An Approximate Analysis of Expected Cycle Time in Business Process Execution

Byung-Hyun Ha[1], Hajo A. Reijers[2], Joonsoo Bae[3], and Hyerim Bae[1,*]

[1] Dept. of Industrial Engineering, Pusan National Univ.,
San 30, Jangjeon-dong, Geumjeong-gu, Pusan, 609-735, Korea
{bhha, hrbae}@pusan.ac.kr
[2] Dept. of Technology Management, Eindhoven Univ.,
P.O. Box 513, 5600 MB, Eindhoven, The Netherlands
h.a.reijers@tm.tue.nl
[3] Dept. of Industrial & Sys. Eng., Chonbuk National Univ.,
664-14, Duckjin-dong, Duckjin-gu, Jeonju, Jeonbuk, 561-756, Korea
jsbae@chonbuk.ac.kr

Abstract. The accurate prediction of business process performance during its design phase can facilitate the assessment of existing processes and the generation of alternatives. In this paper, an approximation method to estimate the cycle time of a business process is introduced. First, we propose a process execution scheme, with which Business Process Management Systems (BPMS) can control the execution of processes. Second, an approximation method for analyzing its cycle time, based on queueing theory, is presented. We consider agents as queueing servers with multi-class customers and predict the response time of the agents. The cycle time of the whole process is calculated using the expected response time and process structure, taking into account parallel process execution. Finally, the results from the analytical approximation are validated against those of a simulation. This analysis can be used to obtain an optimal process execution plan.

1 Introduction

To secure advantage in today's competitive and customer-oriented business environments, it is necessary to maintain the effectiveness of business processes. Efficient management in rendering business processes effective is a key element of competitiveness. Business Process Management Systems (BPMS) were introduced in an effort to manage business processes efficiently. BPMS is an information system for designing, administering, and improving intra/inter-organizational business processes. As a result, BPMS has become a core engine for integrating enterprise information systems in a process-oriented way [11]. One of the most important reasons for employing BPMS is that it can be a sound basis for improving business processes. Integral to this end is performance analysis.

* Corresponding author.

J. Eder, S. Dustdar et al. (Eds.): BPM 2006 Workshops, LNCS 4103, pp. 65 – 74, 2006.

A performance index of a business process can be determined according to customers, internal processes, suppliers, finance, and employees [13]. We consider as our performance index cycle time, which has commonly been used to define the period of time between the receipt of an order from a customer and the completion of the order. Since business processes managed by BPMSs are very dynamic, complete information is rarely known before executing them. Hence, if cycle time can be predicted at process design time, it can facilitate the assessment and streamlining of existing process as well as the outlining of new processes. In this paper, a queueing model for estimating the cycle time of business processes is introduced.

Employing stochastic models as analytic models for business processes has been researched in numerous ways for various purposes. Early research has examined the assertion that queuing theory can be used to redesign business processes [2,10]. Narahari *et al*. have analyzed the cycle time of the New Product Development (NPD) process by modeling an organization's departments as queueing servers, and proposed several ways to reduce cycle time by means of queueing theory and a simulation method [7]. Son and Kim have suggested a capacity planning scheme to satisfy due dates by modeling tasks of business processes as queueing servers [12]. Another extensively researched model based on a well-defined theoretical foundation is Stochastic Workflow Net (SWN), the results of which can be used to analyze the performance of business processes and to plan agent capacity, among other ends [1,9]. However, in most of the previous studies it was presumed that the capacity of the agents is *infinite* and that an agent is dedicated to only a single task. These assumptions can hinder a more accurate description and analysis of business processes, which is the main motivation for us to devise a more realistic approach.

2 Models for Business Process Analysis

Process models used by commercial BPMSs usually include detailed information on the automatic execution of the processes involved. However, since the purpose of our research is to analyze process efficiency, it might not be necessary to consider all business information, e.g. business rules. Therefore, we provide, as required to analyze processes, a simplified process model. Our model includes three aspects of process information: process structure, resource capacity, and statistical information. The following is a definition of a process model.

