

# 최소 엔진을 가진 동적 워크플로우 관리기의 개발

배혜림, 김영호, 강석호  
서울대학교 산업공학과

## 초 록

오늘날 비즈니스 환경은 전역적인 경쟁으로 인해 비즈니스 수행 비용의 감축, 새로운 서비스와 제품의 빠른 개발을 강조하고 있다. 이러한 환경 하에서 워크플로우 시스템이 솔루션으로 주목받고 있으며, 최근에는 워크플로우에 관련된 표준기술의 정립과 분산객체기술의 발전으로, 이질적인(heterogeneous) 분산환경으로 적용 범위가 넓어지고 있다. 그러나 비즈니스 프로세스의 동적인 특성이 워크플로우 시스템의 실제 응용에서 어려움으로 대두되고 있다.

본 논문은 동적인 환경에 적응성을 가지는 워크플로우 관리기의 요구사항을 파악하고 이를 바탕으로 새로운 시스템 구조를 제안하고자 한다. 제안된 구조는 워크플로우 엔진을 최소화하여 처리 개체들의 자치성을 보장하고, 이를 통해 프로세스 실행시에 발생하는 동적인 상황에 적응성을 제공한다.

## 1. 서론

워크플로우 관리 시스템은 비즈니스 프로세스를 관리하고 지원한다는 오래된 문제에 대한 새로운 접근으로 각광받고 있다. 워크플로우 관리 시스템이 기존의 정보시스템에 비해 새로워진 것이라면 구조화된 작업을 지원하기 위한 정보 기술의 힘을 실을 수 있다는 점이다. 또한 워크플로우 관리 시스템은 프로세스의 태스크들 간의 이동을 제어하고 특정 프로세스 인스턴스에 대해 적절한 태스크들이 수행될 수 있도록 해주며 태스크의 수행에 필요한 자원을 묶어주는 역할을 한다.

현재 많은 워크플로우 제품이 상용화되어 워크플로우를 모델링하고 이를 위해 워크플로우 명세 언어(Workflow Specification Language)를 사용하고 있으며 워크플로우의 실행을 검사하고, 분석하며, 모니터링하는 툴을 제공하고 있다. 그러나 많은 워크플로우 제품들은 단순한 사무(이미지, 문서, 전자우편)에 집중되어 있고 전자우편 등에 의존하는 관리적 워크플로우나 프로세스의 과정들이 사용자의 개입이 없이는 진행되지 않는 비정형의 워크플로우를 지원하고 있을 뿐이다[13]. 워크플로우 관리 시스템들은 제품마다 상이한 구조를 지니고 있으며 그 개발의 철학이 서로 다르므로 인해서 많은 문제를 야기시켰고 이러한 문제들은 WfMC(Workflow Management Coalition)등이 표

준화 작업을 통해 해결하려는 노력을 진행 중이다. WfMC는 Workflow Reference Model을 통해서 서로 다른 워크플로우 관리 시스템들이 공통의 모듈들로 구성이 되며 API를 공유하게 함으로써 서로 상호운영성을 가지도록 하는 것을 목적으로 한다.

워크플로우 관리시스템의 적용에 있어서 문제점은 비즈니스 프로세스의 동적인 수행이 요구되고 있지만, 현재 워크플로우 관리 시스템이 이를 적절하게 반영하지 못하고 있다는 것이다. 이러한 상황에 대한 대처방안으로, 유연한 규칙관리를 통해서 이를 해결하려는 노력이 있었다[1]. 그러나 이 방법은 실행시에 발생할 수 있는 상황에 대해, 미리 규칙을 만들어 두어야 하는 제약이 있어 제한된 유연성만을 제공한다. 보다 근본적인 해결책은 워크플로우 관리기의 시스템 구조를 동적인 환경에 대응할 수 있도록 설계하는 것이다.

본 연구에서는 WfMC에서 표준으로 제안한 Workflow Reference Model에서 엔진의 역할을 축소하고 서버의 개념을 제거함으로써 처리개체(processing entity)들의 자치성 보장을 통해 워크플로우 실행 시에 동적인 변화에 대응할 수 있는 새로운 워크플로우 관리 시스템 구조를 제안하고 현재 워크플로우 관리 시스템이 가지는 문제에 대한 해결책을 제시하고자 한다.

