Business Process Cockpit

Mehmet Sayal, Fabio Casati, Umesh Dayal, Ming-Chien Shan
Software Technology Laboratory
HP Laboratories Palo Alto
HPL-2001-313
November 28th , 2001*

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

workflow, business process, HPPM (HP Process Manager), data analysis, visualization of data

Business Process Cockpit (BPC) is a tool that supports real-time monitoring, analysis, management, and optimization of business processes running on top of HP Process Manager, the Business Process Management System developed by Hewlett-Packard. The main goal of the Business Process Cockpit is to enable *business users* to perform *business-level* quality analysis, monitoring, and management of business processes. The BPC visualizes process execution data according to different *focus points* that identify the process entities that are the focus of the analysis, and different *perspectives* that define a way to look at the information. The BPC also allows users to define new concepts, such as “slow” and “fast” executions, and use those concepts to categorize the viewed data and make it much easier for users to interpret.

* Internal Accession Date Only
Approved for External Publication

Business Process Cockpit

Mehmet Sayal, Fabio Casati, Umesh Dayal, and Ming-Chien Shan
Hewlett-Packard
1501 Page Mill road, 1-U4
Palo Alto, CA, 94304
Email: {sayal,casati,dayal,shan}@hpl.hp.com

1 Overview

Business Process Management Systems (BPMSs) are software platforms that support the definition, execution, and tracking of business processes. They are often used to support administrative and production processes as well as to implement complex web services, delivered by composing existing ones according to some process logic. BPMSs have the ability of logging a lot of information about the business processes they support, including for instance the start and completion time of each activity, its input and output data, the resource that executed it, as well as every event (message) sent or received by a process.

This information is a gold mine for business and IT analysts: in fact, its analysis may *reveal problems* and inefficiencies in process executions and *identify solutions* in order to improve process execution *quality*, both as perceived by the users in terms of better and faster processes (*external* quality), and as perceived by service providers in terms of lower operating cost (*internal* quality). In addition, information on active processes can be used to *monitor* active process instances and be alerted of predicted and actual quality degradations.

This paper describes the *Business Process Cockpit* (BPC), a tool that supports real-time monitoring, analysis, management, and optimization of business processes running on top of HP Process Manager, the BPMS developed by Hewlett-Packard [HPPM]. The BPC is a component of the Business Process Intelligence (BPI) tool suite, a set of tools that support off-line and real-time business and IT decisions to improve business process quality. The BPI suite operates by applying data warehousing and data mining techniques to business process execution data, as shown in Figure 1. An Extract, Transfer, and Load (ETL) application collects data from the log and loads them into a Process Data Warehouse (PWD). Besides performing "traditional" warehousing functions such as data cleaning, the ETL component also labels process executions with quality information, based on user-defined criteria of quality. Details on the ETL, PDW, and process quality labeling are provided in [BCDS02]. Data in the PDW can be directly accessed with a commercial reporting tool (such as Crystal Reports or Oracle Discoverer), and are also used by the BPI simulation component to derive simulation parameters from real data when possible (details on the simulation component are provided in [JCS02]). In addition, data mining techniques can be applied on top of the warehouse to assist analysts in identifying the causes of high and low-quality executions and deriving prediction models that can be used at run-time to predict process execution quality for running processes.

The BPC is a crucial component of the BPI tool suite. It provides the following functionality:

- It allows business and IT users to analyze PDW data according to multiple *perspectives*. The BPC is designed to make it easy for (non-technical) users to define a wide range of queries and, more generally, of analysis and quality criteria, without writing any code. The information is presented in an intuitive and direct format, so that users can easily and immediately get the information they need. In addition, the cockpit is configurable according to the needs and preferences of different users.
- It monitors processes, services, resources, and other process-related entities, and inform users of actual or foreseen quality degradation. BPC can also send notifications to users on the medium of their choice.
- BPC can manage running processes by tuning process and system configuration parameters (such as the process priority) and by notifying *events* to processes.
- Java API that extends HPPM API to process monitoring and analysis

The design and development of such a component present several challenges:

- determining which is the best way to present business process execution data
- designing a way to let users express a large number of business process reporting queries without writing code
- enabling users to easily extract business-level information (i.e., *quality* metrics) from execution data
- provide a monitoring console, through which users can have a view of the system at process status at a glance, and be informed of situations that are critical from both an IT and business perspective (such as bottlenecks or risk of missing service level agreements).
- Identify how to provide feedback to running process instances when such critical situations are detected, in order to correct executions.

