

통계적 모형의 업무부하 균일화를 통한 비즈니스 프로세스의 효율화

하병현¹ · 설현주¹ · 배준수^{2*} · 박용태¹ · 강석호¹

¹서울대학교 산업공학과 / ²성결대학교 전자상거래학부

Workload Balancing on Agents for Business Process Efficiency based on Stochastic Model

Byung-Hyun Ha¹ · Hyeonju Seol¹ · Joonsoo Bae² · Yongtae Park¹ · Suk-Ho Kang¹

¹Department of Industrial Engineering, Seoul National University, Seoul, 151-742

²Division of E-Commerce, Sungkyul University, Anyang, 430-742

BPMS (Business Process Management Systems) is an information system that systematically supports designing, administrating, and improving the business processes. It can execute the business processes by assigning tasks to human or computer agents according to the predefined definitions of the processes. In this research we developed a task assignment algorithm that can maximize overall process efficiency under the limitation of agents' capacity. Since BPMS manipulates the formal and predictable business processes, we can analyze the processes using queuing theory to achieve overall process efficiency. We first transform the business processes into queuing network model in which the agents are considered as servers. After that, workloads of agents are calculated as server utilization and we can determine the task assignment policy by balancing the workloads. This will make the workloads of all agents be minimized, and the overall process efficiency is achieved in this way. Another application of the results can be capacity planning of agents in advance and business process optimization in reengineering context. We performed the simulation analysis to validate the results and also show the effectiveness of the algorithm by comparing with well known dispatching policies.

Keywords: business process management, queuing network, capacity planning, business process reengineering

1. 서론

BPM(Business Process Management)은 업무절차를 관리하기 위한 지속적인 노력으로 탄생했다. 기존의 WFMS(Workflow Management Systems)를 시작으로 조직은 내부의 업무절차를 개발, 실행, 관리하는 초석을 마련하였으며, 최근에는 이러한 결실을 기업 간 업무통합에 적용할 수 있도록 확장한 BPMS(Business

Process Management Systems)를 활용하고 있다(Smith *et al.*, 2002).

BPMS를 구성하는 핵심적인 요소로 업무절차와 작업, 참여자, 역할, 관련개체 등이 있다(Hollingsworth, 1995; Smith *et al.*, 2002). 업무절차는 특정 목적을 위해 기업에서 수행하는 일을 나타내는 것으로, 새로운 업무절차를 수행할 시점이 오면, BPMS는 그 업무절차의 미리 정의된 명세로부터 처리해야 하는 작업들의 순서를 읽는다. 그 순서에 따라 특정 작업을 그것

본 연구는 한국과학재단의 특정기초연구과제(과제번호 R01-2002-000-00155-0)의 지원을 받았다.

*연락처 : 배준수, 430-742 경기도 안양시 만안구 안양 8동 산 147-2, 성결대학교 전자상거래학부,

Fax : 031-449-0529, E-mail : jsbae@sungkyul.edu

을 수행할 수 있는 역할을 가진 참여자에게 부여하여 일을 진행하게 된다.

일반적으로 특정 역할을 수행하는 참여자의 자원은 한정되어 있다. 따라서 한꺼번에 많은 업무를 처리해야 하는 상황이 발생할 경우 참여자의 수에 비해 수행해야 하는 작업들의 양이 훨씬 많아지며, 일정한 작업을 처리할 수 있는 참여자는 역할에 의해 제한되어 있으므로 다양한 작업을 효율적으로 처리하기 위해서는 정교한 업무할당 규칙이 필요하다. 또한 BPM에서는 업무를 진행할 때 수행하는 작업들의 순서가 완전히 결정되어 있는 것이 아니라 일의 진행 상황에 따라 조건적으로 작업들이 선택되는 경우가 대부분이므로 참여자에게 작업을 할당하는 문제는 한층 어려워진다.

한정된 자원을 사용하여 주어진 일을 처리하는 문제는, 자원고갈에 따라 관리해야 할 중요한 작업이 달라지고 이러한 중요한 작업들을 어떻게 발견하고 적절하게 할당할 것인가에 대한 이론인 제약이론의 출현에 따라 새롭게 주목을 받고 있으며, 이를 바탕으로 한정된 자원을 고려한 스케줄링 관련 연구가 이루어지고 있다(Goldratt and Cox, 1992; Simons and Simpson, 1997; Spearman, 1997). 또한 전통적으로 스케줄링 분야로 언급되는 제조 및 생산 시스템 외에 업무절차의 특징들을 고려한 연구들 역시 많이 이루어져 왔다 (Park, 1998; Attie et al., 1996; Chang et al., 2002; Davulcu et al., 1998; Jin et al., 2001; Senkul et al., 2002; Son and Kim, 2001).

