

분산객체와 XML 기반의 생산계획 컴포넌트 개발에 관한 연구

민대기¹ · 장태우^{2*} · 박찬권³ · 박진우²

¹LG CNS / ²서울대학교 산업공학과 / ³영산대학교 정보경영학부

A Study on Software Component Development for Production Management Using Distributed Objects and XML Technologies

Dae-Ki Min¹ · Tai-Woo Chang² · Chankwon Park³ · Jinwoo Park²

¹CIM Team, LG CNS, Yeouido-dong, Yeongdeungpo-gu, Seoul, 150-712

²Department of Industrial Engineering, Seoul National University, Gwanak-gu, Seoul, 151-744

³School of Information & Management, Youngsan University, Yongsan-si, Gyeongsangnam-do, 626-847

New trends such as electronic commerce, virtual organizations, e-business applications, etc. increase the dependence of production management on information software systems and contribute to the needs for global, distributed object systems. This paper presents a component based approach for production management systems under the multi-tier distributed information system architecture using UML(Unified Modeling Language), CORBA(Common Object Request Broker Architecture) and XML(eXtensible Markup Language) technologies, and propose rules for mapping a UML class diagram to a XML DTD (Document Type Definition). And we adapt it to the prototype system implementation. The components are implemented by CORBA and we use XML messages for the information exchange between components.

Keywords: production management component, distributed objects, XML, CORBA

1. 서론

최근 전자상거래, 가상조직, e-Business 등의 개념들이 출현하면서 분산시스템에 대한 관심과 요구가 증가하고 있는 추세이다. 글로벌화된 생산시스템의 장점을 얻기 위해서는 이와 같은 정보기술을 활용한 분산생산시스템의 구축이 필요한데, 이를 위한 기업 프로세스의 확장과 분산화는 단일기업이 갖는 '지식·능력'의 한계를 극복하고, 다른 조직의 핵심 경쟁력을 활용할 수 있다는 것을 의미한다(Klement, 1999).

분산환경의 조직모델을 지원하기 위한 정보시스템 구조는

업무 구조, 어플리케이션 구조, 그리고 맨 아래의 기술 구조 등 세 개의 계층으로 구분할 수 있고(Hasselbring, 2000), 이들 분산 환경에서 조직 간의 정보 및 프로세스를 통합하기 위해서는 조직 내부의 수직적인 통합뿐만 아니라 수평적인 통합도 요구된다(Wil, 2000).

수평적인 통합의 핵심이 되는 어플리케이션 구조 계층의 통합에는 보통 메시지 기반의 서비스를 이용한 정보교환 방법이 사용되고 있으나, 독립적인 어플리케이션들 사이의 상호운용성을 지원하기 위해서는 표준화된 메시지를 정의해야 하는 문제가 존재한다. 최근의 인터넷과 XML 등의 정보기술을 이용한 새로운 정보공유 방안은 이에 대한 하나의 좋은 해결책을

제시해 주고 있다. 특히 인터넷 환경의 비즈니스 응용 분야에서는 분산된 이기종 간의 정보공유와 기존 기술·정보와의 호환성 및 기계 가독성의 측면에서 XML과 같은 기술의 적용이 반드시 필요하다.

또한 단일기업의 수직적인 기업정보시스템 통합 문제와 마찬가지로 네트워크형의 기업 간 구조에서의 정보시스템 통합 문제도 존재한다. 즉 분산환경에서는 이기종 시스템 간의 통합이 어려울 뿐만 아니라 업무 프로세스의 통합 역시 힘들기 때문에 분산기업에서의 조직 모델을 어떻게 시스템 설계와 통합에 반영할 것인가가 중요한 요소가 된다. 이와 관련하여 Lewandowski(1998)는 다양한 환경에서 작동할 수 있는 독립적이고 자기관리가 가능한 분산객체와 컴포넌트를 사용함으로써 커스터마이징을 보다 용이하게 할 수 있고 업무객체 간의 상호작용을 위한 강고한 프레임워크를 제공할 수 있음을 주장한 바 있다.

본 연구에서는 네트워크 구조의 분산기업환경에서 공급사슬을 구성하는 제조기업의 생산계획 프로세스를 반영하기 위한 정보시스템의 구조와 분산환경에서의 효율적인 정보공유 방안을 제시하고자 한다. 즉, 글로벌 생산관리 환경에서 수정 및 확장이 용이한 객체지향 컴포넌트를 설계하기 위하여 UML을 이용하여 설계된 프로세스 모델을 설계하고, 이를 CORBA 기반의 분산 객체로 구현한다. 또한 분산된 이기종 시스템 간에 객체들 사이의 메시지 교환을 위하여 ORB 이외에 XML 메시지를 생성하는 방안을 제시한다. 이하 2절에서는 분산환경과 제조정보시스템 및 기반기술과 관련한 기존의 연구현황을 살펴보고, 3절에서는 대상 생산계획 시스템에 대한 설계 및 개발 방법과 프레임워크를 제시하며, 4절에서는 3절의 프레임워크에 따르는 설계 내용과 그 결과물인 프로토타입 시스템에 대해 설명한다.

2. 연구현황

공급사슬 및 제조정보시스템에서 객체지향 기법을 이용한 기존의 연구들은 주로 단일 기업의 스케줄링 수준에서의 문제를 대상으로 하고 있으며, 분산환경을 대상으로 하는 문제에 대한 연구는 90년대 후반에야 사례를 찾아볼 수 있다. 분산환경을 대상으로 하는 연구는 주로 에이전트 기술과 분산객체 기술 등의 기반기술을 이용하여 네트워크 구조의 분산제조환경에 대한 문제를 다루고 있다(Hardwick, 1996; Alfieri, 1997; Bullinger, 1997; Aguirre, 1999; Lee, 1999; Sheremetov, 1999; Zhou, 1999).

Hardwick(1996) and Zhou(1999)는 ORB와 STEP 기반의 제조정보관리 시스템 구조를 제안하였으며, Alfieri(1997)는 OMT 기법을 이용하여 분산환경에서의 생산 및 분배 계획에 따른 재고 비용의 변화에 대한 시뮬레이션 연구를 수행하였다. Aguirre(1999)는 분산객체(CORBA)와 STEP 등의 기술을 기반으로 하는 제조정보 공유 개발 아키텍처를 제안하였으며, Sheremetov

(1999)는 분산 에이전트 기술을 이용하여 FMS 시스템을 개발하였다. 하지만, 이런 연구들은 주로 제조정보의 공유, 분산 시뮬레이션, FMS 시스템 등의 문제에 초점을 두고 있으며, 공급사슬과 관련된 문제 환경을 제대로 반영하고 있지는 못하다.

