

## 제약이론에 기반한 워크플로우 운영에 대한 연구\*

강 석 호\*\* · 안 동 준\*\*

### 〈目 次〉

- |                               |             |
|-------------------------------|-------------|
| I. 서 론                        | IV. 실험 및 분석 |
| II. 연구 현황                     | V. 결 론      |
| III. 워크플로우 특성에 맞춘<br>제약이론의 적용 |             |

### Abstract

본 논문은 제약이론(Theory of Constraints)의 프로세스 운영 개선기법에 기반하여 워크플로우(Workflow)에서 제약 자원을 찾고, 워크플로우의 운영 성과가 개선될 수 있도록 효율적인 업무 흐름 관리 방법을 개발하는 것을 목적으로 한다. 이를 위해 과거의 워크플로우 로그를 분석하여 워크플로우 운영성과에 결정적인 영향을 미치는 제약업무담당자(Capacity Constraints Participant, CCP)를 찾는다. 제약 업무 담당자가 결정되면 워크플로우에 제약이론의 집중개선 5단계의 절차에 따라 DBR 스케줄링 기법을 참고하여 제약 업무 담당자의 업무처리속도에 발생한 프로세스 인스턴스의 워크플로우 투입속도를 동기화하여 CCP의 업무부하가 일정수준으로 유지되도록 하여 비 제약 업무담당자의 여유능력이 향상되도록 한다. 이를 통해 CCP의 업무부하를 가능한 업무담당자들에 분산시킴으로써 프로세스 인스턴스의 처리가 원활히 이루어지도록 한다. 또한 일정기간마다 CCP를 갱신하여 운영개선기법을 반복적으로 적용한다. 본 논문에서 개발한 알고리즘을 워크플로우 관리시스템에 적용함으로써 전체 워크플로우가 효율적으로 운영되는 효과를 기대할 수 있다.

\* 본 연구는 한국과학재단 목적기초연구(400-20030151) 지원으로 수행되었음.

\*\* 서울대학교 산업공학과

## I. 서론

워크플로우란 문서와 정보, 혹은 업무가 일련의 절차나 규칙에 따라 한 참여자에서 또 다른 참여자로 전달되는 전체 혹은 부분적으로 구현된 비즈니스 프로세스의 자동화를 말하며, 워크플로우 관리 시스템은 IT도구나 어플리케이션을 호출하면서 프로세스 정의를 해석하고, 워크플로우 참여자와 상호작용하며, 하나 이상의 워크플로우 엔진 상에서 실행 가능한 워크플로우를 정의하고 생성하고 관리해주는 소프트웨어 시스템이다(Hollingsworth 1994).

기업간 경쟁이 치열하게 전개되어가는 오늘날 기업은 현재도 미래도 자신의 목적을 달성하기 위해서 끊임없이 노력한다. 기업의 이러한 개선과 혁신의 노력을 지원하기 위해 많은 경영혁신 기법들이 개발되었다. 다운사이징, 리스트럭처링, 리엔지니어링, 6 시그마, TQM, 지식경영, TOC, ABM 등 최근 20년 동안 경영혁신 기법들은 자신의 방법론과 기법을 따르며 기업의 많은 문제가 해결되고 경쟁기업보다 월등한 위치를 차지할 수 있다고 선전하고 있다. 과거에 생산 활동의 혁신과 개선기법으로 사용되던 방법들 또한 그 영역을 넓혀 이제는 비즈니스 프로세스의 개선활동에 응용을 시도하고 있다.

비즈니스 프로세스의 개선을 위해 워크플로우 관리 시스템은 조직의 업무를 체계화된 표현 방법을 통해 구체적인 모양으로 만들고 사람과 기계가 이해할 수 있도록 나타내고, 이를 컴퓨터를 이용한 자동화된 시스템을 통해 업무의 흐름을 관리 통제할 수 있도록 한다. 관리되는 업무 프로세스들은 사전에 정의되어 저장되는데, 여기에는 프로세스 전문가와 관리자가 현행 업무 방식이나 개선된 업무 방식을 명확히 즉, 일어날 수 있는 모든 상황을 다 표현할 수 있다는 것을 전제하게 된다.

워크플로우 관련 기존의 연구들은 주로 프로세스 모델 자체에 관한 것과 컴퓨터를 통해 이를 구현할 수 있는 방안에 대해 진행되어왔다. 이러한 연구들을 바탕으로 워크플로우 관리 시스템은 복잡한 개별 프로세스 모델을 정의하고 정확한 순서대로 수행하는 것을 보장하게 되었다. 그러나 전체 프로세스가 모두 효율적으로 진행되는 것을 보장하지는 못한다(Eder et al. 2001). 그러한 인식을 바탕으로 최근에는 워크플로우 운영상의 문제를 집중적으로 다루는 연구들이 나타나게 되었다.

본 논문은 시스템 개선기법인 제약이론의 개념을 워크플로우에 도입하여 제약이 존재하는 워크플로우의 운영을 개선하고자 한다. 과거의 워크플로우 로그를 분석하여 워크플로우 운영 성과에 결정적인 영향을 미치는 제약업무담당자(Capacity Constraints Participant, CCP)를 찾는다. 제약 업무 담당자가 결정되면 워크플로우에 제약이론의 집중개선 5단계의 절차에

따라 DBR 스케줄링 기법을 참고하여 제약 업무 담당자의 업무처리속도에 발생된 프로세스 인스턴스의 워크플로우 투입속도를 동기화하여 CCP의 업무부하가 일정수준으로 유지되도록 하여 비 제약 업무담당자의 여유능력이 향상되도록 한다. 이를 통해 CCP의 업무부하를 가능한 업무담당자들에 분산시킴으로써 프로세스 인스턴스의 처리가 원활히 이루어지도록 한다. 또한 일정기간마다 CCP를 갱신하여 운영개선기법을 반복적으로 적용한다.

