TOC를 이용한 업무프로세스 효율성 향상

배혜림[a], 이승현[b], 조남욱[c]

[a] *부산대학교 공과대학 산업공학과*
부산 금정구 장전동 산30, 609-735
Tel: +82-51-510-2733, Fax: +82-51-512-7603, E-mail:hrbae@pusan.ac.kr

[b] *서울대학교 공과대학 산업공학과*
서울 관악구 신림9동 산56-1, 151-742
Tel: +82-2-880-7063, E-mail:Shyun.Rhee@gmail.com

[c] *서울산업대학교 산업정보시스템공학과*
서울 노원구 공릉2동 172, 139-743
Tel: +82-2-970-6448, Fax: +82-2-974-2849, E-mail:nwcho@snut.ac.kr

Abstract

본 논문은 BPM 시스템에 의해서 관리되는 비즈니스 프로세스의 효율성을 증대시키는 새로운 방법으로 TOC(Theory Of Constraint) 이론에서 사용되는 DBR(Drum, Buffer, Rope) 개념을 적용한다. TOC 이론은 주로 생산공정의 효율증대를 위한 방법론으로 인식되어 왔지만, 비즈니스 프로세스에서도 적용이 되도록 BP-DBR (Business Process – DBR) 방법론을 개발한다. 또한, 이를 통해 실제 금융권에서 사용되는 프로세스 효율성이 증대되는 것을 시뮬레이션 실험을 통해 보인다.

Keywords:

Business Process Management, Theory of Constraint, Workflow, Process Efficiency

Introduction

오늘날 기업의 비즈니스 상황은 시장 환경이 철저하게 고객중심으로 변화함에 따라 기업 내, 기업 간 거래 등에서 더욱 복잡한 프로세스의 수행을 요구하고 있다[12][13]. 이에 따라, 기업의 프로세스를 효과적으로 관리하는 것이 기업 경쟁력 강화에 필수적이라는 인식이 자리잡게 되었다[12]. 실제로 복잡한 프로세스를 정형화된 형태로 모델링하고 이를 명확히 하여 기업을 운영하면, 시장의 요구에 민첩하게 반응하여 궁극적으로 보다 많은 이윤 창출에 기여할 수 있다[12][32]. 이러한 프로세스 중심 경영 하에서는 참여자들이 전체 프로세스 목표를 이해하고 자신의 업무가 줄 수 있는 영향에 관심을 가지며, 프로세스 효율성과 성과를 염두에 두어야 한다[12][28]. 최근에는 많은 기업들이 업무절차를 체계적으로 설계, 관리, 개선하는 활동을 지원하는 총체적인 관리 방법론으로서 업무프로세스관리(Business Process Management System, BPM)기법을 도입하고 있다. 이를 실현하기 위해 업무프로세스관리 기법을 구체화하고 효율적인 업무 환경을 지원하는 소프트웨어 시스템을 업무프로세스관리 시스템(Business Process Management System, BPMS) 이라고 정의한다[32]. 업무프로세스관리 시스템은 기업 안팎으로 산재한 프로세스를 정의, 구현, 개선하는 등의 모델링을 수행하고, 그 결과를 통해 정확한 순서대로 프로세스를 진행하고, 자동화함은 물론 다른 조직과의 상호연동을 편리하게 해준다[30][32].

기존 업무프로세스관리 시스템이 처음 도입되던 시기에는 관리가 용이한 프로세스에 대해 선별적으로 BPM을 적용하는 시도가 이루어졌으나, 점차 전사적인 프로세스 관리에 대한 필요성이 대두되면서 프로세스의 효율에 대한 중요도가 높게 인식되어 가고 있다. 본 논문에서는 업무프로세스의 효율성을 전사적인 프로세스 관점에서 관리하기 위하여 제약이론(TOC, Theory of Constraints)의 개념을 도입한다. 특히, 지금까지 생산공정의 효율성 제고에 주로 응용되던 DBR (Drum-Buffer-Rope) 방법을 업무프로세스관리 영역에 적용함으로써 프로세스의 전체적인 효율을 높이는데 기여한다.

본 논문에서는 먼저, 본 연구와 관련된 연구를 소개하고, 기존 연구와의 차별성을 언급한다. 다음으로 제안한 방법론의 적용대상이 되는

프로세스를 설명하고, 생산 공정과 BPM 프로세스와의 차이점을 분석한다. 그리고, TOC의 개념을 구체적으로 비즈니스 프로세스에 적용하는 방법론과 이를 통한 프로세스 실행 절차를 설명한다. 마지막으로, 개발된 방법론의 효과를 시뮬레이션 실험을 통해 다양한 관점에서 분석한 후 결론을 제시한다.

Backgrounds

Process Model

BPM 시스템이 프로세스를 자동으로 수행하기 위해서는 컴퓨터가 이해할 수 있는 형태로 프로세스가 표현되어 있어야 한다[16]. 이렇게 표현한 것을 본 논문에서는 '프로세스 모델'이라고 한다. 다시 말해, 업무프로세스들은 BPM 시스템 기반에서 프로세스 모델로 표현된 것으로서, 본 논문에서 언급하는 업무프로세스를 프로세스 모델로 이해할 수 있다. 본 논문은 프로세스 모델 자체의 정확성 문제를 다루는 것은 아니므로 최소화된 형태의 모델을 정의하기로 한다. 본 논문의 프로세스 모델은 프로세스 구조와 속성에 의해 표현되며, 구조는 프로세스를 이루는 단위업무와 이들간의 선후 관계를 표현하는 링크를 말한다[29][36].

Definition 1 (프로세스 모델)
프로세스 모델은 다음을 만족하는 유방향 그래프 $P = (\Omega, A, L, U, R)$이다.

- Ω는 프로세스 인스턴스 발생간격.
- $A = \{a_i \mid i = 1, \dots, N\}$은 단위업무의 집합이다. a_i은 i-번째 단위업무이며, N은 P 에 포함된 단위업무의 총개수이다.
- $L \subseteq \{(a_i, a_j) \mid a_i, a_j \in A \text{ and } i \neq j\}$는 링크 집합이다. 원소 (a_i, a_j)는 단위업무 a_i가 단위업무 a_j보다 바로 앞서 실행돼야 함을 의미한다.
- $U = \{u_k \mid k = 1, \dots, K\}$는 업무참여자 집합. K는 전체 사용자의 수
- $R \subseteq A \times U$는 역할집합, 특정 단위업무에 참여하는 업무참여자 역할.

