

BPM을 이용한 웹서비스 기반의 SCM 프로세스 실행 Web service-based SCM process execution using BPM

배혜림*, 서용원**, 최용선***, 장진영***

* 부산대학교 산업공학과

** 단국대학교 경영학과

*** 인제대학교 시스템경영공학과

초 록

기업간 경쟁의 격화와 정보기술의 발달로 인하여 많은 기업들이 공급사슬관리(SCM)를 도입하고 있다. 이에 따라, 공급사슬상의 관련 기업간의 상호작용과 정보공유에 기반한 공급사슬 통합이 추진되고 있으며, 웹서비스와 같은 시스템에 독립적인 기술 적용의 필요성이 제기되고 있다. 웹서비스를 이용하면서도 높은 수준의 공급사슬 통합이 이루어지기 위해서는, 여러 상호작용주체가 참여하는 글로벌 프로세스 수준에서 SCM 프로세스를 관리하고 제어할 필요가 있으며, 본 논문에서는 비즈니스 프로세스 자동화 및 관리도구로 최근 각광을 받고 있는 BPM을 도입하여 각 업무단위가 웹서비스로 구성된 환경에서 SCM을 구성하는 프로세스를 효과적으로 통합하여 실행하는 방안을 제시한다. 이러한 접근법은 SCM 프로세스의 진행과정을 보다 효율적으로 관리, 모니터링하는 것이 가능하며, SCM 프로세스의 수행결과를 토대로 프로세스의 개선에 대한 방안을 모색할 수 있다. 또한, SCM 프로세스의 진행과정에서 의사결정이 필요한 경우 이에 대한 지원 에이전트를 웹서비스로 구성하여 보다 신속한 프로세스 수행을 가능하게 한다.

주제어: SCM, BPM, Web services, BPEL

1. 연구의 개요

최근의 기업환경의 변화는 경쟁의 심화와 경영활동의 복잡성으로 인해 많은 기업들이 개별적인 경쟁력을 통해 생존하기 어려운 방향으로 진행되고 있다. 따라서, 각 기업들은 비약적으로 발달한 정보기술을 바탕으로 상호 공급사슬(Supply Chain)을 구성하고 상호작용하려는 노력을 기울이고 있다. 이러한 노력의 일환으로 공급사슬관리(Supply Chain Management: SCM)의 개념이 매우 중요한 이슈로 떠오르고 있으며, 국내외의 많은 기업들이 이를 도입하여 공급사슬상의 관련 기업간의 상호작용과 정보공유에 기반한 공급사슬 통합을 추진하고 있다.

이러한 SCM의 도입은 급격하게 발달한

정보기술에 의해 가속화되고 있으며, 최근의 정보기술의 새로운 패러다임으로 등장한 웹서비스가 각광을 받고 있다. 그러나, SCM을 구현하는 데는 단순한 메시지 교환을 통한 이기종 시스템간의 상호작용을 넘어 공급사슬에서 발생하는 프로세스의 원활한 실행과 통합관리를 필요로 한다.

본 논문에서는 비즈니스 프로세스 자동화 및 관리도구로 최근 각광을 받고 있는 BPM(Business Process Management)을 도입하여 각 업무단위가 웹서비스로 구성된 환경에서 SCM을 구성하는 프로세스를 효과적으로 통합 실행하는 방안을 제시한다. SCM프로세스를 실행하는데 BPM을 도입하기 위하여 먼저 요구되는 SCM 프로세스의 모델링을 위하여 웹서비스의 프로세스 기술언어인 BPEL (Business Process Execution Language)을 도입함으로써 이기종 정보시스템을 사용하는 SCM 파트너 간에 원활하게 프로세스를 주고 받을 수 있도록 한다. 이를 바탕으로 SCM 프로세스를 실행하는데, SCM 프로세스의 실행과정은 재주문 여부의 결정, 주문량의 결정과 같은 의사결정을 포함하게 된다. SCM 프로세스에서 발생하는 의사결정은 사람의 개입을 최소화한 에이전트 형태로 구현하는 것이 바람직하며, BPM이 의사결정 에이전트를 활용할 수 있도록 웹서비스 통해 에이전트를 호출하면서 이 결과를 따라 프로세스를 진행한다.

