

SCM 환경에서 XML을 이용한 수주정보 시스템*

박현철^{*} · 이상용^{**}

요 약

현재 기업은 다양한 고객 요구사항을 충족시켜 가면서 급속하게 발전하는 기술을 습득하고 발전과 경쟁력확보를 위해, 협력 업체들과 연계관계를 유지하는 네트워크 형태의 공급사슬을 형성하고 있다. 공급사슬이 성공적으로 유지되기 위해서는 효율적인 정보교환이 필요한데, 기존의 EDI 시스템으로는 구축의 비용과 복잡성, 향후 발전방향에 불확실성 등의 문제점이 있기 때문에 SCM의 통합솔루션을 구축하기 위해서는 새로운 데이터 포맷이 필요하게 되었다.

본 논문에서는 이러한 문제점을 해결하기 위해 XML로 새로운 데이터 교환의 포맷을 정의하고, SCM 환경의 데이터 인터페이스 부분을 XML의 데이터베이스 스키마 정보를 이용하는 방법을 제시한다. 본 논문에서는 수주를 받고 다른 협력업체로 발주하는 시나리오는 다음과 같이 구성한다. 먼저 Web 환경의 클라이언트에서 XML DOM객체를 이용하여 ADODB 서버로부터 XML 수주정보 파일을 생성하고, ERP 서버에서는 수주정보의 XML 파일을 ERP 서버에 저장한다. 저장된 수주정보를 이용하여 납기확약을 위한 생산계획과 구

매계획을 세우고, 구매계획이 세워지면 다른 협력업체의 Web 서버로 발주정보를 보낸다.

1. 서 론

전통적으로 기업 경영자들은 경쟁우위 달성을 위하여 기업 내부의 역량확충에 치중해 왔지만 최근에 이르러 급속한 기술변혁과 다양한 고객 요구사항의 증대로 인하여, 하나의 개별기업이 제품생산과 관련된 모든 가치창출활동을 독자적으로 감당하기는 어렵게 되고 있다. 개별기업은 아무리 효율적으로 운영한다 할지라도 인적 자원, 정보, 기술 등을 충분히 소유하고 있지 못하여 경영환경의 빠른 변화속도나 지구촌 시장의 불확실성에 유연하게 대응할 수 없기 때문에 기업영역을 초월한 기업 상호간의 협력을 추구하는 전략적 경영 기법이 필요하게 되었다. 즉, 많은 기업들이 모든 필요 자원과 역량을 하나의 개별기업이 전부 소유하기보다는 오히려 핵심적인 역량에 집중하고 기타 자원 및 역량은 외부 협력 업체들과 연계관계를 유지하면서 전반적인 경쟁력을 확보하기 위한 노력을 기울이고 있다.

다시 말해, 개별 기업들이 서로 네트워크 형태의 공급사슬(Supply Chain)을 이루어 연계관계를 유지해가고 있는 것이다. 이러한 상황으로 볼 때 오늘날 기업경쟁력은 개별기업의 역량만으로 측

* 가온소프트(주)

** 경남대학교 컴퓨터공학부 교수

※본 연구는 2005학년도 경남대학교 학술논문게재연구비 지원으로 수행되었습니다.

정하기보다는 공급사슬관리의 유효성과 효율성으로서 측정함이 타당할 것이다. 왜냐하면 공급사슬을 이루고 있는 개별기업의 활동은 총체적 가치 창출 프로세스를 구성하는 하나의 특정부분으로서 역할을 수행하고 있으며 이들 활동들이 유기적으로 통합되어질 때 유효성이 실현되기 때문이다 [1]. 이러한 상황에서 SCM이 성공적으로 수행되기 위해서는 효율적인 정보교환을 위한 데이터포맷이 필요하다.

데이터의 전달과 공유의 포맷에 있어 전통적인 EDI(Electronic Data Interchange) 형식의 데이터는 구축에 소요되는 비용, 법적인 문제, 메시지의 불안정성, 시스템 구축의 복잡성, 거래처리 약정 체결과정에서 소요되는 비용과 시간, 비용대비 효과의 미흡, 보안성, 거래업체마다 고유한 시스템 구축, 향후 발전 방향의 불확실성 등의 문제점으로 인하여 기업 변환의 요구사항을 효과적으로 수용하지 못하고 있다. 따라서 이러한 EDI의 단점을 극복하기 위하여 본 논문에서는 SCM환경에서 데이터 포맷을 정의하기 위해 XML을 이용하는 방법을 제시한다.