정의1의 내용과 프로세스 실행에 필요한 각종 속성을 포함한 프로세스 모델은 BPM 엔진(engine)에 의해 실행 및 관리된다[21][37]. BPM 엔진은 프로세스 모델을 바탕으로 실제 프로세스(인스턴스)를 발생시키게 되는데, 수행해야 할 단위업무들을 적절한 업무담당자에게 할당하여, 이들을 처리하도록 한다. 이 때, 개별 업무담당자에 할당된 업무집합을 업무목록(work list)이라 한다. 이러한 업무 할당은 업무할당규칙(Task Allocation Rule)이라는 규칙에 의해 이루어진다. 일반적으로 사용되는 규칙에는 후보 담당자에게 모두 할당한 뒤 한 사람이라도 먼저 완료할 경우, 다음 단계로 진행하는 규칙(경합 규칙)과 가능한 사용자 중 가장 부하가 적은 사용자에게 업무를 할당하는 규칙(최소 부하 규칙) 등이 있다.

Theory of Constraints

제약이론은 시스템의 목적 달성을 저해하는 제약자원을 찾아내고 이를 극복하여 기업의 수익창출에 기여하는 경영기법이다[7][8][9]. 이를 위하여, 현금창출율 증가, 재고 감소, 운영비용 절감에 주력한다[9]. 제약자원이란, 현금창출율을 결정하는 요인이며 나아가서는 기업의 목적인 이익증대에 결정적으로 영향을 미치는 요인이다. 일반적으로 제약이론은 다음 표 1과 같은 5단계의 절차를 거쳐 진행되며, 관련 세부 내용은 [7][8][35]을 통해 참고하기 바란다.

표 1 – 제약이론의 5단계절차

단계	세부 내용
1단계	제약자원(Constraint Capacity Resource, CCR)을 발견한다.
2단계	제약자원을 철저히 활용하는 단계이다.
3단계	비제약자원들을 제약자원에 종속시키는 단계이다.
4단계	제약자원의 숨겨진 능력까지 철저히 활용한 후에 투자를 통해 이 자원 자체의 능력을 향상시키는 과정이다.
5단계	제약자원이 다른 공정으로 바뀌는지 주의해 가며 첫 단계부터 반복한다

한편, 제약이론 가운데 효율적인 스케줄링 관리를 지원하는 방법론으로 DBR (Drum-Buffer-Rope)이 있다[19]. DBR은 생산과정을 군대의 행진에 비유한 방법론으로, 먼저 드럼(Drum)은 제약자원의 업무처리속도를 의미하며, 전체 공정의 생산 속도를 결정한다. 다음으로, 버퍼(Buffer)에는 생산의 통계적 변동으로 인한 제약자원의 작업 중단을 방지하는 제약버퍼(Constraint Buffer)와 조립 공정이 존재할 때 출하 시기가 지연되는 것을 방지하는 조립버퍼(Shipping Buffer) 등이 있다[8][9][19]. 마지막으로, 로프(Rope)는 제약자원과 시작 공정 간의 통신 장치로서 제약자원의 공정 속도에 맞춰 시작 공정에 자재를 투입하는 속도를 조절한다.

DBR은 효율적인 시간관리와 자원관리를 목적으로 하며, 주로 제조 공정의 스케줄링을 위하여 버퍼 크기나 우선순위규칙과 같은 할당 문제와 관련해서 연구되어 왔다[5][6][27]. Process turnaround time[9][35]은 물론이고, 납기 준수율[22], 재고[5], 업무부하[6], 버퍼 사이즈[27] 등이 구체적인 연구 영역에 포함된다. 본 연구에서는 제약이론의 적용 절차에 기반한 DBR 기법을 적용하여 효율적인 업무프로세스 관리 기법을 제시한다.

기존연구 고찰

BPM과 관련하여 최근까지 다양한 관점에서 많은 연구들이 진행되어 오고 있으며, 그 양이 점차 증가되는 추세이다. 표 2는 BPM에서 지금까지 주요하게 다루어져 온 연구들을 요약하고 있다.

표 2에서 보는 바와 같이 다양한 관점에서 BPM에 대한 논의가 진행되어 왔다. BPM 관련 연구 초기에는 시스템을 정착시키기 위한 방편으로 주로 프로세스 표현 방법, 통합 방법 등 시스템을 통해 실제로 프로세스가 실행 가능하도록 하는 영역에 편중되어 연구가 진행되었다. 그러나, BPM이 다루는 프로세스들이 점차적으로 복잡해지고 전사적 BPM의 도입 추세가 생겨나면서 이들의 효율적인 실행이 중요한 문제로 대두되었다. 그러나, 프로세스 효율성과 관련한 구체적인 연구는 활발하게 이루어지지 않고 있다[26][38]. 따라서, BPM 프로세스 실행을 효율적으로 할 수 있는 방법들을 연구하여 BPM 기능을 향상시킬 필요가 있다.

표 2 – BPM 관련 주요 연구분야

연구 영역	연구 내용
프로세스 모델링	현실 세계에서 표현되어야 하는 케이스들을 분류하고 거기에 맞는 프로세스 모델링 방법에 대해 설명한 연구 [1]
	프로세스 예외 상황을 정리한 뒤, 새로운 프로세스 디자인 시 발생 가능한 예외상황에 이를 참고하는 Knowledge-based 모델링 방법을 제안[18]
프로세스 통합	eBusiness 환경에서 기업 간 negotiation을 가능하게 하는 시스템 아키텍처 제안[14]
	로제타넷, ebXML 등 통합 표준을 이용하여 기업 간 프로세스를 상호 운용하도록 하는 연구 [31]
모니터링	사용자 별 필요한 정보를 customizing하고 BPM에서 제공하는 정보와 통합하여 모니터링 정보를 제공하는 연구 [14]
	BPM이 관리하는 통합 정보를 이용하여 BAM (Business Activity Monitoring)을 효과적으로 구체화하려는 연구 [13][23]
BPM 활용	BPM 시스템 자체를 애플리케이션 개발 플랫폼으로 인식하여 이를 기반으로 한 새로운 유형의 애플리케이션 연구 [30]

BPM을 통해 프로세스의 효율성을 다루는 것은 크게 두 가지 관점에서 접근이 가능하다. 하나는 사용자의 업무처리 관점이고, 다른 하나는 엔진의 업무할당의 관점이다. 전자의 경우, 사용자로 하여금 전체 프로세스의 효율성을 증대하는 방향으로 업무를 처리하는데 필요한 정보를 주는 것이다 [29][38]. PERT/CPM 기법을 통해 BPM 프로세스 모델의 주경로와 여유시간을 찾고 사용자들이 Dispatching Rule을 통해 이들을 활용하도록 하여 프로세스를 효율적으로 진행하도록 한 [29]의 연구가 대표적이다. 후자의 경우는 [11]에서와 같이 업무참여자의 능력이 유한하다고 가정하고 이들의 업무부하에 주목하여 균형적인 업무할당을 통해 프로세스를 효율적으로 관리하는 방법 등이 있다. 본 연구는 프로세스 효율을 위하여 TOC의 도입을 후자의 관점에서 프로세스 효율성을 달성할 수 있는 방법론을 개발한다.

