A More Comprehensive Approach to Enhancing Business Process Efficiency

Seung-Hyun Rhee[1], Nam Wook Cho[2], and Hyerim Bae[3,*]

[1] Department of Industrial Engineering, Seoul National University,
151-742, Seoul, Republic of Korea
shyun.rhee@gmail.com
[2] Department of Industrial and Systems Engineering,
Seoul National University of Technology,
139-743, Seoul, Republic of Korea
nwcho@snut.ac.kr
[3] Department of Industrial Engineering, Pusan National University,
635-709, Busan, Republic of Korea
hrbae@pusan.ac.kr

Abstract. Whereas Business Process Management (BPM) systematically guides employee participation in business processes, there has been little support, use or development of user-friendly functions to improve the efficiency of those processes. To enhance business process efficiency, it is necessary to provide automatic rational task allocation and work-item importance prioritization, so that task performers no longer need to be concerned with process performance. In the context of BPM, two different perspectives, the Process Engine Perspective (PEP) and the Task Performer Perspective (TPP), are considered. Accordingly, we developed a comprehensive method that considers those two perspectives, in combination rather than separately. We carried out simulation experiments to show the combinational effect of the two phases.

Keywords: Business Process Management, User-Oriented Support, Process Efficiency, Theory of Constraints, Dispatching Rule.

1 Introduction

Companies have come to perceive the importance of the process perspective in managing intra-organizational resources and inter-organizational relationships [3][6][10][11]. The advent of BPM (Business Process Management) is enabling them to employ their processes as the main backbone of the management of their resources and, through the linking of heterogeneous processes, to enter into partnerships involving Business Process Outsourcing and M&A [2][6][14].

* Corresponding author.

M.J. Smith, G. Salvendy (Eds.): Human Interface, Part II, HCII 2007, LNCS 4558, pp. 955–964, 2007.

Most notably, BPM enables even employees, as well as top executives, to easily trace the progress status of their business processes [15]. However, there has been little support, use or development of user-oriented functions in improving the efficiency of those processes. As explained in this paper, we devised a methodology for providing human-oriented features in the context of BPM. Such innovations ultimately lead to the improvement of process efficiency.

Process execution efficiency has been dealt with according to two different perspectives: one is task allocation to participants, and the other, the priorities of the assigned tasks. However, it should be noted that in order to operate business processes efficiently, those two perspectives must be considered together. This paper proposes a comprehensive methodology of process efficiency that combines the two approaches.

The paper is organized as follows. Section 2 explains process efficiency in the context of BPM. Section 3 discusses previous research on process efficiency. Sections 4 and 5 describe, from the BPM-engine- and user perspective, respectively, methodologies for efficient process execution. Section 6 provides an integrated methodology for business process efficiency. Section 7 offers the results of a simulation study to validate our overall concept.

2 Process Efficiency

As BPM functions become more effective and the number of processes managed by BPM increases, Business Process (BP) efficiency issues are beginning to attract more attention [1][17][19]. BP efficiency can be considered from two different perspectives: the Process Engine Perspective (PEP) and the Task Performer Perspective (TPP).