본 논문에서는 BPM이 대상으로 하는 업무절차의 특징을 반영한 효율적인 업무할당 규칙을 탐구한다. 전반적인 관점에서 업무의 실행을 효율화하기 위해서는 참여자들의 업무부하를 균일화하는 것이 중요하며 (Jin et al., 2001; Senkul et al., 2002), 여유시간을 부여함으로써 다양한 스케줄링 문제에서 시스템 전체적인 성능을 향상시킬 수 있음이 알려져 있다(Baker, 1974; Pinedo, 1995). 따라서 참여자에게 여유시간을 부여하는 방법으로 작업부하를 균일화하고, 그를 통해 BPM의 전반적인 성능을 향상시키는 알고리즘을 개발하였다.

특히 이러한 성능의 향상은 작업부하가 커질수록 보다 두드러진다. 본 알고리즘의 전반적인 절차는 먼저 대기행렬 이론을 사용하여 참여자를 서버로 모형화하고, 그 참여자에게 부여하는 작업의 확률값을 변수로 평균적으로 수행하는 일을 구한다. 이를 바탕으로 모든 참여자에게 할당되는 일의 부하를 계산한 후 그것을 평준화하는 작업할당 확률을 선형계획법을 사용하여 풀이한다.

2. BPMS에서의 일반적인 업무할당과 문제점

<그림 1>은 온라인 대출처리 절차를 간략하게 표현한 업무절차와, 작업을 참여자에게 할당할 때 BPM에서 발생할 수 있는 상황을 간트차트로 보여주고 있다.

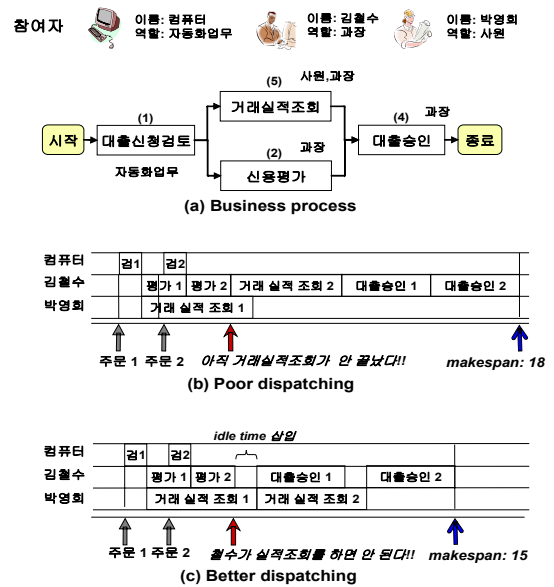


그림 1. 온라인 बैं킹 대출처리 프로세스.

<그림 1>의 (a)에는 컴퓨터, 김철수, 박영희의 세 참여자들과, 그들 각각의 역할이 나타나 있다. 김철수와 같이 특정 참여자는 한 가지 이상의 역할을 가질 수 있다. 온라인 बैं킹에서 대출을 처리하는 업무절차는 대출신청검토, 거래실적조회, 신용평가, 대출승인의 네 작업으로 구성되며, 각 작업의 수행시 소요되는 시간이 각 작업 위에 표시되어 있다. 대출신청검토를 수행한 다음에 거래실적조회와 신용평가를 동시에 수행할 수 있으며, 두 작업의 종료 후 대출승인을 처리할 수 있다는 시간 상의 선후행 관계도 기술하고 있다.