대상 비즈니스 프로세스

일반적으로 비즈니스 프로세스란 입력 요소들(사람, 시스템, 도구 등)을 이용하여 추구하는 결과물(제품, 서비스 등)을 도출하기 위한 일련의 절차이다. 본 논문은 이러한 비즈니스 프로세스의 효율성을 높이는 것을 목적으로 하고 있으며, 전반적인 방법론을 설명함에 있어 그림 1과 같은 실제 금융권에서 사용되고 있는 비즈니스 프로세스를 대상으로 하기로 한다. 해당 프로세스는 은행 여신 심사 프로세스로 각 영업점에서 요청된 심사를 본사가 수행하는 과정을 표현한다. 대출 요청 건이 각 지점으로부터 일정량 모이면 이를 처리하기 위한 프로세스 인스턴스가 생성되고 심사를 진행할 것인지 판단한 뒤, 대출 신청 액수에 따라 해당 대출 심사가 진행되며, 대출 승인이 결정되면 실제 대출을 위한 작업이 진행된다. 이를 위해 관련 서류를 확인하고, 담보를 설정하며 최종 대출 실행 지시가 떨어지면 진행된 관련 서류 및 등기 서류를 접수하고, 채권보전서류를 확인한 뒤 대출을 실행한다.

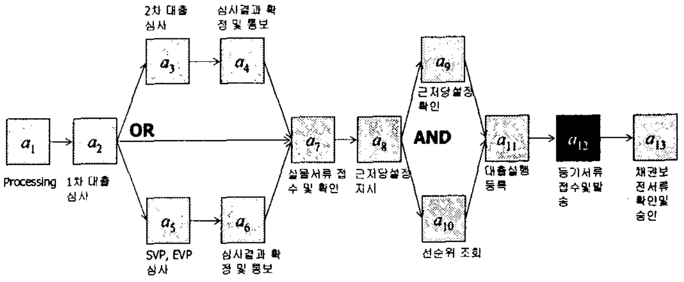


그림 1 – 여신 심사 프로세스

여신심사 프로세스는 이미 상용 BPM이 적용되어 프로세스의 진행과정이 매우 체계적으로 진행되고, 처리 속도 또한 단축되어 많은 이자수익을 창출할 수 있었다. 그러나, BPM을 적용하는 과정에서 "등기서류접수및발송(10여명의 심사자를 거쳐 처리된 심사 결과를 수발하는 곳)"이라는 단위업무 수행과정에서 문서량과 내용이 복잡하여 과도하게 시간이 소요되는 문제점을 낳았고 이러한 효율성 저하 문제를 해결할 필요가 있다. 본 연구에서는 예제에서와 같이 프로세스의 병목현상으로 인해

프로세스 효율성이 떨어지는 경우에 대해 TOC 이론을 적용하여 전체적 관점의 프로세스 관리에 대해 연구하고자 한다.

DBR을 이용한 비즈니스 프로세스 효율화

본 장에서는 업무프로세스와 생산공정 간의 차이를 논의하고, 업무프로세스에서 필요한 DBR 관련 개념을 정의한 후, DBR개념의 적용원리를 설명한다. 생산 공정에는 단위 공정과 필요한 라우팅이 고정되어 있고, 자원도 명시적으로 할당되어 있다. 그러나, 업무프로세스에서는 진행 상황에 따라 다양한 대안이 실행되도록 표현되어야 하기 때문에, 자원이 명시적으로 배치되거나 할당되지 않는다. 한편, 생산 공정에서는 로트 크기를 결정하여 이를 공정에 투입하며, 상황에 따라 예측 생산을 통해 주문을 고정시킬 수 있다. 업무프로세스에서도 로트는 존재하는데, 고유한 인스턴스들이 로트의 역할을 하며 그 크기는 대부분의 경우 1이라고 가정할 수 있다. 그리고, 업무프로세스에서 자원은 주로 업무담당자 혹은 업무관련 애플리케이션인 반면에, 생산 공정은 생산설비를 비롯하여 공정 참여자 등 다양하다. 이러한 자원들에 업무를 할당할 때, 업무프로세스에서는 정의시 1차 할당을 명세한 뒤, 실행시에 업무를 실제 수행할 업무담당자에게 전달한다. 생산 공정에서는 생산설비에 고정, 혹은 그룹별 할당이 이루어진다. 또한, 생산공정은 많은 연구를 통해 효과적인 스케줄링 방법들이 고안되어 계획적이고 체계적으로 관리되고 있지만, 업무프로세스는 체계적인 관리 방법이 부족하며, 특히 전체 프로세스 관점에서 관리가 제대로 이루어지지 않고 있다[4]. 지금까지 언급한 바를 토대로 업무프로세스에 DBR을 적용하는 구체적인 방법론을 기술한다.

비즈니스 프로세스의 제약자원(CCR)

본 연구에서 고려하는 업무 환경은 많은 프로세스 인스턴스가 발생하는 환경이다. 즉, 많은 인스턴스가 발생하고 전체의 효율을 증진시키는 것이 중요한 업무환경을 고려한다. 이러한 환경에서는 전체적인 업무 효율성에 중요한 변수로 작용하는 자원이 존재하게 되는데 본 절에서는 이러한 제약자원을 설명한다. 비즈니스 프로세스에서의 자원은 업무담당자에 한정하며, 따라서, 제약자원 역시 업무담당자이다.

업무담당자는 많은 프로세스 인스턴스로 인해 동시에 다수개의 업무를 수행하는 것이 일반적이다[37]. 그리고, 같은 업무를 서로 다른 담당자가 수행한다. 따라서, 인스턴스 별로 제약자원도 달라질 수 있기 때문에, 다수 개의 인스턴스가 발생하는 실행 환경에서 제약자원이 의미를 갖는다. 일정량의 인스턴스가 발생하면 하나의 인스턴스 완료시간은 다른 인스턴스들의 진행 상황에 영향을 받는다. 업무담당자가 처리하는 단위업무 완료가 지체되면, 할당된 또 다른 단위업무들은 대기 해야 하므로, 결과적으로 다른 인스턴스들의 진행을 방해하게 된다. 이를 통해 현 시점에서 업무목록에 많은 단위업무가 대기하고 있는 업무담당자가 프로세스 진행에 장애가 된다고 이해할 수 있다. 본 연구에서는 업무프로세스의 제약자원을 'BP-CCR'이라 명명하고, 주어진 프로세스 환경에서 업무부하가 가장 큰 업무담당자로 정의한다.

Definition 2 (제약자원, BP-CCR). $WL_{CCR} = max\ \{WL_i \mid WL_i$는 i-번째 업무담당자의 업무부하, $i=1, \ldots, I\}$ 라고 할 때, BP-CCR은 업무부하가 WL_{CCR} 이 되는 업무담당자(u_{CCR})이다.

