

BPM에서 업무할당방식이 업무효율성에 미치는 영향*

왕진국** · †배혜림***

Analysis of Business Process Efficiency Based on Task assignments

Zhenguo Wang** · †Hyerim Bae***

■ Abstract ■

This paper purposes analyzing how task assignments influence on business process efficiency, and improving business process efficiency in Business Process Management (BPM) environments. For this purpose, task assignment is categorized into dynamic assignment and static assignment. Dynamic assignment binds a user to a task at process run-time, whereas static assignments assigns a task to a user at process build-time. So far, the influence of task assignment methods has not been studied. We, in this paper, analyze the business process efficiency using two different assigning rule, in terms of cycle time as a efficiency measure. The comparison result can provide a guideline for a company who wants to employ commercial BPM systems.

Keywords : BPM, Simulation, Task assignment, BP efficiency

1. 서 론

최근 BPM(Business Process Management)이

업무 프로세스를 바탕으로 기업정보시스템을 통합하는 새로운 인프라로 자리 잡음으로써, 많은 기업들이 다양한 업무프로세스를 BPM을 통해 관리해

논문접수일 : 2008년 11월 07일 논문게재확정일 : 2009년 02월 28일

* 이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음.

** 스페이스 솔루션

*** 부산대학교 산업공학과

† 교신저자

나가는 추세에 있다[7]. 그 동안 BPM에 대한 개념이 1990년대 말에 처음 소개된 이후, 학계와 업계가 BPM의 성공적 정착에 주로 역량을 투입해왔으며, BPM 환경에서 어떻게 업무효율성을 증진시킬 것인지에 대한 문제는 다소 등한시 되어 왔다.

본 논문에서는 업무할당방식이 효율성에 미치는 영향을 분석함으로써, BPM 업무환경에서 효율성을 증진시키기 위한 방법론을 모색한다. 일반적으로 업무할당방식은, 프로세스 실행 전에 해당 단위 업무에 구체적인 업무 담당자를 할당하는 정적 할당(Static Task Assignment, STA)과 프로세스 모델링 시에는 역할과 같은 사용자의 집합에 업무를 할당 후 프로세스 실행 시에 구체적 담당자를 할당하는 동적 할당(Dynamic Task Assignment, DTA)으로 크게 구분할 수 있다. 대체로 상용 BPMS(BPM System)들은 DTA를 채택하여 사용함으로써, 실행시에 예기치 못하게 발생할 수 있는 문제들에 대처하고, 유연성을 높이는 쪽으로 활용하여 왔다.

그러나 지금까지 이런 업무 할당방식이 업무효율성에 어떤 영향을 미치는지에 대한 연구가 없었으며, 본 논문은 BP(Business Process) 효율성 척도에 대해서 이들 업무할당방식을 비교 분석한다.

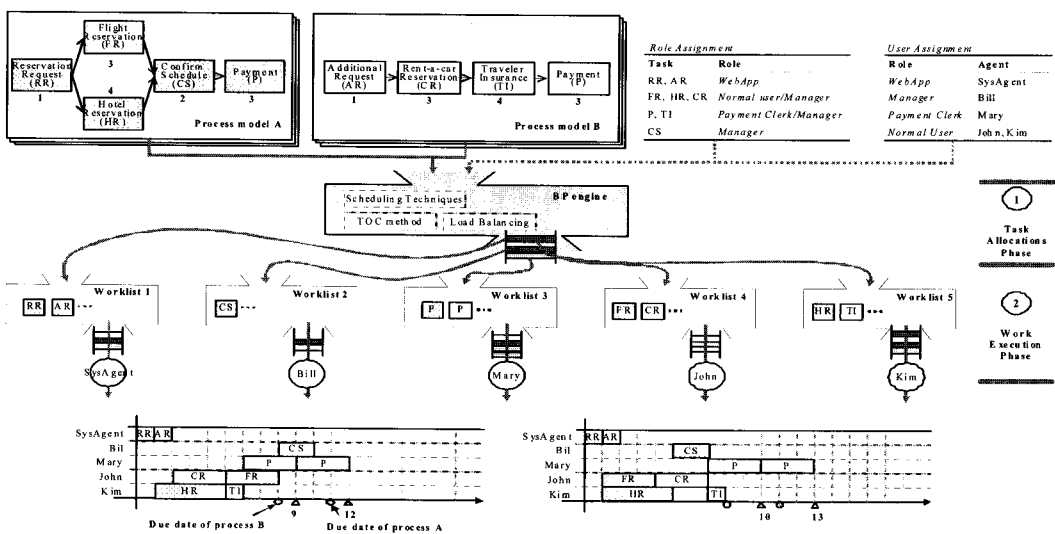
이러한 비교 분석의 결과는 상용 BPM를 도입하여 운영하는 기업에게 업무효율성을 높이는 가이드라인을 제시할 수 있다는 점에서 의미가 있겠다.