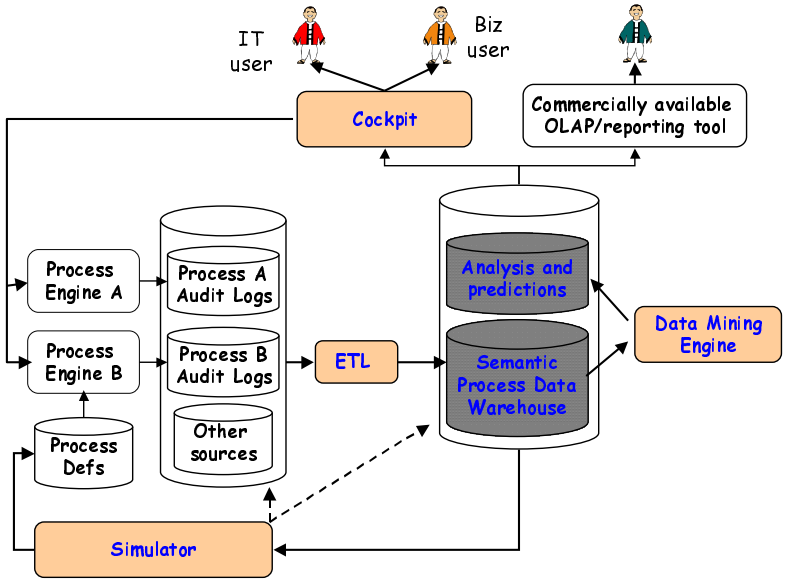

Figure 1 - Architecture of the Business Process Intelligence tool suite

In the following we describe the concepts, features and architecture of the BPC, thereby describing how we addressed these challenges.

We also observe that similar concepts and architectures can be reused and extended to manage, in a uniform fashion, web services, enterprise applications, and in general any e-business system. We initially focused on business process because of its value, both in general and for Hewlett-Packard in particular, and because we had access to customers' requirements and customers' data, a crucial help in the design and development of such a system.

2 Concepts and Features

The main goal of the Business Process Cockpit is to enable *business users* to perform *business-level* quality analysis, monitoring, and management of business processes. Targeting business users implies keeping the user interface as simple and immediate as possible, without limiting the flexibility and the functionalities provided by the tool. Providing *business-level* analysis and monitoring involves instead the development of techniques to enable users to define, monitor, and measure business quality metrics. This section describes the main BPC concepts and the features that the tool provides in order to achieve this goal.

2.1 Focus points, perspectives and restrictions

The BPC visualizes process execution data (and related quality metrics) according to different *focus points.* A focus point identifies the process entity that is the focus of the analysis. For example, under the service focus, the user will be able to see statistics and metrics about the web services invoked during the execution of business processes. After discussing with consultants and customers, we have identified the following foci as being relevant in business-level analysis[1]:

- *Process*: displays information about a specific process or set of processes.
- *Resources*: shows data related to individual resources or group of human or automated resources.
- *Services*: displays data related to the web services invoked within the execution of processes.

For each focus point, the BPC can present basic statistical information (such as average execution time and performance), value-related information (associated revenues and costs), and quality information, such as resource performance and ratings.

For every focus point, the information can be analyzed according to different *perspectives*. A perspective defines a way to look at the information of the selected focus point. For example, looking at the *process* focus point under the *time* perspective, users can view process statistics by year, week, day of the week, etc. BPC includes the following perspectives:

---

[1] One could think at many other foci. These are designed with the purpose of achieving a compromise between expressiveness and ease of use.

- *Resources*: shows data depending on the resource that took part in the execution. For example, by looking at the service focus point under the resource perspective, users can service execution data aggregated depending on which resource (i.e., service provider) executed the service.
- *Services*: shows data aggregated depending on the service that has been executed.
- *Time*: shows data by time. This perspectives has several sub-perspectives, allowing users to view data aggregated by calendar or fiscal year, quarter, or month, by week, by day of the week, hour of the day.
- *Processes*: displays information aggregated by process.
- *Taxonomies*: this is the most innovative aspect of the BPC, and allows users to view data aggregated on the basis of user-defined taxonomical information. This aspect is described in more detail below.