BP-CCR을 정의하기 위해서는 각 업무담당자의 업무부하를 계산할 수 있어야 한다. 정확한 업무부하는 프로세스가 실제로 수행이 된 후 산출된 데이터를 통해 계산될 수 있으나[28], 정의 시에 업무부하를 정의할 수 있다면, BP-CCR을 발견하는 목적 이외에 프로세스 개선 및 모델링의 목적으로도 이를 다양하게 활용할 수 있다. 따라서, 본 논문에서는 정의 시에 업무부하를 계산하기 [11]에서 소개한 방법론을 도입한다. 해당 방법론 하에서 업무프로세스에 참여하는 각 자원들의 능력이 유한하며, 다양한 업무에 참여한다고 가정하고 있다. 이러한 가정은 본 논문의 접근과 동일하기 때문에, 정의한 프로세스 모델을 토대로 [11]의 방법론을 도입하여 업무부하를 계산할 수 있다. 먼저, 업무부하를 계산하기 위하여 필요한 개념을 정의한다.

Definition 3 (기대실행빈도, Expected Execution Frequency) 프로세스 인스턴스가 한번 실행될 때, 액티비티 a가 처리될 빈도수를 기대실행빈도라 하며 f_a로 표현한다.

이 때, 각 단위업무의 도착률은 다음과 같이 정의된다.

Definition 4 (단위업무의 도착률, Task Arrival Rate) 프로세스가 꾸준하게 생성될 때, 단위시간 당 단위업무 a가 수행되는 횟수는 λ_a 라 하며, $\Omega \cdot f_a$로 계산된다.

한편, 특정 단위업무가 특정 업무담당자에게 할당될 확률을 정의하면 다음과 같다.

Definition 5 (업무할당확률, Task Assignment probability) 업무담당자 u에게 단위업무 a가 할당될 확률을 $p_{a,u}$라 한다. (단, $(a, u) \in R$)

업무담당자의 업무부하는 정의한 개념을 이용하여 활용률(utilization rate)을 통해 계산할 수 있다. 본 논문은 정확한 업무부하량보다는 상대적인 업무부하 측정을 통해 CCR을 발견하는 것이 목적이기 때문에, 업무부하의 의미에 부합한 측정 지표를 활용해도 무방하기 때문이다.

Definition 6 (업무부하, Workload)
업무담당자 u_i의 업무부하(WL_k)는 다음과 같이 계산된다.

$$WL_k = \sum_{a \in A_u} \frac{\lambda_a \cdot p_{a,u}}{\mu_{a,u}} = \Omega \cdot \sum_{u \in A_u} \frac{f_a \cdot p_{a,u}}{\mu_{a,u}}, \forall u \in U.$$

$$A_u = \{a \mid (a, u) \in R\}, \text{ for } u \in U.$$

위와 같은 방식으로 업무부하를 계산하면, 프로세스 실행에 앞서 BP-CCR을 비롯하여 각 업무담당자의 업무부하를 예측할 수 있다. 이러한 정보는 전술한 바와 같이 프로세스 설계 또는 개선 시 유용하게 활용될 수 있으며, 본 연구에서 BP-CCR을 정의하는 데에 적용하였다.

비즈니스 프로세스의 제약자원(CCR)

BP-CCR을 바탕으로 비즈니스 프로세스를 위한 드럼 및 버퍼, 로프 등을 정의한다. 드럼(Drum)은 BP-CCR의 업무처리속도로 정의하며 전체 공정의 생산 속도를 결정한다. 한편, 버퍼는 BP-CCR이 예기치 못한 상황에서도 끊임없이 가동될 수 있도록 하는 기능을 가져야 하므로, 적절한 양의 단위업무가 적재되어 있는 BP-CCR의 업무목록으로 정의한다. 업무프로세스의 로프는 BP-CCR의 업무부하 상태에 따라 프로세스 인스턴스 발생을 조절하는 역할로서, 프로세스 인스턴스 발생량에 따라 각 업무담당자의 업무부하가 영향을 받기 때문에, 결과적으로 BP-CCR의 업무부하를 조절하게 된다. 로프의 기능은 BPM의 엔진을 통해서 구현할 수 있다.

Drum

먼저 드럼을 계산하는 방법을 설명한다. 드럼을 계산하는데 있어서 BP-CCR에 해당하는 업무 담당자(u_{CCR})가 업무를 수행하는 두 가지 다른 환경을 고려해야 한다.

- Case 1: u_{CCR}이 속한 담당자 그룹에 단일 업무를 할당하는 경우
- Case 2: 엔진이 두 개 이상의 업무를 u_{CCR}에게 할당하는 경우

가 있다. 이러한 두 가지 경우, 드럼의 계산식은 달리 접근된다. 한편, u_{CCR}이 처할 수 있는 업무환경은 제안한 두 경우를 기반으로 다양하게 확장될 수 있으나, 본 연구는 드럼의 상세한 계산에 초점을 두지 않기 때문에, 이는 추후 연구를 통해 다루기로 한다.

먼저, Case 1을 설명한다. u_{CCR}이 k명이 포함된 업무그룹에 포함되어 업무를 할당 받는 경우, 하나의 단위업무를 a_i라고 하자. 이 때, a_i의 예상처리시간을 ET_i라고 하면, u_{CCR}이 업무에 참여한 시간 $T_{u_{CCR}}$ 동안 BP-CCR은 최대 $\frac{T_{u_{CCR}}}{ET_i} \times k$ 개를 처리할 수 있다. 본 케이스에서는 u_{CCR}이 단일 업무에만 참여하고 있기 때문에, k명은 1개의 인스턴스를 각각 처리할 수 있다. 따라서, 한 사람이 T 동안 처리할 수 있는 개수에 k배가 u_{CCR}이 최대로 처리할 수 있는 단위업무 인스턴스의 개수이다. 따라서, 이 때 드럼은 다음과 같이 계산된다.

$$\text{드럼}(d) = \frac{k}{ET_i} \text{ (개/시간)} \qquad \text{식 (3)}$$

드럼은 프로세스 인스턴스 발생간격을 통제하는 정보로 활용된다. 본 연구에서 드럼은 될 수 있는 대로 크게 잡아 인스턴스 발생 간격이 통제되는 경우를 최소화한다.