Papazoglou(2000)는 공급사슬과 관련한 기업의 어플리케이션은 다음의 네 가지 특성을 가져야 한다고 말하고 있으며, 이런 특징을 반영하는 시스템의 핵심 기술로 분산객체를 이용한 방법론을 제시하였다.

- 새로운 업무모델(New business models): 기업 내부 프로세스와 전체 공급사슬 프로세스의 연계
- 기업 간 공동운영(Cross-enterprise interoperability): 정보의 원활한 교환
- 변화 관리(Change management): 유연하고 융통성 있는 프로세스 및 레거시 시스템과의 연동
- 조직적인 기반구조(Organization infrastructure): 분산시스템과 중앙집중시스템 사이의 조화

Bullinger(1997)의 연구는 객체지향 기법을 이용하여 공급사슬의 유형과 비슷한 복수지역 생산계획 시스템을 설계하였으나, 프로세스 연계와 정보 공유 문제 등에서 Papazoglou(2000)가 제시한 시스템의 특징을 모두 반영하지는 못하고 있다. 여기에는 분산환경에서 내·외부 프로세스가 표준 업무 모델에 따라 연계될 수 있고, 정보의 교환을 원활히 할 수 있으며, 컴포넌트를 사용하여 다양한 업무프로세스를 유연하고 효율적으로 처리할 수 있는 정보시스템과 관련된 추가적인 연구가 필요하다.

최근의 정보시스템은 인터넷 기반의 분산환경을 지원하기 위하여 분산객체기술을 중심으로 한 객체지향 설계와 조직 간 프로세스 통합 기술을 기반으로 하고 있으며, 그 발전 방향은 <표 1>과 같이 요약할 수 있다(Sutherland, 1999). 즉, 컴포넌트 기술과 XML 기술이 중요한 요소가 되고 있음을 알 수 있는데, 본 연구에서도 이들 요소기술을 적극적으로 반영하였다.

분산환경에서의 정보시스템 구조는 분산객체를 기반으로 하는 다계층 클라이언트-서버 구조(Multi-Tier Client-Server Architecture)로 발전되어 가는 추세이다. 다계층 구조는 프론트

표 1. 정보시스템의 발전 방향

정보 시스템의 변화	상호 운용 방안
단일 시스템에서 모듈화된 시스템으로	API
모듈화된 시스템에서 객체지향 시스템으로	RMI
객체지향 시스템에서 컴포넌트 시스템으로	유연성 있는 결합 (Loosely coupled)
컴포넌트 시스템에서 워크플로우 시스템으로	XML 메시지
워크플로우 시스템에서 에이전트 기반 시스템으로	목표 탐색, 중개(brokering) 등

엔드의 클라이언트, 백 엔드의 데이터 관리 시스템, 그리고 두 계층 사이에 존재하는 하나 이상의 중간계층(Middle-Tier)으로 구성된다(Sun, 2000). 중간계층은 클라이언트로부터의 요청을 처리하기 위한 다양한 분산객체들로 구성되는데, 분산객체를 이용하여 다양한 업무 프로세스를 유연하고 효율적으로 처리할 수 있기 때문에 공급사슬에서의 정보시스템이 가져야 할 특징을 적절하게 지원할 수 있다. SAP R/3의 구조, IBM의 샌프란시스코 프로젝트 등의 개념은 모두 기업의 프로세스를 나타내는 중간 계층을 미리 생성해 놓은 객체 컴포넌트의 저장소 형식으로 구성하여, 이들의 결합에 의하여 다양한 영역의 업무 프로세스를 지원할 수 있는 유연한 구조를 제시한다.

한편 전통적인 EDI의 단점을 보완하기 위한 전자적 정보교환 시스템은 XML/EDI의 발전과 객체 클래스 및 분산컴퓨팅 기술을 웹과 연계시키는 방향으로 발전할 것으로 예상되고 있다(NCA, 1999). 또한 XML/EDI 시스템 프레임워크의 표준에 대한 논의가 진행된 후 여러 단체들을 중심으로 ebXML, RosettaNet 등의 연구가 진행중이며, 객체지향 EDI 시스템과의 통합에 관한 연구도 진행중이다. 아울러 XML을 이용한 정보·지식 공유시스템 설계, XML 문서의 저장 및 검색 방안, XML/EDI 시스템 개발 등과 XML 기반 프로세스 통합 방안에 관련된 연구를 중심으로 이루어지고 있다(Kip, 1999). 본 연구에서도 이와 같은 최근 기술 발전 추세를 감안하여 각 컴포넌트 사이의 정보교환 구조로서 XML 기술을 활용함으로써 유연하고 빠른 시스템 설계 및 개발을 가능하게 하고자 한다.

3. 시스템 설계 및 개발 방안

3.1 분산객체 기반의 다계층 구조

본 연구에서 제안하는 시스템 구조는 <그림 1>과 같이 업무객체를 이용한 분산객체 기반의 다계층 구조를 기반으로 하고 있다.

시스템 어플리케이션은 ORB로 연결된 분산 업무객체를 기반으로 이들 객체 집합을 이용하여 설계하고, 각 업무객체는

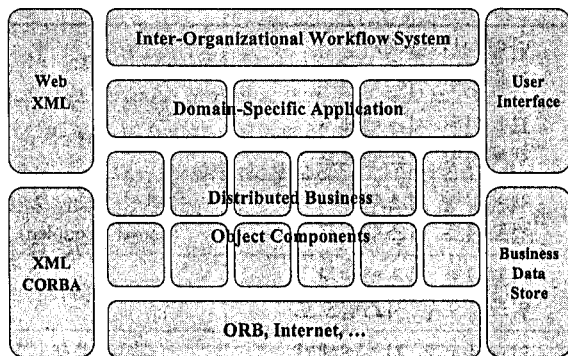


그림 1. 분산객체 기반의 시스템 구조.

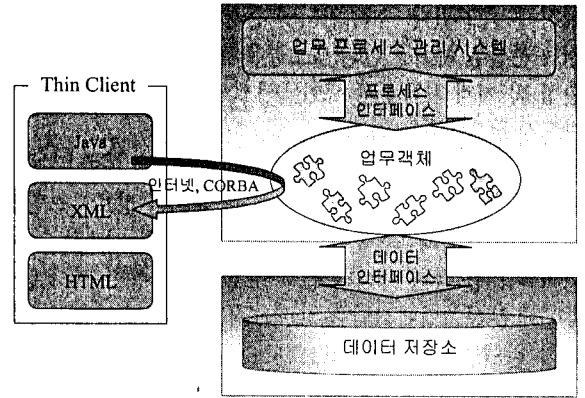


그림 2. 어플리케이션 구조.

UML을 이용하여 설계한 후, CORBA를 이용하여 구현된다. 사용자 인터페이스를 포함하는 모든 컴포넌트는 ORB 이외에 XML 메시지를 이용하여 필요 정보를 공유하도록 한다.

