한국정보시스템학회(KAIS)
03 추계학술대회 논문집, pp. 55~63

Business Process Efficiency in Workflows using TOC

Hyerim Bae [1*], Seung-Hyun Rhee[2]

[1]Department of Internet Business
Dongeui University
Pusan Korea 614-714
*Corresponding author's email: hrbae@dongeui.ac.kr

[2]Department of Industrial Engineering
Seoul National University
Seoul, Korea 151-742

Abstract: Workflow Management System (WFMS) is a software system to support an efficient execution, control and management of complex business processes. Since traditional commercial systems mainly focus on automating processes, they don't have methods for enhancing the task performer's efficiency. In this paper, we propose a new method of executing business processes more efficiently in that a whole process is scheduled considering the degree of the participants' workload. The method allows managing the largest constraints among constituent resources of the process. We utilize DBR scheduling techniques to develop the method. We first consider the differences between workflow process models and DBR application models, and then develop the modified drum, buffer and rope. This leads us to develop WF-DBR (WorkFlow-DBR) that can control the proper size of the task performers' work list and arrival rate of process instances. Use of WF-DBR improves the efficiency of the whole process as well as the participants' working condition. We then carry out a set of simulation experiments and compare the effectiveness of our approach with that of scheduling techniques used in existing systems.

Keywords: Workflow, Business Process, TOC

1. INTRODUCTION

Most companies recently do business with other companies as well as within themselves. These situations result in more complex business processes for the companies. Consequently, it has been conceived that efficient management of the complicated processes is essential for strengthening competitiveness. The companies expect that Workflow Management System (WFMS), which manages a business process efficiently and effectively, can provide solutions for the management. WFMS is a software system to automate and manage a business process (Hollingsworth, 1994).

Existing WFMSs execute a business process in the right order exactly following the process model. However, it cannot be assumed that the automation of a process is nothing but efficient management. Task performers usually participate in a number of different processes, and oftentimes deal with numerous process instances that belong to an identical process model. It is typical that each participant deals with multiple work items at any point in time. Each participant's work can have an impact on the execution of a whole process as well as other participants' existing work. Therefore, WFMS necessitates consideration for the task performer's efficiency as well as monitoring the participants' existing work.

In this paper, we propose an efficient method of process management from the viewpoint of both process and task performer. In order to do this, we employ DBR (Drum-Buffer-Rope) in TOC (Theory of Constraints), a concept for business innovation. However, since manufacturing processes typically utilized in DBR technique are different from the workflow processes in many ways, this technique cannot be utilized directly. In this research, we devise a modified DBR that allows focusing on the largest constituent constraints of a process for the improvement of both process and participant efficiency. The proposed approach has been simulated to demonstrate its effectiveness.

2. WFMS & TOC

Workflow is a business process, automated in whole or part, during which documents, information or tasks are delivered to an appropriate task performer according to a set of procedural rules (Hollingsworth, 1994). WFMS is a software system to execute, control and manage the workflow automatically (Lawrence, 1997). For the automatic execution of a process, it is prerequisite to represent the process in computer-understandable form. The representation is called '*process model*' (Rhee, 2003). There is a large body of literature on process models, which readers can refer to (Kim, 2002), (Rhee, 2003) for further details. The process model specifies the tasks and the relationships among tasks for the achievement of business processes' goal (Lawrence, 1997).

Generally, a process model is considered in '*build-time*' and '*run-time*' respectively. The build-time is a stage when a process is defined with of the constituent tasks of the process, the precedence relationships among them and the attributes for the performance of each task. In the run-time, actual processes are executed, controlled and managed by interpreting the process model prepared in the build-time (Rhee, 2003). The portion of WFMS responsible for the run-time is called '*workflow engine*' (Lawrence, 1997). Based on the process model, the workflow engine can create a process that will actually be performed, called the '*process instance*' (Hollingsworth, 1994).Once a process model is prepared, process instances can be generated repeatedly whenever necessary.

Once a process model is launched to automate actual processes, WFMS interprets these to allocate tasks to participants, so that the participants can perform the tasks assigned. The task allocation is usually carried out using some rules. The task allocation rules can be broadly categorized by group standards and personal standards, and referred to (Rhee, 2003) for further details. Allocated tasks by the task allocation rules, task performer possess some degrees of their workload. The workload has an influence on the performance of the processes.