한편, Case 2에서는 u_{CCR}은 다수 개의 단위업무에 참여하고 그 처리시간이 서로 다르다. 따라서, 속도를 계산함에 있어 단순히 업무의 개수를 적용하는 것은 불합리하다. 이를 해결하기 위하여 단위업무의 처리 개수는 처리시간이 가장 짧은 단위업무의 시간으로 나눈 값을 적용한다. 만일 처리시간 1인 단위업무(a_1)가 최소시간을 가진 단위업무라면 처리시간 2인 단위업무(a_2)는 2개로 계산되어야 한다. 그러므로, 단위업무 a_i의 단위개수는 다음과 같이 환산하여 사용한다.

$$a_i\text{의 단위개수}(UN_i) = \frac{ET_i}{ET_{\min}} \qquad \text{식 (4)}$$

ET_i: a_i의 예상처리시간

ET_{min}: u_{CCR}에 할당된 최소 단위업무 처리시간

이 때, 특정 단위업무의 처리개수는 이 단위업무의 단위개수와 실제 처리된 단위업무 인스턴스 개수의 곱으로 계산된다. 주어진 시간 내에 J개의 업무를 처리한다고 가정하고, u_{CCR}에게 단위업무 a_i가 발생될 확률을 p_i라 하면, 드럼(d)은 다음과 같이 계산한다.

$$\text{드럼}(d) = \sum_{i=1}^{J} P_i \cdot \frac{1}{ET_i} \cdot UN_i \text{ (개/시간)} \qquad \text{식 (5)}$$

Rope

드럼이 계산되면, BPM 엔진은 이를 고려하여 프로세스 인스턴스 발생을 조절할 수 있다. 즉, BPM 엔진이 DBR에서 로프(Rope)의 기능을 수행하게 된다. 로프는 제약자원과 시작 공정 간의 통신 장치로서 제약자원의 공정 속도에 맞춰 시작 공정에 자재를 투입하는 속도를 조절하는 데, 로프로서 엔진의 기능은 크게 두 가지로 요약할 수 있다. 먼저, 인스턴스 투입 속도를 드럼과 동기화하여 BP-CCR u_{CCR}의 과부하를 조정한다. 다음으로, u_{CCR} 이외의 업무담당자들이 처리할 단위업무의 프로세스 인스턴스 순서를 u_{CCR}이 처리할 단위업무의 프로세스 인스턴스 순서로 일치시킨다.

그림 2와 같이, BPM 엔진은 로프로서의 역할을 하기 위해 프로세스 인스턴스의 투입속도(λ)를 u_{CCR}의 업무처리속도인 드럼(d)와 동기화한다.

$$\lambda = d(\text{개/시간}) \qquad \text{식 (6)}$$

한편, 프로세스가 진행됨에 따라 드럼은 지속적으로 변화하기 때문에, 업무량과 처리시간에 따라 투입속도도 갱신한다. 이와 같은 방식으로 로프는 실행시에 지속적으로 프로세스의 효율적인 진행을 지원한다.

그림 2에서 BP-CCR인 u_2는 인스턴스 2번의 a_3을 처리한 후 인스턴스 1번의 a_I를 처리한다. 여기에 맞춰 u_1 역시 인스턴스 2번의 a_I을 처리한 후에 인스턴스 1번의 단위업무들을 처리하도록 한다. 이는 u_2가 처리한 단위업무가 속해 있는 프로세스 인스턴스의 진행이 다른 업무담당자로 인해 지연되는 것을 방지하기 위함이다.

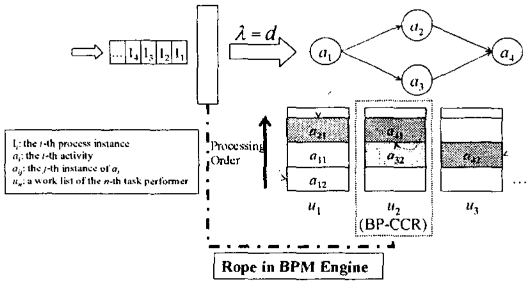


그림 2 - 로프의 기능

Buffer

본 연구에서 버퍼는 BP-CCR 앞에 쌓이는 업무목록으로 정의하며, 이 버퍼를 관리하면 전체 프로세스를 효율적으로 운영할 수 있다. 제약이론에서 제시하는 버퍼는 프로세스가 지속적으로 진행하되, 재고를 최소화하고 운영비용을 절감하기 위해 존재한다[7]. 그러므로, 원활한 프로세스 운영을 위해 BP-CCR 앞에 시간과 비용의 관점에서 적정한 크기의 버퍼를 두어야 한다 버퍼크기 계산은 이미 많은 문헌 등을 통해 연구되고 있으며[7][8][19][25], 제약자원의 버퍼크기를 비용의 관점에서 계산한 연구[27] 또한 존재한다. 그러나, 프로세스 효율의 관점에서 접근하지 않아 본 연구에 적용하기에는 어려움이 있고, 프로세스 효율을 최적화하는 버퍼크기는 상황과 관점에 따라 달리 접근하여 계산되는 성질을 지닌다. 따라서, 본 논문에서는 상황에 적절한 버퍼 크기를 경험적으로 찾고 이를 토대로 프로세스를 관리한다.

BP-DBR 적용절차

업무프로세스에 TOC 개념을 적용하는 것은 곧 드럼 버퍼, 로프의 개념을 이용하여 효과적으로 참여자에게 업무를 할당하는 것이다. 특히, 가장 주요하게 다루어야 하는 것은 BP-CCR의 업무이다. 본 연구에서는 BP-CCR의 업무부하의 범위(Workload Bounds, WB)를 상한(WB_U)과 하한(WB_L)으로 정의해서 범위 내에서 업무할당이 이루어지도록 한다. 즉, 해당 범위의 상한을 벗어나면 이 자원으로의 업무할당이 일시 중단된다. 그리고, 대체자원에 업무를 할당한다. BP-CCR의 능력은 한정되어 있기 때문에, 여유능력을 갖춘 대체자원이 BP-CCR의 업무를 일부 처리함으로써 BP-CCR의 능력을 향상시킨 효과를 기대할 수 있다.

앞서 정의한 BP-DBR방법론과 업무부하 범위를 적용하는 절차를 요약하면, 다음 그림 3과 같다.

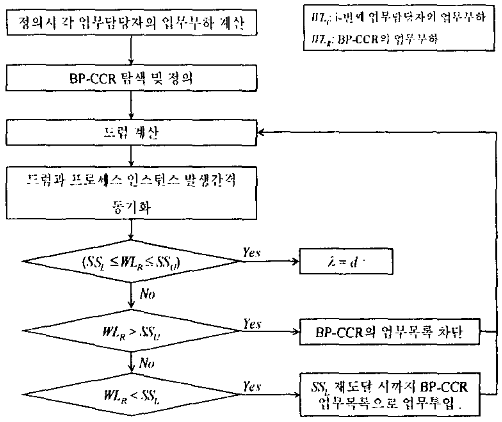


그림 3 - BP-DBR의 적용절차

Simulation Experiments

본 논문에서는 3절에서 설명한 여신 심사 프로세스를 이용하여 BP-DBR의 효과를 보이기

위하여 시뮬레이션 실험을 실시하였다. 본 실험에서는 시뮬레이션 구현에 아레나 6.0을 사용하였으며, 업무처리시간과 프로세스 인스턴스 발생간격이 지수분포를 따른다고 가정하였다[4]. 각 시뮬레이션 시나리오는 총 10000분(약 1개월) 동안 진행되었으며, 실험 시작 직후 500분의 warming up 기간을 두어 안정적인 데이터를 도출할 수 있도록 설계하였다.