XML은 데이터의 접근이 용이하고, 데이터 포맷의 통일성을 가지고 있어 다른 응용 프로그램에 용이하게 사용될 수 있는 장점을 가지고 있다. 특히 XML DOM 객체에서는 데이터베이스 스키마(Schema) 정보가 생성되기 때문에 본 논문에서는 SCM 환경의 정보공유와 전달의 데이터 포맷을 XML으로 처리하여, XML 데이터를 ERP 서버의 수주정보로 활용하여 납기확약을 위한 생산계획 및 구매계획 구현을 목적으로 한다.

2. 관련 연구

2.1 SCM

(1) SCM(Supply Chain Management)의 개념

기존의 기업들이 도입하여 운영하고 있는 JIT, Lean제조방식, MRP II, ERP 그리고 TQM 등은 기업내부에서 실현 가능한 생산성 및 비용 감축에 초점을 둔 방식이지만, SCM은 사업 동반자들의 전략적 제휴를 바탕으로 통합효율성을 추구하는 경영 방식이다. 개별기업은 아무리 효율적으로 운영한다 할지라도 인적 자원, 정보, 기술 등을 충분히 소유하고 있지 못하여 경영환경의 빠른 변화속도나 지구촌 시장의 불확실성에 유연하게 대응할 수 없기 때문에 기업영역을 초월한 기업 상호간의 협력을 추구하는 전략적 경영기법이 필요하게 되었다.

시장 주도자(market leader)가 되기 위해서는 개별기업으로 경쟁하는 것이 아니라 마치 단일 기업처럼 형성된 통합 공급체인에서 주도적 멤버로 경쟁하여야 함을 강조되고 있다. 단순히 정보기술 차원의 접근방법이 아닌 정보기술과 경영철학의 조화를 이루려는 경영 전략 차원의 개념이며 기업간 협력(partnerships), 전략적 제휴(strategic alliance), 프로세스 리엔지니어링(BPR), 활동기준 회계(activity cost management), 그리고 상거래(B2B, EC등) 등이 포함된 광의의 경영 개념이다.

정보기술은 기업간 전략적 제휴를 가능하게 하는 중요한 매개 요인이며, 전략적 제휴의 형성은 기업간 정보기술 채택에 관한 의사결정을 이해하는 바탕이 된다. 그러므로 SCM은 전자 카탈로그, 판매시점정보관리(Point of Sales; POS) 시스템, 전자 자금 이체(Electric Fund Transfer; EFT) 그리고 전자문서교환(Electric Data Interchange; EDI) 등의 첨단 기술을 사용하고 기업간 전략적 제휴(strategic alliance)에 의한 정보공유(information sharing)를 통하여 기업간 업무 기능들이 효율적으로 발휘될 수 있도록 지원하는 어플리케이션 집합이다.

경영전략 측면에서는 시간경쟁전략을 적용하여 납기단축, 비용절감, 재고감축을 이루기 위한 적시생산시스템(JIT), 연속보충시스템(Continuous Replenishment Program; CRP), 효율적 고객반응(Efficient Customer Response; ECR), 즉각 반응(Quick Response; QR), 공급자재고관리(Vendor Managed Inventory; VMI)등과 기업간 성과평가를 효율적 배분하기 위한 활동 기준 경영(Activity Based Management; ABM)이 포함된 경영기법이라고 말할 수 있다.

(2) SCM의 정의

SCM은 공급자에서 고객까지의 Supply Chain 상의 정보, 물자, 현금의 흐름에 대해 총체적 관계에서 체인간의 인터페이스를 통합하고 관리함으로써 효율성을 극대화하는 전략적 기법이다. 지금까지의 경영혁신 기법은 주로 기업 내의 전략, 구조, 기능 등의 개선 및 혁신에 중점을 맞추어 왔다. 그 결과 정보시스템 등 기업 내에 한정된 시스템 구축에 그 중점을 두어온 반면 SCM은 기업 내 뿐만 아니라 기업간 부문까지를 관심 영역으로 두고 있다.

SCM은 그림 1에서 보는 바와 같이 공급자, 자사, 고객을 모두 통합하여 하나의 파이프라인으로 연결하고자 하는 것이다. 불확실성 및 변동에 대비하기 위한 각 체인단계의 과잉 재고가 항상 요

구되고 제조업체의 내부 프로세스합리화로는 대응의 한계점에 봉착함에 따라 불확실성 및 변동의 원인을 능동적으로 제거하고자 하는 적극적 사고가 필요하게 되었고 이는 기업 내에서 기업간으로 시야의 확대를 요구하게 되었다[2].