본 연구를 통하여, BPM을 통한 효과적인 공급사슬 주체 통합과 높은 수준의 프로세스 공유 및 공급사슬 내의 운영성과정보를 통해 보다 정확한 운영상의 의사결정을 수립할 수 있을 것으로 기대한다.

2. 연구의 배경

2.1 공급사슬관리의 개념

1980년대에 접어들면서, 기업들은

새로운 제조 기술과 전략을 도입함으로써 비용을 절감하고 시장에서의 경쟁력을 강화하였다. 그러나, 1990년대에 들어서면서, 많은 기업들은 이와 같은 개별 기업단위의 비용 절감 노력이 한계에 달하였음을 인지하게 되었으며, 글로벌 시장의 경쟁 격화, 제품 수명주기의 단축, 고객 요구 수준과 같은 시장 환경의 변화에 따라, 기업들은 공급 사슬범위의 효과적인 관리의 필요성을 느끼게 되었다.

공급 사슬은 공급자, 제조업체, 창고, 분배센터, 소매점 및 이들을 통해 움직이는 자재, 공정중 재고, 완성품을 포함한다. 다수의 기업이 관여하는 이러한 공급 사슬에서는 비용을 줄이고 서비스 수준을 높이기 위해서 각 단계 사이에서 일어나는 상호작용을 고려하고 이들에 대해 조정이 수행되어야만 한다. 이러한 공급 사슬 전반의 효율적인 관리를 목표로 하는 활동을 공급 사슬 관리라 한다. 공급 사슬 관리는 “정확한 양의 상품이, 정확한 장소로, 정확한 시간에 생산되고 분배되며, 요구되는 서비스 수준을 충족하면서 시스템 비용을 최소화하기 위해, 공급자, 제조업체, 창고, 설비들을 효과적으로 통합하기 위한 일련의 접근 방법[Sims2000]”으로 정의된다.

2.2 BPM의 개념

BPM은 기업의 생산성을 제고하기 위해 업무프로세스를 체계적으로 설계, 관리, 개선하는 활동을 지원하는 총체적인 관리 방법론이다[Smit2003]. 미리 정의된 정형화된 프로세스를 따라 실제 업무를 참여자원에 부여함으로써 업무를 진행하는 방식으로, 기술적으로는 비즈니스 프로세스의 자동화를 의미하는 워크플로우(Workflow)를 포괄하고 있다. BPM 시스템은 이러한 업무프로세스관리 방법을 구체화하여 효율적인 업무 환경을 지원하는 소프트웨어 시스템이다[Smit2003].

BPM 시스템에서 컴퓨터가 이해할 수 있는 형태인 프로세스 모델은 정의시(build-time) 단계와 실행시(run-time) 단계로 나누어 고려하는 것이 일반적이다[배2002]. 정의시 단계는 프로세스 수행을 준비하는 과정으로, 프로세스를 이루는 단위업무와 이들 간의 선후 관계 그리고 각 단위업무를 수행하는데 필요한 세부 속성들을 정의한다. 실행시 단계에서는 정의시 단계의 프로세스 모델을 해석하여 실제 프로세스를 실행, 관리, 통제한다.[Lawr1997].

최근의 BPM을 구축하는데 있어 가장 큰 이슈 중 하나는 이기종 시스템간의 상호운영성(interoperability) 문제이다. 즉, 서로 다른 환경에서 실행되는 개별 비즈니스 프로세스를 하나의 글로벌 프로세스로 구성하고 이를 실행하는데 있어 발생하는 기술적 문제를 해결하는 것이다. SCM은 여러 기업이 상호작용하면서 프로세스를 진행하는 대표적인 문제영역으로 볼 수 있으며, 이러한 SCM 프로세스를 모델링하고 실행하는데 있어 BPM을 도입함으로써 프로세스 실행 자체의 효율성 제고는 물론이고, 체계적인 관리, 감독을 통한 지속적 개선을 기대할 수 있다.