2. 배경이론

2.1 BP효율성

그동안 BPMS에 대한 다양한 연구들이 이루어지고, 상용 BPMS들이 활발하게 도입되었으나, 초기 접근법들은 프로세스를 정확하게 모델링하고 이를 오류 없이 실행하는데 초점을 맞추어 왔다. 그러나 BPM가 보편성을 확보하고, BPMS가 관리하는 프로세스의 절대적 수가 증가하면서, 프로세스 효율성 문제가 대두되기 시작하였다.

<그림 1>은 BPM 환경에서 프로세스 라이프 사이클 상에서 비즈니스 효율성이 어떻게 결정되는지를 보여준다. 먼저, 업무 프로세스 모델링 도구에 의해 다양한 프로세스 모델이 마련되면, 각 단위 업무를 처리할 수 있는 역할 그룹이 정해져서 이 역할 그룹에 소속된 사용자는 해당 단위업무를 실행 시에 처리할 수 있는 가능성을 가진다.



<그림 1> BP 라이프 사이클에서의 효율성

정의시에 마련된 이러한 정보를 토대로 엔진은 각 단위 업무를 적절한 사용자에게 일정한 규칙을 통해 할당하게 되는데, 그림에서 프로세스 A의 'Reservation Request(RR)' 업무를 시스템에이전트(Sys Agent)에게, 'Flight Reservation(FR)' 업무를 John에게 할당하였다. 이에 따라, 각 사용자는 자신의 Worklist에 쌓여있는 작업아이템들 중에서 우선순위를 정하여 작업을 수행하게 된다.

지금까지 설명한 프로세스의 실행과정은 그림 아래쪽에 있는 바와 같은 서로 다른 실행결과를 낳을 수 있다. 그리고 이러한 서로 다른 실행결과는 크게 두 가지 요인에 의해 결정된다. 먼저, 준비된 프로세스 모델을 토대로 실행시에 비즈니스 프로세스 엔진이 어떻게 자원을 할당하고 스케줄 하느냐가 실행효율성을 좌우한다. 만약 두 번째 작업 결과에서 FR과 HR이 모두 John에게 할당되었다면, Bill이 CS를 시작할 수 있는 가능시간이 7로 늦어지게 되고 이는 프로세스 효율을 떨어뜨리는 결과를 가져온다.

다음으로 개별 작업자가 어떤 우선순위를 가지고 자신의 Worklist에 쌓인 작업을 수행하느냐가 BP 효율성을 결정한다. 그림에 나타난 두 가지 서로 다른 결과는 엔진이 수행한 동일한 할당에 대해 각 사용자가 다른 우선순위를 적용했을 때, 나타난 효율성의 차이를 보여준다.

BP 효율성과 관련하여 주목해야할 또 다른 점은 어떤 효율성 척도를 사용할 것인가 하는 점이다. 프로세스 수행시간을 척도로 하는 경우는 왼쪽의 결과가 우수하지만, 만약 프로세스 납기 준수율을 수행도 평가척도로 삼는다면, 오른쪽의 결과가 더 우수하다고 볼 수 있다.

2.2 기존 연구

1990년대 후반 이후 BPM은 학계와 산업계가 공통으로 관심을 가지면서 급격하게 발전해왔다. 그러나, 초기의 BPM 관련 연구들은 프로세스 모델링과 실행의 정확성에 초점을 맞추면서 BPM의 성

공적 정착에 많은 노력을 기울였다. 그러나 점차 BPM이 관리하는 프로세스가 늘어나고 프로세스도 복잡해지면서 효율성에 대한 관심이 높아지기 시작했다[6].

Rhee et al.[4-6]의 연구를 통해서 기존 생산 영역에서 사용되던 다양한 방법론을 BP 영역에 적용하여 BP-DBR, PERT/CPM 등의 방법론을 고안하여 BP 효율성을 증대할 수 있는 방법을 제시하였다. 또한, Ha et al.[3]에서 업무부하 균형화와 업무 효율성간의 관계를 밝히고, 업무효율성을 높일 수 있는 실행규칙을 발견하였다. 그 이후에도 다양한 연구자들이 BP 효율성에 대한 연구들을 진행해왔다[8-10].

그러나 업무 할당방식이 구체적으로 BP효율성에 어떤 영향을 주는지에 대한 연구는 없었다. 그림에도 불구하고, 이러한 기존 연구들은 본 연구의 좋은 이론적 기반을 제시해주고 있다.

