator could be useful not only to explore dynamic characteristics of an existing business process through "what-if" experiments, but also to study potential behaviors of a newly created process before involving costly process deployment procedure. In the following we describe how we extended HPPM design tools in order to provide process designers with advanced simulation capabilities.

We implemented a business process simulation environment that based on top of iGrafx Process, a discrete event simulation engine (see [Banks00] for an overview of discrete event simulation techniques, and [iGrafx] for the specifics of the iGrafx tool). The simulation environment changes from one state to another at discrete points in time as the result of simulation events. Simulation events could be: a resource starts its work, a resource finishes its work; or a pre-scheduled work is due.

This simulation environment also integrates process and service/resource analysis, and derives simulation parameters from past executions (see Fig. 2), by querying a process data warehouse (PDW) [Bonifati01]. When the process to be simulated includes nodes, resources, or services that already exist and for which execution parameters are available, then these parameters are derived from the execution traces. This approach enables more accurate simulation with respect to the case in which parameters are inserted by the process designer based on guesses or estimates. In addition, it can load a very high number of detail parameters that describe time- and case-based characteristics (such as the performance of a resource in processing travel request on weekends). As a minimum, this would be a very challenging and time-consuming should users have to input those parameters themselves.

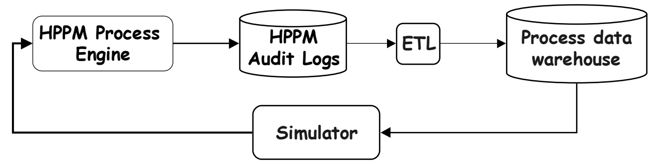

Fig. 2. HPPM Process simulation architecture

The process simulation model includes four major parts: *workload generator(s)*, *resource arrangement*, *schedule definitions* and *activity behavior* parameters (such as duration type and values for work-nodes and outlet ratio for route nodes).

The HPPM process simulator provides various ways to generate workload for process simulation. First, a process instance can be generated only after the previous one has completed. Through this type of workload, process developers can estimate average cycle of the process they are working at. Second, a process instance can be generated whenever the start node has an available resource. This type of workload can help process developer find out capacity of his or her process under a certain resource configuration. Third, process instances can be generated in an inter-arrival way. The inter-arrival time can be set as a constant. It can also be set to follow an expression. Process instances can be generated at the occurrence of a specific time event such as 8:30 am every workday or 10:00am every Tuesday. Process instances can also be generated based on a timetable that specifies the numbers of process instances to be issued in each specified time period. A timetable can be generated from process execution log information. Then the HPPM process simulator can repeat the activation pattern that occurred in a past time period.

HPPM simulator allows process developer to define a schedule for simulation execution in a flexible way. A schedule is a list of active hours and days. When creating a simulation scenario, process developer can set the simulation to begin Monday morning at 8:00 am (which is the default) or he can designate the exact days and hours during which the simulation takes place. A *resource schedule* determines the times that the resource is available. It is set at the resource definition stage. For instance, a manager is available from 9:00 am to 5:00 pm Monday through Friday. A security guard is available from 6:00 pm to 6:00 am every weekday night. An *activity schedule* determines the times that the work node that represent this activity can be fired. For example, a staff meeting is set on 10:00 am on each Tuesday. During this meeting time, all workforces are subject to hold their current tasks and attend the meeting.

There are two major activity (service) behavior parameters: *duration* and *cost*. The duration can be a constant, or it can be defined by an expression. There are two ways to set up durations for worknodes in a process: it can be manually inserted by a user or it can be automatically derived from information contained in the Process Data Warehouse. In fact, the PDW includes duration information for each work node execution. Therefore, if the process to be simulated includes a work node that has been already executed in the past, its characteristics can be derived from the PDW. Analogous techniques are used for determining resource data.