분산객체를 이용하여 설계된 업무 프로세스는 상위의 워크플로우 시스템에 의하여 관리되며, 객체는 이를 위한 인터페이스를 가져야 한다. 한편 어플리케이션은 <그림 2>와 같이 분산객체로 구성된 서버와 웹 기반의 클라이언트로 구성된다.

3.2 UML을 이용한 업무모델 설계

본 연구에서는 업무모델 설계를 위하여 객체지향 시스템의 분석 및 설계도구로서 업계 표준으로 자리잡아 가는 UML을 이용하였다.

업무모델은 업무객체(Business Object), 업무프로세스(Business Process), 업무이벤트(Business Event) 객체를 이용하여 나타내며, 업무객체는 업무의 논리적 흐름 및 정보를 포함한다. 이 업무객체를 중심으로 업무객체 사이의 관계를 나타내는 업무프로세스객체, 그리고 업무객체와 업무프로세스객체를 연계하는 업무이벤트객체가 존재한다(<그림 3>). 이같은 분류는 OMG

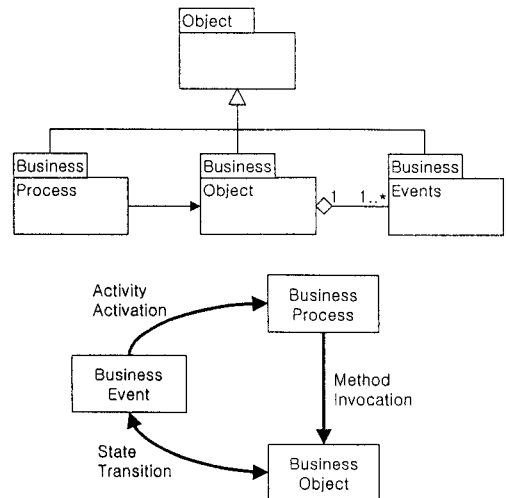


그림 3. 객체의 분류와 객체 간 관계.

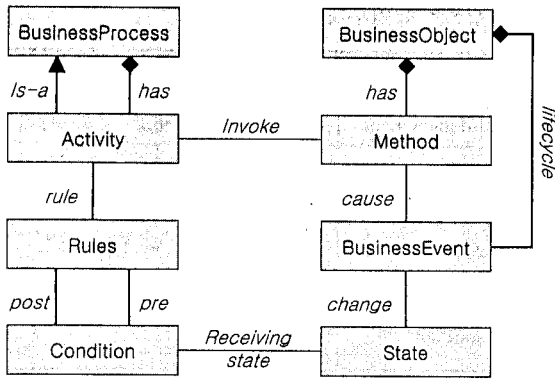


그림 4. 업무프로세스관리를 위한 시스템 설계.

(Object Management Group)의 업무객체 분류기준을 참조한 것이다(OMG, 1995).

업무프로세스객체는 업무객체에 영향을 주며(Method Invocation), 개별 업무객체에 포함된 여러 개의 업무이벤트는 업무객체의 상태에 따라서 특정 이벤트를 발생시킨다(State Transition). 업무객체의 이벤트에 의하여 업무프로세스 객체와 관련된 여러 업무객체가 다시 영향을 받게 된다. 이와 같이 업무모델을 업무객체와 업무프로세스 그리고 업무이벤트와 같은 객체를 이용하여 설계함으로써 물리적인 어플리케이션 시스템의 구현에 이를 반영하는 것이 용이해지며, 분산된 업무객체에 의하여 구성되는 어플리케이션들 사이에서 효율적인 업무프로세스의 관리가 가능해진다.

업무프로세스 관리를 위한 업무객체와 업무프로세스, 업무이벤트 객체 사이의 관계를 UML을 이용하여 객체모형(Class Diagram)으로부터 결정하여 나타내면 <그림 4>와 같다(Gillibrand, 2000; Sims, 1997).

객체모형으로부터 서로 독립적으로 존재 가능하고, 프로세스와도 분리 가능한 객체집합을 업무객체 컴포넌트로 구성하며, 이것을 CORBA를 기반으로 한 분산객체로 구현한다.

3.3 XML을 이용한 정보교환

CORBA를 이용하여 구현한 업무객체 사이의 정보교환은 ORB 이외에 XML 메시지를 이용하여 이루어지는데 XML 메시지의 구조 및 내용은 UML의 객체모형으로부터 결정된다. 이를 위하여 본 연구에서는 업무객체 사이의 정보교환을 위한 UML 정보모형을 XML 메시지로 변환하기 위한 규칙(UML2XML)을 제안하였다. <그림 5>는 UML 객체 사이의 일반화 관계에 대한 변환규칙에 대한 예를 제시한 것이다.

기본적으로 XML 문서 구조는 객체지향적 특성을 가지고 있으며, 본 연구에서는 XML의 이러한 특성을 이용하여, UML 객체모형의 객체(Class)를 XML의 ELEMENT로 변환하는 과정을 기반으로 규칙을 생성한 것이다. 이와 관련하여 Webber (1998)는 객체와 ELEMENT를 1:1 대응시키는 기본적인 아이디어를

UML2XML 규칙 : 일반화(Generalization)

- Generalization은 명시적으로 나타나지 않는다.
- 상속받는 Class의 경우 상위 Class에 대한 nested Element의 구조로 나타난다.
- 동일 수준의 Class는 OR("|")로 처리한다.

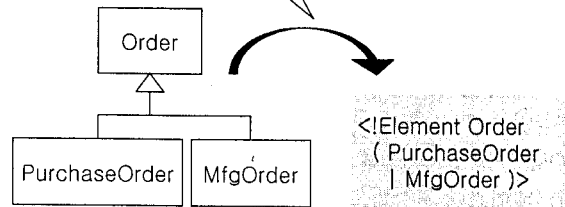


그림 5. 일반화(Generalization) 변환 규칙 예.

제안한 바 있으나, 본 연구에서는 이를 확장하여 각 객체를 XML의 ELEMENT로 변환하는 과정을 기본으로 하여 객체 사이의 관계, 객체의 속성 등에 대해서도 XML DTD로 변환하였다.

3.4 시스템 개발 방안

지금까지 분산객체 기반의 시스템 아키텍처를 기반으로 UML을 이용하는 분산객체 컴포넌트 시스템 설계 방안에 대해서, UML을 통하여 표현된 시스템을 CORBA를 이용하여 분산객체 컴포넌트로 구현하고 컴포넌트 사이의 정보교환을 위한 XML 메시지로 변환시키는 과정에 대해서, 그리고 XML 메시지의 생성을 위해 앞서 제시한 예와 같은 UML2XML 변환 규칙을 적용하는 것에 대해 설명하였다.

이상의 설계 및 개발 과정을 요약하면 <그림 6>과 같다.