TOC proposed by Dr. Goldratt emphasizes the systematic management of a whole process by discovering constraints hindering the achievement of the process' goal and conquering them (Goldratt, 1992). Doing this requires increasing throughput, reducing inventory and cutting down operating expense (Radovilsky, 1998). It utilizes CCR (Constraint Capacity Resource) thoroughly to satisfy the goals (Duclos, 1995). In TOC, DBR has been introduced to various kinds of manufacturing processes' scheduling. The concept of drum, buffer and rope is originated from a method of managing an army on the march, and has been extended to manage manufacturing processes. Each concept in the processes is defined as following (Daniel, 1997), (Duclos, 1995): Drum is the pace of production in the system determined by the constraint. Buffer is an inventory or a time offset to protect throughput of the system by maintaining a supply of material available for production at the constraint. Rope is a communication device to limit or choke the flow of material into the system to match the actual production of the constraint

Fundamentally, DBR is based on the systematic approach of TOC. For the application of DBR to manufacturing processes, it is requisite to determine the CCR first. The CCR is generally a resource with the highest workload among those resources whose workload is more than 100% (Goldratt, 1992). Once CCR is identified, the resource is thoroughly utilized to improve the productivity of the manufacturing processes (Duclos, 1995). If the processes are delayed, it is necessary to disperse the CCR's workload or raise the CCR's capacity. In this paper, we propose an efficient method of managing a workflow process from the viewpoint of TOC concepts.

3. CONSIDERATIONS FOR THE INTRODUCTION OF DBR

A workflow environment is different from a manufacturing process environment. First, their structures are different with each other. The workflow process includes every possible alternative path, so that it can be executed automatically and exactly. On the other hand, the manufacturing process expresses only the paths to actually execute. While order of manufacturing process is fixed by forecasting or calculating the proper LOT size, WFMS focusing on the automation of processes cannot fix process order. The LOT size in the manufacturing process is generally one or more, but that in the workflow is one, a process instance, which is distinguished from another instance. In general, the resources in the workflow are task performers, while in the manufacturing process there are various resources including facilities, participants and so on. Consideration needs to be given to these differences.

In this paper, for ease of understanding DBR technique, we ignore the situation where market demands lack. In other words, a lot of instances exist at any time.. In this situation, it is usual that each task performer takes part in one or more work items (Kim, 2002). This implies that an individual performer can have a multiple number of task instances at any point in time.

In this paper, we define the CCR in a workflow environment, the ' *W-CCR*', as a task performer with the largest workload, who affects the completion of ongoing instances. From the viewpoint of a participant, when a task instance is delayed, other task instances allocated to a work list inevitably have to wait. This results in the delay of process instances. In other words, the size of a work list determined by the task performers' ability has an influence on the completion of ongoing process instances.

4. WF-DBR

Taking into account the aforementioned considerations, we propose WF-DBR (WorkFlow-DBR), which aims at improving both workflow process and participant efficiency through the DBR technique. This section explains how WF-DBR is utilized in WFMS.

We define the W-CCR first to run the method. Since each instance is unique, W-CCR for an instance is of no significance. To identify W-CCR for a multiple number of instances, the term when multiple instances are generated is required. To discover W-CCR, it is prerequisite to calculate the workload of each task performer's work list. The workload of the i-th task performer's work list, WL_i, is calculated as the summation of the expected processing time of each task allocated to the task performer.

$$WL_i = \sum_{j=1}^{J} ET_j \tag{1}$$

ET_j : Expected processing time of the j-th task, J: Number of tasks allocated to the i-th task performer

Then, W-CCR becomes a task performer whose workload is WL_R.

$$WL_R = max\{ WL_i | \, i \text{ is the } i\text{-th participant in the process, } i=1, \ldots, I\} \tag{2}$$

I: Number of task performers who take part in the process

After that, we can calculate the drum, which is the total number of processed task instances over the total time when W-CCR participates in the process. When W-CCR is responsible for m tasks, the drum is defined as follows:

$$\text{Drum} = \frac{\sum_{j=1}^{m} \frac{ET_j}{ET_{\min}} \cdot n_j}{\sum_{j=1}^{m} PT_j \cdot n_j} \tag{3}$$

ET_{min}: The minimum value among ET_j, PT_k: Actual processing time of the k-th task

n_j: Actual process instance number of the i-th task

After the drum is calculated, the arrival rate of a process instance can be determined. The workflow engine generates a process instance considering the drum. In the run-time, the drum at any point in time can be updated recalculating PT_j, and n_j in equation (3).