2.2 SCM에서 XML의 활용

(1) SCM 환경을 위한 XML의 3-Tier 구조
XML을 이용한 SCM 환경을 구축하기 위하여, XML의 사용 가능한 부분을 3-tier 모델에 입각한 시스템 개요도를 살펴보면 그림 2와 같다.

먼저, 데이터베이스의 관점에서 볼 때, 데이터를 XML로 통일하여 저장하는 것은 매우 합리적인 일이라 할 수 있다. 왜냐하면, XML은 범용적인 데이터 교환 포맷(Universal Information Exchange Format)으로 작용하기 때문에, 하나의 저장된 문서를 이용하여 기존의 HTML 문서로는 물론이고, 휴대폰에서 사용하는 WML로도 변환이 쉽고, Voice Web Browser에서 사용하는 SpeechML로도 변환이 쉽다. 이러한 변환을 위해서 필요한 규칙들을 XSL로 저장하여, 필요에 따라 각각 적용하여 원하는 결과를 비즈니스 로직에게 넘겨주면 된다.

비즈니스 로직에서도 데이터베이스로부터 오는 XML 정보들은 XML로 직접 처리하여 필요한

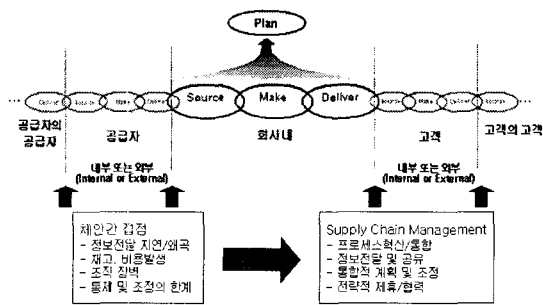


그림 1. 공급체인(Supply Chain) 프로세스

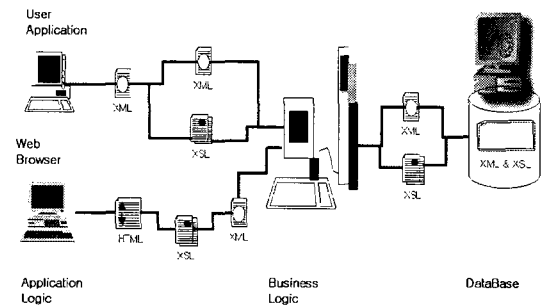


그림 2. XML의 3-Tier 구성도

부분만 추려서 프레젠테이션 층에 전송하여 이를 사용자에게 보여줄 수 있다. 또한 프레젠테이션 층에서는, 사용자에게서 받은 FORM 등의 정보를 처음부터 스트링이 아닌 XML로 전송 받아서, 이를 비즈니스 로직에서 완전한 형태의 XML로 취합한 뒤에 데이터베이스에 반영시킬 수 있다.

프레젠테이션 층에 있는 응용 프로그램에서는 자체적으로 XML 처리 능력을 부여해서 필요한 형태로 가공하거나, 웹 브라우저에서 XML 처리 능력이 있을 경우에는 비즈니스 로직에서 보낸 XML 문서를 바로 사용자에게 보여줄 수도 있다. 또한 사용자가 웹 페이지 등에서 반응한 결과를 종합적으로 취합해서 XML 형태로 다시 비즈니스 로직에게 보내서 데이터베이스에 그 결과가 반영되도록 할 수 있다.

이렇듯, 3-tier 모델로 본 XML 시스템은 웹 브라우저 같은 thin-client를 가정하여, XML 데이터를 보여주고 사용자의 입력을 받아 들여 전송하는 기능에 아주 적합한데, 그 이유는 이때에 보내고 받는 메시지 기능의 데이터도 XML이기 때문이다. 또한 영구적인 저장과 처리를 위한 데이터베이스의 XML에 있어서도 그 저장 내용을 별도의 양식으로 변환하는 과정 없이도 그대로 활용할 수 있는 장점이 있다. 혹시 변환이 필요하다고 하더라도 변환 규칙은 XSL 파일에 저장하여 놓으면, 여러 번 반복할 수도 있고 변환하는 난이도도 타사 제품의 데이터베이스로 포팅하는 노력보다 훨씬 쉽다. XML은 일괄처리, 상호대화, 실시간 문서 교환 기능이 통합된 형태로 제고되며, XML의 문서 구조 정보를 이용하여 더욱 정확하고 효율적인 문서 저장 및 검색이 가능하다. 또한, 문서를 구성하는 각각의 요소들을 객체단위로 처리하여 시스템의 확장성을 제공할 수 있다.