4. 생산계획 컴포넌트 개발

4.1 생산계획 시스템

생산계획 및 통제 프로세스는 고객주문으로 생성된 수요 정

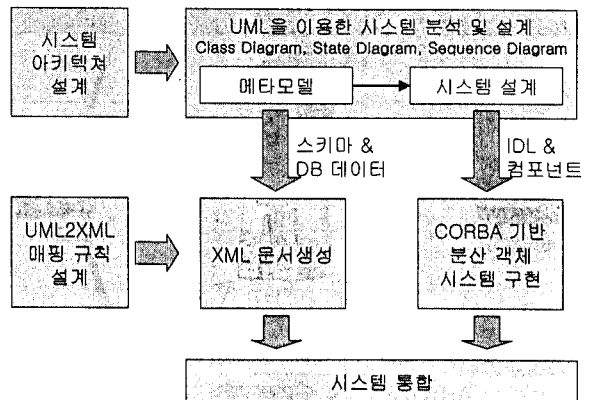


그림 6. 시스템 설계 및 개발 방안.

보를 이용하여, 제품별로 MRP, CRP를 수행하고, 그 결과로 생산·구매주문을 생성한 후에 주문에 대한 세부 통제를 하는 과정을 대상으로 한다(Scheer, 1994).

4.1.1 단일기업 모형

전통적인 생산계획 및 통제 시스템은 고객의 주문을 이용하여 용량을 고려한 자재소요계획을 수립하고, 자재소요계획 수립 결과 생성한 생산주문에 대하여 스케줄링을 수행하는 과정으로 구성된다(Vollmann, 1998). 단일기업을 대상으로 하는 전통적인 ERP 시스템과 같은 경우 생산계획 생성에 필요한 BOM, 공정(Routing), 재고 등의 모든 정보를 하나의 통합된 데이터베이스에 저장하고 관리하게 된다. 이와 같은 단일기업을 대상으로 하는 경우 시스템을 컴포넌트로 구성함으로써 높은 재사용성과 유지보수성 등의 장점을 얻을 수 있다(Yun, 1999).

4.1.2 분산기업 모형

기업의 운영시스템 중에서 ERP 시스템은 과거 독립적으로 수행되던 기업의 기능을 통합하여 관리하는 기능을 수행한 데 비하여 SCM 시스템은 공급사슬에서 발생하는 부품, 기자재, 원료 등의 구매 및 조달뿐 아니라 제조, 보관, 판매 등을 포함한 모든 활동을 효과적으로 운영하기 위한 의사결정 기능을 제공한다. 그 의사결정 과정에서 공급사슬상의 정보흐름과 상호작용에 대한 이해는 매우 중요하다. 공급사슬을 공급업자, 제조업자 그리고 고객으로 단순화한 경우 각 기업들 사이의 협력 과정과 공유가 필요한 정보는 <그림 7>에서 보는 바와 같다.

공급사슬상의 세 업무객체를 가운데 고객은 제조업자가 제공하는 제품 정보에 접근하여 필요한 제품을 주문할 수 있어야 하며, 주문의 처리과정에 대한 정보를 얻을 수 있어야 한다. 제조업자는 고객의 주문에 대해 납기약속 등의 정보를 제공함으로써 고객의 요구에 대응할 수 있어야 한다. 또한 구매 주문에 대한 생산 주문을 연계시켜 MRP 전개 결과를 이용할 수 있어야 한다. 공급업자는 제조업자의 재고정보와 생산계획 정보

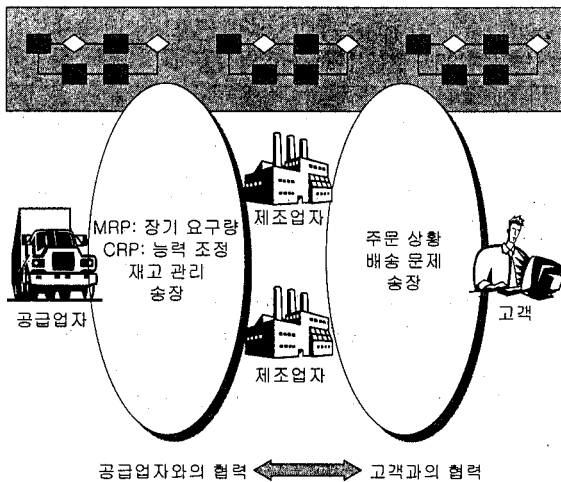


그림 7. 공급사슬상의 협력과정.

표 2. 생산계획 업무객체

업무객체	설명
MaterialManagement	Item 정보 및 재고 정보 관리
Order	고객주문, 생산주문, 구매주문 등의 주문 정보 관리
Equipment	자원정보 관리 및 자원 용량 계획 수행
ProductionPlan	분산자원의 용량을 고려한 생산 계획 수립

를 공유함으로써 제조업자의 요구에 안정적으로 대응할 수 있고, 제조업자 역시 공급업자의 가용성 정보 등을 공유함으로써 안정적인 계획을 수립할 수 있을 것이다. 이와 같이 개별 기업의 최적화 계획결과를 공급사슬 내에서 서로 공유함으로써 공급사슬 전반에 걸친 기업 간 협업이 가능해진다. 또한 기업 간 정보 공유를 통한 협업은 궁극적으로 분산기업의 업무 프로세스 최적화에 대한 방안을 제시한다(Jeong, 2000).

따라서 본 연구에서 제시한 업무객체를 이용한 생산계획 컴포넌트 설계 결과는 단일기업 모형뿐만 아니라 분산기업 환경에서도 매우 효율적이고 유연하게 사용될 수 있을 것이다.

4.2 업무 객체 설계

본 연구에서 대상으로 하는 생산계획 프로세스는 분산된 업무객체 사이의 상호작용에 의하여 이루어진다. 이를 위해 본 연구에서는 <표 2>와 같이 4개의 업무객체와 클라이언트 컴포넌트를 설계하였다.

위의 네 가지 업무객체 중에서 주문 및 생산계획과 관련된 정보를 각각 관리하기 위한 Order 및 ProductionPlan 업무객체 컴포넌트에 대한 UML 모델은 <그림 8>에 나타난 바와 같다. Order 객체에서 상속받는 Planned_Order 객체를 생산주문(Mfg_Order)과 구매주문(Purchase Order) 등의 객체가 다시 상속받을 수 있다. 이 모델로부터 CORBA를 이용한 분산객체를 구현하기 위한 IDL 파일을 생성한다.

4.3 업무프로세스 설계

업무객체를 기반으로 설계한 시스템의 경우, 프로세스는 업무객체 사이의 관계를 의미하며(Riemer, 1998; Klement, 1999), UML Sequence Diagram은 시스템의 시나리오를 객체 사이의 메시지 흐름을 중심으로 표현하기 위한 방법을 제시한다(Hardwick, 1996). 따라서 본 연구에서는 앞서 설계한 업무객체를 기반으로 업무객체들 사이의 관계를 이용하여 시스템의 프로세스를 설계하며, 이를 Sequence Diagram을 이용하여 표현하였다.

