「정보시스템연구」 제10권 제2호
한국정보시스템학회
2001년 12월, pp. 103-128.

항만물류 EDI 시스템의 최적 구현을 위한 비교 분석*

김 현 수	동아대학교 경영정보과학부	hskim@daunet.donga.ac.kr
박 남 규	동명정보대학교 유통경영학과	nkpark@tmic.tit.ac.kr
한 계 섭	동아대학교 경영정보과학부	kshan@daunet.donga.ac.kr
최 형 림	동아대학교 경영정보과학부	hrchoi@daunet.donga.ac.kr
조 재 형	동아대학교 경영정보과학부	b990008@daunet.donga.ac.kr

<목 차>

Ⅰ. 서론
Ⅱ. 항만물류 EDI 시스템의 현황 및 문제점
2.1 항만물류 EDI 시스템 현황
2.2 항만물류 EDI 시스템의 문제점
Ⅲ. 항만물류 EDI 시스템의 프로토타입 구현
3.1 어플리케이션 구조
3.2 항만물류 EDI 시스템의 유형에 따른 프로토타입 구현
Ⅳ. 항만물류 EDI 시스템의 비교 분석 및 최적 구현 방안
4.1 항만물류 EDI 시스템의 비교 분석
4.2 사용자관점에서 항만물류 EDI 시스템의 최적 구현방향
Ⅴ. 결론
참고문헌
Abstract

Ⅰ. 서 론

정보기술 패러다임의 주요 흐름을 살펴보면 오랜 기간동안 주도했던 메인 프레임 위주의 시스템이 90년대 중반에 들어서 클라이언트/서버(Client/Server) 시스템이 주도하는 환경으로 전환되었으며, 90년대 후반 들어서는 인터넷의 활용증대로 네트웍 컴퓨팅 등에 대한 관심이 증대하면서 웹(Web) 기반환경으로 지속적인 확장을 하고 있다. 웹 환경의 클라이언트/서버 시스템은 분산객체의 컴포넌트(component)기술과 접목되어 분산객체 지향 서비스로 한차례 더 발전되고 있다.

* 본 연구는 한국과학재단 지정 동아대학교 지능형통합항만관리연구센터의 지원에 의한 것입니다.

그러나 클라이언트/서버 시스템의 경우 2-tier, 3-tier, 그리고 4-tier 등의 변화 속에서 서비스 제공자들이 개발방법론상 혼동을 경험하였고 채 10년도 지나지 않은 상황에서 인터넷, 인트라넷, 엑스트라넷, 그리고 네트웍 컴퓨팅 등 인터넷 관련 용어가 범람함에 따라 혼란은 더 심화된 것으로 보여진다. 예를 들면, 인터넷 관련 기술을 어느 업무에 적용해야 하는지, 또한 자바(Java)로의 시스템구축을 어디에 적용해야 하는지 등 의사 결정 상에 많은 어려움을 가지고 있다. 항만물류 산업에 적용되고 있는 응용시스템을 살펴보면, 해운물류 업무만을 고려하였을 때 클라이언트/서버 시스템이 6개, 메인 프레임의 Host기반이 2개, 웹 환경이 이루어진 시스템이 2개 뿐인 것으로 나타나 항만물류 산업이 다른 산업에 비해 정보기술의 활용화가 미진한 것으로 나타났다(해양수산부,한국전산원, 1998).

이러한 가운데 항만물류 응용시스템으로서의 EDI 시스템은 대부분의 서비스와 접목되어 필수적인 요소이나 사용자의 측면에서는 전송시간의 지연, EDI 소프트웨어의 미비, 투자비용 과다 등 EDI 시스템의 도입 및 이용에 많은 어려움을 겪고 있는 실정이며 EDI 시장의 독점체제로 인해 전체 항만 EDI 사용이 매우 저조한 것으로 나타나고 있다(전자신문, 2001. 9. 26일자).

정부·기업의 측면에서는 EDI 시스템의 이러한 문제점을 개선하고자 하나 사용자의 다양한 요구사항, 국내 항만업계의 환경, 현재 보유자원, 개발비용, 구축 시 고려사항, 구축 후 장단점 등 많은 고려사항 하에서 서비스의 개발·시행에 어려움을 겪고 있다. 또는 IT기술의 급격한 변화속에서 무조건적인 기술의 도입은 위험요소의 증대와 더 나아가 경쟁력 상실로 인한 EDI 시스템 운영회사의 생존문제로 악화될 수도 있다. EDI 시스템구축의 어려운 결정속에서 도입·적용 후에도 유지관리와 데이터 통합이라는 필수적 고려사항도 존재하고 있다. 클라이언트/서버 시스템이 보편화되고 웹 환경이 접목되면서 EDI 서비스도 자연스럽게 이러한 기술을 받아들여 발전되고는 있으나, 지금까지의 국내외 연구에서 정보기술을 직접 상용화하는 과정에서 종합적이고 다양한 프로토타입에 근거한 EDI 시스템의 비교에 대한 연구가 미미한 실정이다. 그러므로 본 연구에서는 항만물류 산업의 정보화 문제점을 개선하기 위해 다양한 개발방법론을 제시함과 아울러 이를 토대로 EDI 시스템의 다양한 프로토타입을 제시하고 이를 상호 비교하고자 한다. 본 연구에서 제시 할 EDI 시스템들은 크게 스크립트(script)를 이용한 웹 EDI 서비스, XML 기반의 XML/EDI 서비스, 3-tier 기반의 분산객체 지향서비스이다. 이렇게 구축되어진 EDI 시스템을 현재까지 사용되고 있는 기존의 항만물류 전용 VAN/EDI 서비스와 비교·분석해 보고, 각각의 장단점을 살펴보도록 한다.

본 연구에서 제시한 EDI 시스템 비교 분석은 정부·기업이 EDI 시스템 구축을 위한 시작단계에서 어떤 시스템을, 어떻게 구축할 수 있는지에 대한 의사결정의 혼선을 줄이고, 항만물류에 있어 사용자관점에서 최적의 EDI시스템을 구축하기 위한 기준을 찾을 수 있게 할 것이다.

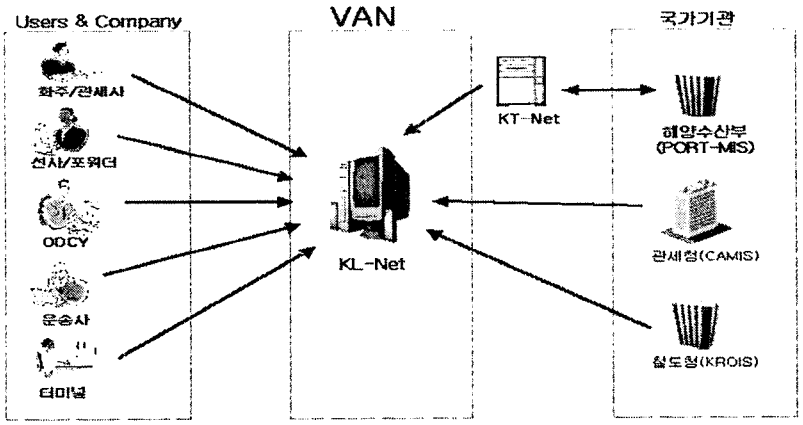
Ⅱ. 항만물류 EDI 시스템의 현황 및 문제점

2.1 항만물류 EDI 시스템 현황

우리나라에서는 항만관련 물류업무처리의 효율화를 위해 1996년부터 EDI를 도입하여 현재 수출입 분야의 각 기관과 관련업계에서 많이 활용하고 있으며, 전용터미널의 정보시스템과 각 해운기업의 운영정보시스템들이 널리 사용되고 있다. 우리나라의 항만물류 관련 EDI는 (주)한국물류정보통신(KL-Net)을 중심으로 해양수산부의 항만운영정보시스템(PORT-MIS), 관세청의 통관시스템(CAMIS), 철도청의 철도화물운송정보시스템(KROIS) 등의 정부기관 시스템과 관련업계들인 해운업계, 선박 대리점 업계, 항만운송 업계, 국제 복합 운송업계들이 연결되어있고 한국 컨테이너부두관리공단, 부산컨테이너부두운송공사가 실 사용자로 참여하고 있다. VAN업자로는 (주)데이콤과 (주)한국무역정보통신(KT-Net)이 참여하고있다.

해양수산부의 PORT-MIS는 입항예보서, 화물반출입현황 등 22개 서식을 EDI로 처리하고 있으며, 철도청의 KROIS는 화물운송장, 화물운송통지서, 화차배분 등을 처리하고 있다. 관세청의 CAMIS에서는 하선신고 및 적하목록 등의 업무를 처리하고 있다. 아래의 <그림 1>은 KL-Net을 중심으로 한 항만물류 관련 기관들의 개괄적인 망 구성도를 보여주고 있다.

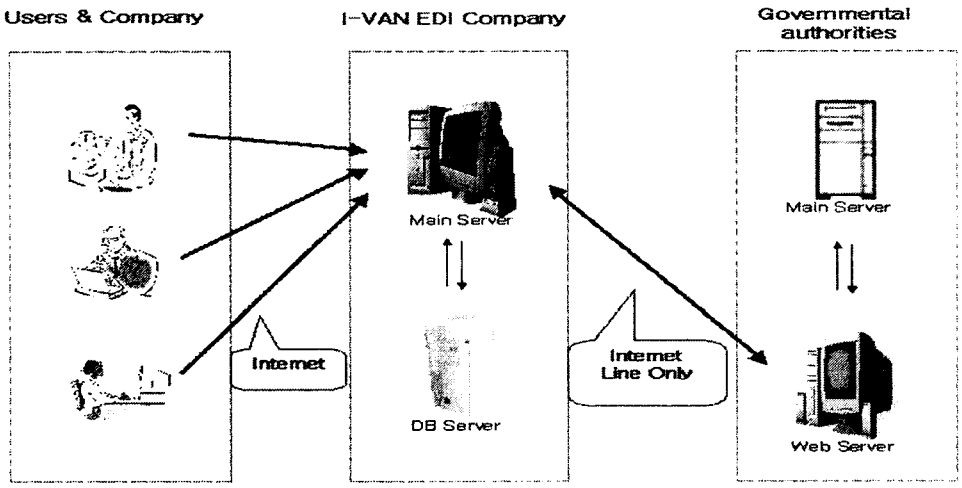
<그림 1>에서와 보는 바와 같이 현행 EDI는 KL-Net을 통하여 모든 업무가 이루어지고 있다. KL-Net의 중요성이 바로 여기에 있는데 수출입과 관련된 모든 정보가 KL-Net을 통해 전달된다는 것이다.