Definition 1 (Process Model). A *process model* is defined as a tuple $\langle T, SB, L, A, \mu, rp, pe, \Phi \rangle$ which is characterized as follows:

i) T is a set of *tasks*.
ii) $SB = \langle B_S, B_R, B_P, s_o \rangle$ is a tuple of blocks, where B_S, B_R, B_P, and s_o are a set of *sequence blocks*, a set of *repeat blocks*, a set of *parallel blocks*, and an *outmost sequence block*, respectively. Each block can be nested, that is, a block can include tasks and internal blocks as its members.
iii) $L \subset \cup\{B \times B \mid B \in B_S \cup B_R\}$ is a set of *links*.
iv) A is a set of *agents*.
v) $\mu : T \times A \rightarrow \mathrm{R}^{+} \cup \{0\}$ is a function of *average service rate*.

vi) $rp : B_R \rightarrow \mathrm{R}^+$ is a function of *repeat probability*.
vii) $pe : \cup\{2^P \mid p \in P\} \rightarrow \mathrm{R}^+ \cup \{0\}$ is a function of *parallel execution probability*, where $pe_{\varnothing} = 0$.
viii) Φ is *customer arrival rate*.

A sample process is represented in Fig. 1 (a), which illustrates an 'Internet loan process.' After a customer requests a loan, a clerk first checks the loan application. Then, two tasks, 'History Review' and 'Credit Inquiry,' are executed simultaneously if their respective preconditions are met. The probability of each task's execution is marked on a split arrow. Taking the results of these tasks into account, the task 'Loan Grant' is executed next. Note that this appraisal may lead to a repeated execution of the history review and credit inquiry tasks.

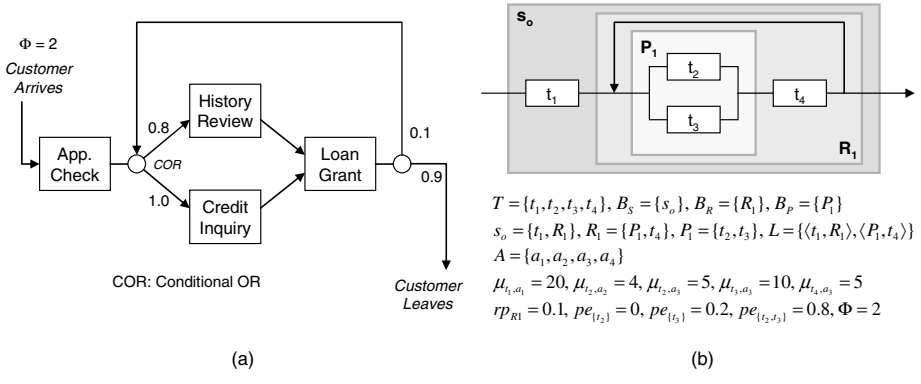

Fig. 1. A simple process model (a) a sample process (b) a process model

The sample process can be mapped onto our process model as shown in Fig. 1 (b). All tasks establish a task set, T, and also defined are sets of blocks (B_S, B_P, and B_R), links (L), and agents (A). Block execution probabilities for parallel and repeat blocks are derived from the task execution probabilities of the original process. For example, the probability that both t_2 and t_3 will execute ($pe_{\{t_2,t_3\}}$) is 0.8, because t_3 is always executed and t_2 is executed with probability of 0.8. In this sample process, customers arrive every 2 time units on the average, that is, Φ=2. For an agent a and a task t, a positive value for average service rate $\mu_{t,a}$ indicates the average number of t that can be performed by a per unit time. Otherwise, $\mu_{t,a}$ equals zero. This kind of statistical information can be estimated by domain experts in the design phase or just collected from the execution history of the business process as registered by the BPMS.