프로토타입 시스템 구현을 위한 예로 4.1에서 언급한 바와 같이 공급업자, 제조업자, 고객 등의 다수 참여자가 존재하는

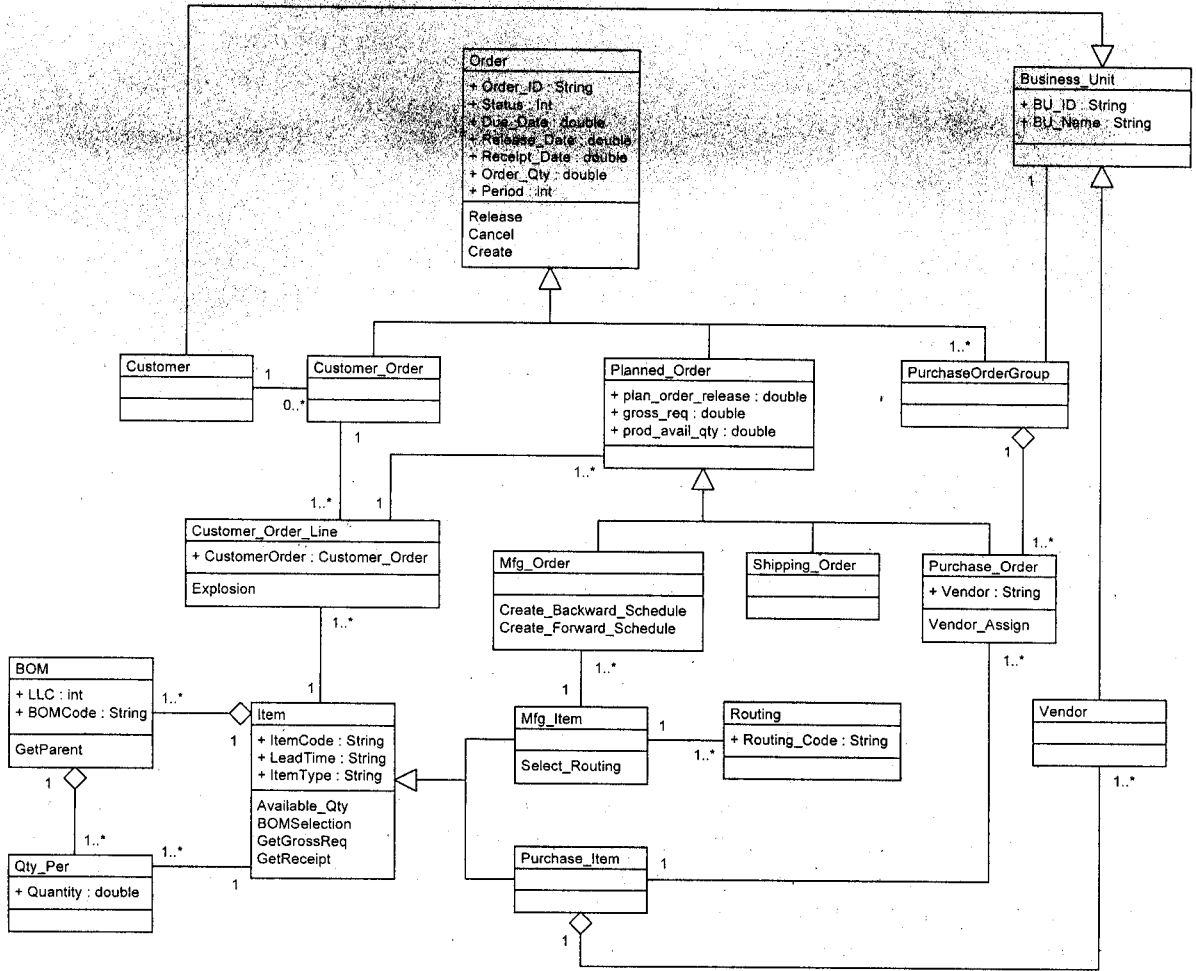


그림 8. Order 및 ProductionPlan 업무객체에 대한 Class Diagram.

생산계획 생성 프로세스에 대한 분석 및 설계 과정을 제시한다. 생산계획 시스템은 XML 문서로 전달되는 고객의 구매주문 내역을 이용하여 용량을 고려한 자재소요계획을 수립하고, 자재소요계획 수립 결과 생성한 생산주문에 대하여 스케줄링을 수행하는 과정으로 구성된다.

일반적인 생산계획·통제 시스템을 대상으로 본 논문에서 고려하는 기능과 앞서 설계한 네 가지의 업무객체와 전통적 생산계획 시스템 사이의 관계는 <그림 9>와 같다.

생산계획생성 프로세스는 생산계획 업무객체 컴포넌트의 인터페이스를 호출하면서 시작된다. 생산계획 업무객체 컴포넌트는 다수의 주문 업무객체 컴포넌트들, 자재관리 업무객체 컴포넌트들, 그리고 생산설비 업무객체 컴포넌트들로부터 정보를 받아들이어 생산주문(Mfg_Order)과 구매주문(Purchase_Order)을 생성한다. 생산설비 업무객체 컴포넌트는 설비에 대한 용량계획을 수행한다. 업무객체 컴포넌트에서 생성된 정보는 XML 메시지로 변환되어 전달된다. 이와 같은 프로세스를 Sequence Diagram으로 표현하면 <그림 10>과 같고, 생산계획 생성을 위한 업무 객체 컴포넌트 사이의 운영 프로세스를 Activity Diagram으로 표현한 것이 <그림 11>과 같다.

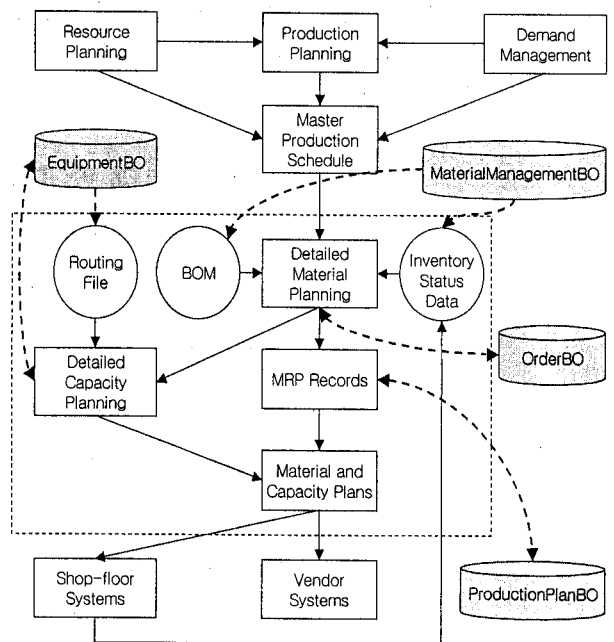


그림 9. 전통적 생산계획 시스템과 업무객체와의 관계.