<그림 1> KL-Net EDI 망

우리 나라의 항만물류산업의 EDI 시장은 EDI 전담사업자에 의해 서비스가 제공되고 있으며, 부가가치통신망(VAN) 을 경유하여 데이터가 전송되고 있어 폐쇄시장이라 불리운다. EDI 사용자들은 전자상거래의 발전으로 인해 VAN을 매개체르 하는 전통적 EDI 방식에서 벗어나 새로운 EDI 방식인 인터넷을 매개로 하면서 자사와의 프로세스의 통합이 가능한 EDI 방식을 요구하고 있다. 사용자의 불만사항을 조사한 결과를 보면 전송시간의 지연, 클라이언트 소프트웨어 기능미비, 문서수신확인의 지연, 표준문서에 대한 이해 부족, 통신료 및 유지보수비 과중 등이 불만사항으로 조사된 바 있다(박남규, 최형림, 1998a).

사용자들의 요구사항에 대한 해결책으로 지금까지 제시된 방식은 월드와이드웹과 파일전송프로토콜(FTP)을 활용한 인터넷 EDI 방식이었다. 이를 인터넷 기반 하이브리드(hybrid) 방식이라 불렀다. 그런데 왜 이와 같은 방식이 지금까지 실용화 과정에서 산업계에서 수용되지 않는 것인가. EDI는 쌍방간에 거래방식을 개선하는 것이다. 기업체의 입장에서는 상대방이 필요한 것이다. 거래 상대방이 인터넷 기반 하이브리드 방식을 수용할 수 없는 상황이면 인터넷 기술이 개발되었다 하여도 기술이 동작할 수 있는 상황이 될 때까지 기다려야 한다. 다시 말하면 인터넷 부가가치 서비스(I-VAN) 체제를 구축하여도 FTP 방식으로 전송되는 문서를 처리할 수 있는 장치가 거래상대방에 마련되어 있지 않으면 전송할 수가 없다.



<그림 2> I-VAN 기반 EDI서비스 개요도

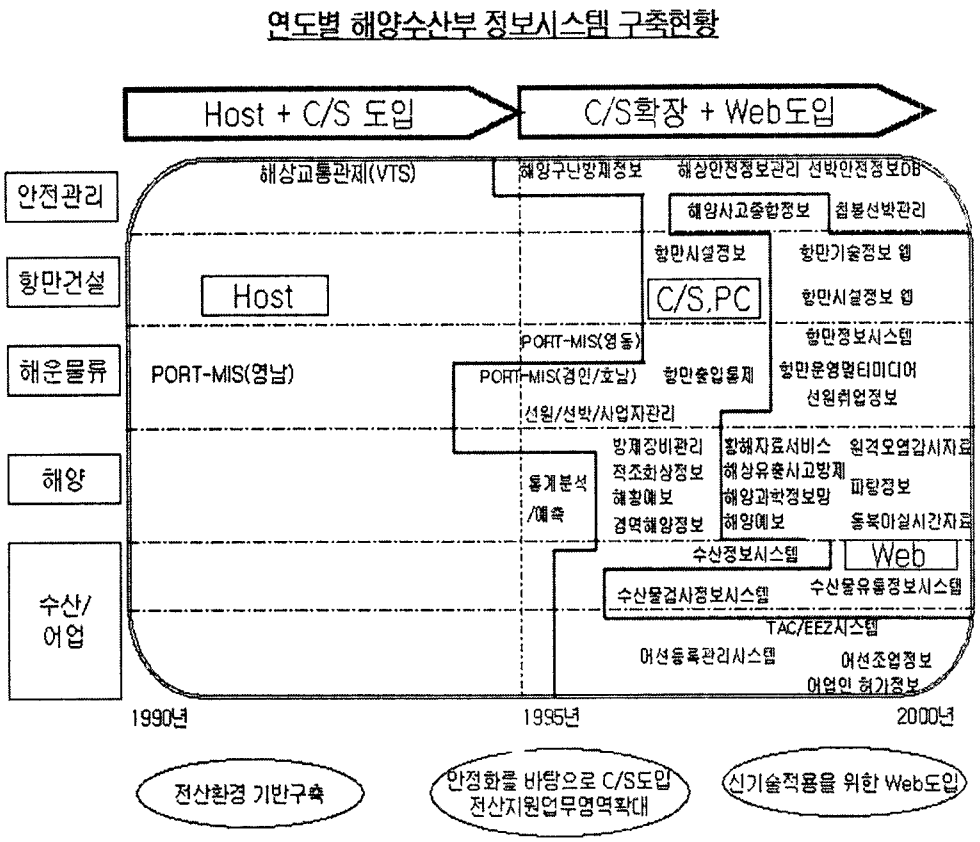
따라서 I-VAN 시스템 개발과 동시에 통관시스템과 항만정보시스템의 FTP 방식에

의한 메시지 수신 시스템의 개발프로젝트가 병행해서 진행되어야 한다. <그림 2>는 인터넷 EDI 사용자, 인터넷 VAN 업체, 해양수산부 등 EDI수신자로 구성된 인터넷 EDI 서비스의 개요도이다. 여기서 인터넷 EDI 방식이 제대로 동작되기 위해서는 정부기관인 해양수산부나 관세청에 웹서버 시스템의 설치와 웹기반 번역기(EDI Translator)가 사전에 설치되어 있어야 한다.

그리고 FTP에 의한 메시지를 수신할 수 있는 체계가 마련되었다 하여도 보안 문제, 오류 체크, 시스템 유지보수 등 제 문제를 감안하여 사용자가 정부기관의 정보시스템에 직접 접속하는 방식보다는 I-VAN 중개자를 거쳐 접속할 수 있는 체제가 구축되어야 한다. 이러한 I-VAN 사업자는 정부기관이 자격을 검증하여 기술적·재정적으로 일정 기준을 만족하면 사업자로 지정될 수 있어야 항만물류 EDI 서비스시장이 경쟁시장 체제로의 전환이 될 것이다. 아직 이에 대한 명확한 정책방향이 설정되어 있지 않아 신기술을 개발하려는 사업자가 등장하기 어려운 상황이 되어 있다. 위의 논의한 몇 가지 사항이 충족된다 하여도 현행 UN/EDIFACT 메시지 교환 방식이 지니고 있는 복잡성 및 프로세스와의 통합성의 문제점은 그대로 남아있게 된다.

2.2 항만물류 EDI 시스템의 문제점

먼저 해양수산부의 해양개발기본계획(Ocean Korea21)에서 선진정보화 체계 확립을 위해 정보기술의 현황에 대한 분석자료를 제시하고 앞으로의 개선·추진방향을 제시하고 있다. 여기서 나타난 항만물류 분야의 응용시스템에 대한 분석결과를 살펴보도록 한다. 현재 EDI 어플리케이션에 대한 시스템 분석에서는 시스템간 연계 미흡, H/W·S/W 표준화 미흡, 기 구축 시스템의 유지보수 미흡, 시스템 응답속도 신속성 미흡, 사용자 위주의 인터페이스(interface) 미흡, 다양한 정보제공 미흡, 데이터 정확성 미흡, 유지보수 비용 과다, 잦은 에러 발생 순으로 주요현안의 문제점을 뽑고 있다(해양수산부,한국전산원, 1998).



<그림 3> 연도별 해양수산부 정보시스템 구축현황

위 <그림 3>에서 살펴볼 수 있는 것처럼, 항만물류 분야 정보화에 있어서 개발환경이 표준화되어 있지 않아 응용시스템간 유기적이고 체계적인 유지보수에 많은 애로사항을 겪고 있다. 또한 아직 DOS모드의 서비스로 사용자 인터페이스에 불편한 점이 지적되는 서비스가 존재하고, 하나의 정보를 입력 또는 조회하기 위해서 여러번의 단계를 거침으로써 시간 낭비 및 사용자 불편을 초래하고 있다. 항만물류의 응용시스템 구축현황으로 아직 인터넷을 통한 서비스가 2개뿐인 것으로 조사되었다(해양수산부,한국전산원, 1998). 그러므로 정보화 추진전략의 개선·발전방향으로 종합적인 정보연계 체계 구축, 웹 기반의 시스템 구축, 사용자 중심의 시스템 구축, 통합시스템 구축, 객체지향 개발 방법론 도입이 요구된다.

앞서 제시된 시스템의 한계를 극복하고 항만물류 산업에 적용, 시행할 수 있는 EDI

시스템 프로토타입을 제안하기 위해 그동안 인터넷 VAN업체에서 사용할 수 있는 개방형 EDI시스템의 개발이 진행되어져 왔다(박남규 외, 1999; 박남규, 김현수 외, 2000; 박남규 외, 2000b; 최형림, 박남규 외, 2001). <표 1>은 지금까지 개발되어온 EDI 송수신 문서들로써, 크게 해양수산부 신고 문서들과 관세청 신고문서들로 분류될 수 있다.