The HPPM simulator extracts parameters from PDW by computing a distribution from the available data. For example, it processes the duration data about the executions of a node to compute what is the probability distribution of that node's duration. The distribution parameters (e.g., normal distribution with a certain average and variance) are inserted as simulation parameters for that node. The simulator also informs the user if the confidence of the derived distribution is below a certain, specified threshold.

For route nodes, behavior parameters are ratios (probabilities) that each outlet (i.e., output arc) has. Again, these parameter can be manually set of it can be derived from the PDW. By changing these ratios in several different ways, process developers can analyze several different execution cases, to understand the behavior of the process in many different conditions.

The simulation results are collected in a *report* file. These results cover the dynamic aspects of a business process. There include process times, costs, resource utilization, and work queues. Process developers can use custom statistics to create customized combinations of the standard statistics (for example, setting costing information or revenue formulas). Several sets of named simulation data can be compared to evaluate different scenarios.

IV. CASE STUDY

We use the process in Fig. 1 as an example to demonstrate how the HPPM process simulator works. This is a simplified version of a corporation procurement process. It includes one start node, eight worknodes, two route nodes and one complete node. We set the duration of start node equal to 5 minutes, worknode “Submit PR” equal to 10 minutes, worknode “Get PR List” equal to 5 minutes, and worknode “Get Approval” equal to 10 minutes. All other email and notification type worknodes have a one-minute duration. The outlet ratio for both route node are all 50 to 50 percent. The workload-generator produces total 1000 transactions for each simulation run. Those transactions are coming in a uniform distribution with random spread in a range between one to thirty minutes.

The elapsed time for one thousand transactions is 44.21 days. The resource utilization is shown in table 1.

We change the approval ratio from 50/50 to 70/30 in favor of disapproval. Then we have the same elapsed time for one thousand transactions. Table 2 shows the changes of resources utilization.

Resource Utilization %

CeRes_ProcureMent_Start_Point	32.05
CeRes_Submit_PR	64.10
CeRes_Send_Email_To_Approver	4.28
CeRes_Send_Email_To_Purch_Dept	4.27
CeRes_Send_PO	4.27
CeRes_Get_Approval	42.75
CeRes_Delivery_Notice	4.27
CeRes_Send_Email_To_User	2.14
CeRes_Get_PR_List	21.38

Table 1. Simulation run 1

Resource Utilization %

	Sim #1	Sim #2
CeRes_ProcureMent_Start_Point	32.05	32.04
CeRes_Submit_PR	64.10	64.07
CeRes_Send_Email_To_Approver	4.28	3.77
CeRes_Send_Email_To_Purch_Dept	4.27	3.77
CeRes_Send_PO	4.27	3.77
CeRes_Get_Approval	42.75	37.74
CeRes_Delivery_Notice	4.27	3.77
CeRes_Send_Email_To_User	2.14	2.64
CeRes_Get_PR_List	21.38	18.87

Table 2. Simulation run 1 and 2.

What will happen if the resource CeRes_Get_Approval, usually a role that is performed by a manager, reduce its time cost by half and how will this resource level performance change impact the whole process? Simulation results show that the average cycle for simulation run 1 and 2 are 2.09 hours and 2.13 hours respectively. After the resource CeRes_Get_Approval reduce its time cost, the average process cycle drops to 1.89 hours. The resource utilization for CeRes_Get_Approval drops to 18.88 from 37.74. It means that this resource has more potential capability.

The HPPM process simulator can also report total cost and cost distribution in a process; queue size of each resource; activity statistics and some other performance information.

V. CONCLUSION

The HPPM Process simulator is a very useful tool for business process designers, developers, and administrators. It offers them a way to understand the behavior of newly-defined processes without involving in costly deployment procedures. It can also help them to verify the output of a major process modification before actual implementation. This simulator gets extended capability, accuracy and usability through a bridge to historic process execution data and resource performance information, stored in a Process Data Warehouse.