본 논문은 다음과 같이 구성된다. 먼저 이어지는 2장에서는 워크플로우 효율적 관리와 제약 이론의 연구 현황을 알아보고, 계약이론 도입의 필요성을 살펴본다. 다음으로, 3장에서는 비즈니스 프로세스가 생산 프로세스와 비교 분석한 후, 계약이론에서 소개되는 집중개선 단계와 DBR 스케줄링 기법을 참고하여 개발한 워크플로우 운영개선 알고리즘을 자세히 설명한다. 4장에서는 기존의 워크플로우 운영방식과 본 논문에서 개발한 운영방식을 시뮬레이션 실험을 통해 비교 분석하며, 5장에서 계약이론의 워크플로우 도입 효과를 정리하고 결론을 맺는다.

## II. 연구 현황

### 1. 워크플로우의 효율적 관리

워크플로우의 운영상의 문제를 다루는 연구들에는, 프로젝트 관리기법을 도입하여 복잡한 워크플로우에서 핵심관리 대상이 되는 주공정경로(critical path)를 찾는 방안(Chang et al. 2002), 워크플로우 로그나 현장의 데이터를 수집, 분석하여 업무 처리 시간의 추정을 통한 워크플로우 라이프 사이클 시간관리(Eder et al. 2001), 워크플로우 관리 문제를 스케줄링 모델에 연관시키려는 시도(Smith et al. 1999) 등이 있다.

워크플로우에서 주공정경로를 찾기 위해서 워크플로우의 각 액티비티를 M/M/1 대기행렬 모델로 분석하여 각 워크아이템의 발생률을 바탕으로 액티비티의 수행시간을 평가하여 여유 시간이 없는 주공정경로를 설정하여 집중적으로 관리하고자 하는 것이다. 그러나 워크플로우에 참여하는 업무담당자들은 여러 프로세스의 업무에 참여하는 것이 일반적으로 그 액티비티를 수행하는 업무담당자의 부하를 고려하지 않고 단순히 액티비티의 기대시간만으로 주공정경로는 찾는 것은 비현실적이다. 또한 주공정경로를 활용한 구체적인 워크플로우 운영개선방안이 부족하다.

워크플로우 라이프 사이클 시간관리에서도 프로젝트 관리기법을 도입하여 각 액티비티에 낙관적인 시간, 비관적인 시간 등 여러 가능한 시간을 열거하여 실제 수행시간을 모니터링하며 적절한 이벤트의 발생을 통해 워크플로우의 수행을 철저히 관리하고자 하는 것이다. 그러

나 이러한 방법은 워크플로우의 전체적인 운영상황을 고려하기보다 개별 주문 프로세스 각각에 대해 지연이 발생하였을 때, 워크플로우 관리시스템이 그에 상응하는 조치를 취하고자 하는 것으로 짧은 시간 다양한 프로세스 주문이 발생하는 환경에서 개별 프로세스 주문을 모두 통제하는 것은 시스템에 상당한 부담을 가져오게 되고 운영성과를 고려한 장기적인 안목에서 오히려 좋지 않은 결과를 가져올 수도 있다.

워크플로우의 운영을 생산프로세스 스케줄링 문제로 분석할 수는 있지만, 프로세스의 주문이 예측 가능한 것이 아니고 미리 재고를 쌓아 둘 수 있지도 않다. 이러한 비즈니스 프로세스의 특성을 철저히 고려하여 일정계획의 문제를 따져 보아야 한다. 또한 최적의 스케줄링을 찾기 위해서는 주문의 발생간격이 빨라지거나 누적되면 탐색해야하는 대안이 기하급수적으로 늘어나기 때문에 오히려 발견적 기법을 사용하거나 Dispatching 문제로 접근하는 것이 바람직하다.

프로세스를 효율적으로 운영하고자하는 이러한 연구들을 통해 워크플로우 관리시스템은 프로세스를 정확히 운영하는 것뿐만 아니라 프로세스를 재설계하며 효율적으로 운영하는 경영혁신 도구로서 고려 대상이 되게 되었다. 워크플로우 관리 시스템이 좀 더 지능적으로 작동하여 업무의 흐름을 최적화하기 위해서 철저히 업무 진행을 통제하여 가용한 자원을 효율적으로 관리해야 한다. 제약이론은 생산 프로세스에서 제약 자원을 찾고 그 자원의 집중 관리를 통해 작업 진행을 효율적으로 통제하여 시스템의 성과를 극대화 하고자 하는 방법론이다. 이러한 제약이론을 활용하면 자원의 효율적 관리와 업무 흐름 최적화에 많은 도움을 줄 수 있다.

## 2. 제약이론

제약이론은 1983년 이스라엘의 물리학자 Eli Goldratt의 *The Goal*(1992)이라는 저서를 통해 세상에 알려지기 시작하여 이후 체계화된 이론으로 정립되었다. 제약이론은 "시스템의 목적(goal)달성을 저해하는 제약조건을 찾아내어 그것을 극복하기 위한 시스템 개선기법"으로 정의 될 수 있다. 시스템의 산출물을 제약이론에서는 throughput이라고 한다. 이 throughput을 증대시키는데 가장 약한 부분이 어디인지를 찾아내고, 그 약한 부분을 보호함과 동시에 능력을 최대한 활용하는 것이다. 시스템의 가장 약한 부분을 제약(constraints)이라고 한다. 제약이론이 생산 프로세스에 적용된 것이 DBR(Drum Buffer Rope) 스케줄링이고 프로젝트 관리에 적용된 것이 Critical Chain 프로젝트 관리이다.

이러한 제약이론은 초기에 DBR 스케줄링 기법을 정밀한 공정관리를 필요로 하는 반도체 산업에 적용하여 많은 효과를 보였다는 사례를 보였으며(Kayton 1998), ABC(Activity Based Costing), TPM(Total Productive Maintenance) 등의 다른 경영혁신 기법들과 비교되며

함께 적용됨으로써 개선의 효과가 증대된다는 것이 알려졌다(Dedera 1995, Rose 1995). 최근에는 기업내부 뿐만이 아니라 기업간 프로세스로 확장된 공급사슬에서 제약을 활용하는 방안이 연구되었다(Yeo 2002, Perez 1997). 또한 제약을 활용하는데 중요한 역할을 하는 버퍼관리에 대한 수리적 모형이 연구되어 현장의 감각에 따라 정하던 버퍼를 정확히 산정할 수 있게 되었다(Radovilsky 1997).

