

# BPM에서의 업무효율성 향상을 위한 포괄적 접근법

## A More Comprehensive Approach for Enhancing Business Process Efficiency

이승현(Rhee, Seung-Hyun)\*, 조남욱(Cho, Namwook)\*\*, 배혜림(Bae, Hyerim)\*\*\*

### 초 록

세계 주요 기업들을 중심으로 비즈니스 프로세스를 보다 표준화하고, 명확화하려는 움직임이 어느 때보다 두드러지고 있다. BPM은 이러한 시도를 구현할 수 있는 적합한 도구로서 주목 받고 있다. 최근, BPM으로 구현된 프로세스를 보다 효율적으로 실행하는 것이 이슈로 대두되고 있다. 본 논문에서는 프로세스 효율성을 크게 프로세스 엔진과 업무담당자 관점에서 이를 해결하고자 한다. 프로세스 엔진 관점에서는, 엔진이 사용자에게 업무를 얼마나 효율적으로 전달하는지 바라본다. 업무담당자 관점에서는, 전달된 업무를 각 업무담당자가 어떻게 처리하는지에 주목한다. 본 연구는 각 단계에서만 효율성을 고려해왔던 과거 연구와는 달리, 두 단계에서 모두 효율적일 수 있는 방법론을 제안하며, 이를 실험을 통해 유효성을 검증한다.

### ABSTRACT

To survive in a global competition, many companies are trying to standardize and visualize Business Process (BP) by implementing Business Process Management (BPM). Recently, enhancing business process efficiency has become one of critical success factors. In this paper, we introduce a two-phase perspective of BP efficiency: Process Engine Perspective (PEP) and Task Performer Perspective (TPP). The former is related to allocation function of BP engine; it is mainly concerned with efficient task allocation to users. The latter phase influences efficiency depending on how users execute tasks assigned to them. Instead of considering each phase separately, we develop a comprehensive method considering the two-phase together, which is more effective for the BP efficiency. We carry out simulation experiment to show the combinational effect of the two phases.

키워드: 프로세스 효율성, 프로세스 엔진 효율성, 업무담당자 효율성, 시뮬레이션 실험  
Business Process Efficiency, Process Engine Perspective, Task Performer Perspective, Simulation

---

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

\* 서울대학교 산업공학과

\*\* 서울산업대학교 산업공학과

\*\*\* 부산대학교 산업공학과

## 1. 서 론

기업들은 업무프로세스(Business Process, BP)를 체계적으로 관리하기 위한 시스템 도구로 BPM(Business Process Management)을 도입하고 있다. 초기의 BPM 기술은 프로세스의 모델링과 실행에 있어서의 정확성 확보와 업무프로세스관리 자체의 올바른 구현에 초점을 두었다 [5][6][10]. 그러나, BPM 관련 기술이 더욱 성숙하고, BPM 시스템이 관리해야 하는 프로세스가 늘어나면서 업무 프로세스의 수행도 프로세스 통합, 업무 프로세스 효율성 증대와 같이 부가적인 이슈들이 발생하게 되었다.

본 연구는 효율적인 프로세스 실행을 위한 관점을 체계적으로 정리하고 이를 바탕으로 효율성을 증대시키는 포괄적 방법론을 개발하는 것을 목적으로 한다. 지금까지, 프로세스 실행의 효율성은 크게 두 가지 관점에서 다루어져 왔다고 볼 수 있다. 첫 번째로는 업무를 적절한 사용자에게 할당하는 엔진의 성능과 관련된 관점이며, 두 번째는 할당된 다수의 업무를 실제 처리하는 업무 담당자의 관점이다. 이러한 두 가지 측면에서 어떤 규칙을 사용하느냐 혹은 어떤 접근법을 사용하느냐는 업무 효율성에 영향을 미치게 된다. 그러나, 지금까지의 연구결과들은 이러한 개별 접근법의 효율성을 다루었을 뿐이며, 이들 접근법을 종합적으로 고찰하지 못했다. 따라서, 업무 효율성 증대라는 목적을 극대화 하기 위해서는 두 가지 접근법을 모두 고려한 좀 더 포괄적인 연구 방법론이 필요하다.

본 연구의 의미는 프로세스 효율성을 위해 사용자와 엔진의 두 가지 관점을 동시에 고려한다는 점에 있다. 각 관점에서는 여러 방법론들이 연구되고 있는 만큼, 본 논문에서 소개할 두 관점에서의 방법론은 하나의 조합 예가 된다. 따라서, 본 논문 이외에 효율을 추구할 수 있는 다양한 조합을 만들 수 있음을 강조한다. 보다 근본적인 것은 이러한 조합이 실제로 프로세스 개선으로 이어진다는 점으로서, 본 논문은 이러한 인식을 '프로세스 효율의 포괄적 접근법 (Comprehensive Rule for Process Efficiency, CRAPE)'이라 제안한다. 따라서, CRAPE는 여러 방법론들의 조합이 가능하며, 본 연구는 그러한 조합 중 한 가지 예를 보인다.

본 연구의 구성은 다음과 같다. 먼저 제 2장에서는 프로세스 효율성의 척도를 개괄하고 이와 관련된 연구들을 살펴보기로 한다. 다음으로 제 3장과 4장에서 프로세스엔진의 성능에 의한 프로세스 효율성 향상 방법과 사용자 처리 규칙에 의한 프로세스 향상 방법론을 각각 살펴본 후, 제 5장에서 이를 포괄적으로 고려하는 방법론을 제시한다. 마지막으로 본 연구의 타당성을 위한 시뮬레이션 실험결과와 결론을 각각 6장과 7장에 제시하기로 한다.