위의 예에서 발생하는 대출을 처리하는 참여자들이 충분히 존재한다고 가정한다면, 주문과 관련된 모든 작업들을 필요할 때 즉시 처리할 수 있다. 하지만 일반적인 상황에서는 항상 자원이 충분히 존재하는 것은 아니며, 때로는 원가절감을 목적으로 자원을 효율적으로 활용하는 것 역시 중요하다. 즉, 참여자들이 필요 이상으로 많은 경우라면 그들 대부분의 시간을 처리할 작업이 없이 보내게 될 것이며, 반대로 참여자가 부족하면 고객의 주문에 충분히 대응하지 못하는 결과로 이어진다. BPMS를 사용하면서 가질 수 있는 이점들 중 한 가지는 이러한 정보들을 체계적으로 파악할 수 있는 가능성이 생기는 것이다 (Smith et al., 2002). 전통적으로 이러한 자원과 업무, 제약조건에 따른 업무할당과 관련된 부분은 스케줄링 분야에서 연구해왔으며(Baker, 1974; Pinedo, 1995), BPMS가 참여자에게 작업을 할당하는 것도 스케줄링이 취하는 접근방법과 동일한 관점에서 접근할 수 있다.

<그림 1>의 (b)와 (c)는 작업을 참여자에게 할당할 때 BPM에서 발생할 수 있는 상황을 나타낸다. 이 예는 잘못된 할당이 시스템의 성능을 얼마나 저하시킬 수 있는지 보여준다. (b)와 (c)는 모두 주문이 시점 0과 2에서 두 번 발생하고 있다. BPMS가 작업을 참여자에게 할당해가는 과정에서 시점 5에서의 상

황을 보면, 첫 주문은 대출승인검토와 신용평가 작업이 수행 완료되었고 거래실적조회 역할을 수행하는 박영희가 거래실적 조회를 하고 있는 것을 알 수 있으며, 두 번째 주문은 대출신청검토와 신용평가 완료되었음을 알 수 있다. 이 때 BPMS가 특별한 작업할당정책을 가지고 있지 않다면, 두 번째 거래실적 조회를 김철수도 수행할 수 있으므로 현재 일이 없는 김철수에게 거래실적 조회의 수행을 부여한다. (b)는 이러한 수행의 결과를 나타내고 있으며, 두 대출 관련 업무를 모두 완료하는 시점이 18이 되는 것을 알 수 있다. (c)는 비록 김철수가 거래실적조회를 처리할 수 있어도 그것을 수행하지 않고, 박영희가 수행하고 있던 거래실적조회를 마친 후 다시 대출의 거래실적 조회를 수행토록 하고 있으며, 이 경우에는 시점 15에서 두 대출에 대한 처리가 종료된다는 것을 알 수 있다. 즉, 비록 당장의 시점에서 아무런 작업도 수행하지 않는 참여자가 존재해도 장기적인 관점에서는 더 이익이라는 것을 알 수 있다.

이러한 문제는 참여자의 용량이 제한되어 있고 수행할 수 있는 작업이 정해져 있다는 제약조건에서 발생한다. 본 연구에서는 업무를 수행할 때 발생하는 위와 같은 비효율적인 문제를 해결할 수 있는 방법을 모색한다. 먼저 확률을 바탕으로 참여자가 작업을 처리하는 시간과 업무의 수행이 요구되는 주기를 파악하고, 통계 모형을 바탕으로 주어진 업무들을 완료하는 시간을 최소화하는 업무할당 규칙을 개발한다.

3. 확률에 기반한 업무할당 규칙

본 연구에서는 BPM 환경하에서 새로운 업무할당 규칙을 개발하기 위해 확률적인 기대값을 사용한다. BPM 환경은 유연히 개입하는 현실적인 일을 다루는 것이므로 확률을 사용하여 모형화함으로써 보다 더 실제에 잘 접근할 수 있는 가능성을 가진다. 아래는 본 연구에서 사용하는 정의를 나타낸다.

- 정의 1 (Process Occurrence Rate)** 정의된 업무명세를 수행시키는 사건의 발생 빈도수를 Process Occurrence Rate라고 한다. Process Occurrence Rate는 평균이 λ_p 인 포아송 분포를 따른다고 가정한다.
- 정의 2 (Average Task Frequency)** 업무명세를 한 번 수행했을 때 명세에 포함된 특정 업무를 수행하는 빈도수를 Average Task Frequency라고 하고 f_T 로 표시한다.
- 정의 3 (Task Dispatching Rate)** 업무명세에 정의된 특정 업무가 단위시간 당 수행되는 빈도수를 Task Dispatching Rate라고 정의한다. Task Dispatching Rate r_T 는 ($\lambda_p \diamond f_T$)로 구할 수 있다.
- 정의 4 (Agent-Task Probability)** 업무명세에 따라 작업을 수행할 때 특정 업무를 특정 작업자에게 부여할 확률을 Agent-Task Probability라고 한다.