제약이론의 목적은 시스템의 어느 한 부분에 대해 집중투자, 집중관리를 통해 전체 최적화를 이루고자 하는 것이다. Eli Goldratt의 *The Haystack Syndrome*(1991)을 통해 제약이론의 생산개선활동을 다음의 집중개선 5단계로 정리하였다.

- 1) 제약조건을 찾아낸다.(Identifying the constraint)
- 2) 제약조건을 철저히 활용한다.(Exploiting the constraint)
- 3) 제약조건 이외의 것은 제약조건에 종속 시킨다.  
(Subordinating the remaining resources)
- 4) 제약조건의 능력을 향상 시킨다.(Elevating the constraint)
- 5) 타성에 젖지 않도록 주의하면서 제1단계로 되돌아간다.(Repeating the process)

생산 프로세스에 위의 절차에 따라 개선활동을 구체적으로 실현해 주는 도구가 DBR 스케줄링 기법이다. 드럼, 버퍼, 로프는 생산 공정에서 전형적으로 활용되며, 효율적인 시간관리와 자원관리를 목적으로 한다. 이러한 기본 개념은 업무담당자를 자원으로 비즈니스 프로세스를 자동화시키는 워크플로우 환경에서도 적용될 수 있다. 이를 위해서는 워크플로우에 적절한 드럼, 버퍼, 로프가 만들어져야 하며, 이들을 이용하여 효율적인 관리 기법이 고안되어야 한다. 이에 따라, 다음 장부터는 제약이론의 적용환경과 워크플로우 환경을 비교하여 워크플로우 특성에 맞게 수정된 제약이론 기반의 워크플로우 운영개선 알고리즘을 설명한다.

### Ⅲ. 워크플로우 특성에 맞춘 제약이론의 적용

제약이론에서 논의하는 제약의 종류에는 크게 3가지로 시장 제약, 물리적 제약, 방침상의 제약이 있다. 시장이 제약이 되는 상황은 생산능력은 충분하나 시장의 수요가 그 보다 낮은 경우이고, 방침상의 제약은 조직의 전략적, 일상적 의사결정을 지배하는 잘못된 평가지표와 평가방법으로 인한 것으로 이 두 가지는 사고 프로세스를 통해 해결하게 된다. 물리적 제약에 대해서

는 DBR 스케줄링 기법과 Critical Chain 프로젝트 관리기법을 도구로 사용하여 문제를 해결한다. 본 논문은 제약이론의 도구들이 잘 적용되는 구조화된 생산 프로세스처럼 주문의 발생이 일상적, 반복적이며 예견 가능하며 프로세스가 잘 구조화 될 수 있는 프로덕션 워크플로우 (production workflow)나 관리적 워크플로우(administrative workflow)를 대상으로 한다. 이러한 워크플로우에서 운영 성과에 결정적으로 영향을 미치는 업무담당자가 제약이 된다.

### 1. 생산 프로세스와 워크플로우 비교

워크플로우에서 제약을 찾고 그것을 최대한 활용하기 위해서 DBR 스케줄링 기법을 참고하여 집중개선 5단계의 절차를 적용한다. 이에 앞서 생산프로세스와 워크플로우 프로세스의 차이점을 먼저 고찰해야 한다. 아래의 표에 그 차이점을 정리하였다.

〈표 1〉 생산 프로세스와 워크플로우의 비교

항목	생산 프로세스	워크플로우
로트 크기	하나 이상의 부품	하나의 업무단위
경로	고정적 경로 혹은 대체 경로	조건, 확률적 경로 포함
자원	생산설비, 참여 작업자	업무담당자, Agent
주문발생	예측, 계획 가능	알 수 없음
계획생산	가능	불가능
일정관리	계획적, 체계적 관리	무계획적, 긴급 업무 우선 처리
재고유형	유형의 재공품	무형의 대기 업무
결과물	유형의 재화	무형의 서비스

WfMC(Workflow Management Coalition)의 정의에 의하면, 비즈니스 프로세스는 비즈니스 목표나 정책 목표를 선택적으로 실현하는 무형의 업무절차들의 집합이며, 워크플로우는 비즈니스 프로세스의 부분적 혹은 전체의 자동화이다. 따라서 워크플로우에 참여하여 정의된 각 업무를 수행하는 자원은 업무담당자나 소프트웨어 에이전트이다. 워크플로우는 자체적인 조직의 요구 또는 고객의 요구에 따라 무형의 프로세스 수행 주문이 발생한다. 주문의 발생은 미리 계획 되어 있거나 예측할 수 없다.

프로세스 주문에 대해 역할 별로 정의된 업무담당자가 단위 업무를 수행하게 되는데, 단위 업무의 순서에 따라 정확하게 진행될 뿐, 발생된 업무를 체계적으로 관리하지 않는다. 다만 업무의 우선 순위에 따라 긴급한 업무를 우선적으로 처리하게 된다. 워크플로우 업무진행 경로는 고정된 하나의 진행 경로만 있는 것이 아니고 조건이나 확률적인 방법에 따라 여러 대안 경로가

존재하게 된다.

워크플로우에서 재고는 워크리스트에서 처리를 기다리는 논리적인 무형의 대기 업무들이 된다. 생산 프로세스처럼 특정한 물리적 공간과 적체 수단이 필요한 것이 아니다. 다음 장에서는 이상의 워크플로우 특성에 맞는 제약이론 적용 이슈들 각 각에 대하여 설명하며 제약이론의 집중개선 5단계 절차에 따른 워크플로우 운영개선 알고리즘을 자세히 설명한다.