## 2. 배경 연구

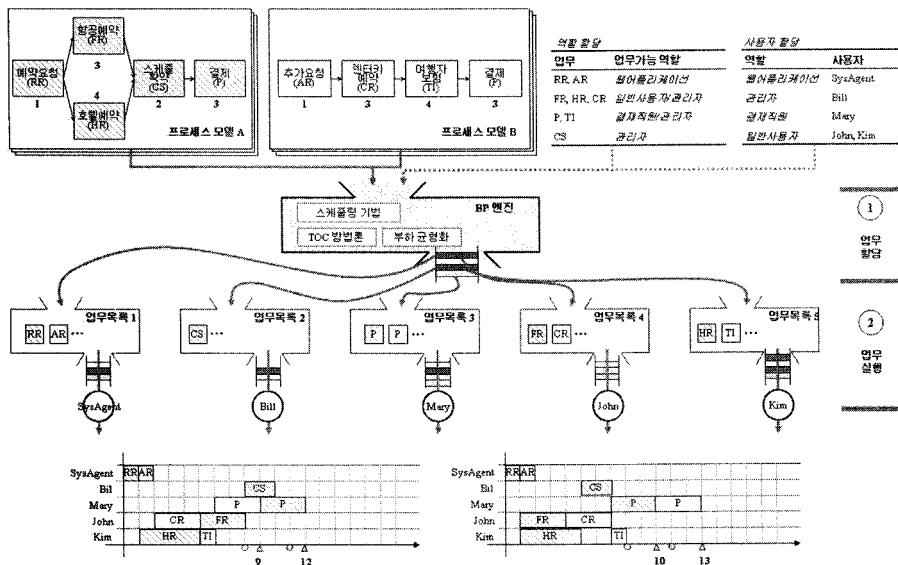
### 2.1 프로세스 효율성

최근까지 기업이 BPM을 도입할 때는

BPM의 대상이 되는 업무를 선별하기 위한 사전 작업을 수행하고 선별된 프로세스를 BPM으로 관리하는 것이 일반적이었다 [1][12]. 이러한 초기 단계에서는 정확한 업무 실행과 편리한 업무 자동화를 통해 BPM 도입의 정당성을 보이는 것이 급선무였다. 그러나, BPM 도입의 정당성이 확보되고 점차 BPM의 대상이 되는 업무프로세스의 양이 증가하면서 업무 프로세스의 효율성이 중요한 이슈로 부각되기 시작하였다 [1][11][15].

BP 효율성은 두 가지 서로 다른 관점에서 고려될 수 있다. 본 논문에서는 이 두 가지 관점을 각각 '프로세스 엔진 관점 (Process Engine Perspective, PEP)' 과 '업무수행자 관점 (Task Performer Perspective (TPP))' 으로 명명한다. <그림 1>은 BPM 시

스템이 BP를 수행하는 과정을 이러한 두 가지 관점에서 보여준다. 먼저, PEP는 BP엔진의 업무할당기능과 관련이 있다. <그림 1>의 좌측 상단에서 보는 바와 같은 프로세스 모델이 준비되면 BPM 시스템의 엔진은 이를 해석하고 프로세스를 이루고 있는 각 단위업무가 끝날 때마다 다음 단위 업무를 수행할 사람을 선택하여 업무를 할당한다[13][14]. 이 시점에서 엔진은 특정 규칙을 사용하게 되는데, 이 때 사용하는 규칙이 업무효율에 많은 영향을 미친다 [4]. BP 엔진이 업무를 할당 하기 위해서는 각 업무에 적합한 역할과 각 역할을 수행 가능한 사용자의 매핑이 <그림 1>의 우측 상단과 같이 마련되어 있어야 한다. BP 엔진은 이러한 매핑을 기반으로 해서 적절한 사용자에게 업무를 할당하게 되는데 이러한 업무



<그림 1> BPM 시스템의 업무처리 과정

할당을 통한 업무처리결과의 예들이 그림의 하단에 표현되어 있다. 만약 주어진 환경에서 엔진이 취하는 규칙이 다르다면 효율성은 달라지게 되며, 이는 효율성을 위한 엔진의 성능과 직결된다. 예를 들면, 그림 왼쪽 하단의 실행결과에서 FR 업무는 현재 John에게 할당이 되어 있는데 이를 Bill에게도 할당할 수 있다. 만약, Bill에게 할당한다면, FR 업무는 좀 더 일찍 시작될 수 있으며, 이는 전체 프로세스의 효율을 높일 것으로 기대 할 수 있다.

다음으로, BP 효율성은 사용자가 자신에게 할당된 업무를 처리하는 방식에 따라 달라질 수 있다. 대개 한 사용자는 특정시점에 다수의 업무 아이템을 자신의 업무목록에 가지게 되는데, 이는 BPM시스템이 여러 프로세스 모델을 관리하고, 하나의 모델로부터 많은 수의 인스턴스가 발생하기 때문이다 [10]. 이러한 환경에서 사용자가 채택하는 우선순위의 규칙은 전체 프로세스 효율에 큰 영향을 미치게 된다. <그림 1>에서 하단의 두 개의 차트는 사용자가 자신의 업무를 수행함에 있어 서로 다른 우선순위 규칙을 적용함으로써, 생기는 업무효율의 차이를 보여주고 있다. 즉, John이 자신에게 할당된 업무들 중 CR과 FR을 모두 수행할 수 있는 때 CR을 먼저 수행하느냐, FR을 먼저 수행하느냐로 인해 프로세스 효율이 서로 달라지게 된 것이다.

업무 효율과 관련하여 한가지 더 언급해야 하는 중요한 이슈는 업무효율의 척도이다. 가장 쉽게 채택할 수 있는 척도는 전통적인 생산분야의 스케줄링기법에서 사용되

어 온 것들인데, 프로세스 완료시간 (Process completion time, service time, mean time), 지연율 (tardiness), 효율 (utilization) 등이다. <그림 1>의 두 가지 실행결과를 다시 살펴보면, 프로세스 완료시간 관점에서는 왼쪽의 결과가 더 효율적이라고 할 수 있다. 반면, 만약 due-date가 중요한 척도라면 오른쪽의 경우가 더 효율적이라고 할 수 있다. 다시 말해, 조직이 추구하는 효율성의 종류에 따라 적절한 전략이 고려될 필요가 있다. 본 논문에서는 1차적으로 프로세스 완료시간의 최소화를 목적으로 한다.