To analyze the cycle time of a business process based on the model above, we first introduce the execution frequency of each task. An *expected execution frequency* of b, denoted by f_b, is the frequency of performing block b while a business process is executed. The expected execution frequencies of blocks are recursively calculated using the following equation:

$$
\begin{aligned}
f_{s_o} &= 1, \\
f_s &= f_S && \forall s \in S \text{ where } S \in B_S , \\
f_r &= f_R/(1-rp_R) && \forall r \in R \text{ where } R \in B_R , \\
f_p &= \sum_{B \in \{S \in 2^P \mid p \in S\}} pe_B \cdot f_P && \forall p \in P \text{ where } P \in B_P .
\end{aligned} \tag{1}
$$

In the sample process depicted in Fig. 1, the expected execution frequencies of the tasks, $\langle f_{t_1}, f_{t_2}, f_{t_3}, f_{t_4} \rangle$ are determined to be $\langle 1, 8/9, 10/9, 10/9 \rangle$.

In addition, the notation below is employed to simplify the formulas in the following sections. The shorthands represent a set of tasks that can be performed by agent a and a set of agents who can perform task t, respectively:

$$T_a := \{t \mid \mu_{t,a} > 0\} , \quad A_t := \{t \mid \mu_{t,a} > 0\} . \tag{2}$$

3 Queueing Analysis of Process Execution

In this section, a method of executing business processes is presented and the main ideas behind the performance analysis of business processes using queuing theory are illustrated. When tasks are required to be performed, a BPMS assigns them to agents by putting them into worklists. The performance of process execution depends on the method of managing worklists and the order of performing tasks in worklists [5]. In this paper, we consider a work environment in which each agent has his own worklist.

When a business process is executed in such an environment, the cycle time depends on the ability of the agent who performs a task. In the process execution phase, a task is assigned to a specific agent with predefined probability as defined below.

Definition 2 (Task Assignment Probability). A *task assignment probability* of task t assigned to agent a, denoted by $p_{t,a}$, is the probability that agent a is selected to perform task t in business process execution, where $\mu_{t,a} > 0$.

In the process execution phase, we apply the following rules: i) when a task is to be assigned, it is assigned to a specific worklist of an agent using the predefined task assignment probability ($p_{t,a}$), and ii) each agent performs the tasks in his worklist using the First-In-First-Out (FIFO) dispatching rule.

To analyze business processes using queueing theory, agents are modeled as queueing servers, which are called *agent servers*. A queueing network can be built by connecting the agent servers. The jobs arriving at an agent server are the tasks assigned to the worklist of the corresponding agent.

The *task arrival rate* of t (λ_t) is the average number of occurrences of task t per time unit when the business process is continuously visited by customers. Similarly, $\lambda_{t,a}$ and $\lambda_{\bullet a}$ are defined as the *task arrival rate* of task t assigned to agent a and the *task arrival rate* assigned to agent a regardless of task type, respectively. The task assignment probabilities can be calculated as follows:

$$\lambda_t = \Phi \cdot f_t \,, \quad \lambda_{t,a} = p_{t,a} \cdot \lambda_t \,, \quad \lambda_{\bullet a} = \sum_{t \in T_a} \lambda_{t,a} \,, \tag{3}$$

and the utilization rate of agent a (ld_a) is derived as [4]:

$$ld_a = \sum_{t \in T_a} \frac{\lambda_{t,a}}{\mu_{t,a}} = \sum_{t \in T_a} \frac{\lambda_t p_{t,a}}{\mu_{t,a}} = \Phi \sum_{t \in T_a} \frac{f_t p_{t,a}}{\mu_{t,a}} \,. \tag{4}$$

The response time (cycle time) of an agent can be obtained by analyzing the corresponding agent server. Let $ST_{t,a}$ and ST_a be the random variables (RV) denoting the service time of agent a for task t and the service time of agent a regardless of task type, respectively. (Note that $E[ST_{t,a}]$ is $\mu_{t,a}$.) Because $ST_{t,a}$ can be assumed to be independent of each other, the moments of ST_a are derived as follows:

$$E[ST_a^{\,n}] = \sum_{t \in T_a} \frac{\lambda_{t,a}}{\lambda_{\bullet a}} E[ST_{t,a}^{\,n}] \,. \tag{5}$$