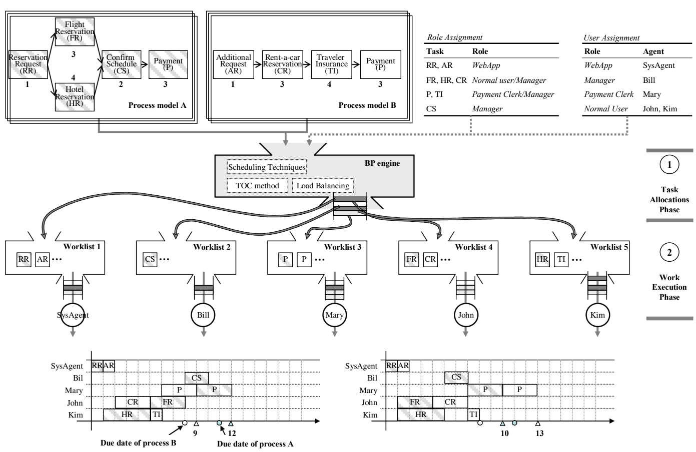

Fig. 1. Overall BP management procedure using BPM

Figure 1 illustrates the overall procedures of BP execution in a BPM system, showing the two perspectives. First, the PEP is the task allocation of the BPM system engine, a core function of BPM. After process models are prepared, the engine starts a process instance based on those models [2]. During process execution, the engine assigns tasks to proper users, using delicate rules [17], and these rules have a great influence on the efficiency of the processes [9]. For example, the FR task, which is assigned to John, can also be assigned to Bill. Once assigned, the FR task can start as early as time point 1, and thereby overall process efficiency, which is shown at the bottom-left of the chart, can be improved.

Second, BP efficiency can vary depending on how a user executes tasks assigned to him. In dealing with the assigned tasks, the priority defined by a user can influence the efficiency of the whole process. The two charts at the bottom of Fig. 1 show the two results from applying two different priority rules in executing tasks on worklists. As can be seen, the user's priority assignment affects overall efficiency by affecting the process completion time.

With respect to BP efficiency, another important issue is the means by which efficiency is measured [2]. A proper strategy has to be considered according to the kind of efficiency desired. The left result in Fig. 1 is more efficient in process completion time. Alternatively, if due-date is the important measure, the right result is better.

3 Related Work

A lot of research has aimed at improving BP efficiency through the use of scheduling technologies, or of frameworks specifically designed for smooth process routing. Baggio *et al.* [1] has provided a method to minimize the number of late jobs in a workflow by applying scheduling techniques. Chang *et al.* [4] proposed the identification and analysis of critical paths to manage time and resources within a workflow process. Zhao *et al.* [19] introduced a concept to predict the turnaround time of a time-driven process and to allocate the expected processing time to each activity in the process. In Kumar's study [12], a general framework was devised for efficient workflow management.

More specific research on task distribution and priority setting has been conducted as well. Ha *et al.* [9] developed a process execution rule that enables the balancing of the workload of agents, each of whom has a worklist. Rhee *et al.* [16] developed a TOC (Theory of Constraints) -based process execution method that contributes to the enhancement of efficiency by controlling task allocation to an overloaded participant and synchronizing process instance release with his/her work pace. Kumar *et al.* [13] dealt with the problem of the trade-off between observing a deadline and offering work items to an overloaded participant at run-time, and proposed a systematic approach to dynamically creating a balance between quality and performance issues in workflow systems.

Regarding priority setting, Eder *et al.* [7] employed a personal schedule that provides information on future work. They showed that, with this information, both the turn-around time and time-constraint violation rate can be decreased. Rhee *et al.* [17] calculated the slack time of tasks in workflow processes. Slack time guides task performers in processing urgent tasks first, which eventually improves BP efficiency. Whereas the previous research has addressed either the PEP or the TPP, this research comprehends both perspectives.

4 Process Engine Perspective (PEP)

As for the PEP, we adopted the TOC [8]-based process execution method developed by Rhee [16]. The TOC method is discussed in detail in the following sections.

4.1 Drum

We introduce BP-Drum, Buffer, and Rope (BP-DBR), modified from the original DBR in the TOC concept. Our method emphasizes the situation in which a specific resource is relatively limited and, therefore, can determine the performance of the entire system. We define a Constraint Capacity Resource (CCR) as a task performer who has the largest workload on his worklist.