본 실험은 크게 세 가지 관점에서 비교 분석한다. 먼저, BP-DBR을 적용한 프로세스가 얼마나 효율적인지 비교, 분석한다. 두 번째, 프로세스의 업무부하에 따른 BP-DBR의 효과 정도를 비교한다. 마지막으로, BP-CCR의 버퍼 크기를 다양하게 하여 이에 따른 BP-DBR의 효과를 비교한다. 세 가지 관점의 실험을 통해 BP-DBR이 효과적인 영역을 구체화할 수 있을 것으로 기대한다.

프로세스 효율성 개선 효과의 검증

본 실험에서는 먼저 BP-DBR를 적용하는 것이 프로세스 효율에 어떤 영향을 미치는지 알아본다. 프로세스 효율은 여러 가지 지표가 있을 수 있으나, 본 연구에서는 프로세스완료시간, 완료인스턴스 수, 그리고 WIP (Work in Process)의 세가지 지표로 정의한다. 프로세스의 효율은 프로세스 인스턴스에서 업무를 누구에게 할당하느냐에 영향을 받는다고 예상할 수 있으며, 이러한 측면에서 실험의 효과를 높이기 위하여 몇 가지 업무할당규칙을 적용하면서 BP-DBR의 효과를 비교한다

본 실험에서는 일반적인 업무할당규칙을 '경합규칙'과 [11]에서 제안한 바 있는 업무부하가 가장 적은 사용자가 업무를 할당 받는 '최소부하규칙'이라고 가정하고, BP-DBR을 적용했을 때와 그렇지 않았을 때를 비교하였다. 실험 프로세스에서 시뮬레이션을 실행한 결과는 아래 표 3과 같다.

표 3 – 각 프로세스 별 효율성 측정 결과

실험방법		평균완료시간 (분)	납기 내 완료인스턴스	평균 WIP*
최소부하규칙사용	BP-DBR 적용	446.888	140	6.351
	BP-DBR 미적용	884.524	78	15.281
경합규칙사용	BP-DBR 적용	517.150	121	6.853
	BP-DBR 미적용	1104.375	58	18.605

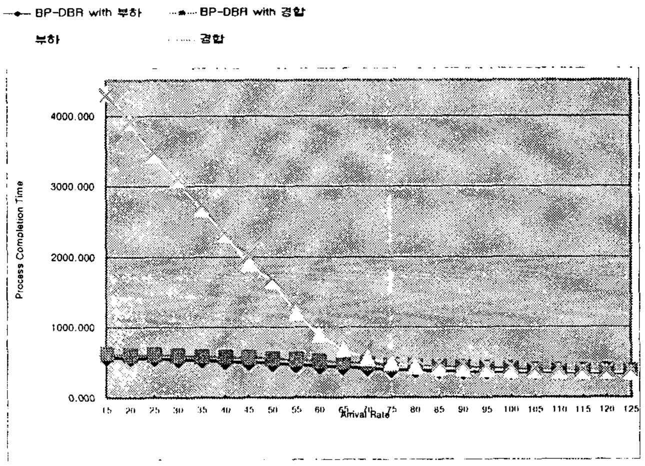
* 평균 WIP: 시뮬레이션 완료 시점에 잔여 프로세스 인스턴스 평균 개수. ** 인스턴스 발생간격: 지수분포 평균 60, 납기: 800, CCR의 버퍼사이즈: 1000

표 3 에서 보는 바와 같이 부하규칙을 사용하고 BP-DBR을 적용하는 경우 프로세스의 효율성이 가장 높은 것으로 나타난다. 특히, BP-DBR을 반영한 프로세스와 반영하지 않은 프로세스 간에 현저한 차이를 보였다. 이는 본 논문의 접근법이 다양한 환경에서 프로세스 효율에 기여할 수 있음을 보인다.

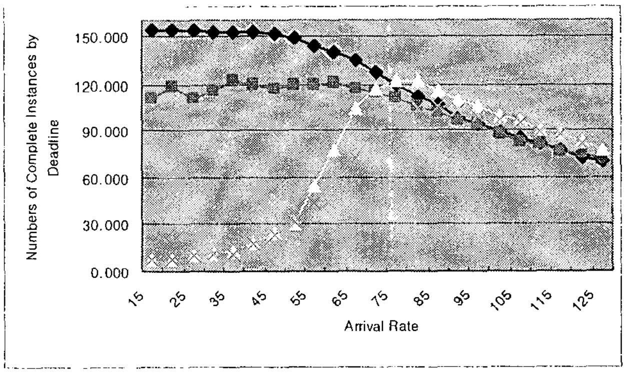
프로세스 업무부하에 따른 효과 비교

본 실험에서는 프로세스 인스턴스 발생량에 따른 BP-DBR의 효과를 비교 분석하였다. 프로세스 인스턴스의 발생량이 많다는 것은 전체적인 관점에서 많은 부하를 유발한다고 볼 수 있다. 실험에서는 프로세스 인스턴스 발생간격을 다르게 하면서 앞 절과 동일한 환경하에서 실험을 실시하였다.

본 실험에서 BP-CCR은 부록에서 보듯이 참여자 l_1, l_2가 해당한다. 이들은 "등기서류접수및발송" 단위업무(a_{12})에만 참여하고 있으며, 처리시간은 130분으로 정의되어 있다. 이러한 정보를 이용하여 드럼의 수치를 예측할 수 있는데, 식 (6)에 따라 최대 1/65(개/분) 의 값을 가지게 된다. 따라서, 본래 프로세스 인스턴스 발생간격에 별도로 로프는 65분 당 1개의 인스턴스를 투입한다.



(a) 프로세스 완료 시간의 변화



(b) 납기준수 완료 인스턴스의 변화

그림 4 – 업무부하에 따른 BP-DBR의 효과변화

위 그림 4는 프로세스 인스턴스 발생간격에 따른 실험 프로세스의 결과를 그래프로 보여준 것이다. 두 그래프에서 x축은 모두 프로세스 인스턴스 발생간격에 해당하며, 값이 60분일 경우 60분 당 1개의 인스턴스가 발생함을 의미한다. 그림 4 (a)의 y축은 프로세스 평균완료시간을 의미하며, 그림 4 (b)의 y축은 납기 내 완료한 프로세스 인스턴스 개수를 가리킨다. 두 그래프에서 공통적으로 해석할 수 있는 것은 프로세스 인스턴스 발생이 빈번할수록, BP-DBR적용의 효과는 증가한다는 점이다.