2.3 DTA와 STA의 정성적 비교

BPMS에서 프로세스를 구성하고 있는 단위업무를 수행하는 사용자를 언제 할당하느냐에 따라 동적할당(Dynamic Task Assignment, DTA)과 정적할당(Static Task Assignment, STA)으로 나눈다. 대부분의 상용 BPM에서 사용하는 업무할당 방식은 프로세스 모델링 단계에서는 각 단위업무를 수행하는 주체를 역할로 명시한 후 프로세스 실제 수행시에 적절한 사용자를 할당하는 DTA 방식을 취하고 있다. 반면, STA 방식은 프로세스를 설계할 때, 단위업무에 사용자를 바로 할당하는 방식이다.

〈표 1〉 DTA와 STA의 정성적 비교

비교기준	DTA	STA
구현의 편의성		○
실행 및 관리의 편의성		○
분석의 편의성		○
유연성	○	
실행효율 평가의 용이성		○
프로세스 엔진 성능에 대한 독립성		○

현재의 상용 시스템들이 DTA를 주로 채택하고 있는 이유는 프로세스를 모델링 하는 단계에서 사용자별 업무부하 등의 정보가 결정되지 않기 때문이다. 그러나 DTA가 BP효율성의 측면에서 STA보다 좋다는 것은 지금까지 입증되지 않았고, STA는 관리상의 목적에서 DTA에 비해 상대적으로 편리성을 제공한다. <표 1>에 DTA와 STA를 정성적으로 비교해두었다.

2.4 프로세스 모델

본 연구에서 사용하는 프로세스 모델은 Bae et al.[1]에서 제시한 다음의 프로세스 모델 표기법을 따른다.

정의 1 : 프로세스 구조모델

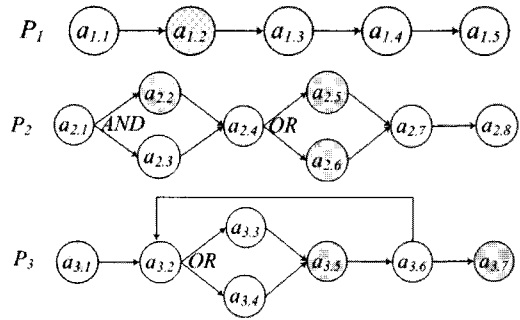
프로세스는 다음과 같이, 유방향 그래프 $p = (A, L)$ 와 분기 및 병합을 위한 라벨링 함수 $f(o)$ 로 정의된다.

- 단위 업무 집합 $A = \{a_i \mid i = 1, 2, 3, \dots, N\}$. 단, a_i 는 i 번째 단위업무이고 N 은 단위업무의 수이다.
- 링크 집합 $L = \{(a_i, a_j) \mid a_i, a_j \in A, \text{ 그리고, } (i \neq j)\}$. 단, (a_i, a_j) 는 단위업무 a_i 는 단위업무 a_j 보다 바로 앞서 실행되어야 함을 의미한다.
- 분기 단위업무 $a_i(S = \{a_k \mid (a_i, a_k) \in L, |S| > 1\})$ 에 대하여, 만일 모든 a_k 가 수행되어야 한다면 $f(a_i) = \text{'AND'}$ 이고 그렇지 않으면 'OR'이다.
- 병합 단위업무 $a_j(P = \{a_k \mid (a_k, a_j) \in L, |P| > 1\})$ 에 대하여, 만일 모든 a_k 가 수행되어야 한다면 $f(a_j) = \text{'AND'}$ 이고 그렇지 않으면 'OR'이다.

2.5 프로세스 예제

본 연구에서 DTA와 STA를 분석하기 위한 예제 프로세스로 정의 1을 따르는 <그림 2>의 세 가지 프로세스를 사용한다.

이 때, 각 단위업무의 수행에 활용될 역할정보와 각 역할이 수행 가능한 단위업무는 <표 2>와 같다.



<그림 2> 프로세스 예제

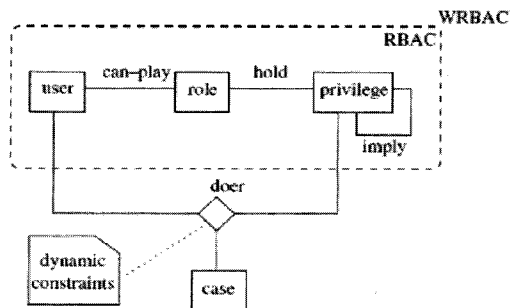
<표 2> 역할별 사용자 및 실행가능 단위업무

역할	사용자	실행가능 단위업무
R_1	w_1	$a_{1.1}, a_{2.2}$
R_2	w_2, w_3, w_4	$a_{1.2}, a_{1.4}, a_{3.2}$
R_3	w_3, w_4	$a_{1.3}, a_{1.4}, a_{2.1}, a_{2.5}, a_{2.6}, a_{3.1}, a_{3.4}, a_{3.7}$
R_4	w_5, w_6	$a_{1.5}, a_{2.3}, a_{2.4}, a_{2.7}, a_{2.8}, a_{3.3}, a_{3.5}, a_{3.6}$

3. DTA에 의한 업무할당 분석

3.1 역할기반 업무할당

BPMS에서 사용하는 역할기반 할당 모형은 RBAC 모형[11, 12]에 기반하고 있다. <그림 3>에서 보는 바와 같이 어플리케이션에서 사용가능한 권한(privilege)을 정의하고 이들의 구성에 따라 역할(role)을 정의하면 이들 역할을 수행할 수 있는 사용자(user)가 할당된다.