First, the Drum is defined as the pace of the CCR. In order to calculate the Drum, two different cases are taken into account. The first case supposes that a CCR user (u_{CCR}) belongs to a workgroup with k ($k \geq 0$) users, and the workgroup processes a single task, a_i, within an expected processing time of ET_i. Since one agent can process $1/ET_i$ tasks per unit time, the Drum can be

$$\text{Drum } (d) = \frac{k}{ET_i} \text{ (task/time).} \tag{1}$$

In the second case, u_{CCR} can deal with multiple tasks in the respective processing times. To deal with this type of case, the concept of unit task was introduced. For example, suppose that a task a_1 has the smallest process time of 1, and that a task a_2's processing time is 2. Then, processing of a_2 corresponds to processing of two unit tasks. Therefore, for a BP-CCP responsible for J tasks, a unit task of a_i can be determined as

$$\text{Unit number of task } a_i \, (UN_i) = \frac{ET_i}{ET_{\min}} . \tag{2}$$

ET_i: Expected Time of a_i, $ET_{min} = \min\{ET_i : j=1,2,\ldots,J\}$

Let P_i denote the probability that a_i be assigned to u_{CCR}; therefore, the Drum can be

$$\text{Drum } (d) = \sum_{i=1}^{J} P_i \cdot \frac{1}{ET_i} \cdot UN_i \text{ (task/time).} \tag{3}$$

4.2 Rope and Buffer

The Rope is defined as a communication tool that controls the speed of input according to the Drum. A BPM engine plays a role as the Rope by controlling the release of a process instance into a system. The BPM engine synchronizes the process instance release rate (λ) with the Drum.

$$\lambda = d \text{ (task/time)} \tag{4}$$

Figure 2 illustrates the function of the Rope. The Rope in the BPM engine controls the entrance of instances so that the process release rate, which can be the instance arrival rate, equals the pace of the CCR.

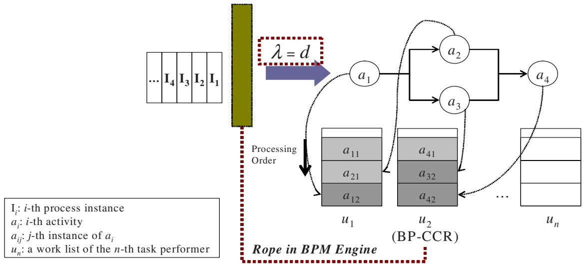

Fig. 2. The Function of the Rope

In the TOC, the Buffer protects the CCR from work starvation, guaranteeing continuous operation of the entire process. In order to achieve maximum utilization in a TOC environment, a system requires a sufficient amount of Buffer, whereas it also tries to minimize inventories and operating cost. In this paper, we suggest a simulation method to determine the size of the Buffer. An overall BP-DBR procedure is presented in Fig. 3.

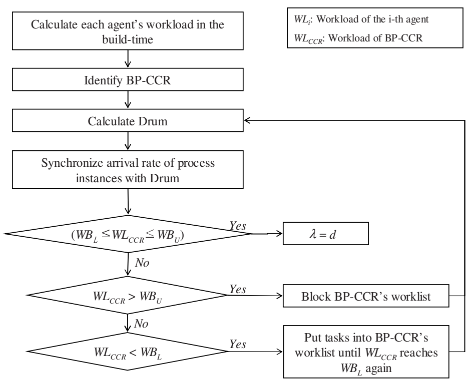

Fig. 3. BP-DBR Procedure

The application of the BP-DBR involves the assignment of tasks to agents through the Drum, Buffer and Rope, with special attention devoted to the BP-CCR. The maximum and minimum workload bounds (*WB*) are introduced in order to control business processes: if a Buffer size is greater than the maximum workload bound (WB_U), tasks are no longer assigned to the BP-CCR or are assigned to an alternative agent, if any, until the Buffer size falls below the WB_U. The minimum workload bound (WB_L) is also defined to maintain the minimum amount of Buffer.

5 Task Performer Perspective (TPP)

This chapter discusses BP efficiency from the TPP. In the present study, Rhee's method [17], was employed to prioritize tasks assigned to a user, based on the

PERT/CPM method. It increases BP efficiency by enabling users to deal with an urgent task earlier.