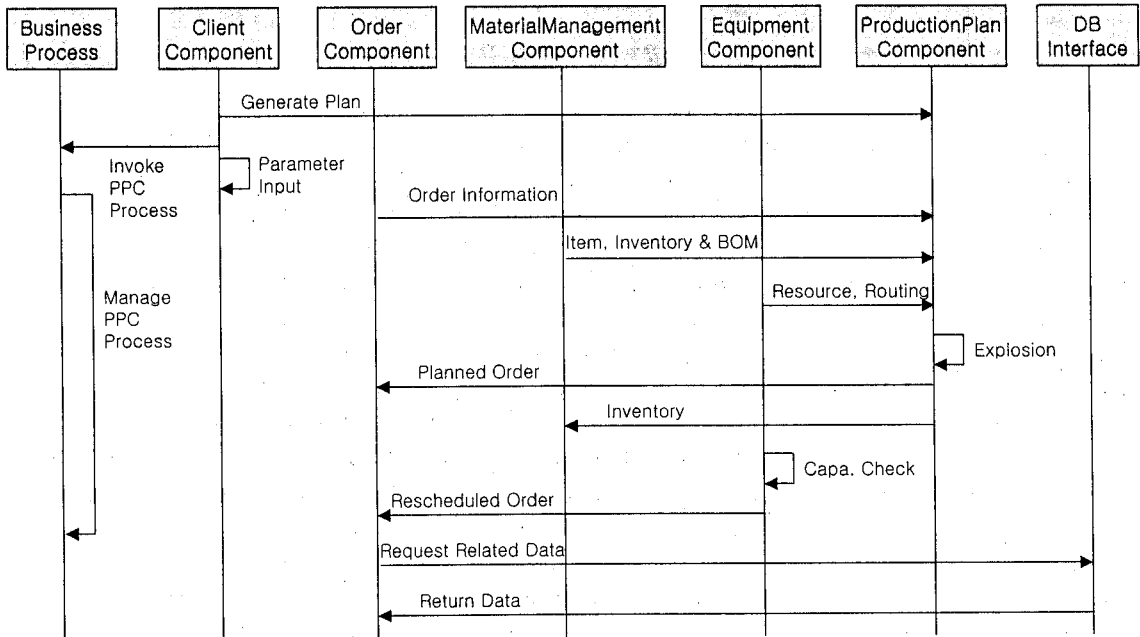


그림 10. 생산계획 생성 프로세스에 대한 Sequence Diagram.

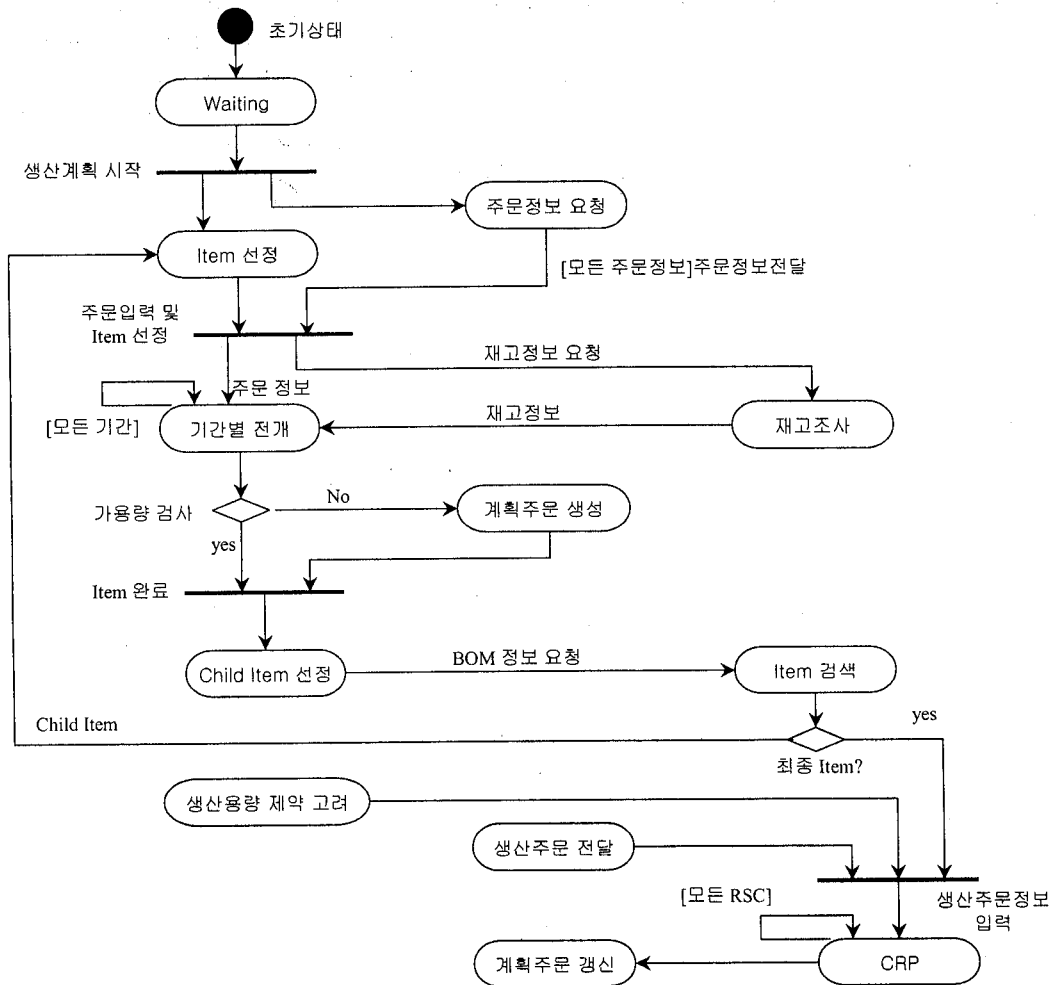


그림 11. 생산계획 운영 프로세스에 대한 Activity Diagram.

```

<!ELEMENT Message (MessageName, MessageNo,
    MsgCreateDate, MessageBody)>
<!ELEMENT MessageName (#PCDATA)>
<!ELEMENT MessageNo (#PCDATA)>
<!ELEMENT MsgCreateDate (#PCDATA)>
<!ELEMENT MessageBody (Order)>
<!ELEMENT Order (CustomerOrder | PurchaseOrderGroup
    | PlannedOrder)>
<!ELEMENT CustomerOrder (#PCDATA)>
<!ELEMENT PurchaseOrderGroup (#PCDATA)>
<!ELEMENT PlannedOrder (PurchaseOrder, MfgOrder+)>
<!ELEMENT PurchaseOrder (#PCDATA)>
<!ELEMENT MfgOrder ( MfgOrderID, MfgItem, OrderQty,
    DueDate, OrderSchedule*)>
<!ELEMENT MfgOrderID (#PCDATA)>
<!ELEMENT MfgItem (#PCDATA)>
<!ELEMENT DueDate (#PCDATA)>
<!ELEMENT OrderQty (#PCDATA)>
<!ELEMENT OrderSchedule (ScheduleID, StartTime,
    EndTime, ProductionQty, ResourceName)>
<!ELEMENT ScheduleID (#PCDATA)>
<!ELEMENT StartTime (#PCDATA)>
<!ELEMENT EndTime (#PCDATA)>
<!ELEMENT ProductionQty (#PCDATA)>
<!ELEMENT ResourceName (#PCDATA)>
    
```

그림 12. 생산 주문에 대한 XML DTD.