## 2.2 관련 연구

1990년대 이후로 업무 프로세스와 관련된 다양한 연구들이 진행되어 왔으며, 이들 연구는 프로세스 모델링, 실행, 모니터링 및 예외처리와 관련된 내용이다. 초기의 BP 효율성에 대한 접근법은 분석 도구들을 사용하였다. Reijers 는 Petri Net에 기반을 둔 SWN (Stochastic Workflow Net)을 이용하여 'throughput time'의 관점에서 효율성을 분석하는 방법론을 제시하였다 [7]. Reijers는 또한 [8]에서 워크플로우 관리 시스템의 적용 효과를 인도기간 (lead time), 서비스 시간 (service time), 대기시간, 그리고 효율 (utilization)의 정량적 관점에서 분석하였다.

이후의 많은 연구들은 업무프로세스에 스케줄링 기법을 도입하는데 초점이 맞추어져 왔다. 먼저, Baggio, Wainer와 Ellis등은 스케줄링 기법을 워크플로우에 도입하여 지연 업무의 수를 줄이고 이를 기존 연구와 비교

하였다 [1]. 하병현 등은 개별적인 워크리스트를 가진 업무수행자들의 업무부하의 균형을 가능하게 하는 프로세스 실행규칙을 개발하였다 [4]. Eder 등은 '개인일정(Personal Schedule)'이라는 새로운 개념을 도입하였고 이는 다가올 작업에 대한 정보를 미리 제공하는 효과를 발생한다 [2]. 이러한 정보를 가지고 turn-around 시간 및 시간제약을 범하는 비율을 줄일 수 있음을 보였다. Zhao는 시간이 중요한 프로세스에서의 turn-around 시간을 예측하는 방법론을 제시하고 이를 이용해 프로세스의 각 단위업무에 예상 수행 시간을 할당하였다 [15]. 이승현 등은 PERT/CPM 법을 활용하여 워크플로우 프로세스에서의 여유시간을 계산함으로써, BP 효율성을 증대시키는 방법론을 제시하였다 [10] [11]. 이 논문에서는 여유시간을 통해 사용자들이 급박한 업무를 우선 처리하도록 유도함으로써, 전체적인 업무효율을 높이도록 하였다. 이승현 등은 또한 [9]에서 BP 효율성을 높이는 다른 방안을 제시하였는데, 이는 TOC 이론에서 제시된 DBR(Drum, Buffer, and Rope) 방법론을 사용하였다. 기존의 생산분야의 스케줄링 등에 주로 응용되던 방법론인 DBR방법론을 활용하여 BPM 엔진이 프로세스 인스턴스의 투입속도를 조절하면서 BP 효율성을 향상시키는 방법론이다.

이러한 기존의 방법론들을 21절에서 제시한 두 가지 단계에 입각해서 분류해볼 수 있다. 먼저, Baggio의 접근법 [1]과 이승현의 두번째 접근법[9]은 PEP에 관여된 방법론으로 업무를 할당하는 엔진의 성능과 관련

이 있다. 다른 방법론들은 대부분 TPP 단계에 해당하는 방법론으로 사용자의 업무처리 기준에 관련된 방법론이라 할 수 있다.

### 3. 프로세스 엔진 관점 (Process Engine Perspective, PEP)

본 연구에서는 전술한 두 개의 관점을 동시에 고려하는 포괄적 방법론을 제시한다. 이를 위하여 먼저, PEP 관점에서는 골드렛 박사가 제안한 TOC 방법론 [3]을 활용하며 [9]에서 제시한 BP-DBR 방법론을 활용한다. 본 장에서는 DBR을 BP에 적용하기 위한 구체적인 방법론을 드럼, 버퍼, 로프 각각의 개념을 중심으로 기술한다. BP-DBR의 개념은 특정 자원의 능력이 제한적인 경우에 매우 유용하다. 이러한 능력부족을 드러내는 특정자원이 곧 전체 시스템의 수행도를 결정하게 된다. TOC에서 이러한 자원을 일컬어 CCR (Constraint Capacity Resource)라고 한다. 일반적으로 생산분야에서는 특정 공정 또는 기계가 CCR로 분류되지만, 본 연구에서는 업무목록에 가장 많은 업무부하를 가진 작업자를 CCR로 정의키로 한다. 이때, 드럼은 CCR의 업무처리 속도가 되며 로프는 드럼에 따라 입력되는 원자재의 속도를 조절하는 의사소통 장치로 정의할 수 있다. 그리고 버퍼는 CCR의 앞에 특정량의 작업을 둬으로써 CCR이 멈춤 없이 작업을 하도록 하는 장치이다. 본 장에서는 BP에 이들 세가지 개념을 적용하는 방법론을 이어지는 절에서 각각 설명한다.

**3.1 Drum**

드럼을 계산하기 위해서는 다음의 두 가지의 서로 다른 경우가 고려되어야 한다.

- CASE I: CCR이 다 수의 에이전트(작업자)를 가진 작업그룹에 소속되어 있고 한 가지 작업만을 수행하는 경우
- CASE II: CCR이 여러 종류의 서로 다른 작업을 처리해야 하는 경우

드럼을 계산하기 위해서는 CCR을 찾아내야 하는데, BP에서 CCR은 업무 부하가 가장 큰 작업자( $u_{CCR}$ )로 정의할 수 있다. 업무 부하를 계산하는 일은 비교적 단순한 계산이므로 [9]에서 제시한 방법을 토대로 계산하기로 한다. [9]에서는 Case II만을 다루었으나, 본 논문에서는 이를 확장하여 두 가지의 경우를 모두 고려하기로 한다.