이처럼 XML은 3-tier 구조의 웹 환경에서 SCM에서 중요한 핵심이 되는 정보의 공유와 전

달 수행 능력을 향상시킬 수 있다. ADO에 의한 데이터베이스 접근으로 스키마를 가진 XML 파일을 자동으로 생성할 수 있다. 생성된 스키마는 인터넷에서 절대적으로 필요한 데이터에 대한 안정적인 구조를 제공하므로 스키마는 연결을 위한 키(Key) 링크로 회사들 간에 자동화된 e-Commerce를 제공하는 열쇠가 된다[6].

3. 시스템 구현

3.1 시스템 개요도

본 논문에서 제시하는 수주정보에 대한 처리 시스템은 그림 3과 같다. A 회사는 사내의 ERP 서버를 운용하여 생산계획과 구매계획을 세운다. 이 구매 정보는 Web서버의 발주정보로 이용되며, A회사의 협력업체들은 A 회사의 웹 서버에 등록하고 자사의 수주정보를 XML파일로 다운 받아 자사의 수주정보로 사용하게 된다. 협력회사 B는 XML로 작성된 수주정보를 응용프로그램을 이용하여 자사의 ERP 서버의 수주정보에 등록한다. ERP 서버는 자재를 확인하고, 필요한 자재가 확보된 경우 등록된 수주정보를 이용하여 생산계획을 세워, 수주에 대한 납기 약속을 발주회사에 메시지로 전달하고, 자재가 부족하면 구매계획을

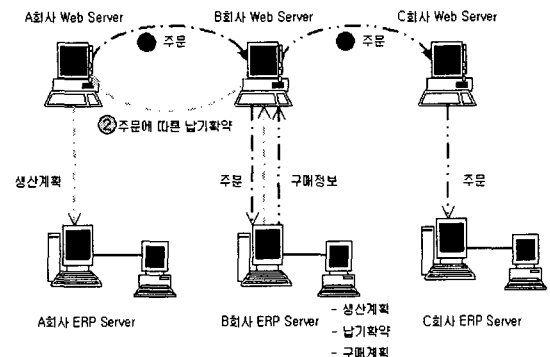


그림 3. 시스템 개요도

세운다. 구매계획에 대한 정보는 협력업체 B의 Web 서버에서 발주정보로 사용된다.

3.2 UML의 순서도(Sequence Diagram)

SCM 환경에서 시스템의 솔루션을 구축함에 있어, 그림 4와 같은 UML 표기법을 이용함으로써 각각의 모델의 문제를 쉽게 이해하고, 참여하는 모든 사람(고객, 해당 분야 전문가, 분석가, 설계자 등)과 프로젝트, 기업 모델링, 문서준비, 프로그램 디자인과 데이터베이스에 대해서 대화하는데 유용하게 사용된다.

UML의 순서도는 분산된 크고 작은 서로 다른 객체들 사이에서의 메시지 전달을 보여주기 위한 표기법으로, 객체 지향 컴포넌트 내에서 서로 다른 객체들 간의 메시지 전달과 통신은 기존의 객체지향 모델링 방법이다[8].

3.3 수주처리 시스템(Order Processing System)

(1) Web에서 XML 수주정보 생성

인터넷을 이용한 기존의 수주처리 시스템은 크

게 두 가지 형태로 나눌 수 있다. 하나는 협력업체의 웹 서버에 접속하여 수주를 등록을 하는 형태이다. 이런 형태의 수주처리는 ERP에서 MRP의 구매계획이 많으면 모든 협력업체를 찾아서 정보를 전달하는 번거로움이 발생하게 된다. 두 번째 형태는 EDI형식의 문서 전달이 있다. 발주업체는 발주에 대한 메시지만 협력업체에 전달하고, 협력업체는 발주하는 회사의 웹 서버에 접속하여 자사와 관련된 정보를 받아가는 형태이다. 이런 EDI 시스템의 단점은 정보의 포맷을 정의해야 하고, 또한 정보형태의 변화에 응용프로그램이나 타 프로그램에 능동적으로 대처할 수 없다는 단점이 있다.

본 논문에서는 이러한 두 형태의 수주정보 교환 방법의 단점을 없애고, 수주정보의 변경에서도 아주 능동적으로 대처할 수 있는 방법으로서 XML을 이용하는 수주처리 시스템을 구축하였다. 수주 처리에 대한 개략적인 구성은 그림 5와 같다.

