INMARSAT-C를 통한 인터넷 기반의 메시지 전송 에이전트 설계 및 구현

朴沇植*, 李泰吾**, 林宰弘***

Design and Implementation of Internet-based Message Transfer Agent via INMARSAT-C

Yeoun-Sik Park*, Tae-Oh Lee**, Jae-Hong Yim***

요 약

본 논문은 INMARSAT-C를 통하여 육상의 인터넷 메일이나 웹 사용자와 선박간 편리한 메시지 송수신 서비스를 제공하는 인터넷 기반의 메시지 전송 에이전트의 설계 및 구현에 대해서 논한다. 이를 위하여 육상과 선박간 통신을 위한 전체적인 시스템 구성, 에이전트의 구성 모듈, 송·수신 데이터의 관리를 위한 데이터베이스를 설계, 구현한다.

본 논문의 타당성 검토를 위하여 메시지 파싱(parsing), 데이터베이스, 메시지 포맷, X.25 인터페이스 등의 모듈을 구현하여 시험하고, 그 결과를 토대로 선박 내 데이터베이스 접근과 더불어 선박 자동화 시스템의 가능성을 논한다.

Abstract

This paper presents the design and implementation of Internet-based message transfer agent via INMARSAT-C between the land and ship for the convenient message transfer service using the Internet mail or Web. In order to achieve these purpose, overall system configuration for communication between the land and ship, agent modules and database for managing transmission and receiving data are described.

* 경상대학교 해양과학대학 정보통신공학과 교수, 해양산업연구소

** 한국해양대학교 공과대학 전자통신공학과 박사과정

*** 한국해양대학교 공과대학 전자통신공학과 조교수

접수일자 : 1999년 2월 13일

For the validity evaluation of this paper, message parsing, database, message format and X.25 interface modules are implemented and tested. Based on the result, the capabilities of ship automation system and ownship database access are discussed.

Ⅰ. 서 론

항해하는 선박에 대해서 안전에 관계된 항해 및 기상 정보를 포함한 해사정보의 서비스를 제공하는 전세계 해상 조난 안전 제도(GMDSS : Global Maritime Distress and Safety System)가 1988년 국제 해사 기구(IMO : International Maritime Organization)에서 도입이 가결되어 1999년 2월 1일부터 일정한 선박에 적용되게 되었다[1, 2]. 그 중에서도 장비의 가격이 저렴하고, 소형, 경량의 장비인 국제 해사 위성 기구(INMARSAT : International Maritime Satellite)-C는 양방향의 메시지 통신을 위한 것으로 전화 통신은 할 수 없지만 국제 텔렉스 및 데이터 송수신, 컴퓨터 데이터베이스를 구축할 수 있다[3].

현재 우리 나라의 육상과 선박간 통신은 텔렉스, 공중전화망(PSTN : Public Switched Telephone Network), 공중데이터망(PSDN : Public Switched Data Network)과 같은 육상의 공중망 이용자가 금산지구국을 통하여 선박내의 INMARSAT 단말기와 통신이 이루어지고 있다. INMARSAT의 여러 방식 중 INMARSAT-B의 경우는 실시간 전송이 가능하기 때문에 선박에서 육상지구국을 통하여 전화나 인터넷과 같은 데이터 전송에도 이용될 수가 있지만, 통신사용료가 워낙 비싸기 때문에 특별한 경우를 제외하고는 거의 사용되지 않는 실정이다. 따라서 상대적으로 통신사용료가 저렴한 축적 전송 방식을 사용하는 INMARSAT-C가 일반적으로 많이 사용되고 있는데, INMARSAT-C의 통신형태로는 육상에서 금산지구국을 경유하여 선박과 직접 통신을 하는 일반서비스(one stage service) 방식과 공중데이터망을 이용하여 금산지구국에 접속한 후 제공되는 서비스 메뉴를 선택하여 통신이 이루어지는 부가서비스(two stage service) 방식이 있다. 일반서비스는 부가서비스보다 통신비용이 비싸기 때문에 일반적으로 부가서비스 방식이 사용되고 있는데, 부가서비스를 사용할 경우 육상의 사용자는 다이얼업(dial up) 형태로 금산지구국에 접속하여 선박에 메시지를 전송할 수는 있으나, 선박에서 육상의 사용자에게 메시지를 전송하기 위해서는 육상의 사용자가 금산지구국과 전용선으로 연결되어 항상 온라인 상태가 되어 있어야만 가능하게 된다. 따라서 현재 각 해운사에서는 선박에 메시지를 전송할 수는 있으나, 선박으로부터의 메시지 수신은 텔렉스를 이용하고 있는 실정이다[4, 5].