<표 1> 본 연구를 위해 개발된 EDI시스템이 포함하는 EDI문서와 신고기관

신고기관		개발 문서	개발환경
해양수산부		입항신고서, 선원명부, 승객명부, 선용품목록, 강제도선면제신청서, 내항선입출항보고서, 예선사용신청서, 도선지정신청요청서, 무전검역신청서, 항해일지, 적하목록	NT/MS SQL 7.0 IIS/ASP
관세청	수출	수출신고서(통관)	H/WServer:Compaq 8000 Windows2000/ MS SQL7.0 XML/Java/IIS
	수입	통관 : 수입신고서 화물 : 입항보고서, 선원명부, 승객명부, 선용품신고서, 승무원휴대품 목록, 적하목록, 하선신고서, 이적허가신청서	H/WServer:Compaq 8000 Windowss2000/ MS SQL7.0 Delphi/ASP/DCOM/ ActiveX

지금까지 개발 프로젝트에 의해 진행되어온 프로그램은 클라이언트/서버구조에서 웹 환경과 접목되어 ASP를 이용한 웹서비스나 XML/EDI의 서비스로 개방형 EDI 시스템을 구축하는데 주력하였다. 항만물류의 EDI 서비스는 신고의 절차가 까다롭고, 업무를 이해하는데 오랜 시간과 노력이 필요했으며, 신고사항에 조금의 오차도 허용하지 않는 등 개발과정에 많은 어려움이 있었다. 또한 이렇게 구축된 시스템이 상용화의 과정에서 시행착오를 겪거나, 기대이하의 효과를 내기도 하였다. 본 연구는 이러한 항만물류에 있어서 EDI시스템의 개발 경험을 통해 사용자 입장에서의 최적의 시스템 구조를 위해 각 접근법의 장단점을 비교·분석하고자 한다.

Ⅲ. 항만물류 EDI 시스템의 프로토타입 구현

3.1 어플리케이션 구조

앞에서 살펴 본 항만물류 EDI 서비스 환경은 폐쇄형 시장의 한계를 극복하고, 개방형을 지향하기 위해 I-VAN업체들로 구성된 중계자를 이용하여 서비스가 이루어지도록 제시하고 있다. 현재 구축되었거나, 이러한 I-VAN 구도에서 개발 진행 중인 EDI 서비스들은 시스템 측면에서 클라이언트/서버 시스템 하에서 개발되었다. 먼저 항만물류 EDI 서비스의 어플리케이션의 적합한 구조를 재정립하기 위해 클라이언트/서버 시스템의 정의 및 유형을 정리하였다.

3.1.1 클라이언트/서버 시스템 유형분석

클라이언트/서버는 일반적으로 토폴로지(topology) 측면에서는 n-tier까지 가능하겠지만 보편적으로 채택되는 2-tier, 3-tier를 비교해 보았다. 프리젠테이션, 비즈니스 로직, DBMS(데이터관리층)를 어떻게 클라이언트/서버 시스템에 분산시키느냐에 따라서 다양한 구조가 발생하게 된다. 항만물류 EDI 서비스의 시스템 모델을 재정립하기 위해 기존 연구로 일반적 분산모델, 가트너그룹의 분산모델, Winsberg의 분산모델을 고찰하였다(임강진 외, 2000). 기존의 클라이언트/서버 유형에 대한 이론들과 항만물류 EDI서비스 계층구조는 제시되어진 모델간 명칭에서 조금씩 차이가 날 뿐, 계층의 기능면에 있어서는 유사한 점이 많다. 이러한 특징들을 하나로 묶어 나타내면 <표 2>와 같다.

클라이언트/서버 시스템 모델을 항만물류 EDI 시스템의 분류에 맞추어 재 정의 해보면, 2-tier fat client, 2-tier plump client, 3-tier thin client로 구성할 수 있다. 이러한 분류는 2-tier fat client와 2-tier plump client를 일반적 분산모델의 2-tier인 fat client 모델과 Gartner그룹의 2-tier 모델에서 참조하였으며, 3-tier thin client는 일반적 분산모델과 Winsberg의 응용로직을 참조하였다.

먼저 2-tier fat client는 모든 어플리케이션 프로그램 논리가 클라이언트에 있고 서버에서는 DBMS가 운영되는 형태이다. 다음으로 2-tier plump client는 일부 어플리케이션 논리가 클라이언트에 있고, 데이터 액세스 논리를 포함한 나머지 논리는 서버에 있는 형태이다. 마지막으로 3-tier thin client는 일부 어플리케이션 논리가 DBMS와는 분리된 서버에서 운영되는 형태로 분류할 수 있다. 이러한 분류의 관점은 클라이언트가 어떤 비중을 가지고 있느냐에 관한 기준이다.

<표 2> 클라이언트/서버 유형의 기존 모델 비교

	일반적모델	가트너그룹모델	Winsberg 모델	항만물류 EDI 시스템
계층유형	물리적 2-tier과 논리적·물리적 3-tier 모델로 구성	-분산표현 -원격표현 -분산로직 -원격데이터관리 -분산데이터관리	2-tier은 -원격표현 -분산로직 -원격데이타접근 3-tier 모델은 응용로직을 포함	물리적 2-tier과 논리적·물리적 3-tier 모델로 구성 -2-tier fat client -2-tier plump client -3-tier thin client
기능영역	-표현층 -응용층 -데이터관리층의 위치에 따라 모델 유형 구성	-Presentation -Application -Database중 Application과 Data의 위치에 따라 물리적 2-tier 모델 유형을 구성	-표현층 : 표현관리자, 표현로직 -응용층 : 응용로직 -데이터관리층 : 데이터로직, 데이터관리자	-Presentation -Application : 응용로직, 데이터액세스로직 -DataBase
특징	-2-tier, 3-tier 구조를 논리적구조와 물리적구조로 세분 -3-tier구조를 이상적 모델로 다루었으며, 응용층을 비즈니스 로직과 데이터처리 로직으로 두고 구성	-물리적인 2-tier 모델만을 다룸 -분산처리와 분산데이터개념의 혼재 -부서단위 시스템이나 워크그룹단위의 시스템에 적합	-표현층과 데이터관리층의 위치에 따라 2-tier 모델을 구성 -3-tier구조의 응용로직에 집중 -대규모의 전사시스템을 구축하는데 적합한 분산기술구조	-client의 역할비중에 따라 분류 -client를 3단계의 모델로 구성

3.2 항만물류 EDI 시스템의 유형에 따른 프로토타입 구현

항만물류 EDI 시스템의 비교를 위해 유형별 항만물류 EDI 시스템의 프로토타입을 개발하였고 서비스 대상이 되는 EDI문서를 이용하였다. 해양수산부 신고용인 입항신고서를 비롯한 10여개의 문서들과 관세청 신고용인 수출신고서 및 수입신고서 그리고 화물신고를 위한 입항보고서, 적하목록 등을 대상으로 하였다. 그 외 항만물류 EDI 시스템으로서 현재 상용화되고 있는 EDI형 해상수입화물관리 시스템인 Seacargo와 분산형 네트워 프로그램인 KCS 2002를 포함시켰다.

앞서 이러한 EDI 시스템들이 클라이언트/서버 시스템 하에서 개발되었으며 유형으로는 2-tier fat client, 2-tier plump client, 3-tier thin client로 재정립할 수 있음을 제안하였다. 다음으로 다양한 개발환경과 개발툴 및 기술들을 이용한 서비스들을 다음과 같이 분류하여 보았다.

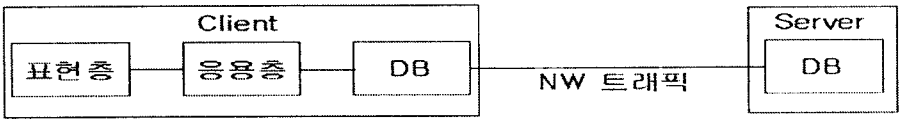
① 클라이언트 어플리케이션 EDI 시스템
② 스크립트를 이용한 웹 EDI 시스템
③ XML 기반의 XML/EDI 시스템

④ 3-tier의 분산객체 지향 EDI 시스템이다.

이렇게 4가지 유형으로 분류된 기준은 기술적·시스템적인 측면에서 클라이언트/서버 시스템의 유형과 개발환경 및 개발기술을 기준으로 지금까지 상용화되었거나 본 연구를 위해 개발된 항만물류 EDI 시스템들을 재분류하는 과정을 통해서 마련되었다. 첫 번째 유형인 클라이언트 어플리케이션 EDI 시스템은 현재까지 업계에서 사용 중인 대표적인 EDI 전용 프로그램이다. 웹 환경이 지원되지 않고, 유지보수의 어려움을 가지고 있다. 앞서 제시한 사용자 중심의 시스템과 웹 기반의 시스템 구축 및 통합시스템 구축을 이루기 위해 제시된 시스템이 두 번째부터 네 번째 유형으로써 본 연구를 위해 개발된 시스템들이다. 모두 웹 환경을 지원하며 개방형 EDI 시스템 구축, XML을 통한 해운 e-Marketplace 통합시스템 구축, 네트웍의 성능향상과 유지보수의 자원을 최소화 하고자 개발된 시스템들이다. 이제, 4가지 유형의 시스템에 대해 상세히 살펴보기로 하자.