4.4 XML 메시지 생성

다음으로는 앞서 3.3에서 설계한 UML 정보모형의 XML 메시지 구조 변환 규칙을 이용하여 XML 메시지를 생성한다. 여기서는 4.3의 생산계획 프로세스 중에서 생성되는 생산계획을 대상으로 하는 생산주문(Mfg_Order)을 표현하기 위한 XML DTD의 예를 <그림 12>와 같이 제시한다. 메시지는 이름, 번호, 생성일 등을 표시하기 위한 Header 부분과 실제 요구되는 정보를 포함하는 Body 부분으로 구성된다.

4.5 프로토타입 시스템 구현

앞에서 제시한 방법을 이용하여 구매주문을 생성하는 과정에 대한 프로토타입 시스템을 구현해 보았다. 시스템 구현을 위하여 사용한 개발환경은 다음의 <표 3>과 같다.

표 3. 시스템 개발 환경

시스템	개발 환경
CASE 툴	Popkin System Architect 2001
서버	Java (JDK1.2.2)
클라이언트 (Viewer)	MS Visual C++ V6.0 MS Internet Explorer V5.0
ORB	Inprise Visibroker for C++ V3.3 Inprise Visibroker for Java V3.4
XML 파서	Sun JAXP V1.0.1 (서버용) MS XML SDK V2.5 (클라이언트용)
DBMS	Oracle 8i

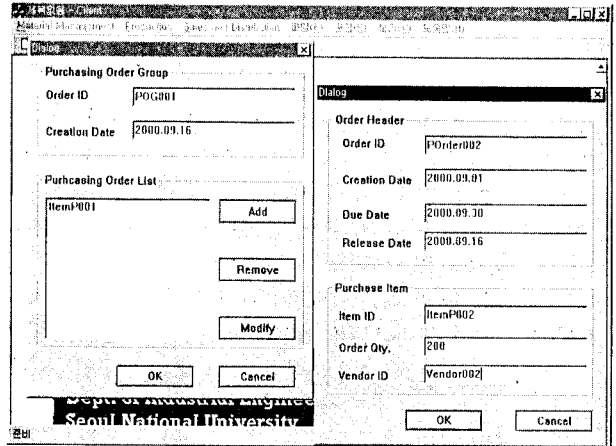


그림 13. 클라이언트의 정보입력 화면.

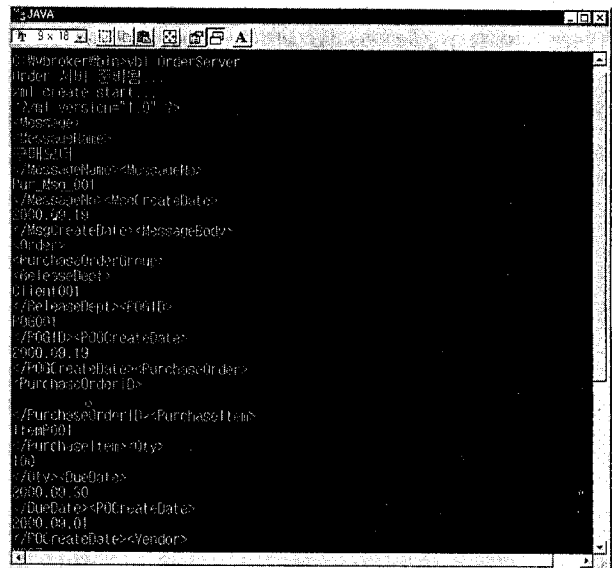


그림 14. 주문 업무객체 서버의 XML 문서 생성.

위와 같은 환경에서 개발한 프로토타입 가운데 클라이언트와 업무객체 서버, 결과 XML 메시지에 대한 예를 각각 <그림 13>, <그림 14>, <그림 15>에 나타내었다.

<그림 13>은 클라이언트에서 구매주문(Purchase_Order)을 생성하기 위하여 주문번호, 아이템, 수량, 공급업자, 일자 등의 정보를 입력하는 상황에 대한 화면이며, 이렇게 입력받은 정보가 CORBA ORB를 통해 주문업무객체 서버로 이동되고 서버에서 XML 문서로 생성되는 과정을 보여주는 것이 <그림 14>이다.

<그림 15>는 구매주문의 특정 아이템에 대하여 자재소요 계획을 통한 전개결과를 XML 메시지로 나타내고 있다. BOM 정보를 이용하여 생산계획(ProductionPlan) 업무객체에서는 MRP를 수행하고, 결과로 생성된 계획주문(Planned_Order)에 대한 정보를 이와 같은 XML 메시지를 이용하여 클라이언트에 전달하게 되는 것이다.


```

<?xml version="1.0" encoding="UTF-8" ?>
<Message>
  <MessageName>MRP Results</MessageName>
  <MessageNo>mrp001</MessageNo>
  <MsgCreateDate>2000.11.13</MsgCreateDate>
  <MessageBody>
    <Order>
      <PlannedOrder>
        <PurchaseOrder>
          <PurchaseOrderID>po4</PurchaseOrderID>
          <PurchaseItem>itemA</PurchaseItem>
          <OrderQty>15.000000</OrderQty>
          <Period>3</Period>
          <DueDate>4.000000</DueDate>
        </PurchaseOrder>
        <PurchaseOrder>
          <PurchaseOrderID>po5</PurchaseOrderID>
          <PurchaseItem>itemA</PurchaseItem>
          <OrderQty>5.000000</OrderQty>
          <Period>4</Period>
          <DueDate>5.000000</DueDate>
        </PurchaseOrder>
      </PlannedOrder>
      <MfgOrder>
        <MfgOrderID>mo2</MfgOrderID>
        <MfgItem>itemB</MfgItem>
        <OrderQty>10.000000</OrderQty>
        <Period>0</Period>
        <DueDate>2.000000</DueDate>
      </MfgOrder>
    </Order>
  </MessageBody>
</Message>
  
```

그림 15. MRP 전개결과 XML 메시지.