첫 번째 Case에 대해서  $u_{CCR}$ 이  $k$  명 ( $k \geq 0$ )의 에이전트를 가진 작업 그룹에 속해있고  $a_i$ 이라는 작업을 수행한다고 가정한다. 또한, 해당 작업의 예상 수행시간은  $ET_i$ 이라고 가정한다. 하나의 에이전트는 단위시간당  $1/ET_i$ 개의 작업을 처리할 수 있으므로 드럼은 다음과 같이 정의할 수 있다.

$$Drum(d) = \frac{k}{ET_i} \cdot (task/time) \quad (1)$$

반면, 두 번째 Case에서는  $u_{CCR}$ 이 각각의 예상수행시간을 가지고 있는 여러 업무에 관여하므로 드럼을 계산하는 식이 좀 더 복잡해진다. 이러한 경우의 드럼계산을 위해서는 단위업무수 (unit number of task)의 개념 도입이 필요해진다. 단위업무수란, 단위

시간에 처리하는 업무의 수를 가장 짧은 예상시간을 가지는 업무의 상대적인 숫자로 표현한 것이다. 예를 들어, 작업  $a_1$ 은  $u_{CCR}$ 이 관여하는 작업들 중에서 가장 짧은 시간인 1을 예상수행시간으로 가진다면, 예상수행시간이 2인 작업  $a_2$ 는 단위업무수가 2가 된다. 따라서,  $u_{CCR}$ 이  $j$ 개의 작업에 관여하는 경우에는 각 작업의 단위업무수는 다음과 같이 계산된다.

$$Unit\ number\ of\ task\ a_j\ (UN_j) = \frac{ET_i}{ET_{min}}$$

(2)

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

만약,  $P_i$ 를 작업  $a_i$ 가  $u_{CCR}$ 에게 할당될 확률로 정의한다면, 드럼은 다음과 같이 정의될 수 있다.

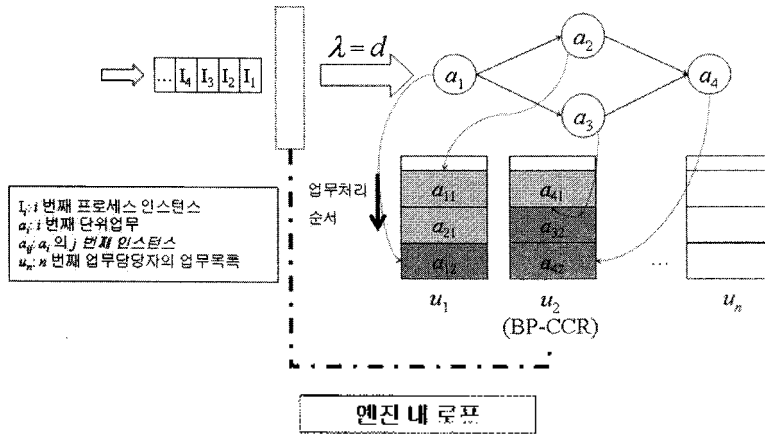
$$Drum(d) = \sum_{i=1}^j P_i \cdot \frac{k}{ET_i} UN_i = 1/ET_{min}(task/time) \quad (3)$$

**3.2 Rope 와 Buffer**

일단 드럼이 결정되고 나면, BPM 엔진은 드럼에 맞추어 프로세스 인스턴스의 투입속도를 제어하게 된다. 이를 DBR관점에서 해석하면, BPM엔진이 Rope로서 자재투입 속도 조절 역할을 하게 되는 셈이다. BPM엔진은 다음과 같이 프로세스 인스턴스의 속도( $\lambda$ )를 드럼과 동기화함으로써 로프의 역할을 한다.

$$\lambda = d \text{ (instances/time)} \quad (4)$$

Rope의 역할에 있어 인스턴스 투입속도



〈그림 2〉 로프의 기능

외에도 작업 처리 순서를 통해서 업무의 효율성을 증대시킬 수 있다. 즉, CCR의 인스턴스 처리순서를 가능한 한 다른 작업자들도 지키도록 하는 것이다. 이를 통해서 CCR자원의 업무 부족을 최소화하는 것이 필요하다. 〈그림 2〉는 이러한 Rope의 역할을 구체적으로 보여주고 있다.

버퍼는 CCR이 처리할 업무가 없어 작업을 중지하게 되는 경우를 배제하여 전체적인 프로세스의 작업이 계속되도록 유지하기 위한 장치이다. 그리고, 이 버퍼의 크기는 충분히 클수록 좋지만, 운영비용과 재고비용을 고려하여야 한다. 제약이론에서 버퍼의 크기는 전체적인 효율성에 영향을 미친다는 점에서 이를 결정하는 것은 중요한 주제이지만, 작업환경에 따라서 매우 가변적이고 수치적으로 결정하기는 매우 어려운 문제로 알려져 있다. 따라서, 본 연구에서는 버퍼의 크기를 결정하는 것은 경험적 방법

론을 이용해서 주어진 환경에서 가장 효율적인 버퍼의 크기를 제시하고자 한다.

#### 4. 업무 작업자 관점 (Task Performer Perspective, TPP)

본 장에서는 TPP관점에서 BP의 효율성을 다룬다. 본 연구에서는 이를 위하여 [10]에 의하여 제안된 우선순위 규칙을 활용한다. BP를 실행하는 동안 업무를 수행하는 작업자는 자신의 업무목록에 리스트된 업무들 중에서 선택하여 작업을 수행한다. 즉, 나름대로의 우선순위를 부여하여 작업을 수행한다. 만약, 이러한 작업들이 가지는 상대적 중요도를 표현할 수 있다면, 중요한 작업을 먼저 수행하게 함으로써 업무의 효율성을 높일 수 있다.