3.2.1 클라이언트 어플리케이션 EDI 시스템

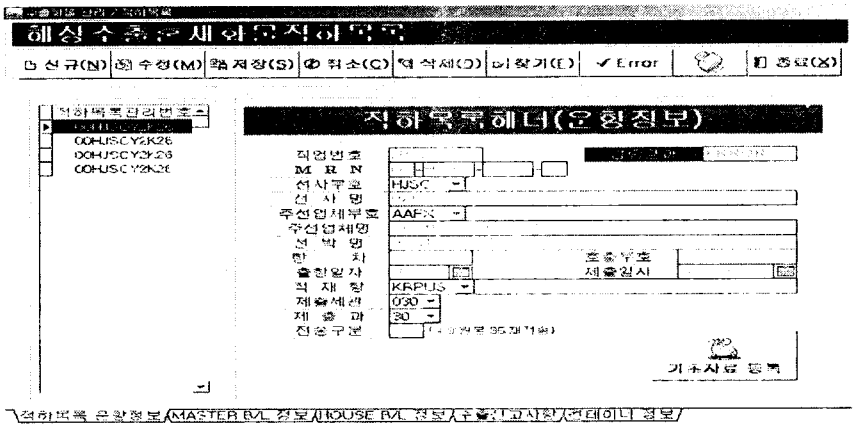
첫째, 클라이언트 어플리케이션 기반의 시스템이다. 1997년부터 상용화된 이 시스템은 전통적 EDI시스템으로 EDI 전용 클라이언트 소프트웨어이다. 현재, 수출입과 관련된 거의 모든 EDI 신고문서들에 사용되고 있으며, 클라이언트 어플리케이션으로 명명한 것은 전형적인 2-tier fat client로서 사용자들은 본 프로그램을 사용하기 위해 하나의 단일 프로그램을 배포 받거나 기술자가 직접 client 컴퓨터에 설치해준 후 사용되기 때문이다. 거의 모든 업무로직 및 어플리케이션 로직이 client 프로그램에 내재되어 있으며 서버에서는 DBMS만이 연동되는 형태이다.



<그림 4> Client Application 서비스의 구조

<그림 4>에서 클라이언트에는 표현층, 응용층, 데이터 접근을 위한 일부 구조가 들어가 있고, 서버에는 데이터관리층(DBMS)이 포함된다. 전체적인 업무의 처리속도는 빠른 편이나, 데이터베이스 프로그램 개발자나 시스템관리자는 DBMS 클라이언트 소프트웨어를 각 클라이언트 컴퓨터마다 설치하고 설정을 올바로 잡아서 데이터베이스에 접근할 수 있게 만들어야 하며, BDE(Business Database Engine)와 데이터베이스 서버에 대한 SQL Link를 함께 설치하여 데이터베이스 서버에 대한 일리아스(alias)를 만들고 환경 설정을 올바르게 해줘야하는 복잡한 작업이 반드시 필요하다. 클라이언트와 서버와의 통신은 전통적인 Networked SQL을 사용한다. 웹 환경은 이루어지지

않으며, 개발 툴 및 기술은 델파이(Delphi)를 사용하였다. 본 유형에 해당되는 방식을 이용한 서비스가 EDI형 해상수입화물관리 시스템인 Seacargo이다. <그림 5>에서는 Seacargo에서 해상수출혼재화물적하목록 서비스를 보여주고 있다.

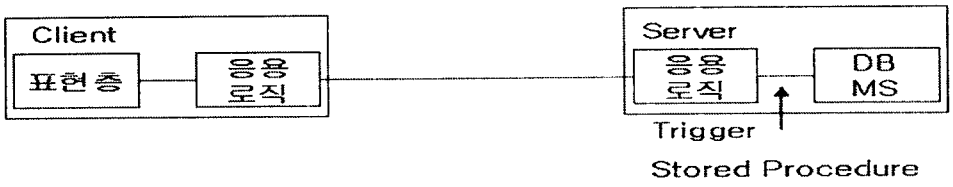


<그림 5> 클라이언트 어플리케이션 유형의 서비스

3.2.2 스크립트를 이용한 웹 EDI 시스템

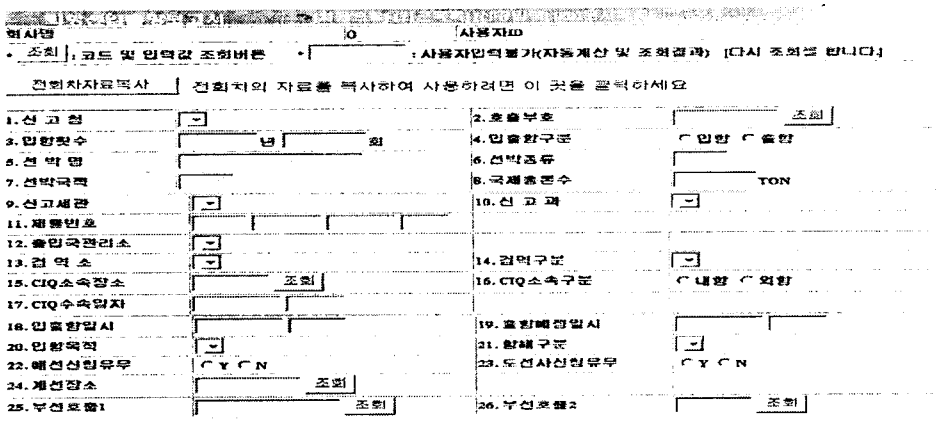
CGI(Common Gateway Interface) 및 스크립트를 이용한 웹 서비스는 아직 수출입과 관련하여 전체적으로 상용화되어 있지는 않다. 클라이언트 어플리케이션 기반의 서비스가 폐쇄 시장이라고 불리우는 반면, 본 시스템은 이러한 문제점을 해결하고자 제시된 웹과 파일전송프로토콜(FTP)을 활용한 인터넷 EDI 방식, 즉 하이브리드 방식을 위해 개발되어진 시스템이다. 대표적인 웹 서비스 방식으로 클라이언트/서버 유형은 2-tier plump client에 해당한다. 개발 기술로 사용된 것은 ASP(Active Server Page)로 서버 스크립트의 대표적 기술인데, 하나의 프로세스 내에서 사용자의 요청이 쓰레드 방식(thread)으로 처리되기 때문에 서버에 걸리는 부하를 줄여 줄 수 있다. 그러나 2-tier의 구조이므로 표현층 로직과 비즈니스 로직을 모두 포함하고 있고, 프로그램의 유지보수와 재사용을 어렵게 만든다. 그리고 사용자 커넥션 및 자원의 한계 때문에 데이터베이스의 확장성이 제한을 받으며, TCP/IP 네트웍을 통해 너무 많은 데이터들이 처리된다는 단점을 가지고 있다. 이것은 곧, 클라이언트의 컴퓨팅환경에 따른 현저한 속도의 차이로 나타난다. 동시 사용자가 집중되거나, 데이터베이스에 접근율이 높아질 경우 서버의 과부하현상을 일으킬 수 있으며, 이는 곧 서버의 다운이나 속도의 저하를 발생시킬 수 있다. 이러한 문제를 다소 해결하기 위해 저장 프로시저(stored procedure)를 사용하였다. 저장 프로시저는 미리 데이터베이스 서버에 일련의 SQL명령을 생성해 놓고, 프로시저를 실행하여 몇 개의 SQL명령을 간단히 실행할 수 있도록 처리되므로 네트웍을 통해서는 적은 양의 데이터만 이전되고, 데이터

처리내용을 수정하는 것이 클라이언트 코드를 수정하지 않고도 가능하다. 또한 데이터베이스가 제공하는 코드를 이용해 효율적으로 데이터가 처리되기 때문에 데이터베이스의 성능이 향상되는 이점을 가지고 있다. 서버개발환경으로는 Windows NT와 IIS 4.0, MS SQL 7을 사용하였다.



<그림 6> ASP를 이용한 웹 서비스 구조

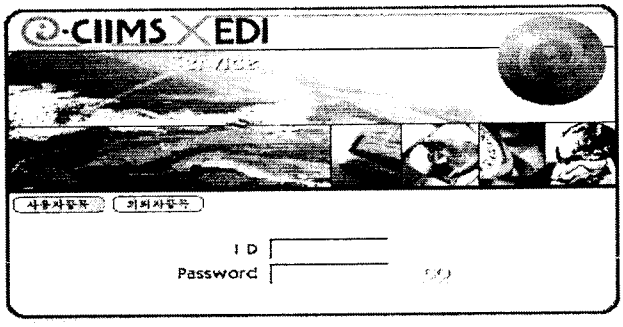
클라이언트에는 표현층과 응용로직의 일부가 있는데, 응용로직은 문서검증을 위한 Java Script와 VB Script가 포함된다. 서버에는 응용로직의 나머지 부분과 데이터관리층이 존재한다. 서버의 응용로직은 ASP의 플랫폼이 들어가 있어 문서의 업무로직을 담당하고, 트리거(Trigger)나 저장프로시저를 사용할 수 있다. <그림 7>는 ASP로 개발 된 외항선 입출항 보고서의 입력화면이다.



<그림 7> ASP를 이용한 웹 EDI 시스템

3.2.3 XML 기반의 XML/EDI 시스템

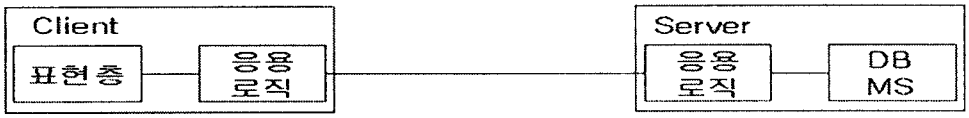
본 시스템은 상용화되고 있지는 않으나, 수출통관과 관련하여 수출신고서, 정정신청서, Invoice, Packing list, 수출신고수리, 오류통보, 수출면허, 정정오류, 정정접수 등을 포함하고 있다.