<그림 3> 워크플로우를 위한 RBAC 모형[11]

DTA의 프로세스 모델링 단계에서는 각 단위업

무의 수행주체로 역할을 할당하고, 실행시에 단위 업무 수행자가 결정된다.

3.2 DTA사용자 할당 규칙

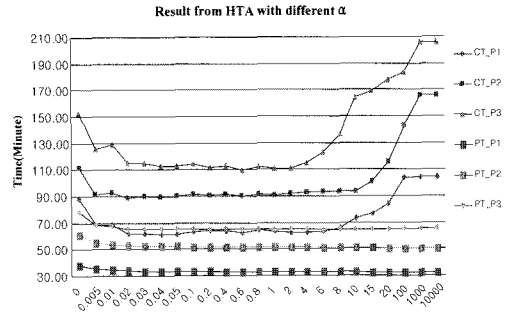
일단 프로세스 모델링 단계에서 단위업무를 수행할 사용자 그룹이 역할에 의해 지정되면, 프로세스 엔진은 실행시에 이들 사용자 중에서 다음과 같은 규칙들 중 하나를 이용해서 구체적인 사용자에게 업무를 할당한다.

- 무작위할당(RA, Random Access) : 사용자 그룹 모두에게 같은 확률로 무작위적으로 할당하는 방식
- 라운드 로빈(RR, Round Robin) : 사용자 그룹에 우선순위를 두어 우선순위가 높은 사용자가 할당될 확률을 높게주어 할당하는 방식
- 최소작업부하 할당(LW, Least Workload) : 해당 그룹의 사용자들 중에서 가장 적은 업무부하를 가진 자에게 해당 업무를 할당하는 방식
- 최소 작업시간 규칙(SPT, Shortest Processing Time) : 해당 작업자들 중에서 해당 작업을 최소 시간 내에 끝낼 수 있는 작업자에게 할당한다.

일반적으로 가장 많이 사용되는 방식은 LW 규칙이며, 본 연구에서는 다음과 같이 LW 규칙을 기본으로 하고, SPT 규칙을 조합한 하이브리드(h_i)값을 통해 업무를 할당하는 방식(DHTA)을 채택하기로 한다.

$$h_i = w_i + \alpha \cdot p_i \tag{1}$$

DTA의 결과를 확인하기 위하여 Arena 6.0 도구를 사용하여 제 2.4절에서 제시한 예제 프로세스를 시뮬레이션한 결과를 <그림 4>에 도시하였다. 그룹에서 세가지 프로세스(P1, P2, P3)에 대하여 α 값이 2부근에서 Cycle Time(CT)과 Processing Time(PT)의 효율성 척도에서 가장 좋은 결과를 보이는 것을 알 수 있다.



주) CT : Cycle time, PT : Processing time.

<그림 4> DTA에서의 BP효율성

4. STA에 의한 업무할당 분석

DTA를 적용하여 업무를 할당하는 가장 기본적인 기준 중의 하나는 업무부하를 평균화하는 것이다. STA는 프로세스 모델링 단계에서 업무를 사용자에게 할당하기 때문에 최적화 모형을 통해 다양한 목적식을 가지고, 할당을 결정할 수 있고, 본 장에서는 이러한 관점에서 STA를 분석하고자 한다.

4.1 STA를 위한 분석모형

STA에서 각단위업무의 작업자를 프로세스 모델링시에 정의하기 하는 데에는 해당 업무할당이 BP 효율성을 최적화하는 방향으로 할당되어야 한다. 이를 위해서는 다양한 목적함수가 고려될 수 있겠지만, 본 연구에서는 DTA와의 비교를 위해 업무부하의 평균화 관점에서 최적화 문제를 풀기로 한다.

먼저 STA에서의 업무 할당을 위해서는 다음과 같은 프로세스 분석 모형을 필요로 한다. 본 연구에서는 Ha et al.[2]에서 제시한 다음의 프로세스 분석 모형을 사용하기로 한다.