Let CT_a be the RV denoting the cycle time of an agent a. Recall that in this research, an agent handles tasks in a worklist with the FIFO rule. And jobs can be assumed to arrive at agent servers according to the Poisson process [12]. Hence, the moments of the cycle time can be derived as follows [4]:

$$E[CT_a] = \frac{\lambda_{\bullet a} E[ST_a^2]}{2(1-ld_a)} + \frac{ld_a}{\lambda_{\bullet a}} \,, \quad E[CT_a^2] = \frac{\lambda_{\bullet a} E[ST_a^3]}{3(1-ld_a)} + \frac{\lambda_{\bullet a}^2 E[ST_a^2]^2}{2(1-ld_a)^2} + \frac{E[ST_a^2]}{(1-ld_a)} \,. \tag{6}$$

The cycle times of agents are independent of each other and tasks are assigned to agents with predefined task assignment probabilities ($p_{t,a}$) regardless of their arrival order. Therefore, the moments of the cycle time CT_t of task t are expected to be:

$$E[CT_t] = \sum_{a \in A_t} p_{t,a} \cdot E[CT_a] \,, \quad E[CT_t^2] = \sum_{a \in A_t} p_{t,a} \cdot E[CT_a^2] \,. \tag{7}$$

4 Estimating Cycle Time

The expected cycle time of each block is derived using those of its internal blocks, and the cycle time of a process is that of the outmost block s_o. In other words, the cycle time of the whole process is recursively calculated from the innermost blocks, that is, the tasks. The results of this section are based mainly on queueing theory and, thus, on the assumption of a steady state system.

Sequence blocks. The cycle times of internal blocks that structure a sequence block are not independent of each other. In other words, if one of the internal blocks has a long cycle time, there is also a very high probability of other blocks in the same sequence having a long cycle time. This kind of dependency can be modeled using the coefficient of correlation. Commonly, the coefficient of correlation varies with the structure of the queueing network and the server utilization rate [3]. However, we fix the coefficient of correlation of sequence blocks (ρ_s) as 0.5, based on comprehensive experiments, and assume that only adjacent blocks are correlated.

Let CT_S be the RV denoting the cycle time of sequence block S; then it is straightforward to obtain the mean and variance of the cycle time:

$$\begin{aligned} E[CT_S] &= \sum_{b \in S} E[CT_b], \\ Var[CT_S] &= \sum_{b \in S} Var[CT_b] + \sum_{\langle b_1, b_2 \rangle \in L \cap S^2} 2\rho_s \sqrt{Var[CT_{b_1}] Var[CT_{b_2}]}. \end{aligned} \tag{8}$$

Repeat blocks. The number of executions of a repeat block R depends on its repeat probability rp_R. Let CT_R and $CT_{R,1}$ be the RV denoting the cycle time of repeat block R and the cycle time of R when it is executed only once, respectively. Then,

$$CT_R = CT_{R,1} + rp_R CT_{R,1} + rp_R^2 CT_{R,1} + \cdots, \tag{9}$$

and the statistics for $CT_{R,1}$ are not different from those of the sequence block.

As in the sequential case, the cycle times from the repeated execution of a block are not independent of each other. We also fixed the coefficient of correlation of the repeated execution (ρ_r) at 0.4. Note that the coefficient of correlation of repeat blocks is different from that of sequence block. That is, the former represents inter-block dependency, while the latter represents the dependency of adjacent inner blocks. The mean and variance of the cycle time of repeat blocks are derived as follows:

$$\begin{aligned} E[CT_R] &= E[CT_{R,1}] / (1 - rp_R), \\ Var[CT_R] &= \sum_{i=1}^{\infty} rp_R^{2(i-1)} Var[CT_{R,1}] + 2\sum_{i=1}^{\infty} rp_R^{2i-1} \rho_r \sqrt{Var[CT_{R,1}] Var[CT_{R,1}]} \\ &= (1 + 2\rho_r rp_R) Var[CT_{R,1}] / (1 - rp_R^2). \end{aligned} \tag{10}$$

Parallel blocks. The cycle time of a parallel execution is the maximum of the cycle times of inner blocks. Given that in general it is not easy to obtain accurate performance measures of parallel blocks, we employ an approximation method for Fork-Join Queue [8] and adapt it to business processes.