2.3 웹서비스와 BPEL

웹서비스는 XML기술을 기반으로 조직의 어플리케이션을 서비스화하여 인터넷을 통하여 제공하는 기술이다. 기업은 일반 사용자의 필요와는 달리, 이기종 기업간 정보시스템을 보다 원활하게 통합하기 위한 필요성을 지니고 있다. 웹서비스는 이러한 상호운영성 문제를 해결하기 위한 현실적인 대안이 되고 있으며, 최근에는 웹서비스를 활용하여 기업간의 프로세스를 통합하려는 노력들이 이루어지면서 WSCI (Web Services Choreography Interface)이나 BPEL4WS (Business Process Execution Language for Web Services)과 같은 웹서비스를 기반으로 한 비즈니스 프로세스 수행 언어들이 등장하고 있다. 특히, BPEL4WS 표준으로 기술되는 BPEL 언어는 XML기반의 웹서비스 정의 언어이다 [Snel2002]. BPEL은 비즈니스 프로세스를 추상모드와 실행모드의 두가지 측면에서 지원함으로써, 웹서비스에서 비즈니스 프로세스를 조회/검색하는 작업뿐만 아니라 웹서비스로 구성된 비즈니스 프로세스를 실제 수행할 수 있도록 하고 있다.

3. 공급사슬에서 정보공유의 필요성 및 BPM의 적합성

공급 사슬 관리를 구성하는 활동들 가운데 가장 중요한 것 중 하나는 공급 사슬 전반에 분포하고 있는 재고에 대한 효과적인 관리이다. 공급 사슬 관리가 일반화되지 않았던 과거에는, 설비들간의 원활한 물류를 위한 완충장치로서의 안전재고의 개념이 중시되었다. 그러나, 이러한 방식의 재고관리는 서로 다른 재고관리단위의 연결부위마다 쌓여 있는 안전재고의 양이 상당한 재고부담을 야기하였다. 또한, 소위 채찍효과(bullwhip effect)[Lee1997]라고 하는

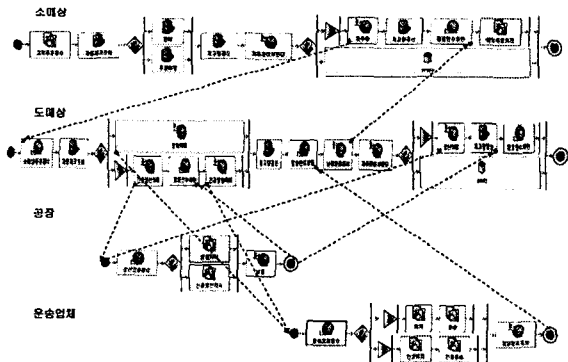
공급 사슬 상위 단계로 올라가면서 왜곡을 발생시키는 문제가 있다. 효과적인 공급 사슬 관리를 위해서는 공급 사슬 전반에 걸쳐 재고 및 최종 수요 정보의 실시간의 공유가 필요하며, 이에 입각한 실시간 의사결정과 통합적 운영이 필요하다.

최근 발전하고 있는 BPM 기술은 이러한 이기종 시스템의 공급사슬의 통합적 운영을 구현하는 데에 효과적으로 이용될 수 있다. BPM은 정보공유를 원활하게 해주며, BPM Engine은 공급 사슬 프로세스 내의 각 활동을 모니터링하여, 운영상의 의사결정을 위한 실시간의 정확한 정보를 제공할 수 있다. 따라서, BPM과 이것이 제공하는 운영정보를 기반으로 하는 의사결정 agent를 결합함으로써, 공급사슬 내의 서로 다른 기업간의 운영을 통합하는 시스템을 효과적으로 구현할 수 있다.