정의 2 : 프로세스 분석 모형

BP를 위한 분석모형은(ϕ, A, L, R, Re)와 같이 5개의 튜플로 구성한다. 단, A와 L은 각각 정의 1에서 설명한 단위업무와 링크집합이다.

- ϕ 는 프로세스 인스턴스의 도착율이다.

- R 단위업무 수행의 주체가 될 수 있는 사용자 및 소프트웨어 에이전트의 집합이다.
- $l \in L$ 일 때, e_l 은 구조적실행확률(Structural execution probability)를 의미한다.
- $Re \subseteq A \times R$ 는 단위업무(A)와 해당 단위업무의 수행이 가능 사용자의 집합이다. 그리고, $(a, r) \in Re$ 일 때, $t_{a,r}$ 은 r 이 a 를 수행하는데 소요되는 시간(Expected activity processing time)이다.

정의 3 : 기대실행빈도

단위업무 a 의 기대실행빈도(Expected execution frequency) f_a 는 프로세스가 한 번 수행되는 동안 기대할 수 있는 단위업무의 실행횟수이다.

정의 4 : 단위업무 도착율

단위업무 a 의 단위업무 도착율(Activity instance arrival rate) λ_a 는 프로세스들이 지속적으로 실행되는 환경에서 단위시간당 발생하는 단위업무의 발생빈도이다.

위의 정의에서 기대실행빈도와 단위업무 도착율 사이에는 다음의 식이 성립한다.

$$\lambda_a = \Phi \cdot f_a \quad (2)$$

4.2 가정과 표기법

STA 분석을 위해서는 제 2.4절에 제시한 프로세스 예제를 사용하며 다음과 같은 가정에 기반을 둔다.

- (1) 하나의 단위업무는 한 사용자에게 의해서 처리된다.
- (2) 단위업무 수행자는 한 번에 하나의 단위 업무를 처리한다. 따라서 수행자가 하나의 단위업무의 처리를 시작하면 끝내기 전까지는 다른 업무를 받을 수 없다.
- (3) 각 사용자는 하루에 주어진 시간동안만 일할 수 있다.

STA 분석을 위해 사용되는 변수들은 다음과 같다.

- i 프로세스 인덱스($i = 1, 2, \dots, I$),
- j 단위업무 인덱스($j = 1, 2, \dots, J_i$),
- k 단위업무 수행자 인덱스($k = 1, 2, \dots, K$),
- f_{ij} j -번째 단위업무의 기대 실행빈도,
- ϕ_i i -번째 프로세스의 도착률,
- l_k 수행자 k 의 용량,
- t_{ijk} i -번째 프로세스에서 j -번째 단위 업무를 수행자 k 가 실행할 때의 기대단위업무실행 시간,
- a_{ijk} 단위업무 실행가능 여부. 만약 업무수행자가 해당 단위 업무를 수행할 수 있다면 1, 그렇지 않으면 0,
- w_k 수행자 k 의 업무부하,
- $\max W$ 모든 업무수행자들의 업무부하의 최대치
- w 모든 업무 수행자들의 업무부하 평균,
- v 업무 수행자들의 업무부하 분산,
- h_i i -번째 프로세스의 업무부하
- x_{ijk} 단위업무의 실행여부, 만약 i -번째 프로세스의 j -번째 단위업무가 수행자 k 에 의해 실행되었다면 1, 그렇지 않으면 0.

4.3 수리모형

4.3.1 프로세스 전체업무부하최소화(MWW)

첫 번째로 사용될 STA의 업무할당 수리 모형은 MWW(Minimize Whole Workload)이다. MWW의 목적은 프로세스 전체의 업무부하를 최소화하는 업무할당을 찾아내기 위함이다. 이를 위해서 다음과 같은 수리 모델을 적용한다.

$$\min \sum_{i=1}^I h_i \cdot \phi_i$$

$$\text{s.t.} \quad \sum_{k=1}^K x_{ijk} = 1 \quad \forall i, j \quad (1)$$

$$x_{ijk} \leq a_{ijk} \quad \forall i, j \quad (2)$$

$$h_i = \sum_{j=1}^{J_i} \sum_{k=1}^K t_{ijk} \cdot f_{ij} \cdot x_{ijk} \quad \forall i \quad (3)$$

$$w_k = \sum_{i=1}^I \sum_{j=1}^J t_{ijk} \cdot \phi_i \cdot f_{ij} \cdot x_{ijk} \quad \forall k \quad (4)$$

$$w_k \leq l_k \quad \forall k \quad (5)$$

$$x_{ijk} = (0,1) \quad \forall i, j, k \quad (6)$$

식 (1)은 하나의 단위업무가 한 단위업무 수행자에 의해서만 실행될 수 있음을 의미하고, 식 (2)는 단위업무는 해당 업무를 수행 가능한 사용자에게만 할당되어야 함을 표현하고 있다. 식 (3)은 각 프로세스의 업무부하를 계산한다. 식 (4)는 각 업무수행자의 업무부하를 계산한다. 그리고 식 (5)는 각 업무수행자의 업무부하 상한을 표현하며, 마지막으로 식 (6)은 결정변수 값의 범위를 표현한다.