그림 6은 업체의 등록정보에 대하여 생성된 XML파일의 소스이다. XML Recordset객체에서 고려해야 할 것은 XML-Data 스키마는 텍스트에 포함되어있고, 그 뒤에 데이터 섹션이 따라오는

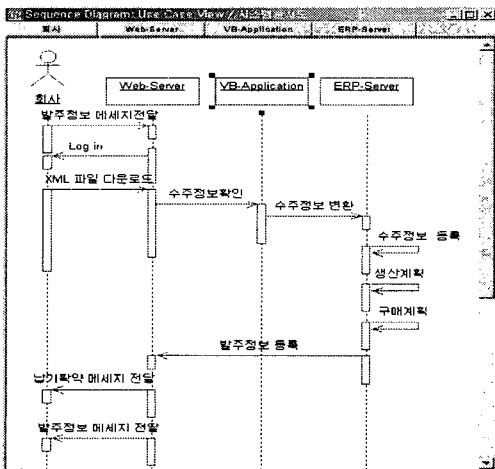


그림 4. UML의 순서도(Sequence Diagram)

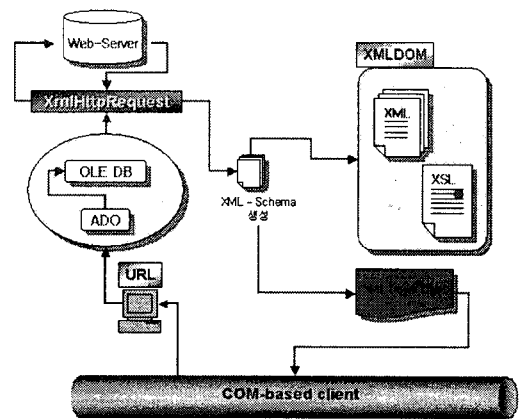


그림 5. XML 문서의 생성

```

<xml xmlns:s='uuid:BDC6E3F0-6DA3-11d1-A2A3-00AA00C14882'
xmlns:dt='uuid:C2F41010-65B3-11d1-A29F-00AA00C14882'
xmlns:rs='urn:schemas-microsoft-com:rowset'
xmlns:z='#RowsetSchema'>
<s:Schema id='RowsetSchema'>
<s:ElementType name='row' content='eltOnly' rs:CommandTimeout='30'>
<s:AttributeType name='c0' rs:name='거래업체' rs:number='1' rs:writeunknown='true'>
<s:datatype dt:type='string' dt:maxLength='25' rs:maybenull='false' /> </s:AttributeType>
.....
<s:extends type='rs:rowbase' />
</s:ElementType>
</s:Schema>
<rs:data>
<z:row c0='한국소니전자(주)' c1='2-166-682-01' c2='0Z-3779' c3='2001-11-10T00:00:00'
c4='50' c5='7453' c6='19' c7='ironspot@orgio.net' c8='2001-10-20T00:00:00' />
.....
</rs:data></xml>
    
```

그림 6. ADO에서 생성된 XML 소스

점이다. 포함된 스키마의 정의는 XML-Data 스펙에 따른다. ADO에서 만들어진 XML 데이터는 같은 스키마를 가지기 때문에 필드는 항상 같은 방식으로 정의된다. 그림 6에서는 네 개의 네임스페이스가 정의되어 있다. 각각의 네임스페이스는 XML 문서의 섹션을 포함하고 특정한 목적을 위해 사용한다. ADO와 XML간의 변환을 적절히 하기 위해 모든 것이 정확하게 정의해야 한다[9].

본 연구에서는 다음과 같은 Schema, DataType, RecordSet, Rowset의 네임스페이스를 정의하였다.

Schema (s:)

다음의 코드는 스키마의 정보를 위해 네임스페이스를 정의한 것이다.

```

xml xmlns:s='uuid:BDC6E3F0-6DA3-11d1-A2A3-00AA00C14882'
    
```

```

<s:Schema id='RowsetSchema'>
    
```

s:으로 시작하는 모든 태그는 데이터 또는 스키마의 정의와 관련된다. 이것은 속성으로 정의되는 필드 이름뿐만 아니라 요소의 정의를 포함하고, 데이터 타입 헤더를 포함한다. 또한 스키마는<..

id='RowsetSchema'..> 태그에 의해 식별된다.