4. BPEL을 이용한 SCM모델링

4.1 SCM 프로세스

본 논문에서는 SCM프로세스를 BPM을 사용하여 실행하는 모델을 제시한다. 이를 위하여 다음 그림 1과 같은 전형적인 SCM 프로세스를 대상으로 개발된 접근법을 적용하기로 한다.



[그림 1] SCM 프로세스 예제

4.2 BPEL 모델링

본 논문에서는 3.1에서 기술된 SCM프로세스를 BPEL로 모델링한다. 예제에서 보는 바와 같이 SCM프로세스는 공장, 도매상, 소매상, 운송회사의 서로 다른 참여 주체들이 포함되어 있다. 이들 SCM 참여주체들은 '주문처리' 프로세스를 위해서 내부의 프로세스를 수행하고 있으며 이들 내부 프로세스가 합쳐져 하나의 SCM 프로세스를 형성하게 된다. 이러한 과정에서 상호작용이 원활하게 이루어지기 위해서는

프로세스 정보의 상호 교환이 필요하게 되는데, 각 참여주체들이 이기종 시스템을 채용하는 경우 프로세스의 정의에 대한 표준적인 언어가 필요하다.

본 연구에서는 이러한 프로세스의 표준적인 모델링을 위하여 BPEL을 활용한다. BPEL은 원래 웹서비스 기반의 비즈니스 프로세스 기술 언어이지만, BPM에서 사용되는 시스템 의존적인 PDL에 비해 의미적인 표현력이 떨어지지 않은 것으로 알려져 있다 [Wohe2002]. BPM에서 다루는 프로세스 모델을 BPEL로 변환하는 주요 방법은 다음 표1과 같다.

[표 1] BPEL 모델로의 변환방법

패턴	Structured	(Structured) BPEL
직렬	sequence	<sequence> ... </sequence>
조건분기	switch	<switch> <case condition="조건식1">...</case> <case condition="조건식2">...</case> </switch>
병렬	Flow	<flow>...</flow>
반복	loop	<loop condition="조건식">...</loop>
복합	Loop - switch	<loop> <switch> <case> <sequence>... </sequence></case>... </switch> </loop>

5. SCM 의사결정 agent

BPM이 제공하는 실시간의 수요 및 재고 정보를 활용하여, 공급 사슬의 비용을 최소화하는 재주문 의사결정 agent를 구현할 수 있다. 의사결정 agent는 웹서비스로 구현되며, 공급사슬 프로세스에서 의사결정이 필요한 시점에 관련 정보를 agent에 전달하여 의사결정의 결과에 따라 이후의 프로세스를 결정하게 함으로써 공급사슬 프로세스 내에 통합된다.

Agent가 수행하는 재주문 의사 결정은 [Seo2002]의 주문 리스크 정책을 활용하였다. 주문 리스크 정책은 최적의 재주문 시점을 결정하며, 특히 유통시스템에서 일반적으로 나타나는 분배형 시스템에서, 기존의 계층 재고 정책에 비하여 높은 성과를 보임이 입증되어 있다.

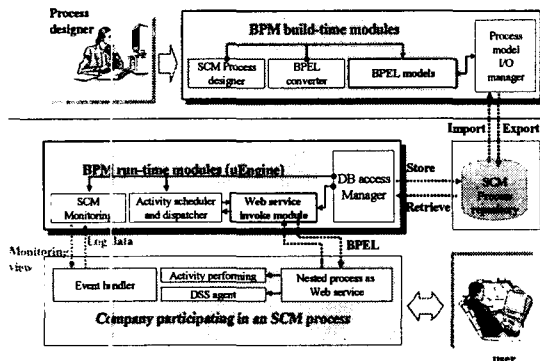
본 연구에서 대상으로 하고 있는 한 개의 도매상과 여러 개의 소매상으로 구성된 2계층

분배형 시스템에서, 주문 리스크 정책에 따른 재주문 시점을 결정하는 모형은 주문 리스크의 정규근사방법[서2002]을 사용하였으며, 구체적인 모형의 세부사항은 해당 논문을 참조하기 바란다.