Let CT_P and PT_B be the RV's denoting the cycle time of parallel block P and the cycle time of inner blocks $B \subset P$ when every block in B is executed, respectively. Then, the mean and variance of block P (with the coefficient of correlation $\rho_p = 0.5$) are given by:

$$\begin{aligned} E[CT_P] &= \sum_{B \in 2^P} pe_B E[PT_B], \\ Var[CT_P] &= \sum_{B \in 2^P} pe_B^2 Var[PT_B] + \sum_{\substack{\langle B_1, B_2 \rangle \in 2^P \times 2^P, \\ B_1 \neq B_2}} 2\rho_P pe_{B_1} pe_{B_2} \sqrt{Var[PT_{B_1}] Var[PT_{B_1}]}, \end{aligned} \tag{11}$$

where $E[PT_\varnothing] = Var[PT_\varnothing] = 0$.

The cycle time of the inner blocks is approximated using a generalized exponential distribution [8]. Let CT_b and AT_b be the RV denoting the cycle time of inner block $b \in B$ and its approximated cycle time using a generalized exponential distribution, respectively. When $E[CT_b]^2 > Var[CT_b]$, the cumulative distribution function of AT_b is given by:

$$F_{AT_b}(x) = 1 - e^{-(x-d_b)/m_b} \quad x \geq d_b$$
$$= 0 \qquad \text{otherwise, where } m_b = \sqrt{Var[CT_b]}\,,\ d_b = E[CT_b] - m_b\,. \tag{12}$$

Note that the mean and variance of AT_b and CT_b are the same. If the cycle times of the inner blocks are assumed to be independent of each other, the cumulative distribution function of PT_B is approximated as follows:

$$F_{PT_B}(x) = \Pr[PT_B \leq x] = \Pr[CT_b \leq x, \forall b \in B] = \prod_{b \in B} \Pr[CT_b \leq x]$$
$$\cong \prod_{b \in B} F_{AT_b}(x) = \prod_{b \in B} 1 - e^{-(x-d_b)/m_b}\,, \qquad \text{where } x \geq \max_{b \in B}\{d_b\}\,. \tag{13}$$

As a result, the first and second moments of PT_B are approximated as follows,

$$E[PT_B] \cong d + \sum_{A \in 2^B \setminus \{\varnothing\}} (-1)^{|A|-1} k_A \exp\left(\sum_{b \in A} -\frac{d - d_b}{m_b}\right),$$
$$E[PT_B^2] \cong d^2 + 2\sum_{A \in 2^B \setminus \{\varnothing\}} (-1)^{|A|-1} k_A (d + k_A) \exp\left(\sum_{b \in A} -\frac{d - d_b}{m_b}\right), \tag{14}$$

where

$$d = \max_{b \in B}\{d_b\}, \quad k_A = \prod_{b \in A} m_b \Big/ \sum_{b \in A} \prod_{c \in A \setminus \{b\}} m_c\,. \tag{15}$$

5 Experimental Results

To validate the accuracy of our method, the analytical results for predicting process cycle time were compared with simulation results. In this validation, we used random processes of which the structure and the parameters are randomly determined by a computer.

Each random process is created with the number of tasks and the number of agents as input data. First, the average service rate of each task is determined using uniform distributions, and service rates of agents were randomly generated based on the average service rate of the task. The service times for tasks were assumed to have gamma distributions with the shape parameter α = 2, an assumption to be known as generally applicable in practice [6]. The simulation was set to prevent any single agent doing more than 10 tasks on the average. Generation of the process structure started from the outmost sequence block, and then the type of inner structure was determined randomly. The numbers of tasks, repeat blocks, and parallel blocks in a sequence or repeat block were respectively set to be 10, 2 and 3 on the average. Task assignment probabilities were generated at random too, and task arrival rates that allow a maximum workload of agents of 50, 60, 70, 80 and 90 % were used for our simulation.