## 2. 워크플로우 집중 개선 5단계 및 워크플로우 운영개선 알고리즘

생산 프로세스와 구별되는 워크플로우의 특성을 감안하여 워크플로우의 효율적 운영을 위해 집중 개선 5단계에 맞추어 DBR 스케줄링을 적용한다. 제약이론에 기반하여 워크플로우 운영개선 알고리즘을 개발하기 위한 이슈는 다음과 같은 것들이 있다. 워크플로우의 제약인 CCP (제약업무담당자)를 찾는 방법, 그러한 제약업무담당자를 찾기 위해 각 업무담당자에 부과되는 업무부하의 계산 방법, 프로세스를 운영성과를 비교하는 측정 기준의 선정, 제약 탐색 기간, 워크플로우에 드럼, 버퍼, 로프를 설정하는 방법, 드럼 버퍼 로프가 설정된 환경에서 발생하는 업무들의 업무담당자 할당 방법, 제약업무담당자를 갱신하는 방법 등이 그것이다.

〈표 2〉 집중개선 5단계에 따른 워크플로우 대응 활동

집중개선 5단계 (TOC Five Focusing Steps)	워크플로우 대응 활동
<ul style="list-style-type: none"> <li>• 제약조건을 찾아낸다.(IDENTIFY)</li> </ul>	<ul style="list-style-type: none"> <li>• 프로세스 인스턴스의 처리를 지연시키는 원인이 되는 참여자를 확인(CCP Identification)</li> <li>• 제약 업무담당자의 업무처리속도에 프로세스 인스턴스 투입속도 동기화(Rope 설정)</li> </ul>
<ul style="list-style-type: none"> <li>• 제약조건을 철저히 활용한다. (EXPLOIT)</li> </ul>	<ul style="list-style-type: none"> <li>• 워크플로우의 안정성, 가시성, 예측성 확보</li> <li>• 비 제약업무담당자의 여유능력을 최대한 활용(Rope의 설정으로 불필요한 추가 업무부하가 줄어듦)</li> <li>• 제약 업무담당자의 워크리스트를 납기위선순위에 따라 Sorting(CCP task dispatching)</li> </ul>
<ul style="list-style-type: none"> <li>• 제약조건 이외의 것은 제약에 종속시킨다.(SUBORDINATE)</li> </ul>	<ul style="list-style-type: none"> <li>• 비 제약업무담당자의 Dispatching은 제약업무담당자의 Dispatching 순서에 종속</li> </ul>
<ul style="list-style-type: none"> <li>• 제약조건을 능력을 향상시킨다. (ELEVATE)</li> </ul>	<ul style="list-style-type: none"> <li>• 제약 업무담당자의 업무부하의 분산 방안 모색</li> <li>• 업무담당자의 역할 재구성 고려</li> </ul>
<ul style="list-style-type: none"> <li>• 타성에 젖지 않도록 주의하며 1단계로 돌아간다.(ITERATE)</li> </ul>	<ul style="list-style-type: none"> <li>• 일정 기간 마다 제약 업무담당자를 다시 조사, 위 활동의 반복적 적용</li> </ul>

계약 업무담당자를 중심으로 업무의 흐름 및 일정을 통제하며 계약 업무담당자가 최대한 활용될 수 있도록 한다. 계약 업무담당자 이외의 참여자들의 업무 처리는 계약 업무담당자의 업무처리를 최대한 지원해 줄 수 있도록 하며, 계약 업무담당자의 업무처리능력을 향상시키기 위한 조치들을 모색한다. 일정 기간이 지나면 계약 업무담당자를 다시 확인하여 위의 내용대로 되풀이한다. 이러한 과정을 계약이론의 집중개선 5단계에 대응하여 정리하면 <표 2>와 같다.

<표 2>에서 각 단계 대응 활동을 토대로 워크플로우 운영개선알고리즘을 정리하면 다음의 5 Step으로 정리할 수 있다.

Step I. CCP 확인 - 계약업무담당자 탐색

현재 시간을  $T_{now}$ 라하고, CCP를 분석하는 기간  $T_{CCP}$ 를 설정한다. 업무담당자의 수를  $n$ 이라 하고,  $T_{CCP}$  동안에 특정 시점  $x$ 에서  $i$  업무담당자의 워크리스트에 적체된 워크아이템의 시간량을  $W_i(x)$ 라고하면  $i$  업무담당자의 시간평균 업무부하  $U_i$ 는 다음과 같이 계산된다.

$$U_i = \frac{\int_{T_{now}-T_{CCP}}^{T_{now}} W_i(x)}{T_{CCP}} \quad (1)$$

식 (1)에서 계산에 따라  $P_{MAX} = \text{MAX}\{U_1, U_2 \dots U_n\}$ 를 계산한다.

$$a \times P_{MAX} \geq \frac{\sum_{i=1}^n U_i}{n} \quad (2)$$

식 (2)에서  $a$ 는 ( $0 < a < 1$ )를 만족하는 값으로 CCP 구별 요소이다. 이 값은 사전에 관리자에 의해 설정되며, CCP로 설정되기 위한 업무부하량의 차이를 나타낸다.  $a$ 가 작을수록 다른 업무담당자보다 시간평균 업무부하가 큰 업무담당자가 CCP로 정해진다. 식 (2)를 만족하는 업무담당자가 발견되면 해당 업무담당자는 CCP로 설정된다.

워크플로우에서 선, 후행 관계에 따라 각 단위업무들이 순차적으로 수행된다. 워크플로우 매니저는 워크리스트 핸들러를 통해 각 단위업무들에 대한 워크 아이템을 해당 업무담당자에 할당한다. 시간평균 업무부하 분석을 통해 계약업무담당자를 찾아내고, 프로세스 인스턴스의 진행이 원활히 일어날 수 있도록 주문에 의해 발생하는 프로세스 인스턴스의 투입을 통제하게 된다.

Step II. Rope의 설정 - 프로세스 인스턴스 투입의 통제

CCP에 적정한 보호시간버퍼,  $B_{CCP}$ 를 설정하고, 현재 CCP의 워크리스트 대기업무부하 시간량이  $B_{CCP}$  이하이면 프로세스 인스턴스가 발생하는 즉시 시작 단위업무에 투입하고, 그렇지 않으면 발생된 프로세스 인스턴스의 시작 단위업무 투입을 막고 투입대기행렬에 쌓아두면서, CCP의 워크아이템 처리종결신호가 발생하면, 투입대기행렬의 대기 프로세스 인스턴스 하나씩 시작 단위업무에 투입한다.