본 논문에서는 이러한 실정을 해결하기 위하여 하나의 통신서버를 금산지구국과 X.25 전용선으로 연결하여 이를 여러 해운사에서 공유하고 선박과 편리하게 메시지를 송수신할 수 있는 인터넷 기반의 메시지 전송 에이전트를 설계, 구현한다. 이와 더불어 향상된 서비스와 송수신한 메시지를 가공하여 보다 효율적인 서비스를 제공하고, 전송되는 메시지, 선박과 해운사에 관련된 정보를 데이터베이스화함으로써 각종 데이터를 효율적으로 관리할 수 있게 한다.

본 논문의 구성은 다음과 같다. II장에서는 전체 시스템 구성 및 에이전트 구성 모듈 등 메시지 전송 에이전트의 설계에 대하여, III장에서는 에이전트의 구현에 대하여 기술하고, IV장에서는 시험 및 고찰에 대하여, 마지막으로 V장에서는 결론 및 향후 연구에 대하여 기술한다.

Ⅱ. 메시지 전송 에이전트의 설계

1. 전체 시스템 구성

그림 1은 전체적인 시스템 구성을 나타낸다. 선박에서 금산지구국까지는 INMARSAT-C 통신을 이용하고, 금산지구국과 에이전트 서버는 X.25 데이터 전용회선으로 연결된다. 이는 현재 금산지구국이 INMARSAT-C를 통해서 가입전신, 데이터통신 서비스를 제공하는데, 이중에서 데이터통신을 위한 통신 시스템이 TCP/IP를 지원하지 않고, X.25만을 지원하고 있기 때문이다. X.25는 광역 통신망

에 이용되는 인터페이스를 지원하는 동시에 음성보다는 데이터를 전송하는데 장점이 있다. 육상의 일반 사용자는 인터넷을 사용할 수 있는 환경이 갖추어져 있다면 인터넷 메일이나 웹을 이용하여 선박과 메시지를 송수신할 수 있고, 인터넷을 사용할 수 없다면 하이텔이나 천리안과 같은 PC 통신을 이용하여 인터넷 메일 형태로 선박과 메시지를 송수신하게 된다.

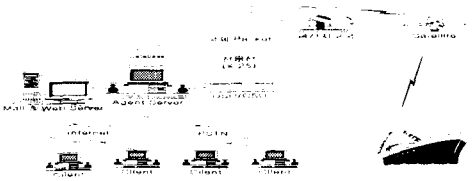


그림 1. 전체 시스템 구성
Fig. 1. Overall system configuration

에이전트를 설계하는데 있어서 고려한 사항은 다음과 같다. 먼저 사용자가 인터넷 메일을 이용하여 메시지를 전송할 경우 메시지의 형식을 준수해야 하는데, 이는 메시지 전송 에이전트에서 인터넷 메일을 파싱(parsing)하고 X.25 형식으로 변환하여 선박으로 전송하도록 하기 위함이다. 이와 더불어 한글코드의 지원, 메시지의 동보전송 지원, 송수신 메시지 및 결과를 데이터베이스에 저장, 메시지 전송에 따른 결과를 사용자에게 통보, 메시지 전송에 관련된 오류나 성공여부 등 상황 기록(log)의 실시간 인쇄 등이 고려되어야 할 중요사항 이다.

2. 에이전트 구성 모듈

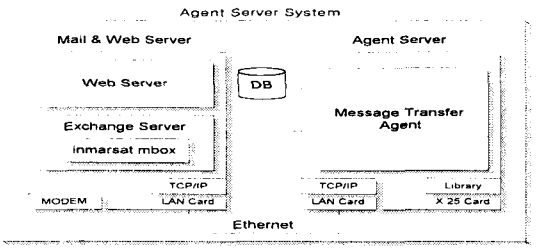


그림 2. 에이전트 서버 시스템 구성
Fig. 2. Agent server system configuration

그림 2는 그림 1에서의 메일 & 웹 서버와 에이전트 서버를 자세하게 나타낸 그림이다. 메일 서버와 에이전트 서버는 이더넷 LAN으로 연결되어 있으며, 에이전트 서버는 금산지구국 서버와 X.25 전용선으로 연결된다. 육상의 사용자가 인터넷 메일을 이용할 경우 선박의 INMARSAT 단말기 ID와 대양코드를 작성하여 메일 서버의 "inmarsat" 메일박스(inmarsat@saraco.co.kr)로 메시지를 전송하면, 에이전트 서버에서는 메시지를 추출하여 파싱하고 X.25 형식의 데이터로 변환하여 선박으로 전송해 주게 된다. 웹을 사용할 경우는 에이전트의 구현을 용이하게 하기 위하여 웹에서 작성된 메시지를 서버에서 자동적으로 "inmarsat" 메일박스로 메시지를 전송하게 함으로써 메일을 이용할 경우의 기능을 그대로 사용하도록 한다. 선박에서 육상의 사용자에게 메시지를 전송할 경우는 메시지 내에 육상 사용자의 인터넷 메일 주소를 작성하면, 에이전트에서 이를 추출하여 인터넷 메일로 육상 사용자에게 전송해 주게 된다.