Once process instances begin to be generated, we can check the progress of a whole process through the W-CCR's workload. When the W-CCR's workload gets large, this implies that the whole process becomes delayed. When the workload gets small, it means that completion of process instances increases. It is necessary to manage the W-CCR's work list by monitoring the workload continuously. In this paper, we specify Upper Standard Size (USS) and Lower Standard Size (LSS) of WL_R to protect W-CCR from retarding process' progress and the starvation of process instances respectively. If WL_R exceeds the USS, the workflow engine temporarily stops allocating task instances to W-CCR. When WL_R falls below the USS again, the arrival rate of process instances is synchronized with the drum at this time. On the other hand, if WL_R is less than the LSS, the engine raises the arrival rate until the workload reaches the LSS again, and synchronizes it with the drum at this point time. The WF-DBR process is represented in Figure 1.

WF-DBR leads us to use a new task allocation rule. In this paper, we propose the " *WL-TAR*" (Work List-Task Allocation Rule), which is an allocation rule to disperse the tasks to other participants which originally would have been assigned to the W-CCR. The WL-TAR is used to complement existing allocation rules applied to task performers, that is, it is not utilized independently. The existing rules are supplemented with the WL-TAR when WL_R is more than the USS. At this time, tasks are assigned to other task performers expect the W-CCR. Existing WFMSs support similar allocation rules to the WL-TAR. The rules, however, assigns tasks ignoring the pace of the W-CCR and comparing workloads only within a group, so that it doesn't consider the workload of a whole process. On the other hand, the WL-TAR specifies the USS of WL_R and prioritizes the W-CCR's tasks.

We can expect improvements as follows. First, concentrating on the important resources for the progress of a process, the WF-DBR will increase the process efficiency. Second, the W-CCR's workload is monitored and controlled, so that it is expected to stabilize the workload. Finally, since task performers except W-CCR are subordinated to the W-CCR, we can predict that the non W-CCRs prioritize tasks which actually contribute to the progress of a process. This can enhance the efficiency of task performers. These expectations are examined by carrying out simulation experiments in the following section.

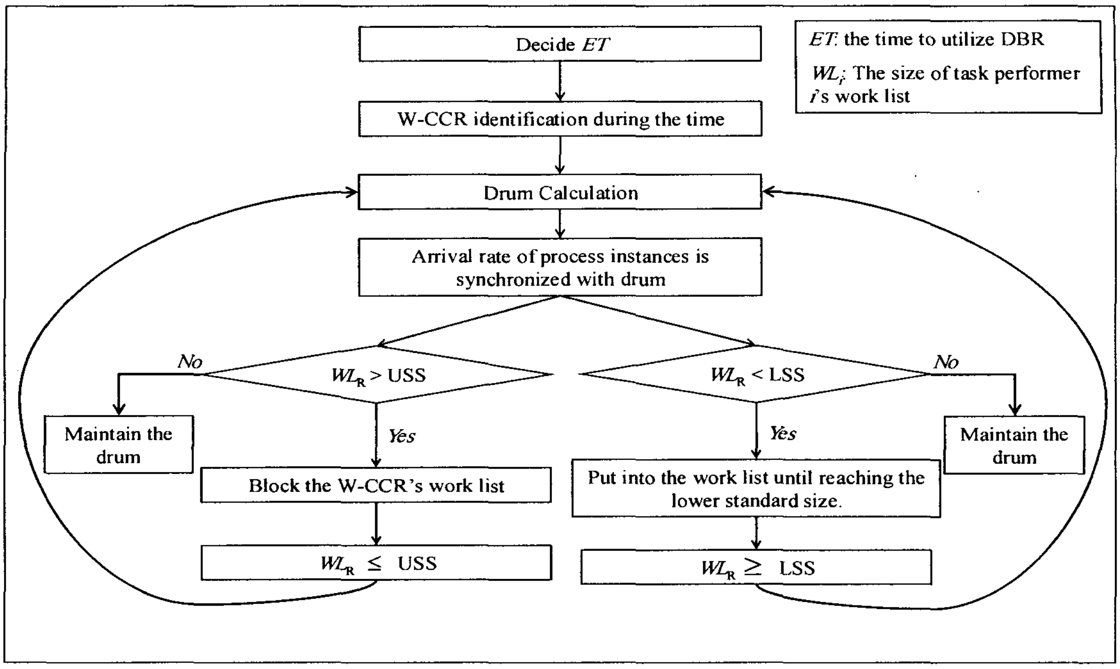


Figure 1. WF-DBR

5. SIMULATION EXPERIMENTS

We have carried out simulation experiments to verify the usefulness of WF-DBR. For the experiments, a process model is prepared. The process with the WF-DBR (WF-DBR process) is compared with the identical process without WF-DBR (non WF-DBR process). The effects of the method are analyzed. Existing WFMSs don't provide standards about prioritization of tasks in each participant's work list. Generally, a task performer prioritizes a task assigned first among tasks in his work list. In our experiments, all the participants in both processes use the general prioritization standard, FIFO.