다음의 <그림 2>는 위에서 정의한 용어를 사용하여 <그림 1>의 (a)의 Task Dispatching Rate를 구한 예를 보여준다.

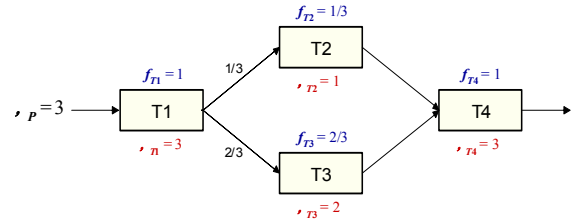


그림 2. Task Dispatching Rate 도출 예제.

Agent-Task Probability는 어떤 작업을 처리해야 할 때 어떤 참여자를 선택하는지 나타내는 확률이다. 예를 들면, 만일 T2 작업을 A2와 A3의 참여자가 처리할 수 있다면 T2를 p 의 확률로 A2에게 q 의 확률로 A3에게 부여하는 것을 의미한다. 다음의 <그림 3>은 <그림 1>에서 보여준 업무절차를 Agent-Task Probability를 사용하여 실제 일을 수행하는 참여자를 명시하여 다시 그린 그림이다.

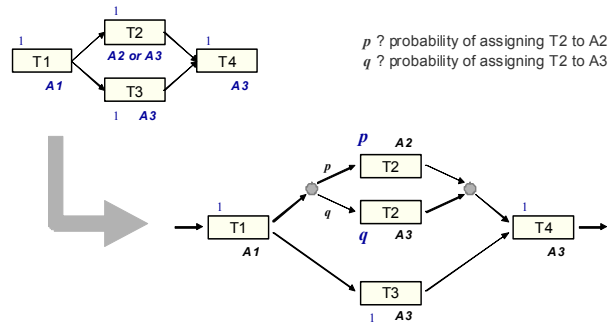


그림 3. Agent-Task Probability를 사용한 명세의 상세화.

본 연구에서는 대기행렬 이론을 사용하여 참여자를 서버로 보고 작업을 고객으로 보아 참여자에게 할당하는 것을 그 참여자에 대응하는 서버에 고객이 도착하는 것으로 분석하였다. Agent-Task Probability를 이용하여 대기행렬 모형을 적용하기 위해서는 작업을 참여자에게 할당할 때 그 참여자의 업무 목록에 작업을 적어두어 후에 그것을 수행하는 방법을 취해야 한다. 작업을 참여자에게 할당하는 여러 방법들 중 업무 목록의 크기를 보고 가장 작은 것에 할당하는 방식도 있지만 이러한 방법은 단기적인 업무 부하를 낮추는 데 유용한 반면 잘못된 작업을 할당할 수도 있으므로 장기적으로 효율을 보장하지는 못한다. <그림 4>는 <그림 1>의 예제를 대기행렬 네트워크로 해석한 것이다. 네트워크를 구성하는 각각의 서버는 참여자를 나타내며 컴퓨터는 A1, 박영희는 A2, 김철수는 A3에 해당한다. 그리고 왼쪽 아래 그림은 대기행렬 네트워크로 변환하기 전인 참여자를 고려해 상세화된 업무명세이다.

<그림 4>의 A3 서버에는 λ_p 의 도착률로 평균 응대시간이 2만큼 필요한 일이 주어지고 $q \cdot \lambda_p$ 의 도착률로 평균 응대시간이 5인 일이, λ_p 의 도착률로 평균 응대시간이 4인 일이 주어진다. 이것은 업무가 λ_p 의 비율로 발생하고 T2를 q 의 확률로 김철수

에게 할당하면 단위시간당 김철수에게 주어지는 일이 위와 같다는 것을 의미한다. 이러한 형태의 서버는 초지수확률변수를 응대시간으로 하는 $M/H_c/1$ 모형을 사용하여 분석할 수 있다 (Lee, 1996).

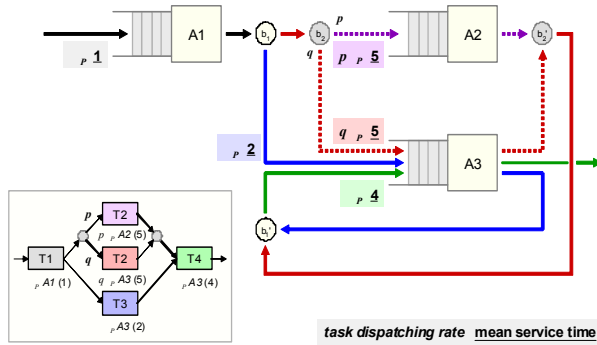


그림 4. 대기행렬 네트워크로 변환.