Data Types (dt:)

다음은 데이터 형식의 네임스페이스를 정의한다.

```

xmlns:dt='uuid:C2F41010-65B3-11d1-A29F-00AA00C14882'
    
```

이 부분은 형식과 데이터 필드의 한계를 정의하는 정보를 포함한다.

Recordset과 Field Properties (rs:)

다음 코드는 Recordset 객체와 Field 프로퍼티 네임스페이스를 정의한다.

```

xmlns:rs='urn:schemas-microsoft-com:rowset'
    
```

Recordset 객체와 Field 프로퍼티는 필드의 순서와 Field의 데이터 프로퍼티, Null값을 가질 수 있는지 없는지에 대한 정보를 포함하는데 이 정보는 ADO Recordset 프로퍼티와 속성들과 밀접한 관련이 있다.

Current Rowset (z:)

Rowset 네임스페이스는 다음과 같이 정의된다. xmlns:z='#RowsetSchema'

이와 같은 정의는 스키마의 id(<s:Schema

```

<?xml version="1.0" encoding="euc-kr" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/TR/ WD-xsl">
<xsl:template match="/">
<HTML>
<BODY>
<TABLE CLASS="Table" ID="RecordTable" bordercolor="#ffffff" border="1" cellpadding="01"
cellspacing="0">
<THEAD><TR CLASS="TableHead">
    <TD CLASS="TableColumnHead">제품코드</TD>
    .....
</TR></THEAD>
<xsl:for-each select="xml/rs:data/z:row">
    <TR CLASS="TableRow">
    <TD><xsl:value-of select="@c1" /></TD>
    .....
    </TR>
</xsl:for-each>
</TABLE>
</div></BODY></HTML>
</xsl:template></xsl:stylesheet>

```

그림 7. XML에 대한 XSL 소스

id='RowsetSchema')를 참조한다. ADO가 이 XML을 적절히 다루기 위해, id는 RowsetSchema가 되어야 하고, #RowsetSchema를 지정해야 한다. 이것이 스키마가 데이터 값을 저장하도록 구현되는 부분이다.

본 논문에서는 데이터 바인딩 기술을 위하여 테이블 모양의 데이터 바인딩을 사용하였다. 그림 7은 XML에 필요한 XSL 파일을 테이블 데이터 바인딩으로 표현하는 소스이다. 각 레코드마다 <xsl:for-each..>를 사용하여 섹션의 내용을 반복시키기 위하여 TableRow 요소를 사용하였고, ADO가 컬럼을 Rowset의 속성으로 취급하기 때문에 각각의 필드 앞에 속성 id와 @문자를 사용하였다.

그림 8은 생성된 XML파일과 XSL을 이용하여 웹 브라우저에 보여주는 소스이다. Stream 객체의 장점을 사용하는 방식으로 XML을 Stream 객체에 읽어 들이고 Stream 객체에서 XMLDOM

문서를 읽어옴으로서 TransformNode 메소드의 장점을 살릴 수 있다. 이렇게 하면, XSL 템플릿을 XML 데이터에 각각 링크를 정의할 필요 없이 데이터에 적용할 수 있다. 이렇게 함으로서 XSL과 XML데이터를 완전히 분리할 수 있다[10].

실제로XMLDOM객체를 이용하여 클라이언트 쪽에서 구현된 화면은 그림 9와 같다. 등록된 정보에 따라 다른 정보들이 생성되기 때문에 업체간의 과열 경쟁이나 정보의 보안 등의 장점을 살릴 수 있다.

```

set xml=Server.CreateObject("Microsoft.XMLDOM")
xml.async=false
xml.load(Server.MapPath("XXX.xml"))

set xsl=Server.CreateObject("Microsoft.XMLDOM")
xsl.async=false
xsl.load(Server.MapPath("YYY.xsl"))

Response.Write xml.transformNode(xsl)

```

그림 8. XML과 XSL을 DOM 객체로 Load 소스

Welcome to order processing system with XML.