We use Arena 6.0 for the simulation. For the example process, an Arena simulation model is prepared, and the probabilistic distribution functions of the processing time for each activity and of the arrival rate for process instances are determined. In our experiments, it is assumed that the processing time of a task and the arrival rate of process instances follow an exponential distribution (Rhee, 2003). Each simulation scenario has been simulated during 96,000 replication time to determine the W-CCR, and this has been repeated 30 times. The length of warming period is set to 3,000, which is about the time when the total number of ongoing process instances becomes constant. The USS and the LSS of the W-CCR's work list is 600 and 400 respectively.

5.1. Process Efficiency

We identify from empirical data that the W-CCR is task performer 7 whose average workload is largest. The workload is 539.47 for the non WF-DBR process and 203.55 for the WF-DBR process. The result implies that WF-DBR controls the release of tasks to W-CCR's work list and encourages the progress of the example process by dispersing tasks to other task performers. On the other hand, for task performers who are not W-CCR, the workload in WF-DBR process has no difference from that in non WF-DBR process. This is because they had capabilities to process the additional, dispersed tasks.

We verify the effectiveness in terms of process efficiency. The process efficiency is measured in two ways: as the average completion time of process instances (PE1), as the average number of process instances completed during a simulation run (PE2). The results are further investigated using the t-test, where the significance level is set to 0.05. The experimental results are summarized in Table 1.

Table 1. Analysis about process efficiency

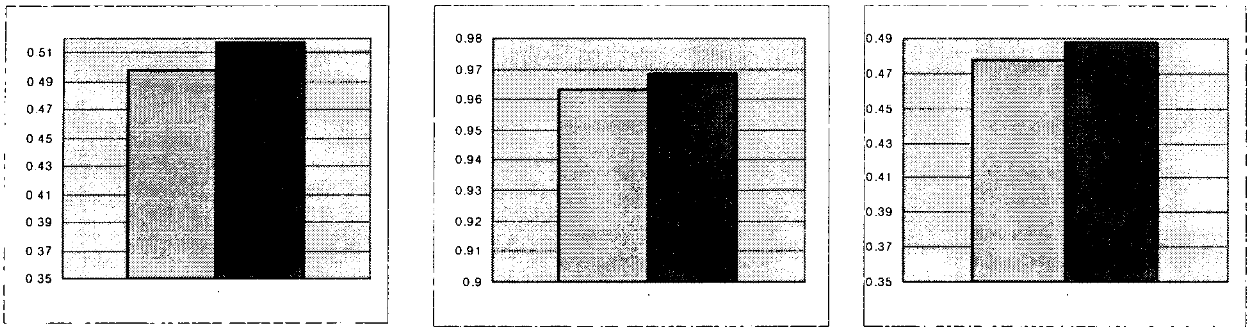
Process Efficiency	**PE1**	**PE2**
non WF-DBR process	1,190.95	743.79
WF-DBR process	758.30	828.27
t statistics* (critical value)	8.52 (1.70)	3.69 (1.69)

* The hypothesis of the t-test is 'H_0:(PEi of a process with the larger value)-(PEi of the other process) = 0 H_1:(PEi of a process with the larger value)-(PEi of the other process) > 0'. i = 1, 2

Table 1 shows that WF-DBR is effective from the viewpoint of process efficiency. WF-DBR gives a 36% reduction in the average process completion time and a 10% increase in the average number of completed process instances. Besides, the results of the t-test show that the improvements are statistically significant.

5.2. Task Performer's Efficiency

Our simulation results are analyzed in terms of the utilization of task performers. The utilization is the rate of actual working time over the total amount of available time. We expected that WF-DBR would have some influence on the utilization. To demonstrate this expectation, we collected the utilization for every performer. The results from only a few of typical task performers are provided in Figure 2. For the other task performers, we could observe almost the same results. In the graph, the results of both processes are put together so that they can be compared easily. The left is the results of WF-DBR process, and the right is those of non WF-DBR process.



(a) Task Performer 3 (b) Task Performer 7 (W-CCR) (c) Task Performer 11

Figure 2. Comparison between the task performers' utilization

Figure 2 shows that for each task performer, the difference of the utilization is not statistically significant. Recall that WF-DBR is effective in the process efficiency. However, it turns out that WF-DBR doesn't provide any variation in terms of the utilization. In other words, the amount of work that the task performers carry out is similar with each other. From this, we can conclude that even though a task performer carries out the same amount of work, the use of WF-DBR allows the task performer to contribute more to the process efficiency.