## 2. 관련연구

### 2.1 워크플로우 관리 시스템

워크플로우란 무엇인가에 대한 명백한 합의는 없지만 워크플로우란 단어는 비즈니스 프로세스, 프로세스의 명세, 프로세스를 자동화하기 위한 소프트웨어, 프로세스를 실현하는 사람들의 협동과 조정을 지원하는 소프트웨어와 관련한 것으로 불려진다. 흔히 워크플로우는 '전체적으로 혹은 부분적으로 비즈니스 프로세스를 컴퓨터를 이용하여 편리하게 하거나 자동화한 것'으로 정의되며 워크플로우 관리 시스템은 '워크플로우 로직을 컴퓨터로 표현하고 이에 따라 생성된 순서를 따르는 소프트웨어의 실행을 통해 워크플로우를 완전하게 정의하고 관리하며 실행하는 시스템'을 말한다[8].

위의 정의에 따라 워크플로우가 가지는 기본 개념을 정리하면 다음과 같다[11].

- 프로세스 로직
- 사람과 태스크의 연결

- 태스크에 정보 자원을 제공
- 프로세스의 관리

이미 많은 상용화 제품이 나와 있는 워크플로우 관리 시스템의 구축과 관련하여 이를 표준화하고 상호 운영성을 가지도록 하고자 하는 노력이 WfMC등을 중심으로 이루어지고 있다. 워크플로우와 관련한 연구들은 조직, 방법, 기술적인 측면에서 진행중이며 앞으로도 환경변화에 적용하기 위한 연구들이 이루어져야 한다.

## 2.2 워크플로우 관련 기술

효율적이고 신뢰할 만한 워크플로우의 구현과 실행은 분산 컴퓨팅 환경에 대한 지원을 요구하게 되었고 기존 시스템과 새로운 시스템을 통합하고 이 시스템이 상호운영성(interoperability)을 갖추도록 하기 위해 분산 객체 관리 기술(DOM, Distributed Object Management)이 필수적이게 되었다. 또한 병행성(concurrency)과 오동작(failure)이 있는 경우에도 시스템이 정확하게 작동하도록 하기 위해 CTM (Customized Transaction Management)기술이 필요하다.

이러한 두 가지 필수 기반 기술 외에도 워크플로우 관리 시스템은 각각의 워크플로우 관리 시스템의 특성에 따라 웹, 데이터베이스, 전자우편 등의 기반기술을 필요로 한다.

## 3. 동적 워크플로우 관리시스템의 요구사항

### 3.1 현재 시스템의 한계

워크플로우 관리 시스템이 프로세스를 지원하고 개선하는 도구로서 많은 성장을 하고 기술적 진보를 이루었지만 많은 제품들이 다이어그램들의 수준에 머무르거나 전자 정보 관리 시스템의 특화된 모습에 그치는 등의 한계를 보이고 있다. 이러한 한계의 측면들을 정리하면 다음과 같다.

- 이종성(heterogeneity)
- 상호운영성(interoperability)
- 정확성(correctness) 및 신뢰성(reliability)
- 시스템의 정적 구조
- 분석, 검사, 디버깅에 대한 지원 부족

### 3.2 동적 워크플로우 관리시스템 요구사항

워크플로우와 관련하여 새로이 대두되고 있는 영역은 적응성이 있고 동적인 워크플로우를 다루는 능력이다. 워크플로우 관리 시스템이 실행환경에서 동적인 변화에 대한 대처로써 프로세스 진행에 따른 정보를 수정하고 의사결정에 반영할 수 있으며 내외부적인 이벤트에 의해 발생하는 프로세스 실행상의 변화에 대응할 수 있느냐 하는 문제가 중요하게 되었다.

워크플로우의 동적인 측면과 관련한 요구사항들을 정리하면 다음과 같다[10][15].

- 태스크간 종속성에 따른 프로세스의 처리
- 멀티 시스템 워크플로우에서의 데이터 관리
- 비즈니스 프로세스의 빠른 변화에 대처
- 실행시의 워크플로우 명세의 변경
- 워크플로우 상태의 저장

- 에러의 인식 및 핸들링
- 적용에 필요한 이벤트 처리
- 태스크간 종속성의 변화:태스크의 추가, 종속성의 추가
- 변화에 의해 발생할 수 있는 에러에 대처

이러한 요구사항들에 의해 필요한 시스템의 특성은 다음과 같다.