5.1 Slack Time Calculation

In order to furnish urgent information for each task, a critical path should be found and slack time should be computed. Computation of the slack time of each task is based on the PERT/CPM method. Whereas a split in PERT/CPM networks always entails use of the AND semantic, BP structures have an alternative path (that is, an OR block is included). In a BP model, the AND parallel structure can be handled in exactly the same way as the PERT/CPM method. However, for the OR structure, the multi-path is consolidated as a single path by a representative activity shown in Fig. 4.

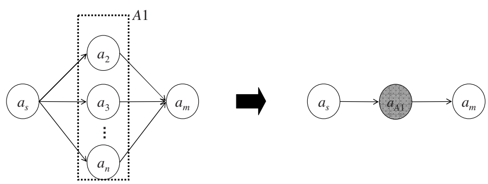

Fig. 4. A representative activity of an OR structure

Prior to finding the critical path and computing the slack time of tasks in an OR structure, the expectation time of each path is required. The expectation time is calculated for three representative types of OR structure; those calculations are summarized in Table 1.

Table 1. Expected processing time of each OR type

Type	Expected processing time	P_i (λ_i)
NOR	$\sum_{i=1}^{n}\prod_{j=1, j\neq i}^{n}(\frac{\lambda_i}{\lambda_i+\lambda_j})ET_i$	λ_i: The service rate with which the distribution of the actual processing time of the *i*-th path follows an exponential distribution
POR	$\begin{cases} P_1 \cdot ET_1 & , n=1 \\ P_1 \cdot ET_1 + \sum_{i=2}^{n}[\{\prod_{j=1}^{i-1}(1-P_j)\}P_i \cdot \{\sum_{k=1}^{i} ET_k\}] & , n \geq 2 \end{cases}$	P_i**:** The probability that, when the *i*-th path, r_i, is determined to be executed, the path is successfully completed
COR	$\sum_{i=1}^{n}(P_i \cdot ET_i)$	P_i**:** The probability that r_i is chosen for execution

With the representative activity of an OR structure, a slack time can be calculated for the structure, and each user prioritizes tasks according to this information. He first carries out an urgent task with the least slack time. We call this rule the Least Slack Time (LST) rule. More details on this rule can be found in [17].

6 Combination of the Two Perspectives

The two-phase perspectives, comprising the PEP and the TPP, have different target areas in improving process efficiency. Whereas the BP-DBR focuses on the engine's function of allocating and distributing tasks, the LST concentrates on fulfilling the assigned tasks. Although each method has been proved to be effective in each phase, the effect of combining the two phases has not yet been considered. The two-phase perspective covers an entire procedure from process launching to execution. We call the combined method *Comprehensive Rule for Process Efficiency* (CR4PE).

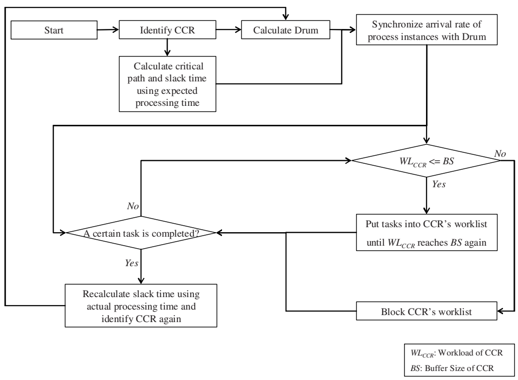

Fig. 5. Comprehensive Rule for Process Efficiency

The overall procedures of the method are illustrated in Fig. 5. Before a process is started, the CCR, the critical path and the slack time of each activity are identified. The Drum is also calculated. Once a process instance is launched, the Rope starts controlling the release of instances. Buffer management is also executed considering the Buffer size of the CCR. The Buffer is controlled with regard to the CCR's workload. In the course of executing a process instance, the completion of a certain task can result in changes to the CCR, the Drum, and the slack time. Whenever a certain task is finished, those data are recalculated in real time.