In the WF-DBR process, the W-CCR's workload is controlled by the USS and the LSS. We can expect that the control results in the steady maintenance of the workload. To confirm these expectations, we have observed the variation of the W-CCR's workload as time goes by during the replication time. Figure 3 shows that the W-CCR's workload in the WF-DBR process remains stable and comparatively smaller than that in the non WF-DBR process. From out experiments, we can conclude that WF-DBR is the method to enhance the process efficiency and assure the steadiness of the W-CCR's workload.

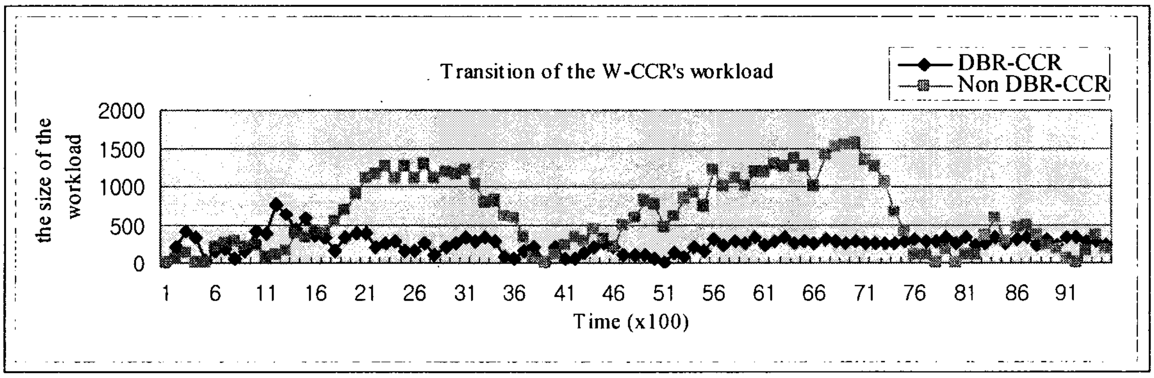


Figure 3. Transition of the W-CCR's workload

6. CONCLUSIONS AND FURTHER RESEARCH ISSUES

This paper shows that our WF-DBR method can manage workflow processes efficiently, and conducted simulation experiments to demonstrate its usefulness. Having focused mainly on the automation of execution, control, and management of processes, existing WFMSs rarely provides any relevant information and methods for the efficiency of process and participant management. We have developed a method that adopts the DBR technique and modifies it to make the technique appropriate for a workflow process. The WF-DBR enables to improve process efficiency by focusing on the workload of a task performer who has the highest impact on the progress of processes, controlling the release of instances and dispersing the W-CCR's tasks to other participants. From the simulation experiment, we could identify several effects of the proposed method as follows. First, the WF-DBR is effective in terms of the process efficiency. Second, the use of the WF-DBR results in the steady workload of the task performer. Finally, the WF-DBR induces each task performer to take part in tasks which actually contribute to the progress of the process.

There are several interesting issues that require further research. First of all, it is very important to take into account various factors, such as number and location of the CCR, dynamic variation of arrival rates, and process types. These certainly require further experiment that needs to be carefully designed. In addition, we can optimize the size of the W-CCR's work list by varying the W-CCR's USS and LSS.

7. REFERENCES

1. Daniel, V and Guide Jr., R. (1997). Scheduling with priority dispatching rules and drum-buffer-rope in a recoverable manufacturing system. International Journal of Production Economics, 53: 101-116.
2. Duclos, Leslie K. and Spencer, Michael S. (1995). The impact of a constraint buffer in a flow shop. International Journal of Production Economics, 42: 175-185.
3. Goldratt, E. M. (1992). The Goal. New York: North River Press.
4. Hollingsworth, D. (1994). Workflow Management Coalition Specification: The Workflow Reference Model, WfMC specification, http://www.wfmc.org.
5. Kim, Y., Bae, J., Bae, H. and Kang, S. (2003). Automatic Control of Workflow Processes using ECA Rules. IEEE Transactions on Knowledge and Data Engineering: To appear soon.
6. Lawrence, P. Workflow Handbook, Wiley, New York, 1997.
7. Radovilsky, Zinovy D. (1998). A quantitative approach to estimate the size of the time buffer in the theory of constraints. International Journal of Production Economics, 55: 113-119.
8. Rhee, S.-H, Bae, H. and Kim, Y. (2003). An efficient method of workflow management using a dispatching rule, Korean Management Science Review: To appear soon.

논문발표 1

• 1C e-Learning •

1C.1 e-learning에서 자기효능감 및 서비스 품질과 기술수용모형의 관계
이웅규, 이종기(대구대)

1C.2 사이버강좌를 위한 e-Learning 시스템 설계 및 구현
서창갑, 박성규(동명정보대)