위의 <그림 4>에서 업무의 발생률 λ_p 가 1/15일 때 각각의 서버에 대한 부하를 구하면 다음과 같다.

$$\rho_{A1} = \lambda_p \cdot 1 = \frac{1}{15}$$

$$\rho_{A2} = p \cdot \lambda_p \cdot 5 = \frac{p}{3}$$

$$\rho_{A3} = q \cdot \lambda_p \cdot 5 + \lambda_p \cdot 2 + \lambda_p \cdot 4 = \frac{q}{3} + \frac{2}{5}$$

업무를 효율적으로 처리하기 위해서는 확률 p, q 를 적당히 조절해서 작업부하가 참여자들에게 골고루 부여되도록 해야 한다. 이것은 사용자들 중 가장 큰 값을 최소화하는 것으로 가능하다. 따라서 적절한 p, q 를 구하기 위해 선형계획법으로 바꾸어서 푼다. 위의 문제를 선형계획법으로 바꾼 것은 다음과 같다.

$$\min \max \{ \rho_{A1}, \rho_{A2}, \rho_{A3} \}$$

$$s.t. \quad 0 \leq p, q \leq 1$$

$$p + q = 1$$

이것을 풀이한 해는 다음과 같다.

$$p = 1, q = 0, \rho_{A1} = 1/15, \rho_{A2} = 1/3, \rho_{A3} = 2/5$$

따라서 위의 업무절차를 가장 효율적으로 수행하기 위해서는 p 는 1, q 는 0으로 주어야 하는 것을 알 수 있다. 이것은 사원인 박영희가 항상 거래실적조회 업무를 수행해야 하고 과장인 김철수는 결코 수행하면 안 된다는 것을 말해준다.

4. 작업부하 균형을 위한 알고리즘

앞에서 제시한 연구결과들을 바탕으로 참여자들의 작업부하

를 고르게 조절하는 Agent-Task Probability를 구하는 과정은 크게 다음과 같은 네 단계를 거쳐 이루어진다.

- i) 업무절차의 각각의 작업에 대해 Average Task Frequency를 구한다.
- ii) 각각의 Agent들에게 걸리는 작업부하를 계산한다. 이 과정에서 Agent-Task Probability를 나타내는 변수들을 사용한다.
- iii) 선형계획법을 사용하여 작업부하를 고르게 만드는 Agent-Task Probability를 결정한다.
- iv) 대기행렬 모형을 바탕으로 전체 시스템의 성능값을 예측한다.

Average Task Frequency를 계산하기 위해 먼저 제시한 선후행 관계와 선행 작업을 수행한 후 후행 작업을 수행할 확률, 그리고 반복구조를 사용한다. 이들 각각을 체계적으로 계산하기 위해 업무명세를 트리 구조로 변형하는 알고리즘을 개발하였다. 이와 같은 계산방법에 대한 연구는(Chang *et al.*, 2002; Disney 1981)에서 연구되었다. (Chang *et al.*, 2002)에서는 일반적인 BPM 시스템이 사용하는 한정된 작업명세에 대해 분석하였으며 본 연구에서는 그들 중 조건분기와 반복구조를 확장해서 사용하였다.

다음의 <그림 5>는 복잡한 업무절차에 알고리즘을 적용하여 Average Task Probability를 구한 결과이다. 구한 결과를 Average Task Frequency Vector F라고 나타내었다.

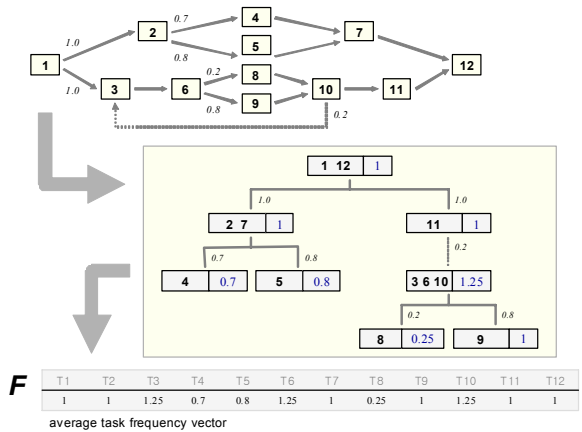


그림 5. Average Task Frequency의 계산.