<그림 8> XML/EDI 시스템의 사용자 login화면

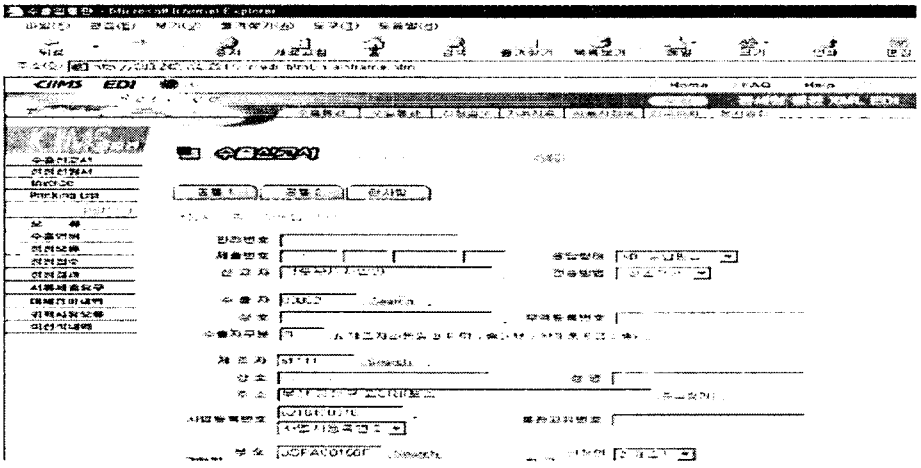
기존 EDI의 문제점으로 지적되었던 새문서의 추가 발생 시 생기는 전용 소프트웨어의 업그레이드(upgrade)와 그에 따른 경비의 지출, 관리의 문제점을 해결하고자 개발되었다. 본 시스템의 특징으로는 웹 브라우저 상에서 신청인으로부터 작성된 수출 관련 문서들을 XML문서와 HTML문서로 작성하여 웹 서버에 전송하고 전송되어진 XML문서와 데이터는 각각의 관리 제어되는 처리과정을 통해 데이터베이스에 저장 과정을 거쳐 최종 전자문서(EDI Message)로 변환하여 전송하는 시스템이다. XML 기반의 XML/EDI 서비스는 궁극적으로 3-tier 기반 하에서 설계되어야 하나, 본 시스템은 2-tier plump client의 유형에 속한다. 그러므로 데이터의 재사용성에 있어서는 타 어플리케이션보다 뛰어났으나 상용화를 위한 테스트과정에서 업무의 처리속도가 현저히 떨어지는 것으로 나타났다. 그 이유는 하나의 XML문서나 HTML로 변환되어진 문서를 보여주기 위한 과정의 복잡성에 있다. 먼저 미리 정의되어진 DTD(Document Type Definition)나 스키마의 구조와 비교하여 유효성을 검증한다. 그 뒤 사용자가 입력할 수 있는 빈 템플릿(Templet)을 생성하고, XSL(Extensible Stylesheet Language)의 자바 애플릿을 이용한 변환과정을 거쳐야만 비로소 사용자가 원하는 화면을 XML이나 HTML로 변환해서 보여준다. 결국, 하나의 화면을 보여주기 위해 XML은 여러번의 과정을 거쳐야한다. 또한 XSL의 강력한 기능을 위해서는 자바와 같은 언어의 연동이 필수적이므로 개발의 과정이 쉽지 않다. 또한 국내 항만물류산업의 경우 아직 EDI 신고를 위한 UN/EDIFACT에 대한 DTD나 스키마의 정의가 이루어지지 않았으며, 앞서 제기한 UN/EDIFACT의 복잡성으로 인해 DTD나 스키마의 정의작업이 이루어진다하여도 많은 혼선과 어려움을 겪을 것으로 보인다. 그러나 항만물류 산업에 있어, 가장 문제시 되고 있는 점은 정보공유체계 미흡이다. 국내 해운관련 전자상거래가 아직 초보적인 단계이며, 여러 중소 업체가 제공하는 서비스도 기본적인 서비스만을 제시하고 있다. 그러나 해외 해운관련 전자상거래 현황을 살펴볼 때, GT Nexus, Inttra.com, Cargosmart.com과 같은 세계적 해운 e-Marketplace 구축작

업이 활발히 진행 중에 있다. 따라서 향후 이러한 해운 e-Marketplace라는 통합 시스템 구축을 위해 XML 기반기술의 활용은 필수적인 것으로 보인다. 현재 전세계적으로 XML을 위한 전자상거래 프레임워크가 제시되고 있으며, 국내 해운·항만업계에서도 2002년도에는 XML을 이용한 시스템이 본격화 될 것으로 전망되어지고 있다. 이러한 관점에서 볼 때 본 시스템은 앞으로 이루어 질 XML/EDI 시스템의 개발에 초석이 되어 질 것으로 보인다.

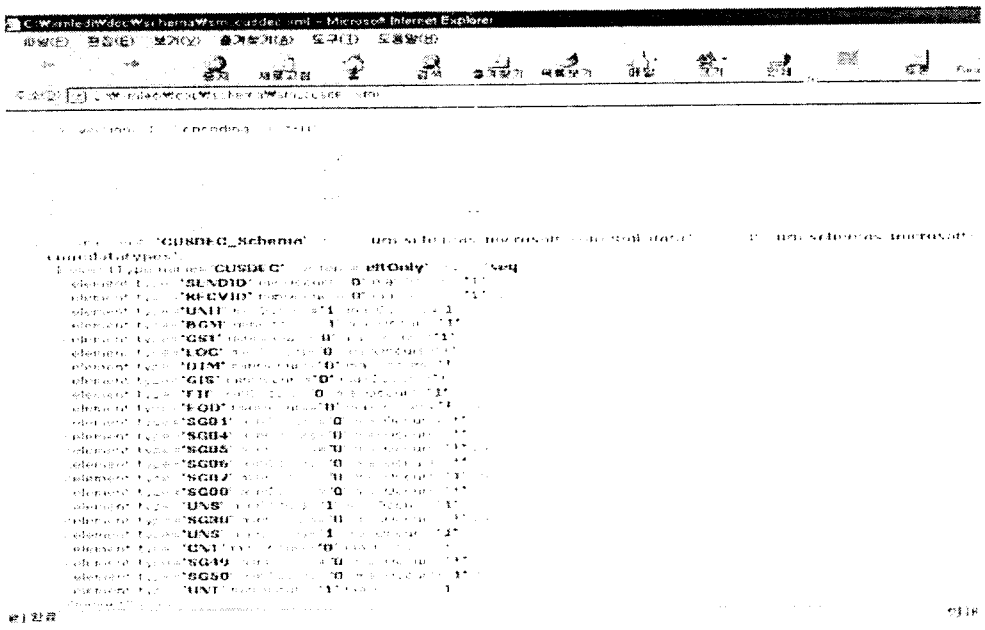


<그림 9> XML/EDI 시스템의 구조

XML/EDI 서비스는 XML의 특성인 계층적 데이터저장소의 특징을 가지고 있으므로 클라이언트/서버 시스템의 유형으로 표현하는데 다소나마 어려움이 있다. 클라이언트에는 표현층과 응용로직을 가지고 있는데, 응용로직은 필요시에만 서버에 접근하여 다운을 받아 표현하는 애플릿이 포함되어있다. 서버에는 다수의 응용로직과 데이터관리층이 존재한다. XML/EDI의 개발환경은 Compaq-Prolient 8000을 이용하였고, Windows 2000 Server와 Java, Servlet, JDBC와 IIS 5.0을 이용하였다. <그림 10>과 <그림 11>은 입력 화면과 XML 구현을 위한 스키마구조이다.