계약 업무담당자를 찾아낸 이후에 이를 중심으로 워크플로우의 운영을 통제하게 된다. 프로세스 인스턴스의 투입을 계약 업무담당자의 업무처리속도에 동기화 시키면 운영성과에 도움이 되지 않으며 불필요하게 낭비되는 비 계약업무담당자의 능력을 줄여줄 수 있다. 남은 잉여능력은 기존에 계약 업무담당자에 할당되어 있던 워크아이템의 처리를 위해 사용될 수 있다. 계약 업무담당자의 업무를 비교적 여유능력이 있는 비 계약업무담당자에 분산시킨다. 또한 새로 발생되는 워크아이템에 대해서도 비 계약업무담당자에 주로 할당되도록 제어하여 계약 업무담당자의 업무부하 수준이 일정하게 유지되도록 한다.

Step III. CCP의 워크리스트 대기 업무량 수준  $B_{CCP}$ 로 유지 - 워크아이템의 할당 통제

발생된 워크 아이템이 CCP가 포함된 역할 그룹에 할당되었을 때, CCP의 워크리스트 대기 업무량 수준이  $B_{CCP}$  이하이면 해당 워크아이템을 CCP에 할당하고, 그렇지 않으면 역할 그룹의 멤버 중에 대기 업무량의 수준이 가장 낮은 업무담당자에 워크아이템을 할당한다.

Step IV. CCP 및 비 CCP의 dispatching - CCP의 워크아이템 처리순서에 비CCP 종속

CCP의 워크리스트의 워크아이템 처리순서를 결정한 후, 비 CCP의 워크리스트에 CCP와 같은 프로세스 인스턴스 워크아이템이 있으면, 최우선적으로 처리되도록 순서를 조정한다. 해당 워크아이템이 다수일 때는 CCP 워크아이템의 처리순서를 따른다.

워크아이템들은 일정한 우선순위를 가지고 있다. 주로 납기에 의해 우선순위가 매겨지는데, 우수고객의 주문이나 매우 중요한 고객의 주문 등 특정한 프로세스 인스턴스에 대해 높은 우선순위가 매겨지기도 한다. 프로세스 인스턴스를 우선순위에 맞추어 원활하게 소통하기 위해서 계약 업무담당자의 워크아이템 처리순서에 비 계약업무담당자의 워크아이템 처리 순서를 동기화 한다. 이러한 Dispatching을 통해 계약업무담당자를 거쳐 가는 프로세스 인스턴스가 비 계약업무담당자로 인해 납기가 지연되는 것을 방지한다.

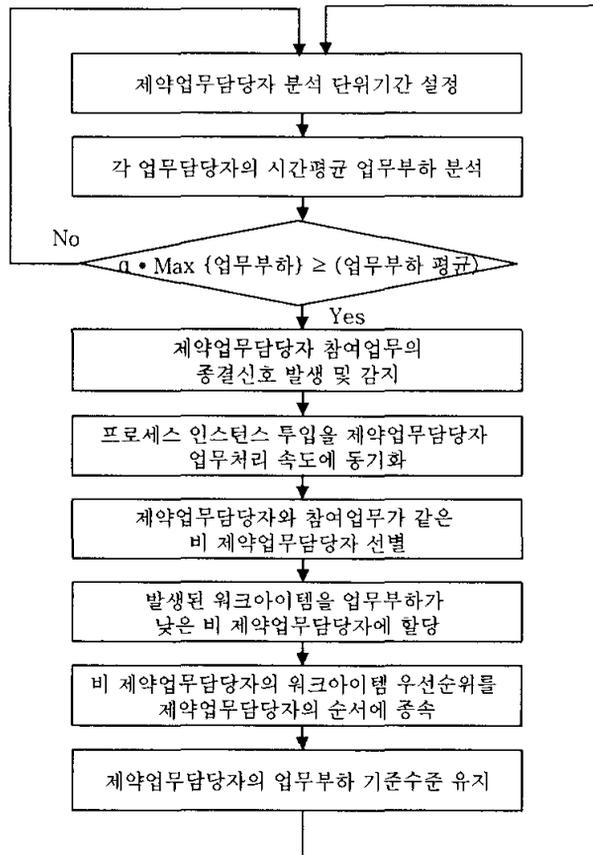
Step V. CCP의 갱신 - CCP에 대한 집중적 모니터링

$T_{CCP}$  동안 (ii), (iii), (iv)의 과정을 반복하고, 다시 (i)의 과정으로 돌아간다.

워크플로우의 운영환경은 고정적이지 않다. 주문의 감소나 증가 등과 같은 외부 환경적 요소나 업무담당자의 교체, 프로세스 구조의 변화 등과 같은 내부의 요인에 따라 제약업무담당자는 바뀔 수 있다. 이러한 환경에서 워크플로우의 개선을 위해서는 지속적인 제약업무담당자의 갱신이 필요하다. 이를 위해 일정한 기간마다 워크플로우 로그를 분석하여 진정한 제약 업무담당자가 누구인지를 판단하여 위의 개선단계를 반복하여 적용해야 한다.

〈그림 1〉은 워크플로우 운영 개선 알고리즘을 순서도로 나타낸 것이다.  $\alpha$ 는 0~1 값을 가지며, 제약업무담당자를 찾는 기준으로 낮은 값을 설정할수록 다른 업무담당자의 업무부하와 큰 차이를 보이는 업무담당자를 제약으로 설정한다.