- 병렬 라우팅의 허용
- 조건에 의한 라우팅
- 자동화 에이전트의 활용
- 늦어진 태스크의 처리
- 서버 프로세스
- 태스크의 거부
- 프로세스 일시중지 기능
- 자동적인 프로세스 인스톨링
- 변화된 내용에 대한 리포트
- 다음 태스크 처리 개체의 선정 규칙

## 4. 시스템 구조

### 4.1 시스템의 구조

동적이고 적응성을 지닌 워크플로우 관리 시스템은 실행 시에 태스크를 처리개체에 재할당할 수 있어야 한다. 이전의 연구들은 이러한 유연성을 확보하기 위해서 대안의 경로(alternative routing)를 미리 설정해두는 방법과 스케줄을 담당하는 엔진을 분산하는 방식을 사용하였다. 이러한 구조는 실행 시에 정확한 프로세스 정의 및 스케줄의 생성을 요구하지만 이러한 정보들은 시작시점에서 결정될 수 없는 경우가 있다. 본 연구에서는 프로세스 정의를 바탕으로 실행 시에 자치성을 지니는 각 처리 개체들이 Bidding Mechanism을 통해 태스크를 다음 개체에 할당하는 분산 구조를 고려한다. 이를 도시하면 다음 그림 2와 같다.

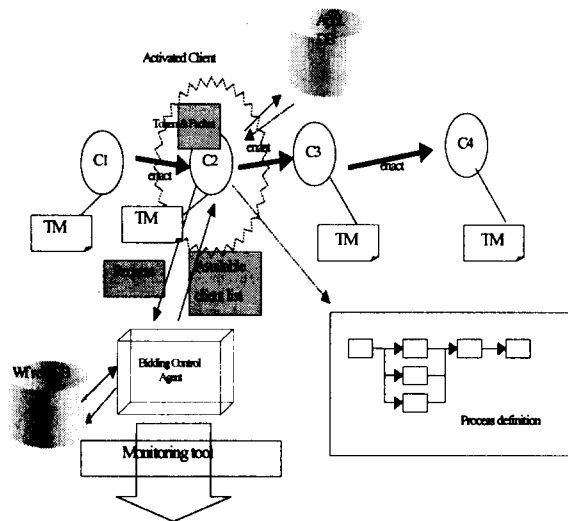


그림 2 전체 시스템의 구조

이를 위해서 주어진 프로세스 정의 도구 (process definition tool)를 사용하여 프로세스를 정의해두고 각 자치 개체들이 실행 시에 해석하며 필요한 데이터와 활성화 정보들을 패킷으로 전달하도록 한다. 이에 따라 서버로써의 엔진의 역할은 Bidding Controller로 바뀌게 되며 처리개체는 자치성을 지니면서 이후 시스템 확장성을 보장받는 모델이 된다.

이에 따른 Bidding Controller와 처리 개체의 역할을 정리하면 다음 표 2와 같다.

표 2. 분산구조에서의 역할

Bidding Controller	처리 개체
<ul style="list-style-type: none"> <li>프로세스 진행의 모니터링</li> <li>다음 태스크를 처리할 개체 리스트의 전달</li> <li>workflow relevant data의 유지</li> </ul>	<ul style="list-style-type: none"> <li>프로세스 정의의 해석</li> <li>다음 처리 개체의 선정 및 enactment</li> <li>워크플로우 인스턴스 데이터의 전달</li> </ul>

## 4.2 Bidding

Bidding Mechanism은 Davis의 협상모형[6]에서 최초로 제시된 후 여러 분야에서 응용되어 왔다. 특히 최근에는 FMS와 같이 동적인 환경에서 다양한 Resource들간의 상호작용에 의하여 작업이 수행되는 환경에서 의사결정을 위한 도구로서 매우 적합한 것으로 알려졌다[2][14]. 생산현장에서 적용하였을 때, 하나의 Controller에 의할 때 보다 신뢰성, 확장성, 수행도, 비용 등의 측면에서 보다 유리하며 각각의 개체들이 자신의 상태와 시스템의 상황에 따라 적절한 목적함수를 설정함으로써 일정한 Heuristics에 의하여 통제될 때보다 전체 시스템 수준에서 더 나은 해를 제공하는 것으로 알려져있다 [2].