A *taxonomy* is a user-defined way to classify instances of a process depending on its characteristics. Many taxonomies can be defined for the same process, and each taxonomy can have several *categories*. For each taxonomy, a process instance can be in one and only one category at any given time. For example, a taxonomy *outcome* may include the categories *success* and *failure*, while the taxonomy *performance* may include the categories *fast*, *acceptable*, *slow*. As an example, Figure 2 shows the distribution of the number of instances of a "Travel Expense Reimbursement" process among the categories of a *duration* taxonomy. The duration taxonomy has four categories: *fast* (defined as processes lasting less than 5 days), *acceptable* (between 5 and 10 days), *slow* (between 10 and 15 days) and *very slow* (more than 15 days). By using taxonomies, business users can analyze the aspects of the process in which they are interested and organize reporting data as they see fit, avoiding focusing on details that may have little relevance to them.

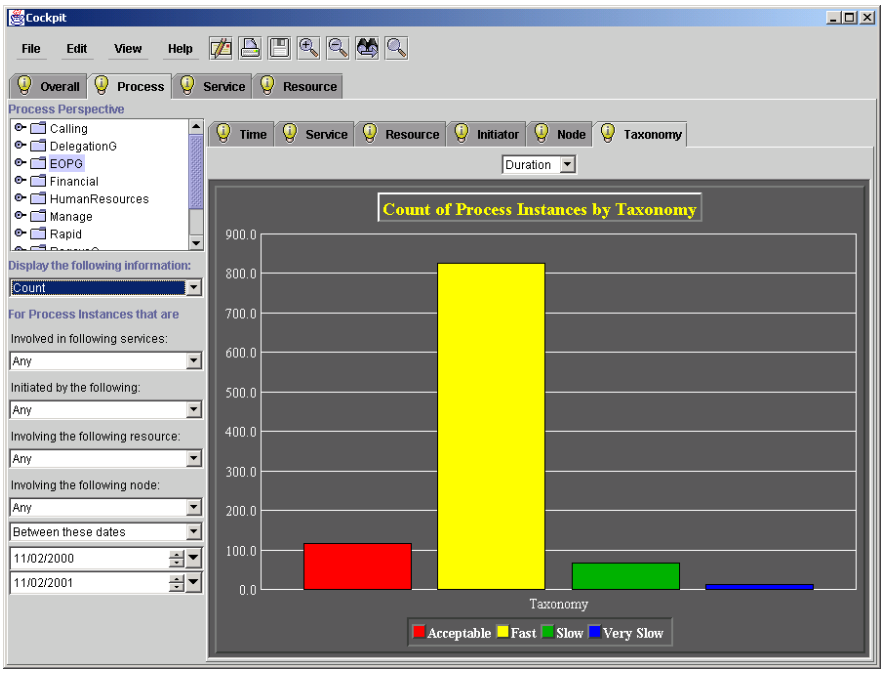

Figure 2 - Screen shot from Business Process Cockpit: *Process* focus and *Taxonomy* perspective

Categories are defined by instantiating a template. The template identifies a generic process behavior, such as: *Processes **P** in which a resource **R** has been involved in the execution*. A template can be then instantiated by specifying the parametric parts (P and R in this example). Complex categories may be defined by conjunction or disjunction of simple categories.
The BPC provides several templates, that can be instantiated by filling a form. In addition, the BPC can update its template library by downloading them from the BPC web site, analogously to how antivirus software updates their virus definition files. Templates are implemented by means of SQL statements that analyze data in the PDW and determine the categories to which a process instance belongs.

Figure 3 shows another example of knowledge that can be extracted by using the BPC. The figure shows the process instance duration depending on the resources that have been involved in process execution. The analyst can use this report to identify the cause of slow process executions.