6. BPM을 이용한 프로세스 실행

6.1 SCM프로세스 수행절차 및 시스템구조

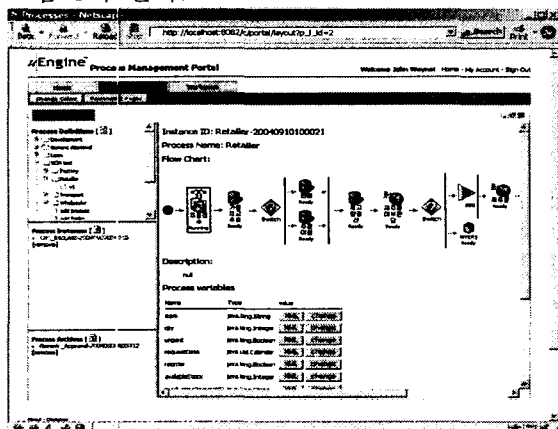
BPM을 이용한 SCM프로세스 실행을 위한 시스템의 전반적인 구조와 절차는 다음 그림 2와 같다.



[그림 2] 시스템 구조 및 실행절차

6.2 uEngine 및 실행결과

본 연구에서는 SCM을 실행하기 위한 BPM엔진으로 uEngine[장2004]을 사용한다. uEngine을 사용하여 본 연구에서 고려하고 있는 SCM프로세스를 실행하는 화면은 다음 그림 3과 같다.



[그림 3] uEngine 실행화면

7. 결론 및 추후연구과제

본 논문에서는 BPM을 이용하여 여러 기업이 관여하는 SCM 프로세스를 실행하는 모델을 제시하였다. 본 논문의 접근법은 SCM 참여주체들간의 정보 공유를 원활하게 하고 전체적인 SCM프로세스의 관리 및 모니터링을 가능하게 한다. 앞으로 BPM에서 관리하는

SCM프로세스의 정보를 바탕으로 프로세스를 개선하고 이를 통하여 체계적인 지식관리를 수행하는 구체적인 방법론으로 발전시킬 필요가 있다.

Acknowledgement

이 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음. (KRF-2003-002-B00118)

참고문헌

- [배2002] 배혜림, e-Business 문서변경관리: 워크플로우 프로세스를 위한 비즈니스문서변경관리, 공학박사논문, 서울대학교, 2002.
- [서2002] 서용원, 성제훈, 정성원, 강석호, 박진우, 복합 포아송 수요를 갖는 2계층 분배형 공급사슬에서 공유 재고 정보를 활용한 재주문 정책, 한국 SCM 학회지, 제1권, 제2호, pp.35-46, 2002.
- [장2004] 장진영, uEngine 사용자/개발자 가이드(ver 1.0), 2004
- [Hogg1970] Hogg, R. V., and A. T. Craig, Introduction to mathematical statistics, 4th ed., Collier Macmillan Publishers, London, 1970
- [Lawr1997] Lawrence, P. Workflow Handbook, Wiley, 1997.
- [Lee1997] Lee, H. L., V. Padmanabhan, and S. Whang, Information Distortion in a Supply Chain: The Bullwhip Effect, Management Science, vol. 43, no. 4, pp. 546-558, 1997.
- [Seo2002] Seo, Y., S. Jung, and J. Hahm, Optimal reorder decision utilizing centralized stock information in a two-echelon distribution system, Computers & Operations Research, Vol.29, No.2, pp.171-193, 2002.
- [Simc2000] Simchi-Levi, D., P. Kaminsky, and E. Simchi-Levi, Designing and Managing the Supply Chain, Irwin McGraw-Hill:Singapore, 2000.
- [Smit2003] Smith, H. and Fingar, P., Business Process Management - The Third Wave, Meghan-Kiffer Press, 2003.
- [Snel2002] Snel, J., Automating business processes and transactions in Web services, <http://www-106.ibm.com>, 2002
- [Wohe2002] Wohe, P., Aalst, W.M.P., Dumas, M., and Hofstede, A. H. M., Pattern Based Analysis of BPEL4WS, Technical Report FIT-TR-2002-4, QUT