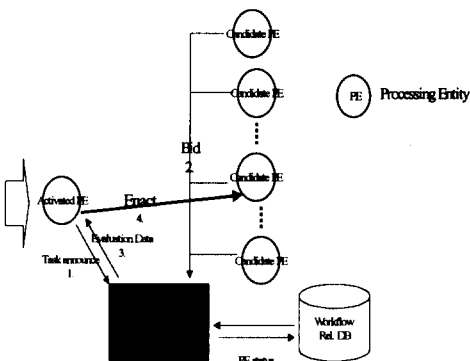


그림 3 개체들의 Bidding에 의한 enactment

워크플로우 관리 시스템의 작용방법은 일정계획에서의 과정과 흡사한 점을 발견할 수 있다. 워크

플로우 관리 시스템에서 엔진의 가장 중요한 역할의 하나는 프로세스의 진행에 필요한 태스크들을 각 처리개체에 할당하는 스케줄링이다. 이러한 관점은 각 태스크를 처리시간을 가지는 작업으로 보고 처리 개체를 기계로 볼 수 있다는 것을 의미한다.

동적인 스케줄의 생성은 오프라인 스케줄에 의해서 보다 온라인 스케줄에 의해 효과적으로 달성될 수 있으며 이는 워크플로우 관리 시스템에도 적용이 된다. 워크플로우 관리 시스템에서 bidding에 의한 처리개체(PE)의 선정은 그림 3과 같이 이루어진다.

최초의 태스크를 수행할 처리 개체가 활성화되고 나면 활성화된 개체들은 작업의 완료시점에서 서버에 작업완료에 따른 다음 태스크 선정에 요청한다. 서버는 데이터 베이스에 있는 각 처리개체들의 상태를 파악하고 이에 따라 가능한 개체들로부터 Bid를 받아서 필요한 데이터의 리스트를 현재의 PE에 넘겨준다. 현재의 PE는 후보들을 평가하고 다음 태스크를 수행할 PE를 선정하고 작업을 끝낸다. 이때, 다음의 태스크를 처리할 PE의 선정 과정은 다음 그림 4와 같다.

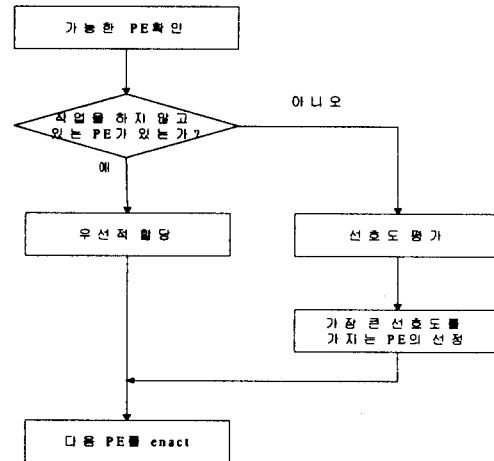


그림 4 다음 처리개체(PE)의 선정과정

선호도의 평가는 현재 후보 PE에 할당되어 있는 태스크들의 처리 시간의 합, 대상 태스크의 Due Date, 두 처리 개체간의 거리의 함수로 표현되며 이 함수는 워크플로우의 특성과 각 태스크별로 다르게 설정할 수 있다.

$$PR_i = f(DD, Pt_i, D_i)$$

DD: 대상 태스크의 due date

Pt<sub>i</sub>: i번째 후보 PE에 있는 태스크들의 처리시간의 합

D<sub>i</sub>: 현재 태스크를 수행중인 PE와 대상 태스크간의 거리

## 4.3 동적 구조의 이점

본 연구에 의한 시스템은 다음과 같은 이점을 지닌다.