이러한 우선순위 규칙가운데서 긴박한 업

무를 먼저 처리하는 것은 시간적인 관점에서 효율을 증대시키는데 합리적인 기준이라 볼 수 있다. 긴급성을 표현함에 있어 프로젝트 관리 방법론에서 자주 사용되어온 PERT/CPM의 여유시간 개념이 매우 효과적이다. 본 절에서는 작업자 관점에서 업무의 우선순위를 결정함으로써 업무 효율을 높이는 방법에 대하여 설명한다.

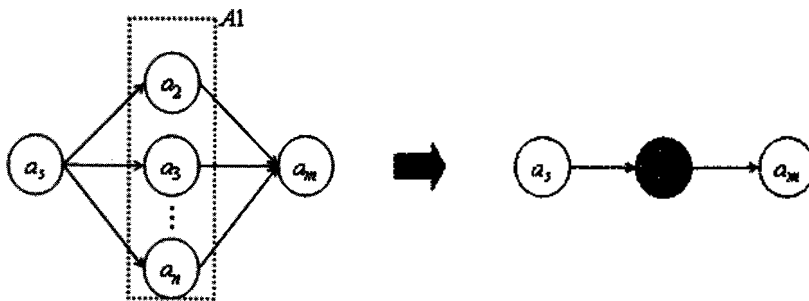
### 4.1 여유시간의 계산

각 작업의 긴급함에 대한 정보를 제공하기 위해서는 먼저 프로세스 구조상에서 주경로(Critical path)를 찾고 여유시간을 계산하여야 한다. 각 작업에 대한 여유시간의 계산은 PERT/CPM법에 기반을 두지만, BP의 구조와 프로젝트 관리에서 다루는 공정 네트워크의 구조는 상이해서 단순 적용에는 문제가 있다. PERT/CPM에서 네트워크에서 병렬로 분기될 때의 모든 공정이 실행되는 것을 의미하지만, BP의 분기는 대안의 경로를 허용한다. 다시 말해서, PERT/CPM에서 다루는 모든 분기는 AND이지만 BP는

OR분기를 포함한다. 따라서, BP의 AND분기는 일반적인 PERT/CPM법으로 계산할 수 있으나, OR분기에 대해서는 특별한 방법이 고안되어야 한다. <그림 3>과 같은 병렬구조가 OR구조라고 가정할 때, 실행시에는 세 개의 경로 중 하나가 실행되는 것이 일반적이므로 오른쪽 그림과 같이 분기된 각 경로를 하나의 경로로 표현한 후 이에 대한 예상 시간을 구함으로써, 여유시간을 계산한다.

OR분기에서 하나로 표현된 경로의 예상 수행시간은 다음 <표 1>과 같은 방법으로 계산한다.

위의 표에서와 같이 OR 병렬구조내의 모든 업무들은 같은 여유시간을 가지는 것으로 계산할 수 있다. 이렇게 계산된 여유시간을 사용자에게 주면 사용자는 자신의 업무 목록에서 여유시간이 가장 작은 업무부터 수행하도록 한다. 이러한 우선순위 규칙을 '최소 여유시간 (Least Slack Time, LST) 규칙' 이라고 명명한다. LST 규칙에 대한 좀더 자세한 사항에 대해서는 [10]과 [11]을 참고하기 바란다.



<그림 3> OR분기와 대표 단위업무



(표 1) OR분기의 경로 예상 수행시간 계산법

형식	예상 수행 시간	의미
NOR	$\sum_{i=1}^n \prod_{j=1, j \neq i}^n (\frac{\lambda_j}{\lambda_i + \lambda_j}) ET_i$	일반적 OR분기로 하나의 경로라도 완료되면 전체가 완료된 것으로 간주
POR	$\begin{cases} P_i \cdot ET_i & n=1 \\ P_i \cdot ET_i \cdot \sum_{j=2}^n [P_j \cdot (1-P_j)] P_i \cdot \{\sum_{k=2}^n ET_k\} & n \geq 2 \end{cases}$	경로실행의 우선순위가 정해져있어 우선순위가 높은 경로부터 실행하고 하나의 경로라도 완료되면 전체가 완료된 것으로 간주
COR	$\sum_{i=1}^n (P_i \cdot ET_i)$	각 경로의 조건이 명시되어 있어 조건을 만족하는 경로가 실행되고 하나라도 완료되면 전체가 완료된 것으로 간주

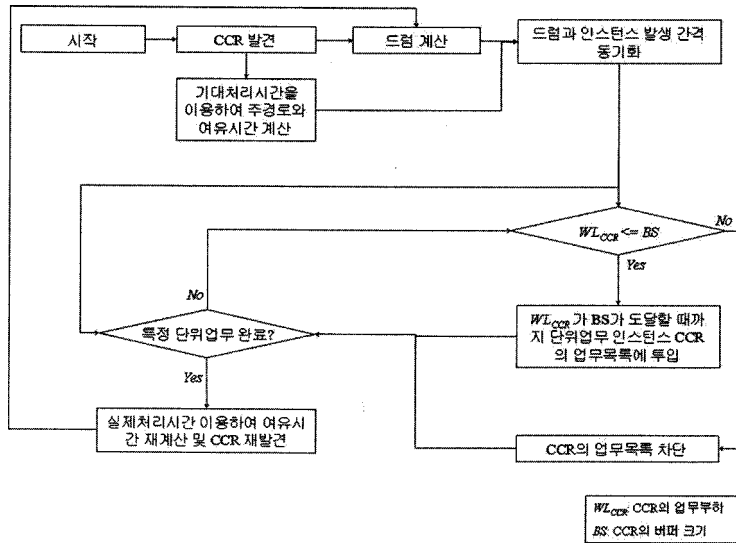
$ET_i$ : 단위업무  $a_i$ 의 예상수행시간,  $\lambda_i$ :  $i$ -번째 경로의 서비스율,  $P_i$ :  $i$ -번째 경로가 수행될 확률

### 5. 두 관점의 포괄적 적용

앞의 두 장에서 살펴본 두 관점, PEP와 TPP는 프로세스의 효율성을 향상시키기 위한 각각의 영역에 집중한다. PEP관점에서는 BP-DBR 방법론을 이용해 업무를 배분, 할당하는 기능에 집중하고 TPP관점에서는 할당된 업무를 어떤 우선순위로 처리할 것인가에 집중하여 전반적인 BP 효율성을 증대시킨다. 개별 관점에서 이미 그 효율성을 검증 받은 바 있지만, 두 방법론을 동시에 적용했을 때의 효과에 대해서는 알려진 바가 없다. 두 가지의 관점이 프로세스의 시작과 실행에 관련된 전체 절차를 확장하기 때문에 본 연구에서는 두 가지 방법의 조합을 '프로세스 효율의 포괄적 접근법 (Comprehensive Rule for Process Efficiency,

CR4PE)라고 명명하기로 한다. 전반적인 절차는 다음 그림과 같다.