참여자들이 수행하는 작업 처리시간의 통계치와 업무명세에서 기술한 선후행 관계의 확률을 사용하여 사용자들을 구한다. 참여자 A_i 가 작업 T_j 를 처리하는 확률인 Agent-Task Probability p_{AiTj} 에 T_i 의 Task Dispatching Rate를 곱하면 A_i 가 T_i 를 처리하는 비율을 구할 수 있다. 따라서 $M/H_c/1$ 모델에서 참여자 A_i 의 사용자들은 그 참여자가 처리할 수 있는 모든 작업 T_j 에 대하여 그 작업이 A_i 의 부하에 기여하는 정도를 합한 값이 된다. 다음은 사용자들을 구하는 식이다.

$$\rho_{Ai} = \sum_{j=1}^{n_r} S_{ij} p_{ij} r_j = \lambda_p \cdot \sum_{j=1}^{n_r} S_{ij} p_{ij} f_j$$

<그림 5>에서 제시한 업무절차의 참여자들이 <그림 6>의 (a)와 같은 업무처리 능력을 가진다고 가정한다면, 본 알고리즘을 적용하여 작업할당 확률을 구하면 <그림 6>의 (b)와 같은 결과를 얻을 수 있다. (b)의 표에서 ×로 표시된 항목은 비록 수행할 수 있지만 현재의 자원을 고려하면 업무의 효율을 위해 수행하면 안 된다는 것을 의미한다. 예를 들면 T4를 수행할 때 수행할 수 있는 능력이나 자격은 A2, A3, A4, A5 네 참여자가 가능하지만 A2에게 91%의 일을 맡기고 A3에게 9%의 일을 그리고 A4와 A5는 작업을 할당하지 말아야 한다는 정책을 알려준다. 결론적으로 참여자들의 작업부하는 모두가 70% 정도로 고르게 부여되었음을 알 수 있다.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
A1	7	-	-	-	5	8	-	2	4	5	-	6
A2	-	3	5	2	-	-	8	1	-	-	3	-
A3	-	2	7	2	-	-	8	1	-	-	2	-
A4	-	-	6	2	-	6	7	1	-	3	2	-
A5	-	-	7	3	5	-	8	1	2	-	1	8

(a) 특정 작업에 대한 참여자의 평균 수행시간 S

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	Utilization r(%)
A1	1.00	-	-	-	?	0.09	-	?	?	?	-	0.43	0.7027
A2	-	0.33	0.91	0.70	-	-	0.12	0.74	-	-	0.58	-	0.7027
A3	-	0.67	0.09	0.30	-	-	0.88	0.26	-	-	0.42	-	0.6990
A4	-	-	?	?	-	0.91	?	?	-	1.00	?	-	0.7027
A5	-	-	?	?	1.00	-	?	?	1.00	-	?	0.57	0.7027

(b) 업무발생 비율이 1/15일 때 작업할당 확률

그림 6. 업무처리 능력과 작업할당 규칙.

5. 실험 및 알고리즘 평가

본 연구에서 제시한 작업할당 규칙을 사용하여 <그림 5>의 업무 예제를 시뮬레이션하고 그 결과를 임의로 작업을 할당하는 정책의 결과와 비교하였다. <표 1>은 전체 업무의 평균 수행시간을 업무의 발생간격에 따라 비교해서 나타낸 것이다. 업무부하를 균일화했을 때는 발생간격이 줄어들어도 수행시간이 완만하게 증가하는 것을 알 수 있으나, 임의로 작업을 참여자에게 할당하는 경우는 평균 수행시간이 기하급수적으로 증가하는 것을 알 수 있다.

표 1. 발생간격에 따른 평균 수행시간

발생간격	30	20	17	15	13
임의선택	58	81	120	618	60192
부하균일화	68	94	119	153	241

부하를 균일화한 경우는 모든 경우에 부여되는 작업부하가 동일하다는 결과를 보인 반면, 임의로 작업을 할당한 경우에

는 업무 발생간격이 15일 때도 가장 많은 부하가 걸리는 참여자는 자신의 시간의 99% 이상을 작업에 소요하는 반면 다른 참여자들은 50~80%의 시간을 소요하는 것으로 나타났다. 즉 발생간격이 15 이상이 되면 임의로 할당하는 방식은 시스템의 성능을 전혀 보장하지 못하는 결과를 초래하였다. 발생간격이 긴 경우, 즉 업무의 강도가 약한 경우는 임의로 업무를 부여하는 것이 더 좋은 성능을 보여주는데, 이것은 작업부하를 균일화하지 않고 특정한 참여자에게 더 많은 일을 할당하기 때문이다.