- 오동작에 대한 대처
- 유연한 경로 설정
- 워크플로우의 수정이 용이
- 처리 개체의 추가 등의 확장성
- 수행도의 개선

## 5. 결론 및 추후 연구과제

본 연구에서는 빠른 변화에 대처가 가능한 워크플로우 관리 시스템의 요구사항을 정리하고 자치성을 지닌 개체들이 bidding을 통해 실행 시에 태스크를 할당하는 동적인 구조를 제안하였다. 이 구조에서는 서버의 개념을 없애고, 서버의 역할을 클라이언트들이 담당하도록 함으로써, 프로세스의 동적인 변화에 적절히 대응할 수 있도록 하였다. 이 구조는 서버가 프로세스의 진행에 대한 스케줄을 모두 생성함으로써 발생하는 문제인 비유연성을 극복할 수 있다.

본 시스템에서 제안한 구조는 bidding 메커니즘을 사용하며, bidding 함수의 설계에 따라 수행도가 달라질 수 있다. 다양한 워크플로우 형태에 대하여 bidding 함수의 설계와 수행도에 미치는 영향에 대한 분석이 행해져야 할 것이다.

또한 트랜잭션 처리의 규칙관리를 이용하여, 동적인 환경에 대한 적응성을 제고할 수 있는 방안에 대해서 고찰이 필요하며, 제안된 구조에 대해서, 워크플로우 시스템간의 상호 운영성에 대한 분석과 이를 지원하기 위한 연구가 행해져야 할 것이다.

## 6. 참고문헌

- [1]김성주, 김용훈, 최인준, Workflow상에서의 규칙관리, 한국경영과학회/대한산업공학회추계공동학술대회 논문집, pp606-609, 1997.4.
- [2]문장석, 자치적 객체들간의 Bidding을 통한 FMS 생산현장통제에 관한 연구, 서울대학교대학원 산업공학과 석사논문, 1993.
- [3]G. Alonso and H.J.Schek. Research Issues in Large Workflow Management Systems. Proc. Of the NSF workshop on workflow and process automation in information systems. University of Georgia, May 1996.
- [4]G. Alonso, D. Agrawal, A. El Abbadi, M. Kamath, R. Gnthr, and C. Mohan. Advanced Transaction Models in Workflow Contexts. In Proceedings of the 12th International Conference on Data Engineering, February 1996.
- [5]G. Alonso, M. Kamath, D. Agrawal, A. El Abbadi, R. Gnthr and C. Mohan. Failure Handling in Large Scale Workflow Management Systems. Technical Report RJ9913, IBM Almad Research Center, November 1994.
- [6]Davis,R., and Smith,R.G., 1983,Negotiation as a Metaphor for Distributed Problem Solving, Artificial Intelligence, Vol.20,No.1, 63-109.
- [7]D. Georgakopoulos, M. Hornick, and A. Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. Distributed and Parallel Databases,

3(2):119-154, April 1995.

[8]D. Hollingsworth. The Workflow Reference Model. Technical Report TC00-1003, Issue 1.1, The Workflow Management Coalition, Brussels, Belgium, November 1994.

[9]M. Kamath, G. Alonso, R. Gnthr, and C. Mohan. Providing High Availability in Very Large Workflow Management Systems, Proc. Of The Fifth International Conference on Extending Database Technology, March 1996.

[10]N. Krishnakumar and A. Sheth. Managing Heterogeneous Multi-system Tasks to Support Enterprise-wide Operations. Distributed and Parallel Databases, 3(2): 155-186, April 1995.

[11]P. Lawrence. Workflow Handbook 1997. John Wiley & Sons LTD, 1997.

[12]J. A. Miller, A. P. Sheth, K. J. Kochut, and X. Wang. CORBA-based Run-Time Architectures for Workflow Management Systems. Citedogac, 7(1):16-27, Winter 1996.

[13]C. Mohan, G. Alonso, R. Gnthr, and M. Kamath. Exotica: A Research Perspective on Workflow Management Systems. Bulletin of the Technical Committee on Data Engineering, IEEE, 18(1), March 1995.

[14] Shaw, M. J., FMS Scheduling as Cooperative Problem Solving, Annals of Operations Research, Vol. 17, 323-346. 1989.

[15]A. Sheth and K. J .Kochut, Workflow Application to Research Agenda: Scalable and Dynamic Work Coordination and Collaboration Systems, Technical report of Computer Science Department of The University of Georgia, GA30602-7404.