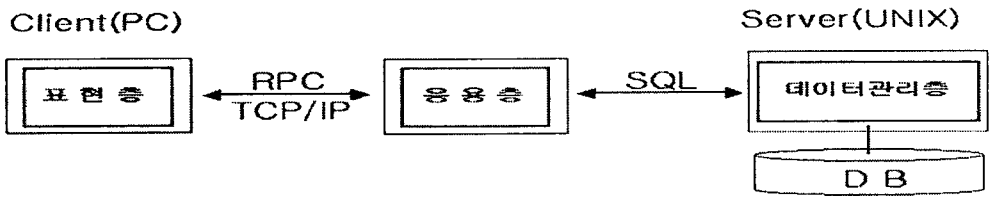
<그림 10> XML로 표현된 입력화면



<그림 11> XML구현을 위한 스키마구조

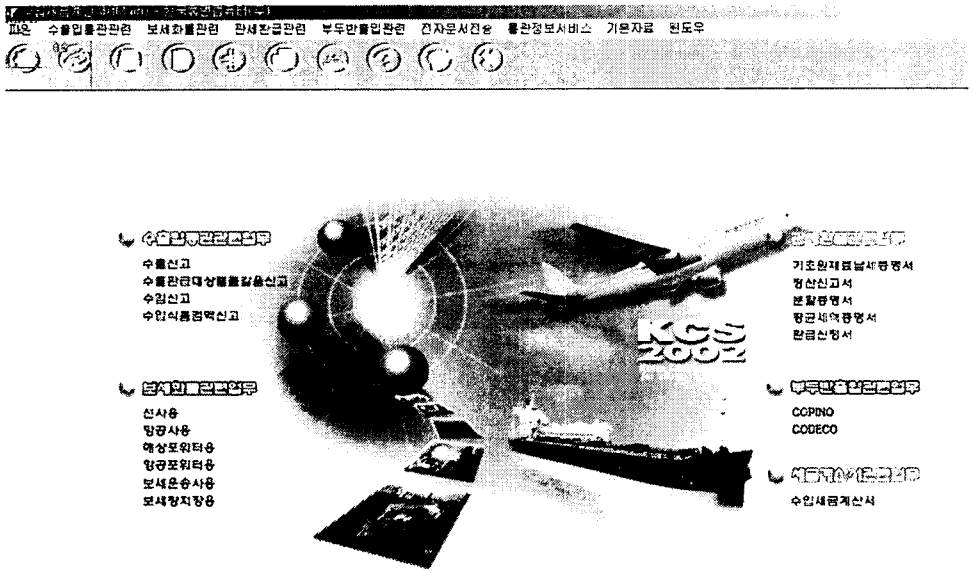
3.2.4 3-tier의 분산객체 지향 EDI 시스템

최신의 네트웍 기술인 DCOM(Distributed Component Object Model)을 이용한 3-tier thin client의 유형이다. 수출입 통관관련업무에 이용되기 위해 개발되었으나, 상용화는 되고있지 않다. 3-tier구조는 데이터베이스 서버에 접근해서 데이터를 처리하는 중개자 혹은 어떤 업무에 필요한 비즈니스 로직들을 별도의 어플리케이션 서버에 넣어두는 것이다. 혹은 클라이언트 프로그램과 데이터베이스 서버 사이에 위치해서 필요한 일들을 대신 처리해 준다고 해서 데이터브로커(Data Broker)라고도 부른다. 데이터 어플리케이션을 만들면 클라이언트 어플리케이션은 단순히 사용자 인터페이스만 가지고 있는 가벼운 클라이언트가 될 수 있다. 또한 데이터베이스 서버쪽에도 그리 많은 부담을 주지 않아도 된다. 본 서비스는 델파이 5.0의 마이다스(MIDAS: Multi-tired Distributed Application Service Suite)로 구현되었으며, MS의 DCOM과 결합된 기술이다. 새로운 네트웍기술인 만큼 개발당시 많은 시행착오를 겪었으며, 개발기간도 상당히 소요되었다. 아래의 그림은 델파이 마이다스로 구현된 사용자 메뉴화면과 서비스이다. 웹으로도 접근가능하며 웹 환경으로 확장하기 위해 Active X 기술을 도입하였다.

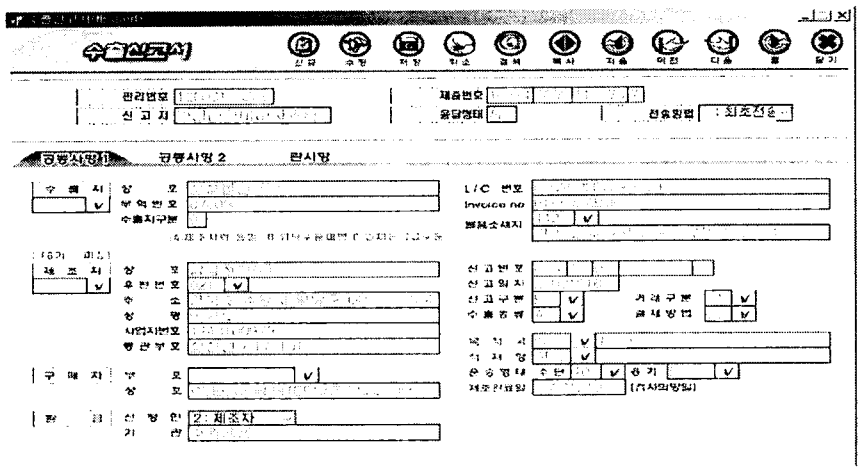


<그림 12> 3-tier의 분산객체지향 EDI 시스템 구조

클라이언트에는 표현층만 존재하고 서버에는 데이터관리층만이 존재한다. 응용층에는 업무로직과 데이터액세스 처리 로직을 포함한다. 클라이언트와 응용서버는 RPC(Remote Procedure Call)를 이용하여 서로 통신한다. 본 방식으로 구현된 EDI 시스템인 KCS 2002의 첫 메뉴화면과 수출신고서 입력화면을 <그림 13>과 <그림 14>에서 보여주고 있다.



<그림 13> 델파이 마이다스로 구현된 EDI 시스템 메뉴화면



<그림 14> 델파이 마이다스로 구현된 EDI 서비스

이제 이상과 같은 4개의 서비스 유형을 <표 3>와 같이 정리하였다.

<표 3> 항만물류 서비스의 유형

	Client Application EDI 시스템	스크립트를 이용한 Web EDI 시스템	XML기반의 XML/EDI 시스템	3-tier의 분산객체 지향 EDI 시스템
Client/ Server 유형	2-tier fat client	2-tier plump client	2-tier plump client	3-tier thin client
개발 환경	OS:Windows NT DBMS: Paradox 개발기술: 델파이 2.0	OS: Windows NT/2000 DBMS:MS SQL6.5-7.0 웹서버: IIS 4.0-5.0 개발기술: ASP, 저장프로시져, Java Script, VB Script	OS: Windows 2000 Server 웹서버: IIS 4.0 DBMS: MS SQL 7.0 개발기술: Java JDK1.2.2, Servlet JSDK 2.0, MS SQL Server JDBC, Connection Pool 1.0.7, Java Mail	OS:Windows2000 Server DBMS: MS SQL 7.0 개발기술: 델파이 5.0 델파이마이다스, MS DCOM, Active X
이용 및 개발 문서	수출입 EDI 신고의 모든 문서에서 이용	해양수산부 신고의 모든문서가 이용되고 있으며, 수출입관련 문서 개발완료. 관세청 문서 중 해상관련 수출입 업무 관련 문서 개발	수출통관 문서업무인 수출신고서 외 3개 문서, 수신내역 조회업무인 수출신고 수리정정/취하업무 외 9개 문서, 신청업무, 기초업무, 사용자정보업무 문서 개발	관세청 수출입신고서 개발

Ⅳ. 항만물류 EDI 시스템의 비교 분석 및 최적 구현 방안

4.1 항만물류 EDI 시스템의 비교 분석

지금까지 살펴본 각각의 서비스들을 비교평가하기 위해 다음과 같은 평가 요소를 제시한다. 본 평가요소는 지금까지 EDI활용 실태분석에 대한 선행 연구로 이루어진 항만관련업체의 EDI활용 현황과 개선방안(박남규,최형림, 1998a)에서 설문조사되어진 사용자 불만사항을 종합한 결과와 해양수산부의 해양개발기본계획에서 선진정보화 체계확립을 위해 제시된 정보기술 개선방안을 중심으로 다음과 같은 6가지의 평가요소를 집약하였다.

첫째, 속도이다. 항만물류 서비스의 이용자는 화주, 선사, 포워드, 운송업자, 국가기관 등 다양하고 폭넓은 편이다. 또한 이러한 서비스의 이용이 모두 분산환경에서 이루어져 신고자와 신고대상이 뚜렷이 구분되고, 한번에 여러 건의 문서가 처리되기도 한다. 대형선사의 경우 화주나 포워드의 업무를 대신하여 신고하기 때문에 한번의 수출입과정에서 수십에서 수백건의 자료가 처리되기도 하고, 영세업자인 경우 몇 건의 자료만을 처리하기도 한다. 이 때 업무의 처리속도는 사용자의 입장에서 제일 중요한 평가사항이 된다.

둘째, 프로그램 배포 및 시스템 유지관리이다. 새로운 프로그램을 개발하였다 하더라도 많은 클라이언트에게 배포하는 문제는 절대 간과할 수 없으며 계속적인 유지보수 및 업데이트의 업무도 포함된다. 현재의 시스템을 파악하기 위한 계속적인 모니터링 기능도 필요하다.

셋째, 개발 용이성이다. 코드의 재사용성은 한번 개발된 코드나 모듈을 다시 이용함으로써 개발기간과 비용을 단축시킬 수 있으며, 기업의 측면에서는 비용과 직결되는 평가사항이다.

넷째, 사용의 편리성이다. 사용자 인터페이스의 구성이 얼마나 효율성을 가지고 있느냐의 문제와 프로그램 구축 및 사용의 편리성이 명료해야 한다. 서비스 이용자는 업무에 밝고 이전의 서비스를 계속적으로 이용해 왔기 때문에 새로운 시스템으로 구축된 서비스는 보다 나은 사용성을 제공해야 한다. 서비스 이용시 사용자의 입장에서 최소의 입력만으로 신고가 가능해야 하며, 하나의 입력 오류도 발생하지 않도록 유도하고 오류발생시 사용자가 충분히 인지하고 재입력할 수 있는 기능이 제공되어야 한다.

다섯째, 보안이다. 수출입과 관련된 업무에서는 기업간 거래, 기업과 정부간의 거래이므로 기업의 정보는 철저한 보안성이 유지되어야 한다. 해킹, 바이러스의 위험에서 벗어나야 하며, 데이터의 손실은 절대 일어나서는 안된다.

여섯째, 로드 밸런싱(load balancing)과 페일오버(fail-over) 서비스이다. 수많은 이용자가 동시 접속할 수도 있으며, 한 건의 수출입신고를 위해서는 몇 개의 신고문서

가 필요하다. 그만큼 서버에 많은 부하가 걸릴 수 있다. 서버의 부하로 인한 다운이나 현저한 속도감소는 많은 이용자들의 불만사항이었다. 적절하게 로드를 분배해야 하며, 서버다운 시 필요한 응급처리는 필수적인 사항이다.

<표 4>에서는 각 항만물류 EDI 시스템에 대한 유형별 일반적 평가를 정리하였고, <표 5>에서는 위에서 언급한 6개의 평가요소를 이용하여 앞서 제시한 4개의 시스템을 평가한 결과이다. 본 연구에서 제시한 평가모델의 객관적인 측정을 위해 현재 항만물류 EDI 서비스를 담당하고 있는 (주)한국휴먼컴퓨터의 서버컴퓨터에서 4가지의 프로그램을 설치한 후 공동 측정하였다.