6. 결론 및 추후 과제

본 연구를 통해 업무부하를 균일화하는 것으로 BPM에서 작업을 할당할 때보다 효율적인 방법을 찾을 수 있는 가능성을 탐구하였다. 비록 수많은 가정을 두기는 했지만 스케줄링 등 관련 분야에서 더 나은 방법을 찾을 수 있는 방법을 모색할 수 있는 기반을 제시하였다. 그리고 업무절차를 실행 전 사전에 자원의 용량까지 고려해서 평가할 수 있는 가능성도 제시하였고, 또한 자원을 계획하고 그를 통해 업무를 수행하는 비용을 절감할 수 있는 가능성을 높이는 데 기여할 수 있을 것으로 생각된다.

이 논문을 실제로 적용하는 데는 아직 해결해야 하는 문제가 많다. 먼저 사용한 대기행렬 모형의 가정이 아직은 현실에 적용하는 데는 많은 한계점을 둔다. 또한 여기서는 발생하는 모든 업무의 우선 순위가 동일한 것을 가정하였다. 하지만 현실에는 높은 중요도를 가지고 빠르게 처리해야 하는 경우도 있다. 이것을 하기 위해서는 보다 정교한 모델이 필요할 것으로 파악된다.

참고문헌

Attie, P. C., Singh, M. P., Emerson, E. A., Sheth, A., and Rusinkiewicz, M. (1996), Scheduling Workflows by Enforcing Intertask Dependencies, *Distributed Systems Engineering*, 3(4), 222-238.

Baker, K. R. (1974), *Introduction to Sequencing and Scheduling*, John Wiley & Sons, Inc, New York, NY.

Chang, D., Son, J., and Kim, M. (2002), Critical Path Identification in the Context of a Workflow, *Information and Software Technology*, 44(7), 405-417.

Davulcu, H., Kifer, M., Ramakrishnan, C. R., and Ramakrishnan, I. V. (1998), Logic Based Modeling and Analysis of Workflows, *Proceedings of the ACM Symposium on Principles of Database Systems*.

Disney, R. L. (1981), Queuing Networks, *American Mathematical Society Proceedings of Symposium in Applied Mathematics*, 53-83.

Goldratt, E. M. and Cox, J. (1992), *The Goal: A Process of Ongoing Improvement*, North River Press Publishing Corporation.

Hollingsworth, D. (1995), *The Workflow Reference Model*, Workflow Management Coalition.

Jin, L. J., Casati, F., Sayal, M., and Shan, M. C. (2001), Load Balancing In Distributed Workflow Management System, *Proceedings of the 2001 ACM*

- symposium on Applied computing.*
- Lee, H W (1996), *Queueing Theory*, Doseochulpan Kisul, Anyangsi, Kyunggido, Korea.
- Park, S D (1998), *Operations Research*, Third Edition, Minyoungsa, Seoul, Korea.
- Pinedo, M. (1995), *Scheduling: Theory, Algorithms and Systems*, Prentice Hall, Inc., Englewood Cliffs, NJ.
- Senkul, P., Kifer, M., and Toroslu, I. H. (2002), A Logical Framework for Scheduling Workflows under Resource Allocation Constraints, *Proceedings of the 28th VLDB Conference*.
- Simons J. V., Jr., and Simpson, W. P. (1997), An Exposition of Multiple Constraint Scheduling as Implemented in the Goal System, *Production and Operations Management*, 6(1), 3-22.
- Smith, H., Neal, D., Ferrara, L., and Hayden, F. (2002), The Emergence of Business Process Management, *A Report by CSCS Research Service*.
- Son, J. and Kim, M. (2001), Improving the performance of time constrained workflow processing, *Journal of Systems and Software*, 58(3), pp. 211-219.
- Spearman, M. L. (1997), On the Theory Of Constraints and the Go System, *Production and Operations Management*, 6(1), 28-33.