Fig. 2 illustrates a sample random process, and the experimental results of the cycle time approximation. Fig. 2 (a) shows that the sample process consists of 12 tasks, 3 parallel blocks, and a repeat block. Five agents participate in the execution of the

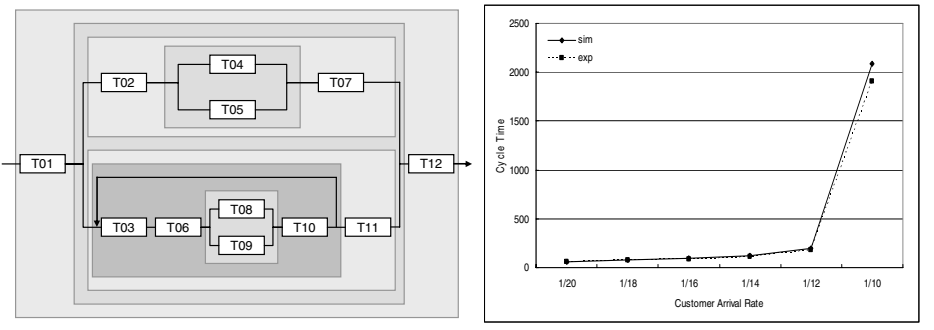

Fig. 2. A sample process for experiments. (a) A random process model. (b) Comparison between the analytical and simulation results.

process. With customer arrival rates varying from 1/20 to 1/10, simulation results were compared with the cycle times calculated by our method. From the result in Fig. 2 (b), it can be seen that the estimation was quite satisfactory across all customer arrival rates.

Table 1. Summary of experiment results with # of business processes*, ME (Mean Error) in %**, and MSE (Mean Squared Error) in %***

# of agents	# of tasks										
	20	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**	**110**	**120**
10	249*	248	249	249	246	245	251	271	271	271	271
	6.84**	6.82	6.89	6.22	6.61	7.50	8.11	8.19	7.88	7.60	8.58
	66.80***	63.34	69.35	52.60	57.34	75.23	91.36	90.26	84.22	78.88	92.97
20	263	264	264	261	260	262	262	251	251	251	251
	3.78	4.64	4.75	5.82	5.14	5.66	6.06	5.71	6.09	5.67	5.83
	23.07	33.37	35.52	54.58	38.69	46.60	54.60	47.69	60.70	49.17	51.15
30	261	259	259	273	272	251	251	251			
	3.86	3.30	4.21	4.19	4.51	4.29	4.85	4.81			
	22.61	16.10	26.56	25.53	32.27	28.35	39.93	32.44			
40	251	251	251	251	251	251	251	251			
	2.68	3.45	3.50	3.66	3.65	3.93	4.28	3.99			
	13.00	21.04	22.36	20.92	19.91	24.18	27.33	28.64			
50	251	251	251	251	251	251	251				
	3.13	2.90	3.22	3.41	3.76	3.78	4.17				
	15.09	14.77	15.79	17.39	23.66	21.55	28.88				

11,503 iterations of the experiments were conducted for the random processes with varying numbers of agents (from 10 to 50) and numbers of tasks (from 20 to 120). The experimental results show that the total mean error is 5.08 (%) and the mean squared error is 41.42 (%). The results are summarized in Table 1.

In Table 1, the second row of each cell shows the percentage of time difference between the predicted cycle time and the simulation result. The results show that our prediction of the cycle time is within a 10% error on average. It can be seen that the

error generally increases as the number of tasks increases, but the trend is not so significant. This fact implies that complex process structures can cause an estimation error. At the same time, the error decreases as the number of agents increases. This is because an increased number of agents likely reduces the variance in the cycle times of tasks.