<표 4> 항만물류 EDI 시스템의 유형별 일반적 평가결과

	클라이언트 어플리케이션 EDI 서비스	스크립트를 이용한 웹 EDI서비스	XML기반의 XML/EDI 서비스	3-tier의 분산객체 지향 EDI서비스
장점	-전반적으로 빠른 속도 -익숙한 인터페이스 -오랜 사용기간으로 인한 서비스의 신뢰성 -개발의 용이성	-미들웨어 사용시 하나의 DBMS벤더로부터 구매사용 -클라이언트와 서버사이의 메시지를 줄임 -개방적 EDI 방식 -개발도구의 보편화 -실시간 처리 확인 가능 -웹 브라우저의 통일된 인터페이스로 인한 편리성	-데이타의 재 사용성 -개방적 구조 채택 -운영비 감소	-네트웍의 성능향상 -확장성 증가 -효과적인 데이터 액세스 -컴포넌트의 이용 -보안의 확대
단점	-소프트웨어 분배 및 버전 컨트롤의 어려움 -온라인 모니터링 기능 없음	-네트웍의 부하 발생 우려 -컴퓨터사양 및 네트웍에 따른 속도의 감소 -클라이언트나 서버의 비대	-속도의 저하 -개발구축 기술의 미미와 사용의 복잡성 및 어려움 -DTD 및 스키마의 표준 미정립	-개발기술의 습득이 어렵고 도입시 개발비용과 개발기간이 높아짐 -다계층 구조의 논리적·물리적 설계의 복잡성

<표 5> 유형별 항만물류 EDI 시스템의 비교평가 결과

	클라이언트 어플리케이션 EDI 시스템	스크립트를 이용한 웹 EDI 시스템	XML기반의 XML/EDI 시스템	3-tier의 분산객체 지향 EDI 시스템
속도	다른 서비스 중 가장 빠르다. 데이터베이스의 연동시 조금 지체현상이 일어난다.	컴퓨팅 환경과 네트웍상태에 따른 속도 차이가 크다. 사용자가 늘어날수록 속도가 현저히 느려진다.	ASP를 이용한 서비스와 속도가 비슷하거나 느린편이다. 다른 서비스와 비교하여 가장 느린 것으로 평가된다.	클라이언트 어플리케이션에 비해 느리지만, 평균속도는 빠른 편이다. 전체적인 업무처리에 있어서는 오히려 가장 나은 속도를 낸다.
	상	중	중하	상
프로그램 배포 및 시스템 유지관리	배포의 과정이 까다롭고, 시스템의 업그레이드 시 많은 어려움을 가진다. 온라인 모니터링의 기능이 불가능하다.	클라이언트는 웹브라우저만을 가지고 인터넷환경만 구축되어 있으면 된다. 시스템의 유지관리는 모두 서버에서 이루어진다. 실시간적인 모니터링 기능도 가능하다.	ASP를 이용한 서비스와 같이 클라이언트는 웹브라우저만을 가지고 인터넷환경에 접속만 하면된다. 유지보수 및 업데이트는 서버에서만 이루어진다.	서버와 클라이언트의 데이터를 미들웨어가 동기화 시켜주고 데이터베이스 접근,유지,보수,관리문제를 해결해준다. 실행파일형태로 배포되기도 하며, 웹 환경에서도 업무처리가 가능하다.
	하	상	상	상
개발 용이성	4GL을 이용한 개발환경은 많이 보급되어 있으며, 솔루션 자체의 강력한 기능으로 GUI 및 네트워크상의 통신이 가능하도록 한다. 여타 서비스에 비해 개발의 과정이 빠르고 쉬운편이다.	웹 환경에서의 작업은 오프라인에서의 작업에 비해 많은 손질이 필요하다. 웹서비스 구현은 두개 이상의 기술이 접목되며, 디버깅작업이 어렵다. 사용하는 기술과 지원되는 환경의 호환유지가 이루어져야한다.	XML개발 시 아직까지 XML의 개발툴이나 솔루션이 부족하며, 복잡한 업무와 다양한 기능을 제공하기 위해 Java와 같은 언어가 필수적으로 필요하다. 상당히 많은 작업과정이 요구된다.	분산환경구축을 위한 솔루션은 있으나, 미들웨어를 지원하는 개발기술을 습득하기 어려우며, 대용량의 작업 범위를 가지고 있다. 그러나 한번 작성한 코드에 대한 재사용성이 보장된다.
	상	중	중	중
사용의 편리성	숙달자는 빠른 입력이 가능하나 전체적인 사용자 인터페이스부분은 미흡한 편이다. 입력에 필요한 모든 기능이 별도로 이루어져 있는데, 입력이 모두 끝난 후에야 오류수정이 가능하다.	입력의 중복을 최소한 줄이고 사용자가 최소한의 입력이 가능하도록 구성하였다. 최대한 사용자가 입력오류발생치 않도록 유도하고 있으며 입력안내가 실시간으로 제공된다.	ASP를 이용한 서비스와 동일한 기능을 제공하며, 입력오류가 발생치 않도록 최대한으로 인터페이스를 구성하였다.	4GL의 단순한 인터페이스 구성과 웹의 실시간인 입력안내 기능 등 다른 서비스의 인터페이스 특성을 모두 살리고 있다.
	중	상	상	상

	클라이언트 어플리케이션 EDI 시스템	스크립트를 이용한 웹 EDI 시스템	XML기반의 XML/EDI 시스템	3-tier의 분산객체 지향 EDI 시스템
보안	기업내·외부적으로 보안상의 문제가 존재한다. 파이얼 월을 사용하고는 있으나 중간과정에서 해킹의 우려가 남아있다.	파이얼 월을 사용하고 있으며, 저장프로시저를 액세스할 수 있는 권한을 제약함으로써 데이터구조에 대한 보안을 증가시킬 수 있다.	인증관리부분의 모듈이 분리되어 있지만, 현재 보안의 효용성을 측정하기는 어렵다. 별도의 보안 시스템이 구축되어야 할 것으로 보인다.	컴포넌트 기반으로 권한을 부여하고 거부할 수 있으며, 관리작업도 단순해진다.
	중하	중	중	상
로드밸런싱/페일오버	네트웍관리를 위해서는 서버의 업그레이드만이 유일한 방안이다.	저장프로시저 기능으로 다소나마 네트웍을 통해서는 적은 양의 데이터만이 이전되며, DB의 성능도 향상시킬 수 있다.	현재, 네트웍 부하를 해결할 수 있는 방안이 모색되지 않고 있다.	여러 컴퓨터에 어플리케이션 서버를 설치해서 작업을 분산시킬 수 있다. 가장 뛰어난 네트웍 관리를 보장한다.
	중하	중	중하	상

(하 : 기능이 없거나 미진하여 만족도가 낮음 → 상 : 기능이 우수하고 만족도가 높음)

이상의 평가결과를 살펴보면 3-tier의 분산객체지향 EDI 시스템이 가장 우수한 시스템으로 평가되었으며, XML/EDI의 시스템은 속도와 개발의 용이성 및 네트웍의 관리면에서 취약한 것으로 나타났다. ASP를 이용한 웹 EDI 시스템은 속도의 문제만 해결할 수 있다면 앞으로 더욱 나은 서비스를 제공할 수 있으리라 본다. 이상의 평가결과가 절대적일 수는 없지만, 앞으로의 다른 연구 및 기업시스템 측정을 위한 지침이 될 수 있으리라 본다.

4.2 사용자관점에서 항만물류 EDI 시스템의 최적 구현방향

지금까지 각 어플리케이션의 비교를 통해 항만물류 서비스와 시스템을 살펴 볼 수 있었다. 평가결과를 한마디로 요약해 보면, 최적의 속도에서 사용자의 원스톱 서비스 구현을 위한 시스템이 구축되어야 한다. 결국 클라이언트/서버 구조의 정보시스템은 기본적으로 다양한 분산객체로 구성되며, 복잡한 인터페이스를 요구하고 있어 메인프레임 컴퓨팅 구조의 정보시스템에 비해 보다 강력한 표준과 구조적 정형화 작업을 요구한다.

사용자의 입장에서는 기업의 서비스 시스템이 2-tier인지 혹은 3-tier로 이루어져 있는지를 알 수 없으며 알고 있을 필요도 없다. 사용자가 요구하는 중요한 사항들을 살펴보면 서비스의 속도, 사용의 편리성, 입력정보의 안정성, 통신료 및 유지보수비의

최소화 등을 들 수 있는데, 기업의 입장에서 이러한 사용자의 요구사항을 만족하기 위해서는 시스템의 환경이 최적화 되어야 하며, 그러기 위해서는 시스템 개발의 용이성, 코드의 재사용, 시스템의 지속적인 유지 및 관리, 서버의 네트웍 성능향상을 위한 로드밸런싱과 페일오버 서비스 등의 요건을 갖추어야 한다. 이러한 요건을 충족할 수 있는 항만물류 EDI 시스템으로 앞서 평가방법을 이용한 결과로 3-tier의 분산객체 지향 네트웍 시스템이 가장 최적의 환경임을 알 수 있었다. 그러나 객체지향 분산 네트웍 시스템을 개발하기 위해서는 여러 가지의 고려사항이 있다. 3-tier의 분산형 네트웍 시스템은 개발 시 많은 비용과 시간을 요구하고 있다. 기존의 2-tier 시스템에서 3-tier로 전환하기 위해서는 업사이징(up-sizing)이나 부분적으로 필요한 사항을 먼저 개발해 나가는 라이트사이징(light-sizing)이 이루어져야 하며, 개발자간의 개발 표준 유지와 강제적 준수의 제도화, 공통적 구성요소에 대한 사전 부품화, 개발자의 개발 지식 공유를 통한 개발 생산성 향상 등을 실현 할 수 있는 방법이 정립되어야 한다.