먼저, 프로세스가 시작되기 전에 CCR을 찾아서 드럼을 계산한다. 그리고, 주경로와 여유시간을 찾는다. 프로세스가 실행을 시작하면, 로프가 작동하기 시작한다. 로프의 역할을 수행하는 BP 엔진은 인스턴스의 투입 속도를 드럼의 속도와 맞추면서 조절한다. 이 때, 버퍼도 관리되는데 버퍼의 크기는 CCR의 업무부하에 따라 조정된다. 사용자는 할당된 업무들 중에서 계산된 여유시간이 가장 짧은 업무부터 처리한다. 각 업무가 완료되면, 실행시에 실시되는 각 기능은 재계산 및 조정된다. 즉, 여유시간이 재계산되고 각 작업자의 업무부하를 계산해서 CCR을 재확인하는 절차를 거친다. 또한 업무의 투입량은 CCR의 업무부하에 따라 적절히



〈그림 4〉 프로세스 효율을 위한 포괄적 방법론

조정된다. 결과적으로 CRAPE 법은 이러한 과정을 통해 지속적으로 업무의 투입과 실행을 조정, 제어한다. 다음 장에서 이러한 포괄적인 접근법의 유용성을 실험을 통해 분석하였다.

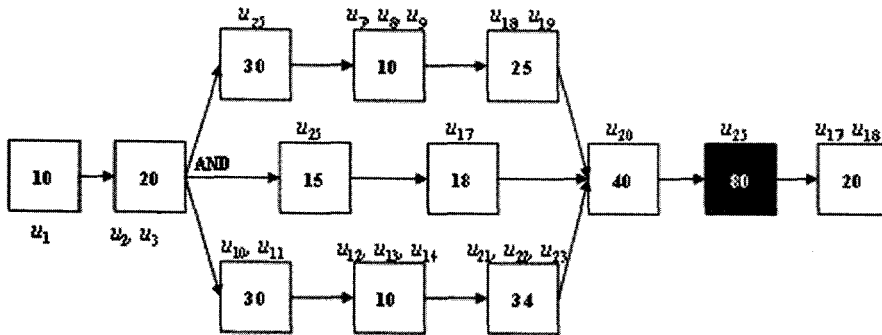
## 6. 시뮬레이션 실험

### 6.1 실험 모델과 환경

이미 수행된 기존 연구에 의해, PEP에서 적용되는 BP-DBR은 상당 규모의 업무부하가 특정 단위업무에 집중될 때 효과적이다. 한편, TPP에서 적용되는 LST 규칙은 업무부하가 크고, 프로세스 구조가 병렬적일수록 효과적이다. 본 실험에서는 두 방법론이 모

두 효과적인 환경을 고려하여 실험 모델을 〈그림 5〉와 같이 설계하였다. 〈그림 5〉에서 각 박스는 단위업무를 가리키고 박스 내에 숫자는 기대처리시간을 의미한다. 또한, 각 단위업무에 대한 업무담당자가 박스 위에 표시되어 있다. 색칠된 박스는 실험 모델 내에 가장 업무부하가 큰 단위업무를 뜻하며, 이에 대한 업무담당자는 CCR로서,  $u_{cc}$ 이다.

시뮬레이션 모델은 Arena 6.0 소프트웨어로 구현하였다. 시뮬레이션은 총 10,000분 실행하였으며, 안정적인 데이터를 확보하기 위해 초기 500분 동안 발생한 데이터는 고려하지 않았다. 인스턴스 발생간격과 각 단위업무 처리시간은 지수분포를 따른다고 가정하였으며, CCR의 버퍼 크기는 1,800으로 설정하였다. 그리고 〈그림 5〉의 모델에 대해 다음 4가지 모델과 같이 적용 규칙을 달리하



〈그림 5〉 시뮬레이션 실험 모델

여, CR4PE의 효과를 비교 분석하도록 하였다.

- 모델 I: CR4PE이 적용된 모델
- 모델 II: FIFO 규칙만 적용된 모델
- 모델 III: BP-DBR만 적용된 모델
- 모델 IV: LST 규칙만 적용된 모델

### 6.2 프로세스 효율성 결과

네 가지 모델에 대해 프로세스 효율성의 관점에서 결과를 측정하고, 이를 비교 분석하였다. 측정에 사용된 지표는 프로세스 완료 시간, 완료 인스턴스 개수, 진행중인 프로세스 인스턴스 개수를 사용하였다. 프로세스 완료 시간은 평균의 개념으로서, 이를 통해 모델 간 시간 효율성을 비교할 수 있다. 인스턴스 개수는 보다 자세한 비교 지표로서, 프로세스의 생산성을 의미하기도 한다. 결과는 <표 2>와 같다.

실험 결과를 살펴보면, 프로세스 완료시간과 진행 인스턴스 개수에서 CR4PE가 가장 좋은 결과를 보였다. 각각에 대해 살펴보면,

프로세스 완료시간의 경우 각 규칙이 독자적으로 적용됐을 때에 비해, 그 효과의 합이 결합된 형태로 모델 I의 결과가 나타났다.