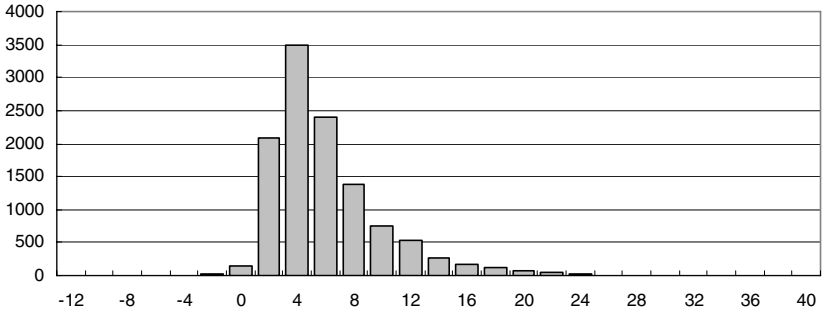

Fig. 3. The histogram of errors in %

Fig. 3, which depicts the overall distribution of error, shows that the variance of error is not so large.

6 Conclusions

In this paper, we provide an approximate analysis of the average cycle time of business processes. For this purpose we first considered a process execution scheme assuming BPMS to control the execution of the processes. Under this execution scheme, an agent is assumed to have an individual worklist and the BPMS assigns tasks to the worklists. An approximation method for the setting of the individual worklist was devised to analyze the cycle time. The method is based on queueing theory, and we considered agents as queueing servers with multi-class customers in order to predict the response time of the agents. The cycle time of the whole process was calculated using the expected response time and process structure, taking into account parallel process execution. We conducted simulation experiments to verify the effectiveness of our approach, and showed that our method can predict cycle time with acceptable accuracy. We expect that the prediction of business process performance in the design phase can facilitate the assessment of existing processes and help to recommend the generation of new designs.

With respect to further research, since we can evaluate a process with respect to its cycle time, it might be possible to find, under the process execution scheme introduced in this paper, execution rules that minimize the cycle time. Though mathematical solutions are often difficult, a meta-heuristic approach to this problem can be very effective.

Acknowledgements

This work was supported by the Regional Research Centers Program (Research Center for Logistics Information Technology), granted by the Korean Ministry of Education & Human Resources Development.

References

1. van der Aalst, W., van Hee, K., Houben, G.: Modelling and analysing workflow using a petri-net based approach. In: Proceedings of the Second Workshop on Computer-Supported Cooperative Work, Petri Nets and Related Formalisms. (1994)
2. Buzacott, J.A.: Commonalities in reengineered business processes: models and issues. Management Science 42(5) (1996) 768–782
3. Daduna, H., Szekli, R.: On the correlation of sojourn times in open networks of exponential multiserver queues. Queueing Systems 34(1-4) (2000) 169–181
4. Gross, D., Harris, C.: Fundamentals of Queueing Theory. John Wiley & Sons, New York (1998)
5. 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) (2006) 64–84
6. Law, A.M., Kelton, W.D.: Simulation Modeling and Analysis. Third edn. McGraw-Hill, Boston, MA (2000)
7. Narahari, Y., Viswanadham, N., Kumar, K.V.: Lead time modeling and acceleration of product design and development. IEEE Transaction on Robotics and Automation 15(5) (1999) 882–896
8. Rajaraman, B., Morgan, T.W.: Approximate analysis of the average delay in parallel program execution. In: Proceeding of the Twenty-Sixth Hawaii International Conference on System Sciences, Hawaii (1993) 584–593
9. Reijers, H.A.: Design and Control of Workflow Processes. Springer-Verlag (2003)
10. Seidmann, A., Sundararajan, A.: The effects of task and information asymmetry on business process redesign. International Journal of Production Economics 50(2-3) (1997) 117–128
11. Smith, H., Fingar, P.: Business Process Management: The Third Wave. Meghan-Kiffer, Tampa (2003)
12. Son, J., Kim, M.: Improving the performance of time-constrained workflow processing. Journal of Systems and Software 58(3) (2001) 211–219
13. Wesner J.W., Hiatt, J.M., Trimble, D.C.: Winning With Quality: Applying Quality Principles in Product Development. Addison-Wesley, Reading, MA (1995)