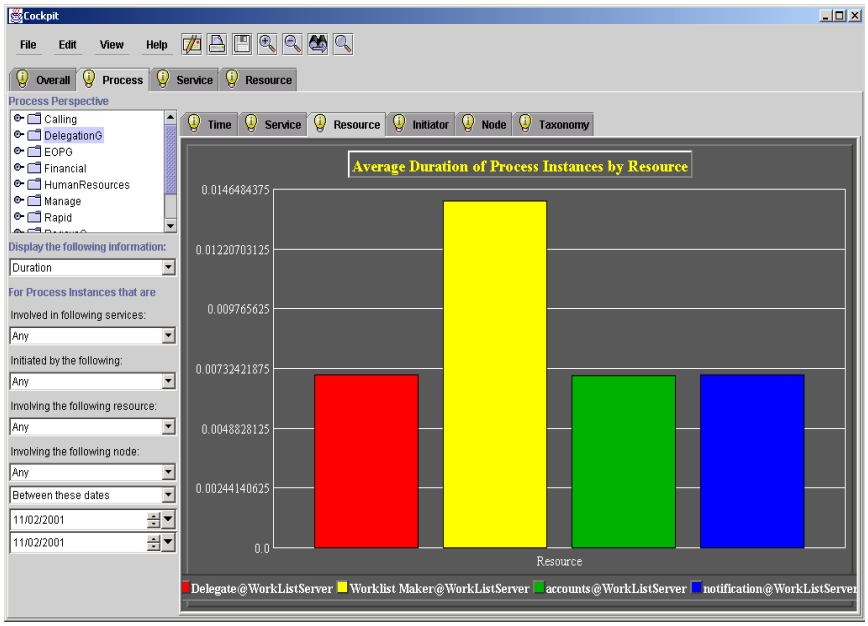

Figure 3 - Screen shot from Business Process Cockpit: *Process* focus and *Resource* perspective

As a final note on perspectives, we observe that not all perspectives are available for a given focus point. For example, the process perspective does not have meaning when observed under the process focus point.

Besides determining aggregation, the characteristics described in the perspective bullets above can be used to *restrict* the data considered in the analysis. For instance, the BPC can show data related to processes started within a given time window, or classified in certain categories, or that involved the execution of a certain service. The restriction attributes are exactly the same as the perspectives. We experienced that this symmetry proved to be helpful in improving the usability of the tool by business users.

2.2 Process monitoring and alerts

Besides providing reports on completed executions, BPC also monitor running executions, particularly to detect the occurrence of critical situations that must be corrected. For this purpose, BPC allows analysts to define *alerts*, i.e., situations that need to be detected and on which actions need to be taken. Alerts are defined in a way analogous to category, i.e., by instantiating templates. An example of an alert template is "processes **P** stopped at node **N** for more than **M** minutes". In addition, users can associate alerts to specific *actions*. For each alert, the following two actions can be defined:

- Send an email, page, or cellphone message to a specified user. The message is defined as a simple string. Depending on the practical feasibility, it may be possible to instead define it as a parametric string, with the parameters referring to the behavior condition values that triggered the alert (for example, for alerts on process duration being more than a certain threshold, it may be possible to include the actual duration in the message).
- Send an *event* to the process instance for which the alert has been detected. In fact, process definitions in HPPM can include event nodes, i.e., points in the process execution when the process stops (along the path where the node is defined) waiting for an event. If process designer want to take advantage of the BPC alerting feature, they can define branches that subscribe to alert events, and that perform suitable actions when the event is received, as shown in the sample process of Figure 4.
- Publish a message on a JMS bus, to be consumed by any interested enterprise management application that subscribed to published alerts.

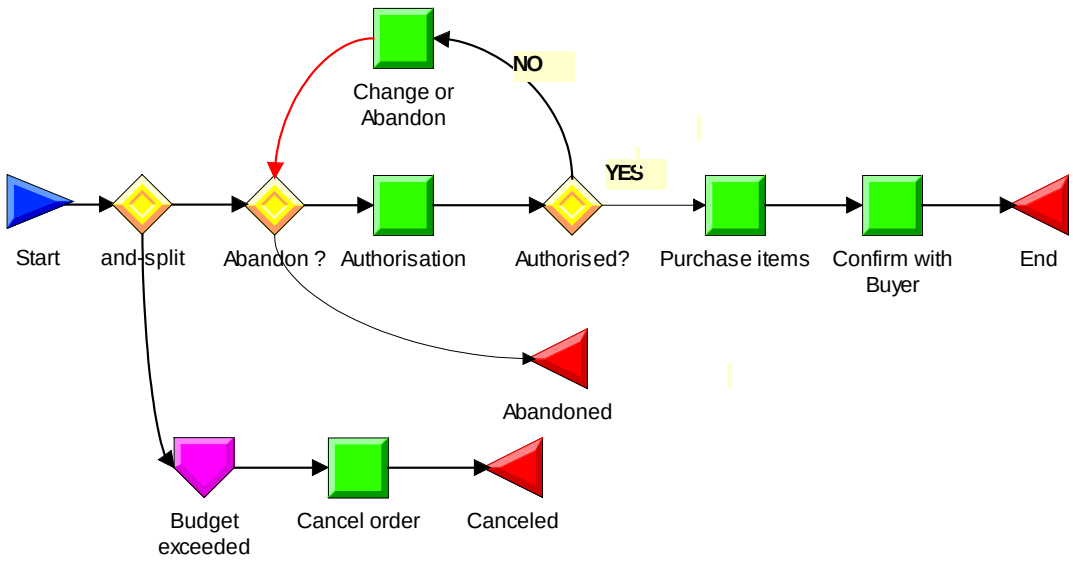

Figure 4 - process including an event node (represented by a pentagon) that monitors for the alert (budget exceeded) and cancel a purchase request if the alert is detected.