주문코드	발행일	발행일	납기일	주문일	주문수량	인기
1000000001	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000002	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000003	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000004	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000005	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000006	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000007	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000008	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000009	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000010	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000011	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000012	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000013	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000014	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000015	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000016	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000017	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000018	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000019	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000020	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000021	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000022	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000023	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000024	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000025	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000026	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000027	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000028	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000029	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000030	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000031	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000032	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000033	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000034	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000035	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000036	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000037	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000038	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000039	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000040	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000

다음은도

그림 9. Web에서 수주처리화면

(2) VB Application에서 XML파일을 ERP에 저장
본 연구에서는 수주 받은 XML파일을 다시 ERP 서버에 저장하기 위해 VB를 이용하여 프로그램을 작성하였다. XML파일은 DB Schema 정보를 모두 가지고 있기 때문에 필요한 프로그램에 확장성이 아주 용이하다. XMLDOM 객체는 C++, Visual C++, Delphi, java, Visual Basic 등 여러 응용프로그램에서 모두 지원이 가능하다 [11]. 수주 받은 XML 파일을 ERP 서버에 저장한 결과를 그림 10과 같이 구현하였다.

주문코드	발행일	발행일	납기일	주문일	주문수량	인기
1000000001	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000002	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000003	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000004	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000005	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000006	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000007	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000008	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000009	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000010	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000011	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000012	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000013	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000014	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000015	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000016	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000017	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000018	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000019	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000020	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000021	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000022	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000023	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000024	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000025	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000026	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000027	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000028	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000029	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000030	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000031	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000032	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000033	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000034	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000035	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000036	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000037	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000038	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000039	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000
1000000040	2006-03-01	2006-03-01	2006-03-01	2006-03-01	1000	1000

그림 10. XML의 ADODB 변환

객체	코드
DOM Object	<pre>Dim objDOM As MSXML.DOMDocument Dim Txtxml As String objDOM.async = False If (objDOM.Load(strFilename)) Then Txtxml = objDOM.xml LoadDocument = True Else LoadDocument = False End If</pre>
ADODB Recordset and ADO Stream Object	<pre>Dim Rs As New ADODB.Recordset Dim St As New ADODB.Stream St.Open St.WriteText Txtxml.Text St.Position = 0 Rs.Open St Set AdoDC1.Recordset = Rs</pre>

그림 11. XML을 ADODB로 저장하는 소스

그림 11은 다운 받은 XML 데이터를 DOM객체로 로드(Load)시키는 것과 다시 ADODB로 저장하는 소스를 보여주고 있다[12].

(3) ERP 서버에서 생산계획 및 구매계획
본 논문에서 시도하려는 ERP 서버에서 생산계획과 구매계획의 알고리즘은 그림 12와 같다. 먼저, XML에서 생성된 수주정보의 주문수량을 받는다. 기존의 주문잔량을 계산하고, 주문잔량을 제품재고와 원재료 재고를 비교하여 기존주문수량에 대한 예상납기일, 생산계획, 구매계획 등을 세운다. 다음으로는 주문수량과 주문잔량의 수를 합하여 제품재고와 원재료 재고를 비교하여 주문수량에 대한 예상납기일, 생산계획, 구매계획 등을 세운다.

ERP에 등록된 수주정보에 대하여 생산계획 및 구매계획 대한 실제 구현 화면은 그림 13과 같다. 수주입력에 대한 수주처리는 선입선출(FIFO)법으로 처리하였으며, 제품재고와 원재료재고는 안전재고를 고려하지 않고 적게 가져가는 방법으로

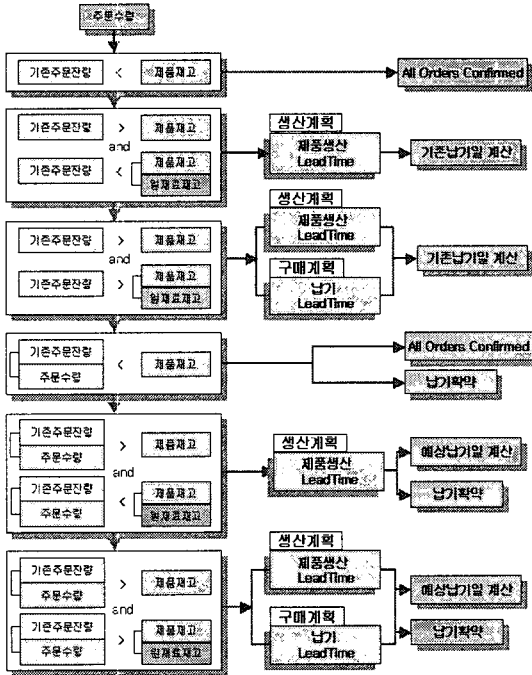


그림 12. 생산계획과 구매계획 알고리즘

4. 결 론

본 연구에서는 급변하는 기술발전과 다양한 고객 요구사항의 증대로, 하나의 개별 기업이 제품생산에 관련한 모든 가치창출활동을 독자적으로 감당하기 어려운 환경에서 외부 협력 업체들과의 연계관계를 유지하여 경쟁력 향상을 시도하는 경영의 패러다임(Paradigm)인 SCM에 관하여 연구하였으며, SCM에서 기업내 및 기업간 서로 상호간 협업관계를 위해 가장 중요시 되는 정보의 전달과 공유에 있어 데이터 포맷이 필요하게 되었다.