5. 결론 및 추후 연구 과제

본 연구에서는 분산객체와 XML을 이용하여 생산계획을 수립하기 위한 업무객체 컴포넌트를 설계하고 구현하기 위한 방법론과 예를 제시하였다. 구체적으로는 UML을 이용하여 분산업무객체를 설계하고, XML 메시지와 ORB를 분산된 업무객체 컴포넌트 사이의 정보교환의 수단으로 하고, CORBA를 이용하는 시스템을 구현하였다.

본 연구에서 수행하고 기여한 점들을 나열하면 다음과 같다.

- 공급사슬상의 주문, 자재관리, 생산설비, 생산계획 업무객체 컴포넌트를 정의하고, 업무객체와 업무프로세스를 UML을 이용하여 설계하였다.
- 컴포넌트의 구조로부터 XML 메시지의 구조(DTD)를 도출해내기 위하여 UML 정보모형을 XML 메시지로 변환하기 위한 규칙을 제시하였다.
- 제안한 방법론의 실용성을 검증하기 위해 생산관리 분야를 대상으로 프로토타입 시스템을 개발하였다.
- 개별 업무객체는 CORBA를 이용하여 구현하였다.
- UML 설계 결과로부터 XML 메시지 구조 변환규칙을 이용하여 XML DTD를 도출하고, 컴포넌트 사이의 정보교환을 위한 XML 메시지의 생성 및 처리과정을 구현하였다.

앞으로의 소프트웨어 개발은 컴포넌트 소프트웨어에 의한 응용특화형 특징을 가질 것으로 예상되며 본 연구는 그러한 소프트웨어 개발 방법의 하나의 가능성을 제시하고 있다. 그러나 본 연구에서는 업무객체의 최소한의 범위를 포함하고 있으나 앞으로 업무의 역할에 따라 더 확장될 수도 있을 것이다.

XML 메시지의 전송 및 관리 메커니즘의 문제, 데이터베이스와의 연계 문제, UML2XML 변환규칙의 확립과 이를 지원하기 위한 자동화된 도구 개발 등도 추후 중요한 연구과제가 될 것으로 사료된다.

참고문헌

Aguirre, O., Weston, R., Martin, F. and Ajuria, J. L. (1999), MCSARCH: An architecture for the development of manufacturing control system, *International Journal of Production Economics*, 62, 45-59.

Alfieri, A. and Brandimarte, P. (1997), Object-oriented Modeling and Simulation of Integrated Production / Distribution Systems, *Computer Integrated Manufacturing Systems*, 10, 261-266.

Bullinger, H. J., Faehrich, K. P. and Laubscher, H. P. (1997), Planning of Multi-site Production - An Object-oriented Model, *International Journal of Production Economics*, 51, 19-35.

Gillibrand, D. (2000), Essential Business Object Design, *Communications of ACM*, 43 (2).

Hardwick, M., Spooner, D. L., Rando, T. and Morris, K. C. (1996), Sharing Manufacturing Information in Virtual Enterprise, *Communications of ACM*, 32, 46-54.

Hasselbring, W. (2000), Information System Integration, *Communications of ACM*, 43(6).

Jeong, Cheongi (2000), Building strategy of the real B2B model using B2BERP, *e-ERP building strategy and solution fair 2000*.

Kip, C. (1999), Improving Interorganizational Data Interchange for Drug Development, *Computers in Biology and Medicine*, 29, 89-99.

Klement (1999), Component Framework Supporting Inter-company Cooperation.

Lee, W. B. and Lau, H. C. W. (1999), Multi-agent Modeling of Dispersed Manufacturing Networks, *Expert systems with Applications*, 16, 297-306.

Lewandowski, S. M. (1998), Frameworks for component-based client/server computing, *ACM Computing Survey*, 30(1), 3-27.

NCA: National Computerization Agency (1999), Trend analysis of the standardization of the next generation EDI (Development of the CALS/EC Standard model II), National Computerization Agency.

OMG (1995), Business Object Management Special Interest Group, OMG Business Application Architecture-white paper drafts 2.

Papazoglou, M. P. et al. (2000), Integrated Value Chain and Their Implications from A Business and Technology Standpoint, *Decision Support Systems*, 29, 323-342.

Riemer, K. (1998), A Process-Driven, Event-Based Business Object Model, *EDOC '98. Proceedings*.

Scheer, A. W. (1994), *Business Process Engineering*, Springer-Verlag.

Sheremetov, L. B. and Smirnov, A. V. (1999), Component Integration Framework for Manufacturing Systems Re-engineering: Agent and Object Approach, *Robotics and Autonomous Systems*, 27, 77-89.

Sims, O. (1997), Business Object Framework, *IEEE Colloquium Distributed Object-Technology and Applications*.

Sun Microsystems (2000), The Java 2 Enterprise Edition Developer's Guide.

Sutherland, J. (1999), "Big Workflow" for Enterprise Application, *OOPSLA'99*.

Vollmann, T. E., Berry, W. L. and Whybark, D. C. (1998), *Manufacturing Planning and Control Systems*, 4th edition, McGraw-Hill Book Co., Singapore.

Webber, D. (1998), OO-edi Compatibility with XML/EDI, *TMWG white paper draft*.

Wil M. P. et al. (2000), Process-Oriented Architecture for Electronic Commerce and Interorganizational Workflow, *Information Systems*, 24(8).

Yun, Seunghyeon (1999), A Study on the Development of Event-driven MRP Components Considering Capacity Constraints, *Master's thesis in the Department of the Industrial Engineering, Seoul National University.*

Zhou, Q. and Besant, C. B. (1999), Information Management in Production Planning for A Virtual Enterprise, *International Journal of Production Research*, 37, 207-218.



민 대기

서울대학교 산업공학과 학사
 서울대학교 산업공학과 석사
 현재: (주) LG CNS 시스템 연구원
 관심분야: ERP, SCM



박 찬 권

서울대학교 산업공학과 학사
 서울대학교 산업공학과 석사
 서울대학교 산업공학과 박사
 서울대학교 자동화시스템공동연구소
 현재: 영산대학교 정보경영학부 조교수
 관심분야: ERP/EC, 생산정보시스템



장 태 우

서울대학교 산업공학과 학사
 서울대학교 산업공학과 석사
 현재: 서울대학교 산업공학과 박사과정
 관심분야: ERP, SCM, 정보시스템



박 진 우

서울대학교 산업공학과 학사
 KAIST 산업공학과 석사
 미국 UC Berkeley 산업공학과 박사
 현재: 서울대학교 산업공학과 교수
 관심분야: 제조시스템공학, ERP/MRP, FMS/
 CIM, 시뮬레이션/일정계획