기업 최적의 환경을 위해서는 3-tier의 구조로 구축함이 가장 이상적임을 알 수 있었지만, 위의 고려사항이 함께 충분히 인식되어야 한다. 그러므로 이러한 고려사항을 충족시킬 수 없다면 다른 유형이 보다 나은 대안이 될 수 있다. 예를 들어, 기업에서는 대략 백 명 내외의 사용자가 하나의 데이터베이스를 이용하는 서비스이거나 신고하는 문서에 입력사항이 적을 때, 또는 처리하는 문서의 양이 적은 영세업체들에게는 2-tier의 시스템이 유리하다고 볼 수 있다. 반면, 수 백명 이상의 사용자가 이용하고, 복잡한 입력사항을 요구하는 문서, 두 개 이상의 데이터베이스가 연결되어 있고, 대형 선사나 포워드 업체의 업무일 경우에는 3-tier의 서비스가 필요하다.

현재 메인 프레임 시스템 구조와 클라이언트/서버 구조가 공존하고 있듯이 네트웍 컴퓨팅환경이 기술적으로 성장하더라도 기존의 환경과의 공존은 불가피하며, 주요 성공요소 중의 하나인 어플리케이션 설계에 대한 관심과 노력이 증대되어야 할 것으로 보인다. 또한 3-tier 구조가 아무리 뛰어나다 하더라도 클라이언트 어플리케이션 EDI 시스템은 지금까지 기존의 조직에서 오랫동안 사용되어져 왔기 때문에 당분간 쉽게 변경되지 않을 것이며, 서버 스크립트를 이용한 웹 서비스도 계속적으로 병행 사용되어 질 것으로 보인다. XML/EDI의 방식은 현재 개발되어진 환경을 3-tier구조로 변경하고, DTD 및 스키마의 정의가 항만산업에서 이루어진다면 더 나은 서비스가 가능해 질 것으로 보인다. 지금도 XML의 기술은 발전되고 있으며, XML을 생성하거나 제작할 수 있는 솔루션, DBMS와의 연동, XML을 다양한 형태로 처리할 수 있는 이론 및 기술들이 제시되고 있어, 2002년부터 우리나라 정부기관의 EDI 시스템이 XML로 전환할 것을 발표했으므로 XML로의 전환은 필수적인 사항이라 할 수 있다.

V. 결 론

본 연구에서는 지난 몇 년간 수행되어온 프로젝트의 결과물인 EDI 서비스와 실무에서 사용되고 있는 EDI 전용 프로그램을 이용하여 각각의 서비스를 시스템 구조측면·기술적 측면·사용 측면을 중심으로 비교 분석해 보았다. 항만물류산업에서 뿐만 아니라 현재 BtoC, BtoB에서 모든 상거래와 문서교환, 정보교환은 다양한 방식으로 구축되어 최적의 서비스를 제공하는데 목표를 둔다. 본 연구에서 제시된 항만물류 EDI 시스템의 유형별 평가는 지금까지 수행되어온 항만물류 EDI 시스템 개발프로젝트에서 여러 가지의 애로사항과 시행착오 속에서 얻어진 경험의 결과이며, 이러한 문제를 해결하고자 항만물류산업의 관계자와 많은 대화와 요구사항을 조사하고, 시스템을 분석함으로써 얻어진 결과이기도 하다. 현재 새로운 네트웍 컴퓨팅의 기술이 도입되고, 웹이라는 필수적인 환경이 접목되면서 여러 가지의 서비스와 시스템 구조가 개발되고 있으며, 이것은 기존의 환경과 공존되고 있다.

앞서 제시한 평가결과에서 제시하고 있듯이, 최적의 항만물류 EDI 시스템 구조는 3-tier가 되겠지만, 2-tier 시스템 구조의 장점은 최대한 유지한 채, 단점을 보완하는 여러 가지 방안이 제시될 수도 있다. 또한 앞서 제기한 바와 같이 기업의 주요 성공요소 중의 하나인 어플리케이션 설계에 대한 관심과 노력이 증대되어야 할 것으로 보인다. 각각의 조직에 적합한 IT구조를 선택하여 꾸준히 추진해나가는 것이 바람직할 것으로 보인다.

본 연구의 결과에서 제시된 다양한 시스템은 첫째, 웹 환경을 기본적으로 제공하고 둘째, 통합시스템 구축을 위해 개발되었으며 셋째, 사용자 위주의 인터페이스와 서비스를 중심으로 개발하였다. 마지막으로 이러한 시스템 구축을 통해 궁극적으로 정보기술의 활용에 있어 원스톱서비스를 이룰 수 있어야 하겠다. 이제 항만물류 EDI 서비스가 효과적으로 사용자를 만족시키기 위해서는 앞으로 많은 개선사항이 필요하겠지만, 기업·정부에서 추구하고자 하는 시스템과 서비스에 대한 하나의 프로토타입으로서 본 연구가 충분한 역할을 할 수 있을 것으로 보인다.

참고문헌

김병곤, 김종욱, "EDI 효과요인의 계층적 모델 개발에 관한 실증적 연구," 경영정보학연구, 제9권, 제3호, 1999, pp. 159-173.

김병곤, 정경수, "한국기업의 EDI구현 결정요인," 경영정보학연구, 제9권, 제1호, 1999, pp. 165-189.

박광호, "정보시스템 개발을 위한 애플리케이션 프레임웍의 설계," 경영정보학연구, 제8권 제1호, 1998, pp. 87-101.

박남규, 최형림, "항만관련업체의 EDI 활용 현황과 개선방안," 정보시스템연구, 제7권 제2호,
1998a, pp. 147-172.

박남규, 최형림, "항만물류 산업에서의 원스톱서비스 시스템구현 방안," 제1회 CIIPMS 연구결과 발표회 논문집, 1998b, pp. 13-27.

박남규, 최형림, 김현수, 박영재, 조재형, 이철우, "수출입 컨테이너화물 원스톱서비스 시스템 개발," 제2회 CIIPMS 연구결과 발표회 논문집, 1999, pp. 63-74.

박남규, 김현수, 조재형, "수출입 화물 원스톱 서비스 상용화에 관한 연구," 한국항만학회 추계정기학술대회, 2000a, pp. 43-52.

박남규, 김현수, 조재형, "항만물류 원스톱서비스 시스템 개발," 제3회 CIIPMS 연구결과 발표회 논문집, 2000b, pp. 279-287.

박남규, 김현수, 최형림, 손형수, 김성훈, 조재형, 박영재, 김영호, 송정훈, 조광민, 주승협, "통관 Web-EDI 서비스 시스템 상용화에 관한 연구," 제3회 CIIPMS 연구결과 발표회 논문집, 2000, pp. 415-428.

신승식, 김수엽, "해운·항만 물류정보화를 위한 기반조성 연구," 한국수산해양수산개발원 기본연구, 2000-05호, 2000.

이재기, "항만 물류 최적 네트워크 개발," 제1회 CIPMS 연구결과 발표회 논문집, 1998, pp. 29-40.

임강진, 신재호, 권병희, "Network Bible 3rd ed," 영진.com, 2000.

윤영수, 서의호, 이원창, "인트라넷 기반의 의사결정을 위한 지식공유 아키텍처," 경영정보학연구, 제9권 제4호, 1999, pp. 89-105.

차윤숙, 정문상, "전자상거래 참조모델 개발 및 활용방안," 정보시스템 연구, 제 8권 제1호, 1999, pp. 65-92.

최형림, 박남규, "항만물류 통합 데이터베이스의 구축방안," 제1회 CIIPMS 연구결과 발표회 논문집, 1998, pp. 3-12.

최형림, 박남규, 김현수, 이현철, 이창섭, 정헌욱, "SCM 방식을 적용한 항만물류 통관 Web-EDI 시스템 개발에 관한 연구," 제4회 CIIPMS 연구결과 발표회 논문집, 2001, pp. 5-14.

최형림, 박경환, 안선하, 황성진, "Biztalk에 기반한 항만물류 XML/EDI 시스템의 개발," 제4회 CIIPMS 연구결과 발표회 논문집, 2001, pp. 15-22.

해양수산부, 한국전산원, "해양수산정보화 기본계획," 1998.

한국전산원, "SGML-XML-EDI 통합 및 연계방안," 1999.

한국전산원, "차세대 EDI 표준화 동향 분석," 1999.

Hilerio, Israel, "XML-RPCs and Java," XML-Journal, Vol. 1, Issue 3, 2000, pp. 32-38.

Webber, David, "Introducing XML/EDI Framework," Electronic Markets, Vol. 8, No. 1, 1998, pp. 38-41.

<Abstract>

A Comparison Study for Optimal Implementation of the EDI Systems of Port Logistics

Hyun Soo Kim	Dong-A University	hskim@daunet.donga.ac.kr
Nam Ku Park	Tongmyong University	nkpark@tmic.tit.ac.kr
Kay Seob Han	Dong-A University	kshan@daunet.donga.ac.kr
Hyung Rim Choi	Dong-A University	hrchoi@daunet.donga.ac.kr
Jae Hyung Cho	Dong-A University	b990008@daunet.donga.ac.kr

At present, the EDI systems are indispensable software in port logistics industry. Currently, a monopolistic VAN/EDI service provider operates the EDI services. The current EDI client software has the 2-tier fat client/server architecture. However, the current EDI software is lack of Web interface and causes lots of cost for maintenance. Therefore, a variety of implementation architecture has been being tried by using script, XML and distributed object-oriented technology. Web/EDI and XML/EDI are the new EDI systems. However, the EDI systems have some limitations such as speed.

This study intends to compare the variety implementation architecture for the EDI systems in the users' perspective and explore the strong and weak points of each architecture. We compared the EDI systems based on our experience of more than 2-years of implementation project for the EDI systems of port logistics.

We categorized the EDI systems as client application EDI, Web EDI using script, XML/EDI, and 3-tier distributed object-oriented EDI system. We compared them with criteria such as speed, program maintenance, easiness of implementation and usage, security, and load balancing and fail-over. Finally we discuss the direction of optimal EDI system architecture for the future.