3 Architecture and Implementation

This section describes the architecture of the BPC and and outlines the features that have been implemented. Figure 5 shows the "external" architecture of the Cockpit, that is, how it interacts with HPPM and other management components. The cockpit reads "live" data about running process executions from the HPPM logs and it reads data about completed process executions from the PDW. The PDW also contains several kinds of aggregate data that could be used for display by the BPC. Databases are accessed through JDBC. Input could also come from JMS bus, for example if the BPC is used in conjunction with forthcoming OV products.

In terms of output, the BPC communicates (in addition to the users) with HPPM in order to optimize executions (for example tuning some parameters, such as the process priority). BPC also outputs to the warehouse, since it may be used to configure some aspects of the warehouse, and it outputs to JMS, for example to notify alerts (that could be captured by other componentsof the management/intelligence solution, such as the OV console, or by any other interested application).

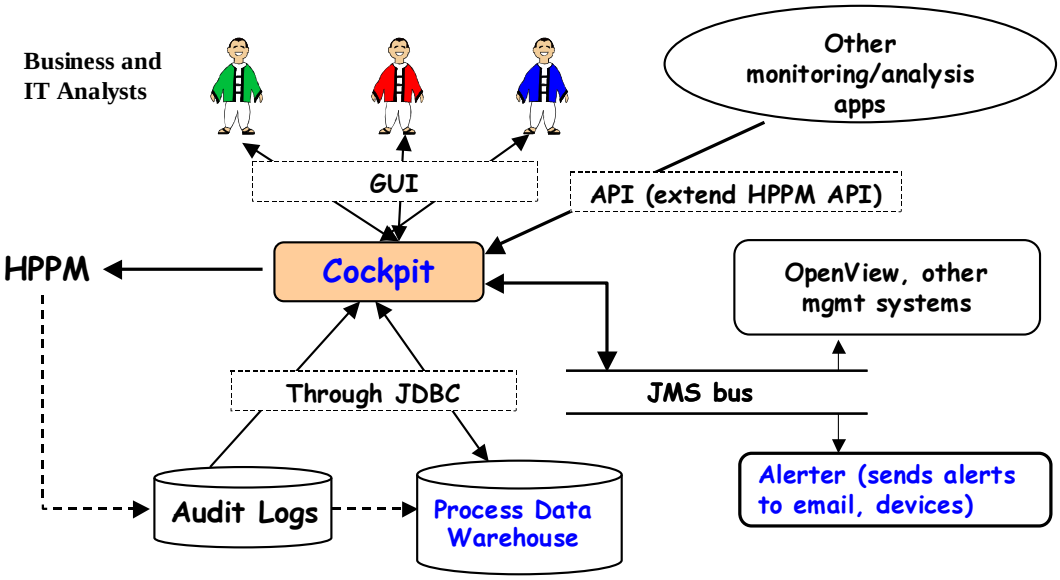

Figure 5 - External architecture of the Business Process Cockpit

BPC implementation consists of two main components: applet and servlet. The BPC applet can be run either as an applet within a web browser, or as a stand-alone Java application from an operating system's shell. When running within a browser, the BPC applet runs inside the standard sandbox of Java Virtual Machine. Therefore, it cannot directly communicate with the PDW and Audit Logs. The BPC servlet is used as an interface between the applet and those other components in that case. The

only job of the BPC servlet is to transmit the SQL queries and their results between the BPC applet and PDW and Audit Logs. When BPC applet is executed as a stand-alone program, it can directly communicate with PDW and Audit Logs through the Internet. BPC is implemented completely in Java, and therefore can be used virtually on any platform. The BPC applet can be run from any machine on the Internet. Similarly, the BPC servlet can be located at any machine on the Internet, but it is preferable to locate it close to the PDW and Audit Logs only for performance reasons.

References

[BCSD02] Angela Bonifati, Fabio Casati, Umesh Dayal, Ming-Chien Shan. Warehousing Workflow Data: Challenges and Opportunities. *Procs of VLDB'01*, Rome, Italy, Sept 2001.

[HPPM] Hewlett-Packard. HP Process Manager Process Design Guide. 2001.

[JCS02] LiJie Jin, Fabio Casati, Ming-Chien Shan. Business Process Simulation with HP Process Manager. *Procs. of the Collaborative Technologies Symposium*, San Antonio, Texas, January 2002 (to appear).