전통적인 EDI 데이터를 데이터포맷으로 정하려고 하였으나, 구축에 소요되는 비용, 법적인 문제, 메시지의 불안정성, 시스템 구축의 복잡성, 거래처리 약정 체결과정에서 소요되는 비용과 시간, 비용대비 효과의 미흡, 보안성, 거래업체마다 고유한 시스템 구축, 향후 발전 방향의 불확실성 등의 문제점으로 인하여 기업의 요구사항을 효과적으로 수용하지 못했다.

이러한 문제점을 해결하기 위해 본 논문에서는 W3C에서 제정한 웹 표준 XML을 이용하여 수주처리 시스템을 XML DOM 객체 스키마를 활용하여 데이터베이스에 저장하는 응용프로그램 구현과, SCM 환경에서 수주정보 시스템 활용으로 납기확약을 위한 생산계획과 구매계획 프로그램을 구현하였다. XML DOM 객체는 데이터의 구조적 접근이 용이하고 데이터 포맷의 통일성을 가지고 있기 때문에 OS가 달라도 다른 응용 프로그램의 개발에 아주 용이한 장점을 가지고 있다. 본 논문에서 XML DOM 객체와 ADODB 객체를 이용한 데이터 전달 방법을 제시하였다.

제품코드	납기일	주문수량	단가
2455-2554	2007-11-30	200	7
2455-2554	2007-11-30	30	4
2455-2554	2007-11-30	30	4
2455-2554	2007-11-30	30	4
2455-2554	2007-11-30	30	4
2455-2554	2007-11-30	30	4
2455-2554	2007-11-30	30	4
2455-2554	2007-11-30	30	4
2455-2554	2007-11-30	30	4

그림 13. 생산계획과 구매계획 구현화면

운영되도록 구현하였다. 등록된 수주정보는 예상 납기일을 확정하고, 이에 대한 응답을 수주업체에 전달하게 된다. 또한 생산계획과 동시에 구매계획을 세워, 자사의 Web서버에 등록하여 협력업체에게 발주에 대한 메시지를 전달하도록 구현하였다.

참 고 문 헌

[1] 유영임, "SCM 개념에 입각한 자재수급관리 프

로세스의 혁신에 관한 사례연구”, 석사학위논문, 충북대학교, 2000.

- [2] 이영희, <http://myerp.com.ne.kr/how>
- [3] 아더앤더슨코리아, <http://www.aak.co.kr/market/scm.htm>
- [4] XML Specification, <http://www.w3.org/TR/WD-xml.html>
- [5] 박병철, “컴포넌트 기반 시스템 개발”, 석사학위논문, 서강대학교, 1999.
- [6] 이종호 저, “XML과 전자상거래”, 정보문화사, 2001.
- [7] 홍동숙, “OpenGIS 사양을 지원하는 XML 데이터 제공자의 설계 및 구현”, 석사학위논문, 건국대학교, 2001.
- [8] 지영수 저, “Rational Rose 2000”, 홍릉과학출판사, 2000.
- [9] XML Articles, <http://www.topxml.com/xml/articles/sql2000/>
- [10] Richard Blair 외 12인 공저, 안성욱 외 3인 역, “PROFESSIONAL ASP XML”, 정보문화사, 2001.
- [11] XML Library, <http://www.msdn.microsoft.com/library/>
- [12] Mark Willson 외 1인 공저, 정희경 역, “VB와 ASP를 이용한 XML 프로그래밍”, 이한출판사, 2000.



박 현 철

- 2000년 경남대학교 산업공학과 졸업
- 2002년 경남대학교 메카트로닉스협동과정 졸업(석사)
- 2002년-현재 가온소프트(주) 재직
- 관심분야 : ERP, SCM, 데이터베이스



이 상 용

- 1982년 : 고려대 산업공학과 졸업
- 1984년 : 고려대 대학원 산업공학과 (공학석사)
- 1992년 : 포항공대 대학원 산업공학과 (공학박사)
- 1992년~현재 : 경남대학교 컴퓨터공학부 교수
- 관심분야 : ERP, SCM, 컴퓨터 비전, 패턴인식