〈그림 1〉 워크플로우 운영개선 알고리즘



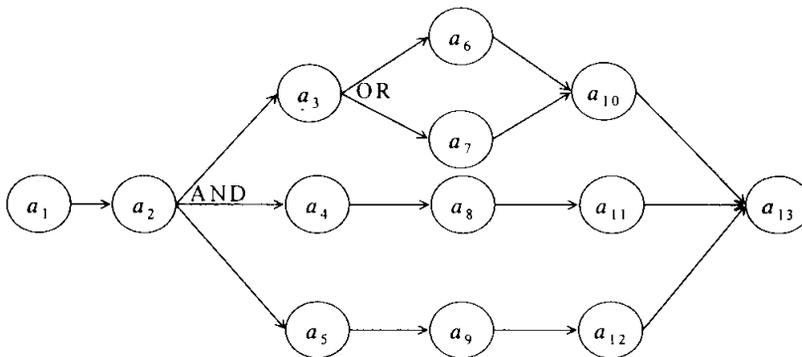
#### IV. 실험 및 분석

워크플로우 운영개선 알고리즘의 효용성을 검증하기 위해서 시뮬레이션 실험을 수행한다. 먼저 실험에 사용될 워크플로우 운영환경을 설정하고, 프로세스 모델을 정의한다. 기존의 방식과 운영개선알고리즘이 적용된 방식으로 정의된 모델에 대하여 시뮬레이션 실험을 수행한다. 시뮬레이션 결과를 프로세스 효율성과 자원의 효율성, 두 가지 측면에서 비교한다.

##### 1. 실험 모델

정의시 단계(workflow build-time)에서 업무담당자들은 일정한 역할에 따라 그룹으로 구성된다. 프로세스 정의문서에 각 단위업무에 참여하는 역할그룹이 기록되어 저장된다. 실행시 단계(workflow run-time)에서 발생된 워크아이템의 할당은 다양한 방식으로 업무담당자에게 할당될 수 있다. 해당 역할그룹의 담당자에게 균등하게 할당하는 방식, 이전업무를 가장 빨리 마친 담당자에게 할당하는 방식, 워크리스트의 워크아이템의 수가 가장 적은 업무담당자에게 할당하는 방식 등이 있다. 이 중에서 본 실험에서는 워크리스트의 워크아이템의 수가 가장 적은 업무담당자에게 할당하는 방식을 기존의 업무할당 방법으로 설정하고 제약업무담당자를 찾는다.

〈그림 2〉 프로세스 모델



위의 세 가지 기존의 방식과 운영개선 알고리즘이 적용된 방식을 비교하게 된다. 각 프로세스 인스턴스에는 일정한 납기가 정해져 있다. 보통 납기에 따라 우선순위가 매겨져서 처리되는데 우수고객의 주문이나 매우 중요한 고객의 주문 등 특정한 프로세스 인스턴스에 대해 높은 우선순위가 매겨지기도 한다. 실험에서 사용되는 프로세스 정의는 〈그림 2〉와 같다. 13개의 단위업

무로 구성되어 있으며, 하나의 AND 블록, OR 블록을 가지고 있다.

OR블록의 OR분기는 조건이나 확률에 따라 분기 되는데, 본 실험에서는 확률적으로 분기되는 상황을 가정한다. 업무담당자는 총 11이며 이들은 각각 5개의 역할그룹에 할당되어 있다. 업무담당자의 구성과 참여 단위업무가 <표 3>과 <표 4>에 나타나 있다.

<표 3> 업무담당자의 참여 단위업무

Task	Task Performer	Task	Task Performer	Task	Task Performer
a <sub>1</sub>	Team 1	a <sub>6</sub>	Team 2	a <sub>11</sub>	Team 1
a <sub>2</sub>	Team 6	a <sub>7</sub>	Team 3	a <sub>12</sub>	Team 6
a <sub>3</sub>	Team 4	a <sub>8</sub>	Team 3	a <sub>13</sub>	Team 2
a <sub>4</sub>	Team 5	a <sub>9</sub>	Team 4		
a <sub>5</sub>	Team 5	a <sub>10</sub>	Team 4		

<표 4> 역할그룹의 업무담당자 구성

Team	Task performer	Team	Task performer
1	1, 2, 3, 7	4	4, 9, 10, 11
2	4, 7, 9	5	5, 7
3	5, 6, 7, 8	6	7

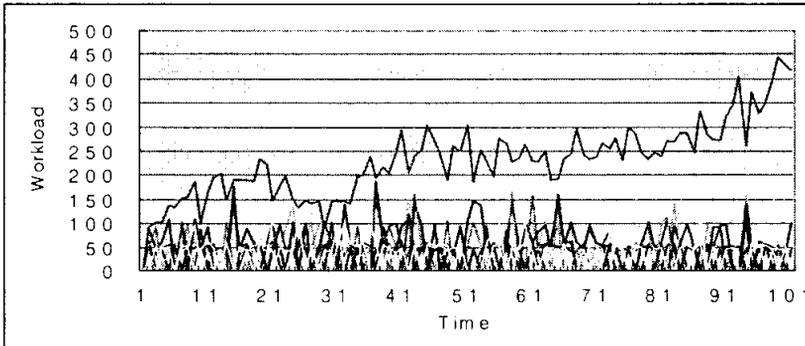
시뮬레이션 입력 파라미터로 업무담당자의 참여 단위업무의 처리시간을 모두  $N(50, 25)$ 로 배정했다. 프로세스 인스턴스의 발생간격은  $Expo(105)$ 를 따르도록 하였다. 하루 일과를 8시간, 한달 운영 일을 30일로 가정하여 약 7개월(100,000 min)을 실험한다. 이러한 실험을 30회 반복 수행하여 그 평균을 사용하여 결과를 분석한다. 시뮬레이션 소프트웨어로는 ARENA 6.0을 사용하여 실험을 수행하였다.