4.3.2 사용자간 업무부하분산최소화(MWV)

다음으로 취할 수 있는 업무할당은 업무부하의 편차를 줄이는 것이다. 다음 수리 모형은 업무부하의 분산을 최소화하는 MWV(Minimize Workload Variance) 할당을 찾는다.

$$\min v$$

$$\text{s.t.} \quad \sum_{k=1}^K x_{ijk} = 1 \quad \forall i, j \quad (1)$$

$$x_{ijk} \leq a_{ijk} \quad \forall i, j \quad (2)$$

$$w_k = \sum_{i=1}^I \sum_{j=1}^J t_{ijk} \cdot \phi_i \cdot f_{ij} \cdot x_{ijk} \quad \forall k \quad (4)$$

$$w_k \leq l_k \quad \forall k \quad (5)$$

$$\bar{w} = \sum_{k=1}^K w_k / K \quad (7)$$

$$v = \sum_{k=1}^K (w_k - \bar{w})^2 \quad (8)$$

$$x_{ijk} = (0,1) \quad \forall i, j, k \quad (6)$$

식 (1), 식 (2), 식 (4)~식 (6)은 MWW 모델과 동일하며, 식 (7)과 식 (8)은 각각 업무부하의 평균과 분산을 구하는 식이다.

4.3.3 사용자 최대 업무부하 최소화(MMW)

MMW(Minimize Maximum Workload)는 모든

수행자들 가운데 최대 업무부하를 최소화하는 전략이다. 다음의 수리모형을 통해 MMW 할당을 찾을 수 있다.

$$\min \max W$$

$$\text{s.t.} \quad \sum_{k=1}^K x_{ijk} = 1 \quad \forall i, j \quad (1)$$

$$x_{ijk} \leq a_{ijk} \quad \forall i, j \quad (2)$$

$$w_k = \sum_{i=1}^I \sum_{j=1}^J t_{ijk} \cdot \phi_i \cdot f_{ij} \cdot x_{ijk} \quad \forall k \quad (4)$$

$$w_k \leq l_k \quad \forall k \quad (5)$$

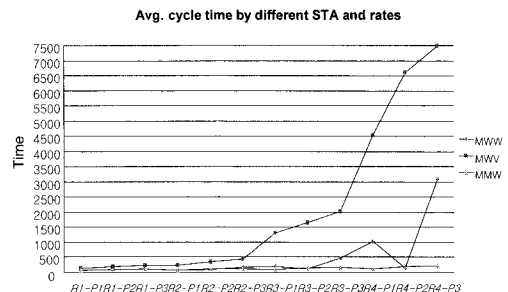
$$\max W \geq w_k \quad \forall k \quad (9)$$

$$x_{ijk} = (0,1) \quad \forall i, j, k \quad (6)$$

식 (1), 식 (2), 식 (4)~식 (6)은 이전 모델과 동일하며, 식 (9)는 최대부하를 제약하기 위해 추가한 식이다.

4.4 STA분석의 결과

ILOG CPLEX 10.2를 이용하여 STA 업무할당을 위한 수리 모형의 최적해를 구한 결과를 <표 3>에 도시하였다. 즉, 프로세스 모델링 단계에서 업무부하와 관련된 세 가지 지표를 최소화할 수 있는 업무할당을 찾을 수 있음을 알 수 있다.



<그림 5> STA에 의한 업무할당 결과

STA에 의해서 할당된 결과에 대해서 cycle time 측면의 효율성을 비교해보면 <그림 5>와 같다. 그래프에서 보는 바와 같이 cycle time에서는 MMV

〈표 3〉 STA에 의한 업무할당 결과

	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$	$a_{2,6}$	$a_{2,7}$
MWW	w_1	w_3	w_4	w_4	w_6	w_1	w_1	w_5	w_6	w_4	w_4	w_6
MWV	w_1	w_5	w_2	w_3	w_2	w_4	w_1	w_2	w_5	w_4	w_3	w_6
MMW	w_1	w_3	w_4	w_4	w_2	w_1	w_1	w_5	w_2	w_4	w_3	w_6
	$a_{2,8}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$	$a_{3,6}$	$a_{3,7}$	Avg.	SD	Max.	
MWW	w_5	w_3	w_1	w_2	w_3	w_5	w_2	w_3	40.8	7.14	52.1	
MWV	w_1	w_4	w_3	w_6	w_1	w_5	w_6	w_4	45.7	0.64	46.3	
MMW	w_5	w_3	w_1	w_2	w_3	w_4	w_6	w_4	43.2	0.94	44.5	

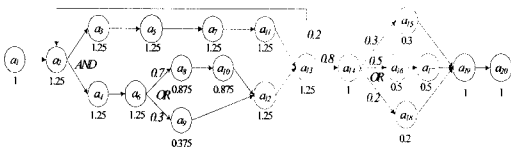
가 가장 좋은 성능을 보이고 있으며, 도착율이 증가할 수록 즉, 많은 수의 프로세스 인스턴스가 발생할 수록 격차는 커지는 것을 알 수 있다. 참고로, X축의 괄호 안의 수치는 프로세스의 도착률을 나타낸다.