REFERENCES

[Bonifati01] A. Bonifati, F. Casati, U. Dayal, M.C. Shan. Warehousing Workflow Data: Challenges and Opportunities. *Procs. of VLDB'01*. Rome, Italy. Sept. 2001.

[Casati00] F. Casati and M.C. Shan. Process Automation as the Foundation for E-Business. *Procs. of VLDB2000*, Cairo, Egypt, 2000.

[Leymann01] F. Leymann, D Roller. Production Workflows. *Addison Wesley*, 2001.

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

[iGrafx] Micrografx, iGrafx Process User's Guide. 1999.

[Banks00] Jerry Banks (Editor), et al, Discrete Event System Simulation, Prentice Hall, August 2000.

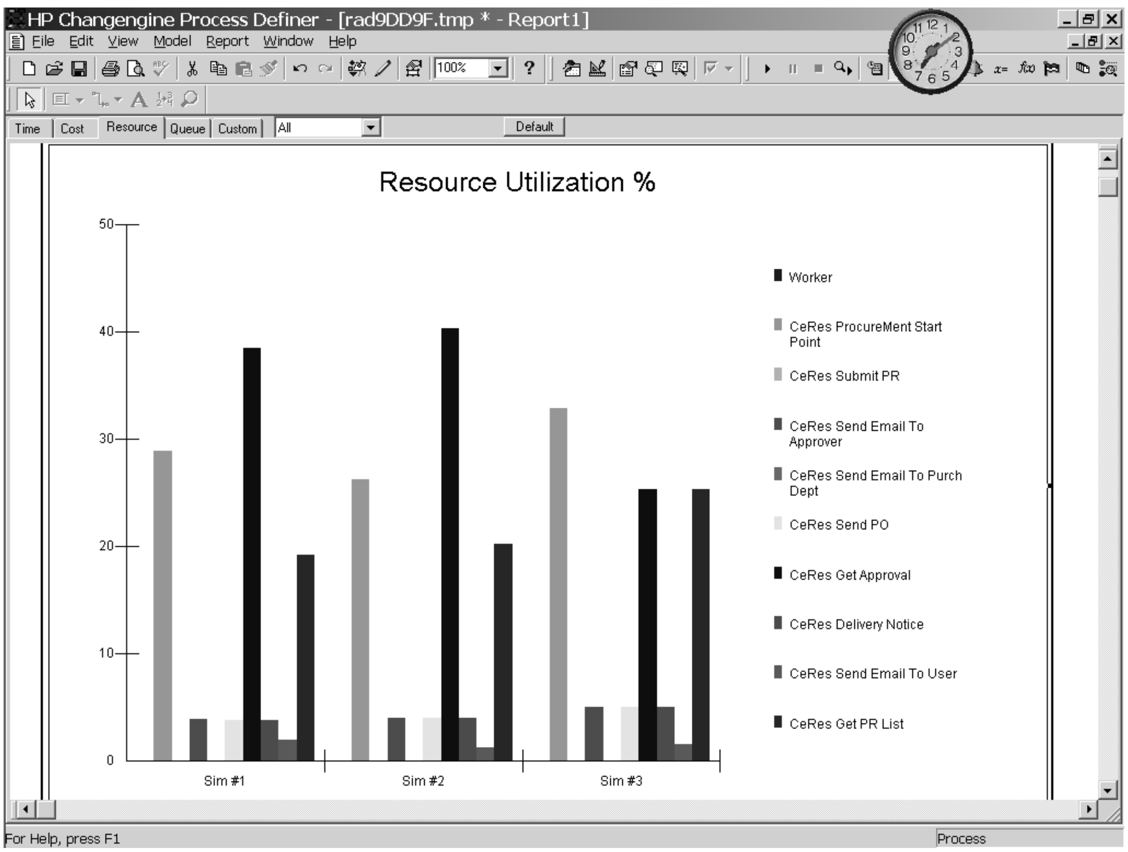

Figure 3 - Sample simulation results