## 2. 실험결과 및 분석

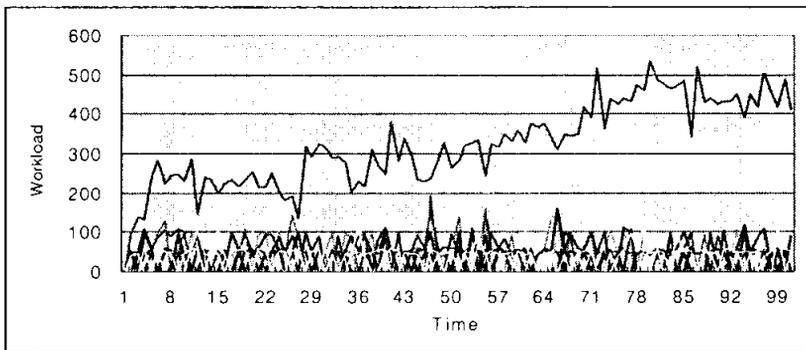
워크플로우 운영개선 알고리즘이 적용된 모델을 실험하기 위해서는 먼저 제약업무담당자를 찾아야 한다. 먼저 기존 워크플로우 운영모델을 실험한다. <그림 3>, <그림 4>는 기존 워크플로우 운영모델의 업무담당자 별 업무부하의 변화를 보여준다. 다른 업무담당자에 비해 7번 업무담당자의 업무부하가 평균 업무부하보다 2배 이상을 나타내며 점차 증가하는 경향을 보인다. 따라서 7번 담당자가 제약업무담당자로 선정되었다. <그림 5>는 운영개선 알고리즘이 적용된 모델의 업무담당자 별 업무부하의 변화를 보여준다. <그림 3>, <그림 4>에 비해 제약업무담당

자 업무부하의 수준이 현저히 떨어져서 일정한 수준을 유지하는 것을 보여 준다. 이는 프로세스 인스턴스 투입을 제약업무담당자의 업무처리속도에 동기화 시키며, 제약업무담당장의 업무부하를 일정한 수준으로 유지하며 비 제약업무담당자에 업무부하를 분산시킨 결과이다.

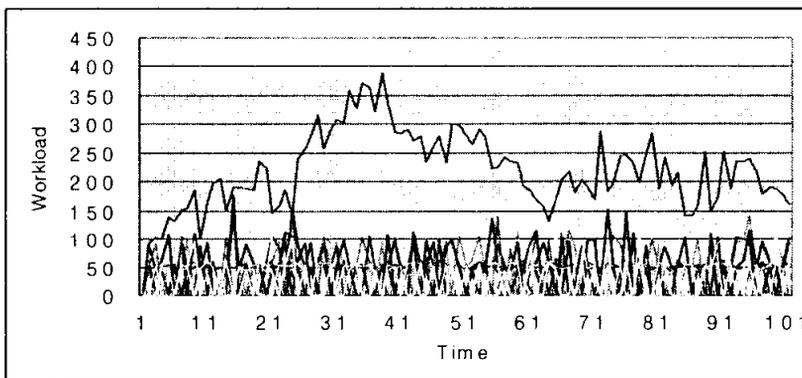
〈그림 3〉 최소업무부하 우선 할당 방식의 업무부하 분석



〈그림 4〉 워크아이템 균등 할당 방식의 업무부하 분석



〈그림 5〉 운영개선 알고리즘 적용 방식의 업무부하 분석



프로세스의 효율성은 워크플로우 관리 시스템이 얼마나 시간을 효율적으로 활용하여 프로세스 인스턴스들을 처리하였는가에 대한 측정 양이다. 이의 측정 기준으로는 일정기간에 대해 완료된 프로세스 주문수, 납기를 어긴 프로세스 주문수, 평균적인 프로세스 리드타임이 있다. <표 5>는 이러한 프로세스 효율성을 비교한 것이다. 각 워크플로우 운영방식에 대해 평균완료 개수에는 거의 차이가 없으나, 프로세스 인스턴스의 평균 리드타임이 워크아이템 균등할당 방식 대비 75.8분 줄었으며, 최소 업무부하 우선할당 방식에 비해 243.4분이 줄었다. 납기를 준수하며 완료된 프로세스 주문수는 워크아이템 균등할당 방식 대비 76.2개 늘었으며 최소 업무부하 우선할당 방식 대비 117.8개가 늘었다. 이는 운영개선 알고리즘을 적용하였을 때 제약업무담당자의 업무부하를 효율적으로 분산시키며 각 프로세스 인스턴스의 리드타임을 줄인 결과이다. 또한 제약업무담당자의 업무처리 우선순위에 비 제약업무담당자의 업무처리 순서를 종속시킴으로 납기를 어긴 주문수가 크게 줄어들었다. 이 결과를 정리하면 납기준수의 개선효과는 25 ~ 46%, 프로세스 인스턴스의 리드타임 개선효과는 7 ~ 22%이다.

<표 5> 프로세스 효율성 비교

워크플로우 운영 방식	운영성과 프로세스주문 평균완료 개수	프로세스주문 평균완료 시간	납기준수 주문수
워크아이템 균등할당 방식	755.37	1166.07	299.07
최소 업무부하 우선할당 방식	761.97	1333.65	257.50
TOC 운영개선 알고리즘 적용 방식	760.47	1090.26	375.30

자원의 효율성은 워크플로우 관리 시스템이 프로세스 효율성 결과에 대해 자원을 얼마나 효율적으로 활용하였는가에 대한 측정 양이다. 업무담당자의 업무활용도로 나타날 수 있다. 이는 총 운영시간 중에서 업무담당자가 워크아이템의 처리에 기여한 시간의 비율로 계산된다. <표 6>에 이 결과나 나타나 있다.

워크아이템 균등 할당 방식에서는 같은 팀에 속한 업무담당자들은 거의 같은 업무활용도록 기록되며 일부 여러 업무에 참여하는 업무담당자는 높은 업무활용도를 기록한다. 그에 비해 최소 업무부하 우선할당 방식이나 운영개선 알고리즘적용 방식에서는 업무담당자들의 업무활용 편차가 높게 나타난다. 이는 인위적으로 워크아이템을 균등하게 할당하는 것이 아니라 업무담당자의 현재 부하 수준에 따라 적절히 업무를 배분한 결과이다. 기존의 워크플로우 운영모델은 다른 업무담당자보다 업무부하가 월등히 높았던 제약업무담당자의 활용도가 거의 1에 가까운 값을 보였다. 운영개선 알고리즘을 적용한 후에도 제약업무담당자의 업무활용도는 줄어들지 않았다.