한편, 완료 인스턴스 개수의 경우는 LST 규칙, BP-DBR의 단독 사용으로도 현저한 개선효과를 볼 수 있다. 두 규칙 간의 결합에 대한 시너지 효과를 크게 볼 수 없었는데, 그 이유는 진행 인스턴스 개수에서 찾아볼 수 있다. LST 규칙에서는 현재 진행 중에 있는 프로세스 인스턴스 양이 많다. BP-DBR과는 달리 로프 기능이 없어 투입 자체를 통제하지 않기 때문에, 인스턴스 발생 간격에 따라 결정되어 모델 II와 크게 다르지 않다. 이러한 상황에서 LST의 효과로 인해 완료 인스턴스 수가 증가하였다. 한편, 모델 III에서는 진행 중인 인스턴스의 수가 현저하게 감소하였는데, 로프의 기능과 버퍼 관리로 인한 결과이다. 두 규칙이 결합되면, 로프와 버퍼로 인해 투입 인스턴스가 통제되어 진행 인스턴스 수가 LST에 관계없이 TOC 규칙에 대해 결정된 것임을 예측할 수 있는데, 이것은 모델 I에서 확인할 수 있다.

〈표 2〉 프로세스 효율성 측정 결과

모델	프로세스 완료시간	완료 인스턴스 개수	진행 인스턴스 개수
모델 I	2747.458	81	26.662
모델 II	4223.30	58	108.27
모델 III	2995.951	77	28.038
모델 IV	3780.851	80	96.663

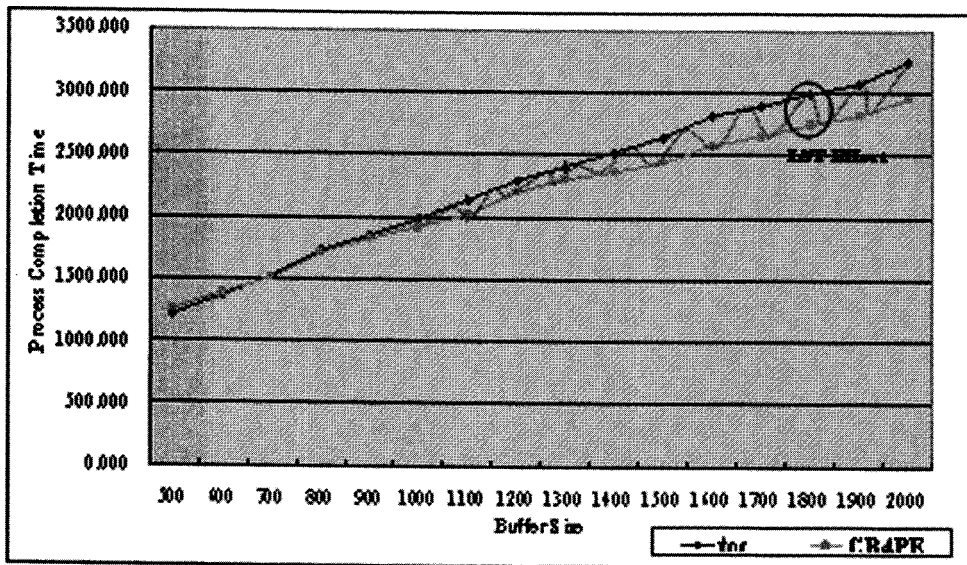
결국, 완료 인스턴스는 그만큼 진행 인스턴스가 많아야 그 개선의 여지가 있는데, 이미 BP-DBR로 인해 그 숫자가 급격하게 감소하였으므로, 진행 프로세스 개수의 개선 폭도 감소할 수밖에 없다. 따라서, 완료 인스턴스에서는 CRAPE가 프로세스 완료시간만큼 효과적이지 않았다. 요컨대, CRAPE는 완료 인스턴스 개수에서 두 규칙 간 시너지 효과를 현저하게 보이지 않지만, 여전히 모델 II보다는 매우 효과적이다.

### 6.3 Influence of Buffer Size

한편, 본 논문에서는 버퍼 크기에 대한 각 모델의 실험 결과를 부가적으로 분석해 보았다. 이미 6.2에서 설명하였듯이, 진행 프로세스 개수에서 LST와 TOC에서는 큰 차이가 있었다. 이는 TOC가 가지고 있는 버퍼와 로프의 기능 때문인데, 특히, LST 규칙은 업무담당자의 업무목록을 관리하는 기능으로

로, 버퍼의 크기가 LST 성능에 영향을 줄 수 있을 것으로 보였다. 즉, 버퍼의 크기가 LST와 TOC의 시너지에 영향을 준다고 예측할 수 있다. 이에 따라, 버퍼의 크기를 다르게 주어 각 모델의 결과를 살펴보도록 한다.

위 〈그림 6〉에서 보듯이, 버퍼의 크기가 커질수록 CRAPE와 TOC 간의 프로세스 완료시간 격차가 벌어지는 것으로 나타났다. 버퍼의 크기가 크게 되면, CCR이 처리하기 위해 대기하는 업무량이 증가하게 된다. 대기하는 업무의 양이 크지 않다면, 각 업무에 대한 여유시간의 다양한 정도도 크지 않을 것으로 볼 수 있다. 즉, LST로 인한 효과를 기대할 수 없다는 것이 되며, 이는 〈그림 5〉에서도 드러난다. 반면, 버퍼의 크기가 크면 업무목록에 존재하는 업무의 양도 많아지고 그만큼 다양한 여유시간의 값이 각 업무에 존재한다고 볼 수 있다. 이에 따라, LST는 버퍼의 크기가 클수록 그 활용 여지가 커지



〈그림 6〉 버퍼 크기의 영향

게 되어 프로세스 효율성에 기여하게 된다. 따라서, CR4PE는 TOC가 제공하는 버퍼의 기능을 LST가 활용할 여지가 클수록 그 시너지가 발휘된다고 할 수 있다.