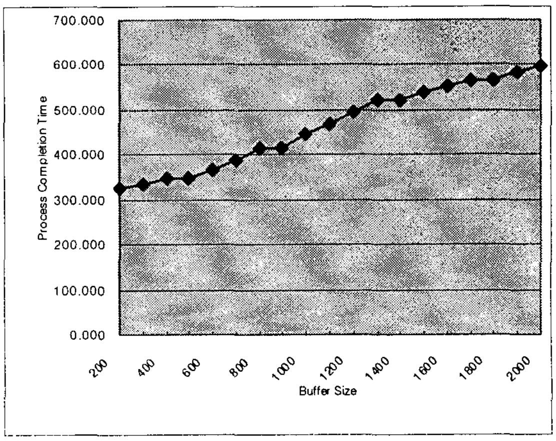
한편, 두 그래프에서 드럼의 값에 주목할 필요가 있다. 본 실험에서 계산된 드럼에 의해 인스턴스는 65분마다 투입된다. 즉, 두 그래프의 x축에서 65분 부근을 기준으로 그 이상이면 BP-DBR 적용 프로세스와 비적용 프로세스 간에 차이가 현저하게 줄어드는 것을 확인할 수 있다. 드럼 즉, BP-CCR의 업무처리속도보다 더 빈번하게 인스턴스가 발생하면 BP-DBR로 인한 효과가 증대되고 그렇지 않으면, BP-DBR에 의한 효과는 기대하기 어렵다. 왜냐하면, 로프는 드럼의 속도에 맞춰 인스턴스를 투입하는데, 본래 인스턴스 투입속도가 드럼보다 작으면 로프는 기능을 수행할 수 없기 때문이다.

적정 버퍼 크기의 결정

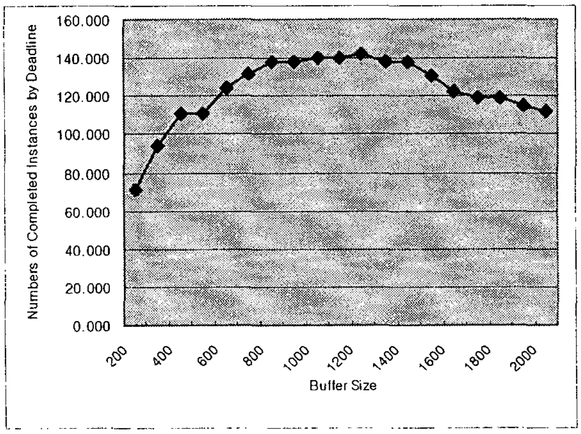
전술한 바와 같이, 다양한 프로세스 환경에서 공히 적용되는 최적 버퍼크기란 있을 수 없고, 수리적으로 계산하기도 힘들 것으로 예상된다. 따라서, 본 연구에서는 현재와 같은 프로세스 환경에서 가장 효과적인 버퍼 크기를 발견하는 것을 목적으로 실험을 진행한다.

그림 5에서 보면, 버퍼 크기에 따라 두 가지 효율성 지표가 달라짐을 확인할 수 있다. 먼저, 그림 5 (a)는 프로세스 완료시간을 보여주는데, 버퍼 크기가 작을수록 프로세스 완료시간이 작아진다. 즉, 버퍼가 없을 때가 가장 효율적이라고 해석할 수 있을 것이다. 그러나, 그림 5 (b)에서 버퍼 크기가 작을수록 때 완료 인스턴스 개수가 많아지지 않는다. 따라서, 프로세스 완료시간이 작더라도, 해당 시간으로 완료된 인스턴스 개수가 작다면 효율성이 높다고 말하기 어렵다. 오히려, 주어진 시간 내에 완료된 프로세스 완료시간은 크더라도 이 시간으로 완료된 인스턴스 개수가 많다면 프로세스 효율성이 높다고 말할 수 있을 것이다. 따라서, 버퍼 크기에 따른 효과는 프로세스 완료 인스턴스 개수의 측면에서 논의한다.

그림 5 (b)에서 버퍼의 크기가 약 1400일 때 완료 개수가 가장 많다. 1400 이상이 되면, 다시 완료 개수가 줄어드는 추세를 보인다. 결국, 1400 근처가 최적의 버퍼 크기가 됨을 확인할 수 있다. 버퍼의 크기가 아주 작으면, 오히려 CCR에게 투입되는 업무량이 작아지고, CCR의 활용률이 저하되어 업무처리 효율성을 떨어뜨릴 수 있다.



(a) 버퍼 크기에 따른 프로세스 완료시간



(b) 버퍼 크기에 따른 납기 내 완료 인스턴스

그림 5 - 버퍼크기에 따른 납기 내 완료 인스턴스

결론 및 추후 연구과제

본 논문에서는 업무프로세스 운영방법론으로서 TOC 개념을 적용한 BP-DBR을 제안하고, 방법론의 효과가 현저하게 나타날 것으로 기대되는 영역을 통해 그 효과를 시뮬레이션 실험을 통해 보였다. 기존 업무프로세스관리 시스템은 기업 내 프로세스를 정의, 실행, 분석하며 타 시스템과의 통합에 주목해왔다. 그러나, 프로세스 실행의 관점에서 이를 효율적으로 운영하는 기능에 대해서는 매우 미흡하다. 그러나, 앞으로 비즈니스 프로세스를 시스템적으로 관리하는데 대한 요구가 증가되는 추세로 본다면 프로세스 효율은 매우 중요한 의미를 가지며, 본 연구의 의미도 이러한 점에서 찾을 수 있다.

BP-DBR은 프로세스 진행에 결정적인 영향을 주는

업무담당자의 업무부하를 지속적으로 관리함으로써, 발생한 인스턴스의 투입을 조절하고, 다른 업무담당자에게 업무를 분산하여 전체 프로세스의 효율성 개선을 가능하게 한다. 특히, 이러한 개선 효과는 업무부하가 특정 자원에 편중되고, 금융권과 같이 사람과 시스템을 통해 처리되어야 하는 프로세스에서 크게 기대할 수 있다. 실험을 통해 확인된 효과로는 먼저, BP-CCR은 프로세스 완료 시간과 완료 인스턴스 개수 등 일반적인 효율성 관점에서 개선의 효과가 있었다. 또한, 업무부하가 클수록 BP-DBR이 효과가 증가했다. 마지막으로, BP-CCR의 버퍼 크기에 따라 프로세스 효율성 정도가 달라지며, 비교적 적합한 버퍼 크기가 존재함을 확인할 수 있었다.

본 논문은 몇 가지 추후 보완 연구를 필요로 한다. 먼저, 제약자원을 주어진 상황에 따라 제약자원의 정의 및 특성은 달라질 수 있음을 인식하고, 각 해당 상황에 맞게 제약자원을 구체적으로 정의하는 방식을 고안해야 한다. 또한, 다양한 업무할당 케이스를 고려하여 드럼계산방식도 세분화할 필요가 있다. 마지막으로, 본 논문에서 접근한 것처럼 다양한 영역에 BP-DBR을 적용하여 개선 효과를 체계적으로 정리할 필요가 있다.