7 Simulation Experiments

7.1 Simulation Model and Experimental Environment

The BP-DBR is known to be effective when a heavy workload is concentrated on a few activities [16]. By contrast, the LST rule is effective when there is a heavy workload for many activities in parallel relations [17]. For simulation experiments, we used the process model shown in Fig. 6, which model includes features of both perspectives. As illustrated in the Figure, each box represents an activity, and the value inside a box represents an expected time. Participants are marked above each activity. A colored box is an activity with the heaviest workload and the activity's user, u_{25}, is a CCR.

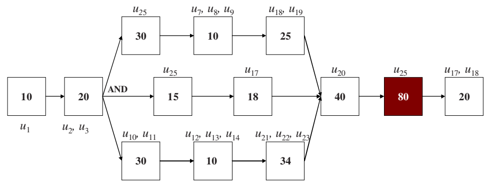

Fig. 6. Simulation Model

The simulation model was implemented with Arena 6.0. Each experiment was warmed up for 500 minutes and run for 10,000 minutes. The arrival rate of process instances with a parameter of 40, and the processing time were assumed to be exponential. The Buffer size was set to 1,800. Finally, thirty simulation experiments were carried out for each the following four models.

· I: Model with **FIFO Rule** · II: with **TOC** · III: with **LST Rule** · IV: with **CR4PE**

7.2 Experimental Results

We measured the Process Completion Time (PCT), the Number of Complete Instances (NCI) and the Number of Ongoing Process Instances (NOCI). The simulation results in Table 2 show that the CR4PE method is the most effective for the PCT and the NCI, which means that the combination of the TOC and the LST was more effective than the respective applications. However, the synergy effect was relatively less effective for the NCI, mainly because the TOC controls a number of instances by using the Drum and the Rope. As can be seen in Table 2, the NOCI of Models II and IV are much smaller than those of Models I and III, which means that in Model IV, a smaller number of instances are assigned to each task performer, and the effectiveness of the LST is relatively limited. We can still state, however, that Model IV is better than Model III, since in Model IV, each transaction can be completed in a shorter time.

Table 2. Summary of BP efficiency results

Model	PCT	NCI	NOCI
Model I	4223.30	58	108.27
Model II	2995.95	77	28.04
Model III	3780.85	80	96.66
Model IV	2747.46	81	26.66

7.3 Influence of Buffer Size

As discussed in Section 7.2, the effectiveness of the LST can be limited by the number of job instances on participants' worklists. To further study the effect of the Buffer, additional simulations were conducted with respect to different Buffer sizes.

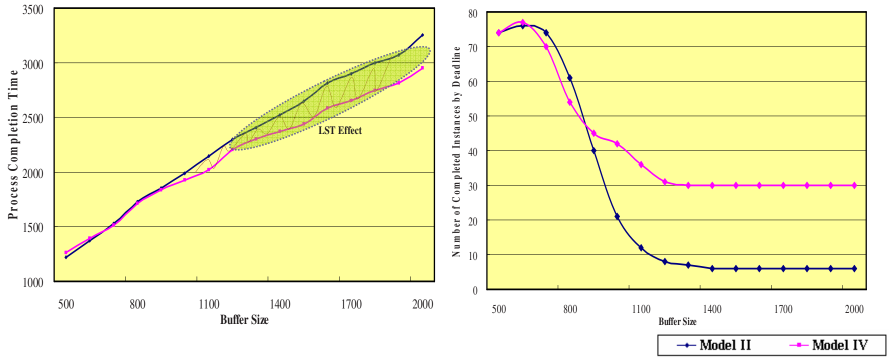

Fig. 7. Influence of Buffer Size

Figure 7 shows that, for PCT and NCI by Deadline, Model IV performs better than Model II as the Buffer size increases. The difference between the two models originates from the effectiveness of the LST Rule. Since a CCR has a greater number of job instances assigned with a larger Buffer size, the LST plays a more important role. The LST would be ineffective if a task performer had few jobs on his worklist. Therefore, the synergy effect becomes greater as the size of the Buffer increases.