## 7. 결 론

본 논문에서는 비즈니스 프로세스의 효율성과 관련된 이슈를 엔진의 효율적인 업무할당의 관점과 작업자의 효과적인 업무처리라는 두 가지 관점으로 분류하고 이들 관점을 종합적으로 고려하여 프로세스 실행 효율을 극대화하는 방법론에 대하여 고찰하였다. 본 연구가 제안하는 방법론은 개별적인 관점에서의 프로세스 효율을 살펴본 과거의 연구들을 확장하여 좀 더 포괄적인 관점에

서 전체 프로세스 효율을 증대시킬 수 있는 방법론을 고안하였다는 점에서 찾을 수 있다. 이러한 효율성 증대에 대한 정당성은 시뮬레이션 실험을 통하여 제시하였고, 각 관점이 미치는 효율성의 의미를 분석하였다.

본 연구는 몇 가지 점에서 추후 연구를 필요로 하는데, 먼저 좀 더 다양한 패턴의 프로세스 모델이 고려되어야 한다. 각 개별적인 관점에서 비즈니스 프로세스의 효율을 극대화할 있는 패턴들이 고려된다면, 좀 더 효과적인 효율성 증대 방안이 강구될 수 있을 것이다. 다음으로 각 관점에서의 좀 더 다양한 규칙들을 활용할 필요가 있다. 본 연구는 한 가지 규칙만을 각 관점에서 살펴보았지만 몇 가지 규칙들이 더 고안될 수 있을 것을 기대한다.

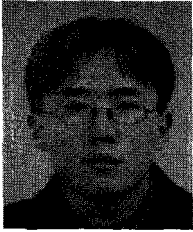
---

## 참 고 문 헌

---

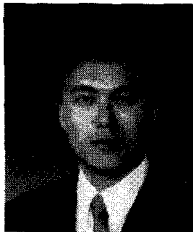
1. Baggio, G., Wainer, J., Ellis, C., "Applying Scheduling Techniques to Minimize the Number of Late Jobs in Workflow Systems," Proceedings of the 2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, 2004, pp. 1396 - 1403.
2. Eder, J., Pichler, H., Gruber, W., and Ninaus, M., "Personal Schedules for Workflow Systems," Lecture notes in computer science, Vol. 2678, 2003, pp. 216-231.
3. Goldratt, E. M.: The Goal, North River Press, New York, 1992.
4. Ha, B. -H., Bae, J., Park, Y. -T., Kang, S. ?H., "Development of process execution rules for workload balancing on agents," Data & Knowledge Engineering Vol. 56, No. 1, 2006, pp. 64-84.
5. Hammer, M., "The Agenda: What Every Business Must Do to Dominate the Decade," Crown Business, New York, 2001.
6. Kim, Y., Bae, J., Bae, H., Kang, S., "Automatic Control of Workflow Processes using ECA Rules," IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 8, pp. 1010-1023, 2004.
7. Reijers, H. A., "Design and control of workflow processes: Business process management for the service industry," Lecture notes in computer science, Vol. 2617, 2003.
8. Reijers, H. A., van der Aalst, W. M. P., "The effectiveness of workflow management systems: Predictions and lessons learned," International Journal of Information Management, Vo. 25, No. 5, 2005, pp. 458-472.
9. Rhee, S. -H., Bae, H., Ahn, D., Seo, Y., "Efficient Workflow Management through the introduction of TOC concepts," Proceedings of The 8th Annual International Conference on Industrial Engineering Theory, Applications and Practice (IJIE2003), Las Vegas, America, 2003.
10. Rhee, S. -H., Bae, H., Kim, Y., "A dispatching rule for efficient workflow, Concurrent Engineering - Research and Applications," Vol. 12, No. 4, 2004, pp. 305-318.
11. Rhee, S. -H., Bae, H., Choi, Y., "Enhancing the Efficiency of Supply Chain Processes through Web Services," Information Systems Frontier: Special Issue on From Web Service to Services Computing, Forthcoming
12. Smith, H., Fingar, P., "Business Process Management - The Third Wave," Meghan-Kiffer Press, Florida, 2003.
13. Workflow Management Coalition, Terminology & Glossary, WfMC-TC-1011, Lighthouse Point, Florida, 1999.
14. Workflow Management Coalition, Process Definition Interchange, WfMC-TC-1016-P, Lighthouse Point, Florida, 1999.
15. Zhao, J. L., Stohr, E. A., "Temporal workflow management in a claim handling system," SIGSOFT: Software Engineering Notes, Vol. 24, No. 2, 1999, pp. 187-195.

## 저 자 소개



이승현  
2007. 2  
관심분야

(E-mail: shyun.rhee@gmail.com)  
서울대학교 산업공학과 (학사)  
SCM, 비즈니스 프로세스



조남욱  
1994. 2  
1996. 2  
2001. 2  
2001~2002  
2003~2004.  
2004. 3~현재  
관심분야

(E-mail: nwcho@snu.ac.kr)  
서울대학교 산업공학과 (학사)  
서울대학교 산업공학과 (석사)  
Purdue University 산업공학과 (박사)  
Lucent Technologies, Microelectronics Group, MTS  
삼성SDS, 전자제조컨설팅팀, 책임 컨설턴트  
서울산업대학교 산업정보시스템공학과 조교수  
정보시스템, 비즈니스 프로세스 관리, e-manufacturing



배혜림  
1996. 2  
1998. 2  
2002. 8  
2002~2003.  
2003~2004.  
2004. 9~현재  
관심분야

(E-mail: hrbae@pusan.ac.kr)  
서울대학교 산업공학과 (학사)  
서울대학교 산업공학과 (석사)  
서울대학교 산업공학과 (박사)  
삼성카드 정보기획팀  
동의대학교 인터넷비즈니스학과 전임강사  
부산대학교 산업공학과 조교수  
BPM, eAI, 웹서비스, 물류 프로세스 관리