5. DTA와 STA 간의 비교

앞서 설명한 바와 같이 BPM에서 업무할당의 주요 목적은 업무부하를 평준화하는 것이다. 그러나 이러한 업무부하의 평준화가 반드시 전체적인 업무효율성을 높이는 결과를 낳는다고는 보장할 수 없다. 특히, BP 효율성의 척도로 cycle time을 고려한다면, [3]에 밝힌 바와 같이 BP 효율성을 위한 새로운 실행규칙을 필요로 한다.

5.1 실험을 위한 프로세스

두 가지 할당방식에서 발생하는 효율성의 차이를 좀 더 명확히 하기 위해서 본 연구에서는 각 할당 방식에서 사용했던 기본 예제보다 좀 더 복잡함 예제를 통해서 실험하기로 한다. 다음 <그림 6>은 이러한 프로세스 예제를 보여주고 있다.

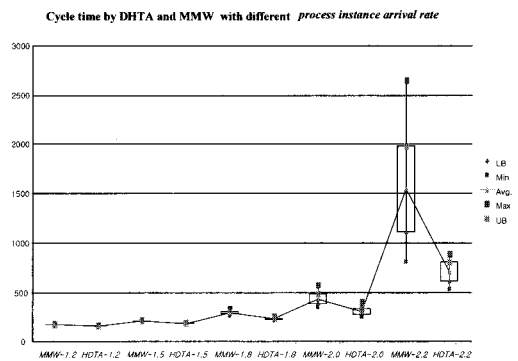


〈그림 6〉 복잡 프로세스 예제

<그림 6>에서 단위업무 아래에 표현한 수치는 각 단위업무의 기대실행빈도이다.

5.2 실험결과 요약

본 연구는 DTA와 STA라는 서로 다른 업무 할당 방식이 Cycle time 측면에서 효율성에서 어떤 영향을 미치는지를 시뮬레이션을 통해 알아보았다. DTA에서는 업무할당 규칙중 가장 일반적인 두 방식을 조합한 DHTA 방법론에서 찾은 최적의 α 값을 사용하고 STA에서는 cycle time 측면에서 가장 좋은 결과를 보여준 MMW를 사용하여 비교하였다. 그림의 X축에서 할당방식의 우측에 기입된 숫자는 도착율이다.



〈그림 7〉 DTA와 STA 간의 cycle time 비교

그림에서 보는 바와 같이 전체적으로는 Cycle time에 있어 DTA가 약간 우수한 성능을 보이고

있다. 그러나 그 차이는 비교적 근소하고, 다만, arrival rate이 커서 많은 프로세스가 발생할 수록 그 차이는 점차 커지는 것을 발견할 수 있다. 즉, 인스턴스가 많이 발생하는 상황에서 DTA가 STA보다 효율적이라고 볼 수 있다.

만약, 프로세스가 많지 않고, 관리상의 목적에 의해 프로세스 모델링 단계에서 사용자를 할당해야 한다면, STA를 사용하는 것이 업무효율을 떨어뜨리지 않는다고 볼 수 있으며, cycle time이 아닌 다른 효율성 척도를 최적화 해야한다면 STA를 활용하는 것이 필요하다.

6. 결 론

본 연구에서는 BPM 환경에서 사용할 수 있는 두 가지 업무할당방식인 DTA와 STA에 대한 분석을 수행하여 비교하고, 각 업무할당방식이 BP 실행 효율성, 특히 cycle time에 어떤 영향을 미치는지를 분석하였다. DTA와 STA는 BPMS에서 업무를 할당하는 가능한 두 가지 대표적인 시나리오이지만, 그동안 대부분의 상용 BPMS들은 모델링 단계에서 결정될 수 없는 요인들이 많다는 이유로 DTA를 주로 사용해왔다. 그러나, 만약 대기모형이나 확률이론을 통해 실행시에 발생하는 상황에 대해 기대할 수 있는 수치를 적절히 활용하여 STA를 사용할 수 있다면, 관리적인 측면에서 편리한 STA가 활용될 여지는 충분히 있다. 특히, STA를 활용하면, 프로세스 모델링 단계에서 특정 BP 효율성 척도를 최적화하는 할당을 미리 결정할 수 있다는 장점도 가진다.