Acknowledgements

이 논문은 교육인적자원부 지방연구중심대학 육성사업(차세대물류IT기술연구사업단)의 지원에 의하여 연구되었음.

References

[1] van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B. and Barros, A.P., "Advanced workflow patterns", Proceedings of 7th Conference on Cooperative Information Systems of Lecture notes in Computer Science, vol. 1901, 2000, pp. 18–19.

[2] Bae, J. –S., Jeong, S. –C., Seo, Y., Kim, Y. and Kang, S. –H., "Integration of workflow management and simulation", Computers & Industrial Engineering, Vol. 37, No. 1-2, 1999, pp. 203–206.

[3] Baker, K. R., Introduction to Sequencing and Scheduling, John Wiley & Sons, New York, 1974.

[4] Chang, D., Son, J.H. and Kim, M.H., "Critical path identification in the context of a workflow", Information and Software Technology, Vol. 44, No. 7, 2002, pp. 405–417.

[5] Daniel, V. and, Guide Jr., R., "Scheduling with priority dispatching rules and drum-buffer-rope in a recoverable manufacturing system", International Journal of Production Economics, Vol. 53, No. 1, 1997, pp. 101–116.

[6] Duclos, L. K. and Spencer, Michael S., "The impact of a constraint buffer in a flow shop", International Journal of Production Economics, Vol. 42, No. 2, 1995, pp. 175–185.

[7] Goldratt, E. M. The Haystack Syndrome, North River Press, New York, 1991.

[8] Goldratt, E. M. The Goal, North River Press, New York, 1992.

[9] Goldratt, E. M. and Fox, Robert E., The Race, North River Press, New York, 1986.

[10]D. Gross, C. M. Harris, Fundamentals of Queuing Theory, John Wiley & Sons, New York, 1998.

[11]Ha, B. –H., J. Bae, Y. –T. Park and S. –H. Kang, "Development of process execution rules for workload balancing on agents," Data & Knowledge Engineering, Online Available, 2005.

[12]Hammer, M., The Agenda: What Every Business Must Do to Dominate the Decade, Three Rivers Press, 2003.

[13]M. Hellinger and S. Fingerhut, "Business Activity Monitoring: EAI Meets Data Warehousing," eAI Journal, July 2002.

[14]W. Hur, H. Bae and S.-H. Kang, "Customizable Workflow Monitoring," Concurrent Engineering Research and Applications, Vol. 11, No. 4, 2003, pp. 313–326.

[15]J. Kim and A. Segev, "A Web Services-enabled marketplace architecture for negotiation process management," Decision Support Systems, Vol. 40, No. 1, 2005, pp. 71-87.

[16]Kim, Y., Bae, J., Bae, H. and Kang, S., "Automatic Control of Workflow Processes using ECA Rules", IEEE Transactions on Knowledge and Data Engineering Vol. 16, No. 8, 2004, pp. 1010–1023.

[17]Kim, Y., Kang, S., Kim, D., Bae, J. and Ju, K., "WW-Flow: Web-based workflow management with runtime encapsulation", IEEE Internet Computing, Vol. 4, No. 3, 2000, pp. 55–64.

[18]M. Klein and C. Dellarocas, "A Knowledge-based Approach to Handling Exceptions in Workflow Systems," *Computer Supported Cooperative Work,* Vol. 9, 2000, pp. 399–412.

[19]Klusewitz, G., Rerick, R., "Constraint management through the drum-buffer-rope system," IEEE/SEMI Advanced Semiconductor Manufacturing Conference, 1996, pp. 7–12.

[20]A. Kumar and J. L. Zhao, Dynamic routing and operational controls in workflow management systems, Management Science, Vol. 45, 1999, pp. 253-272.

[21]Lawrence, P. Workflow Handbook, Wiley, 1997.

[22]Lu, S.H. and P.R. Kumar, "Distributed Scheduling Based on Due Dates and Buffer Priorities," IEEE Transactions on Automatic Control, Vol. 36, No. 12,

1991, pp.1406–1416.

[23]D. McCoy, "The Convergence of BPM and BAM," Gartner Research Note, SPA-20-6074, 2004.

[24]K. L. Myers and P. M. Berry, "At the boundary of workflow and AI," Proceedings of the 16th National Conference on Artificial Intelligence (AAAI-99) Workshop on Agent Based Systems in Business, Orlando, 1999.

[25]Pierce, N. G. and Yurtsever, T., "Dynamic Dispatch and Graphical Monitoring System", IEEE International Symposium on Semiconductor Manufacturing Conference Proceedings, 1999, pp. 65–68.

[26]Pozewaunig, H., J. Eder, and W. Liebhart, "ePERT: Extending PERT for Workflow Management Systems," *The First European Symposium in ADBIS*, 1997, pp. 217-224.

[27]Radovilsky, Z. D., "A quantitative approach to estimate the size of the time buffer in the theory of constraints", International Journal of Production Economics, Vol. 55, No. 2, 1998, pp. 113–119.

[28]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 (IJIE2003)*, Las Vegas, America, 2003.

[29]Rhee, S. -H, Bae, H. and Kim, Y., "A dispatching rule for efficient workflow", Concurrent Engineering – Research and Applications, Vol. 12, No. 4, 2004.

[30]Rhee, S. –H. et al, "Process-Oriented Development of Job Manual System," Lecture Notes in Computer Science, Vol. 3482, Springer-Verlag, Berlin, 2005, pp. 1259-1268.

[31]M. Sayal, F. Casati, U. Dayal, M.-C. Shan, "Integrating Workflow Management Systems with Business-to-Business Interaction Standards," Proceedings of the 18th International Conference on Data Engineering, 2002.

[32]Smith, H. and Fingar, P., Business Process Management – The Third Wave, Meghan-Kiffer Press, 2003.

[33]J. H. Son and M. –H. Kim, "Improving the performance of time-constrained workflow processing," Journal of Systems and Software, Vol. 58, No. 3, 2001, pp. 211-219.

[34]Peter Wegner, "Why Interaction is more powerful than algorithms," Communications of the ACM, Vol. 40, No. 5, 1997.

[35]Wei, Chiu-Chi, Liu, Ping-Hung and Tsai, Ying-Chin, "Resource-constrained project management using enhanced theory of constraint", International Journal of Project Management, Vol. 20, No. 7, 2002, pp. 561–567.

[36]WfMC-TC-1016-P, Process Definition Interchange, Workflow Management Coalition, Lighthouse Point, Fla, 1999.

[37]WFMC-TC-1011, Terminology & Glossary, Workflow Management Coalition, Lighthouse Point, Fla, 1999.

[38]Zhao, J. L. and E. A. Stohr, "Temporal workflow management in a claim handling system," *SIGSOFT: Software Engineering Notes*, Vol. 24, No. 2, March 1999, pp. 187–195.