이는 계약업무담당자의 업무부하를 일정수준으로 유지시키며 계약업무담당자의 능력이 낭비되지 않도록 통제하면서, 비 계약업무담당자에 업무부하를 분산시킨 결과이다. 모든 업무담당자의 활용도가 1에 가까운 값을 가져야 운영성과가 높게 나타나는 것은 아니다. 업무담당자의 활용도를 높이기 위해 워크리스트에 워크아이템을 무리하게 쌓아 놓는다면 오히려 불필요한 능력의 낭비를 유발한다. 프로세스 인스턴스의 흐름을 원활히 소통하기 위해서는 계약업무담당자의 능력을 최대한 활용하고 다른 업무담당자가 그를 받쳐주어야 한다.

〈표 6〉 업무담당자의 활용도 비교

업무담당자	워크아이템 균등 할당 방식	최소 업무부하 우선할당 방식	계약이론 적용 워크플로우
업무담당자 1	0.34	0.55	0.54
업무담당자 2	0.34	0.30	0.30
업무담당자 3	0.36	0.10	0.10
업무담당자 4	0.65	0.82	0.81
업무담당자 5	0.92	0.93	0.94
업무담당자 6	0.35	0.49	0.50
업무담당자 7	0.99	0.99	0.96
업무담당자 8	0.35	0.21	0.20
업무담당자 9	0.65	0.72	0.70
업무담당자 10	0.41	0.43	0.42
업무담당자 11	0.42	0.17	0.17

## V. 결 론

본 논문은 계약이론이 throughput을 증대시키는데 가장 약한 부분이 어디인지를 찾아내고, 그 약한 부분을 보호함과 동시에 능력을 최대한 활용한다는 사실에 착안하여 이를 제약이 존재하는 워크플로우에 적용하였다. 비즈니스 프로세스에서 각 단위 업무담당자들은 개별적으로 업무의 효율을 높이기 위한 노력들을 하고 있다. 그러나 조직의 성과는 프로세스 전체의 효율성과 관계있는 것이다. 국소적인 개별 단위업무담당자들의 업무개선만으로는 성과가 나아지지 않는다. 프로세스의 목적인 높은 성과달성에 도달하기 위해서는 프로세스의 흐름을 방해하는 요소를 찾아내고, 최대한 활용하며, 집중 관리해야 한다. 계약이론의 생산개선활동은 집중개선 5단계를 통해 이루어진다. 생산 프로세스의 제약은 기계설비가 되지만 사람에 의해 단위업무가 수행되는 워크플로우에서는 업무담당자가 제약이 된다. 워크플로우는 단위업무의 선, 후행 관계

로 정의되지만, 업무담당자를 중심으로 업무의 흐름을 분석하면, 발생된 프로세스 인스턴스들은 각 단위업무들이 업무담당자들 사이에서 순차적으로 수행된다. 결국, 업무가 원활히 소통되어 납기준수율과 업무효율을 높이기 위해서는 업무 소통에 방해가 되는 제약업무담당자를 중심으로 워크플로우를 운영해야 한다. 본 논문에서는 워크플로우 운영성과를 좌우하는 제약업무담당자를 찾았다. 제약업무담당자의 업무성과에 워크플로우 성과가 동기화된다는 사실을 밝혔으며, 워크플로우 효율적 운영을 위해 제약이론을 참고하여 운영개선 알고리즘을 개발하여 적용하였다. 본 논문에서 개발한 알고리즘을 워크플로우 관리시스템에 적용함으로써 자동화된 프로세스가 자원의 활용상황을 항상 감시하여, 프로세스의 시작을 적절히 통제하고 제약업무담당자를 최대한 활용하도록 발생하는 업무부하를 분산시킴으로써 워크플로우의 지속적인 운영개선이 달성되어 전체 워크플로우를 정확한 수행을 기반으로 효율적으로 운영할 수 있게 되었다.

### 참 고 문 헌

1. Chang, D-H., Son J. H., and Kim, M. H., Critical path identification in the context of a workflow, Information and Software Engineering, Vol. 44, No. 7, pp. 405-417, 2002.
2. Deder, Cristopher R., Can TOC and ABC Coexist?, IEEE/SEMI Advanced Semiconductor Manufacturing Conference 1995.
3. Eder, Johann., Panagos, Euthimios., Managing Time in Workflow Systems, WfMC, Workflow Handbook 2001, pp. 109-132. 2001.
4. Goldratt, Eliyahu M., The Haystack Syndrome: Sifting Information Out of the Data Ocean January, 1991
5. Hollingsworth, D., Workflow management coalition specification: the workflow reference model, WfMC specification, 1994.
6. Kayton, Dave., Using the Theory of Constraints' Production Application in a semiconductor Fab with a Reentrant Bottleneck, IEEE/CPMT Int'l Electronics Manufacturing Technology Symposium 1998.
7. Kim, Y., Kang, S.H., Kim, D.S., Bae, J.S., and Joo, K.J., WW-flow: A Web-based Workflow Management System Supporting Run-time Encapsulation, IEEE Internet Computing, vol. 4, no. 3, pp. 55-64, May/June 2000.

8. Pozewaunig, H., Eder, J., and Liebhart, W., ePERT: Extending PERT for workflow management systems, The First European Symposium in ADBIS, pp. 217-224, 1997.
9. Radovilsky, Zinovy D., A quantitative approach to estimate the size of the time buffer in the theory of constraints, Production economics, vol. 55, pp. 113-119, 1998.
10. Rose, Ed., Odom, Ray., Dunbar, Randy., Hinchman, Jim., How TOC & TPM Work Together to Build the Quality Toolbox of SDWTs, IEEE/CPMT Int'l Electronics Manufacturing Technology Symposium 1995.
11. Smith, Stephen F., Hildrum, David W., and Becker, Marcel A., Workflow Management from a Scheduling Perspective, Technical Report WS-99-02, The AAAI Press, pp. 68-72, 1999.
12. Yeo, K. T., Ning, J. H., Integrating supply chain and critical chain concepts in engineer-procure-construct(EPC) projects, Project Management, vol. 20, pp 253-256, 2002.