BPMS 업무할당방식이 업무효율성에 미치는 영향분석을 위해 먼저 DTA에서 현재 상용 DBMS가 사용하는 규칙을 조합할 수 있는 업무할당 방식을 제시하여 cycle time을 최소화하는 DTA할당방식을 결정하였다. 다음으로 STA 할당방식을 위해 대기모형에서 사용되는 도착율을 활용하여 프로세스 총부하, 사용자별부하분산, 사용자최대부하의 세가지 목적함수를 최소화하는 수리모형을 제시하고

각 수리모형에 의한 할당결과를 획득한 후, 이들 중 역시, cycle time을 최소화하는 할당방식을 결정하였다. 마지막으로 DTA와 STA에서 cycle time관점에서 효율성을 비교 분석하였다. 분석의 결과를 토대로, 다음의 결과를 도출하였다.

- Cycle time의 관점에서는 DTA가 STA보다 다소 우수한 결과를 보였지만, 그 차이는 크지 않다.
- 도착율이 클수록 DTA와 STA사이의 효율성 차이는 점차 증가한다.

그 동안 BPM에서의 업무할당방식이 BP 효율성에 어떤 영향을 미치는지에 대한 연구가 없었다는 점에서 본 연구가 가지는 의미를 찾을 수 있지만, 몇 가지 추후연구를 필요로 한다. 첫째, cycle time이 아닌 다른 효율성 척도를 가지고 업무할당 방식을 분석할 필요성이 있다. 둘째, 프로세스 모델링 후에 프로세스 인스턴스의 발생시에 사용자를 할당하는 방식과 같은 Hybrid 업무할당 방식을 고려해서 분석할 필요가 있다. 셋째, 좀 더 다양한 업무 환경과 프로세스에 대해서 분석을 함으로써, DTA와 STA가 효과를 극대화하는 환경을 가이드라인으로 제시할 필요가 있다.

참 고 문 헌

- [1] Bae, H., W. Hu, W.S. Yoo, B.K. Kwak, Y. Kim, and Y.T. Park, "Document configuration control processes captured in a workflow," *Computers in Industry*, Vol.53, No.2 (2004).
- [2] Ha, B.H., J. Bae, and S.H. Kang, "Workload Balancing on Agents for Business Process Efficiency Based on Stochastic Model," *Business Process Management, Lecture Notes in CS*, Vol.3080(2004).
- [3] Ha, B.H., J. Bae, Y.T. Park, and S.H. Kang, "Development of process execution rules for workload balancing on agents," *Data and*

- Knowledge Engineering*, Vol.56, No.1(2006).
- [4] Rhee, S.H., H. Bae, D. Ahn, and Y. Seo, "Efficient Workflow Management through the introduction of TOC concepts," *Proceedings of the 8th Annual International Conference on Industrial Engineering Theory, Applications and Practice*, 2003.
- [5] Rhee, S.H., H. Bae, and Y. Kim, "A Dispatching Rule for Efficient Workflow," *Concurrent Engineering*, Vol.12, No.4(2004), pp. 305-318.
- [6] Rhee, S.H., N.W. Cho, and H. Bae, "A More Comprehensive Approach to Enhancing Business Process Efficiency," *Human Interface and the Management of Information, Lecture Notes in CS*, Vol.4558(2007).
- [7] Smith, H. and P. Fingar, "Business Process Management : the Third Wave," Meghan-Kiffer, Tampa, 2003.
- [8] Son, J.H., S.K. Oh, K.H. Choi, Y.J. Lee, and M.H. Kim, "GM-WTA : An efficient workflow task allocation method in a distributed execution environment," *Journal of Systems and Software*, Vol.67, No.3(2003), pp.165-179.
- [9] Song, B., G. Yu, D. Wang, D. Shen, and G. Wang, "An Efficient User Task Handling Mechanism Based on Dynamic Load-Balance for Workflow Systems," *Proceedings of the 5th Asia-Pacific Web Conference*, 2003.
- [10] Sun, Y., X. Meng, S. Liu, and P. Pan, "Flexible Workflow Incorporated with RBAC," *Computer Supported Cooperative Work in Design II*, Vol.3865(2006), pp.525-534.
- [11] Wainer, J. et al., "DW-RBAC: A formal security model of delegation and revocation in workflow systems," *Information Systems*, Vol.32, No.3(2007), pp.365-384.
- [12] Wang, B. and S. Zhang, "An Organization and Task Based Access Control Model for Workflow System," *Advances in Web and Network Technologies, and Information Management*, Vol.4537(2007), pp.485-490.