8 Conclusions

Having perceived the absence of user-oriented support for BP efficiency, we devised a more comprehensive approach to enhancing it. Two different perspectives on BP efficiency, the PEP and the TPP, were identified. Based on our previous research in a single perspective, we combined both perspectives into an integrated model. According to the simulation results, the new methodology shows better performance than utilization of an individual perspective.

Our research has raised issues for future study. First, more process model patterns can be considered in our CR4PE. Second, whereas this research adopted one rule for each perspective, it would be interesting to analyze various rules for each perspective. Finally, if the economic values of process efficiency improvement are measured [5], we can acquire insight into the extent of the benefits our integrated methodology can guarantee compared with single-handed administration.

Acknowledgments. This work was supported by the Seoul R&BD Program. It was also supported by a Korea Research Foundation Grant funded by the Korean Government (MOEHRD) and the Regional Research Universities Program/Research Center for Logistics Information Technology.

References

1. Baggio, G., Wainer, J., Ellis, C.: Applying Scheduling Techniques to Minimize the Number of Late Jobs in Workflow Systems. In: The 2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, pp. 1396–1403 (2004)
2. Basu, A., Kumar, A.: Research Commentary: Workflow Management Systems in e-Business. Information Systems Research 13(1), 1–14 (2002)
3. Buzacott, J.A.: Commonalities in Reengineered Business Processes: Models and Issues. Management Science 42(5), 768–782 (1996)
4. Chang, D.–H., Son, J.H., Kim, M.H.: Critical Path Identification in the Context of a Workflow. Information and Software Engineering 44(7), 405–417 (2002)
5. Davamanirajan, P., Kauffman, R.J., Kriebel, C.H., Mukhopadhyay, T.: System Design, Process Performance, and Economic Outcomes: An Empirical Study of Letter of Credit Systems Integration in International Banking. Journal of MIS 23(2), 65–90 (2006)
6. Davenport, T.: The New Industrial Engineering: IT and Business Process Redesign. Sloan Management Review, 11–27 (1990)
7. Eder, J., Pichler, H., Gruber, W., Ninaus, M.: Personal Schedules for Workflow Systems. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 216–231. Springer, Heidelberg (2003)
8. Goldratt, E.M.: The Goal. North River Press, New York (1992)
9. Ha, B.–H., Bae, J., Park, Y.–T., Kang, S.–H.: Development of process execution rules for workload balancing on agents. Data. & Knowledge Engineering 56(1), 64–84 (2006)
10. Hammer, M.: The Agenda: What Every Business Must Do to Dominate the Decade, Crown Business, New York (2001)
11. Kaplan, S., Sawhney, M.: E-Hubs: The New B2B Marketplaces. Harvard Business Review 78(3), 97–103 (2000)
12. Kumar, A., Zhao, L.: Dynamic Routing and Operational Controls in a Workflow Management System. Management Science 45(2), 253–272 (1999)
13. Kumar, A., van der Aalst, W.M.P., Verbeek, H.M.W.: Dynamic Work Distribution in Workflow Management Systems: How to balance quality and performance? Journal of MIS 18(3), 157–193 (2001)
14. March, S., Hevner, A., Ram, S.: Research Commentary: An Agenda for Information Technology Research in Heterogeneous and Distributed Environments. Information Systems Research 11(4), 327–341 (2000)
15. McCoy, D.: The convergence of BPM and BAM. Gartner Research Note (2004) SPA-20-6074
16. Rhee, S.–H., Bae, H., Seo, Y.: Efficient Workflow Management through the introduction of TOC concepts. In: The 8th Annual International Conference on Industrial Engineering Theory, Applications and Practice, USA (2003)
17. Rhee, S.–H., Bae, H., Kim, Y.: A dispatching rule for efficient workflow. Concurrent Engineering – Research and Applications 12(4), 305–318 (2004)
18. Smith, H., Fingar, P.: Business Process Management – The Third Wave. Meghan-Kiffer Press, Florida (2003)
19. Zhao, J.L., Stohr, E.A.: Temporal workflow management in a claim handling system. SIGSOFT: Software Engineering Notes 24(2), 187–195 (1999)