그림 3은 에이전트 서버의 구성 모듈을 나타내며, 각 모듈의 기능은 다음과 같다.

- Email 송수신 모듈: 육상 사용자가 "inmarsat" 메일박스로 전송한 메시지를 POP(Post Office Protocol)을 사용하여 에이전트 서버로 가져와서 파싱 모듈로 넘겨주는 기능과 선박에서 전송되어 온 메시지를 SMTP(Simple Mail Transfer Protocol)를 사용하여 육상 사용자에게 인터넷 메일로 전송해 주는 기능을 제공한다.
- 파싱 및 변환 모듈: 육상에서 선박으로 메시지를 전송하는 경우는 메시지 내의 INMARSAT 단말기 ID와 대양코드를 파싱하여 메시지 정보

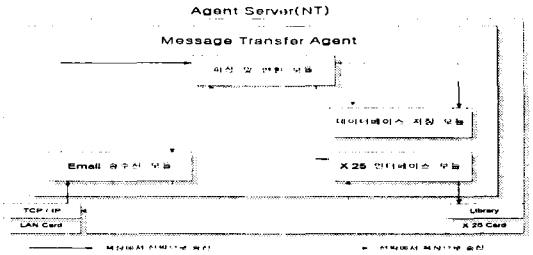


그림 3. 에이전트 구성 모듈
Fig. 3. Agent configuration modules

를 DB에 저장하고 X.25 데이터 형식으로 변환하며, 선박에서 육상으로 전송하는 경우는 메시지 내의 사용자 인터넷 메일 주소를 파싱하여 메시지 정보를 DB에 저장하고 인터넷 메일 형태로 변환하는 기능을 제공한다.

- 데이터베이스 저장 모듈: 송수신한 메시지에 관련된 정보의 저장, 관리 기능을 제공한다.
- X.25 인터페이스 모듈: 금산지구국과의 X.25 연결 및 메시지 송수신 기능을 제공한다.

메시지의 송수신 과정은 다음과 같다. 육상의 사용자가 선박에 메시지를 전송하는 경우 Email 송수신 모듈에서 POP을 이용하여 메일 서버로부터 Email를 받아들이고, 메시지의 <User_ID> 및 <Passwd>를 체크하여 <User_ID>와 <Passwd>가 맞으면 파싱 모듈에서 본문을 추출하여 데이터베이스에 저장한 뒤, X.25 인터페이스 모듈에서 각 선박이동국에 메시지를 전송하고, Email 송신자에게 메시지 전송에 대한 결과를 통보한 후, 메시지 전송 상황을 인쇄한다. 그러나 <User_ID>와 <Passwd>가 맞지 않으면 준비된 오류 메시지를 송신자의 Email로 재전송하고, 선박이동국에는 메시지를 전송하지 않는다. 이 과정이 끝나면 다시 대기 상태로 돌아간다.

선박이동국에서 육상의 이용자에게 메시지를 전송하는 경우에는 선박에서 금산지구국까지는 기존의 INMARSAT-C 통신망을 사용하므로 금산지구국까지 메시지가 전송되며, 금산 지구국에서는 X.25 데이터 전용회선을 통해서 에이전트 서버에 할당된 고유의 ID로 메시지를 송신하게 된다. X.25 인터페이스 모듈에서 받아들인 메시지를 <User_ID>와 <Email 주소>를 확인하여 메시지를 처리한다. 이 때 네 가지의 경우가 발생한다. 첫 번째는 <User_ID>와 <Email> 모두가 지정된 경우인데, 이 경우에는 <User_ID>에 연결된 Email 주소와 <Email>에 지정된 임의의 Email 주소 두 곳으로 메시지를 전송하고, <User_ID>에 지정된 고유의 관리자에게 삐삐호출을 한다. 두 번째는 <User_ID>는 인증되고, <Email>에서 Email은 지정이 되지 않은 경우에는 <User_ID>에 연결된 Email 주소로 메시지를 전송하고, <User_ID>에 지정된 고유의 관리자에게 삐삐호출을 한다. 세 번째는 <User_ID>는 인증되지 않고, <Email>에 지정된 임의의 Email 주소가 지정된 경우에는 <Email>의 지정된 Email 주소로 메시지를 전송한다. 네 번째는 <User_ID>와 <Email>에 Email 주소가 모두 지정되지 않는 경우에는 사용자 가입시 지정된 Email 주소로 메시지를 전송하고, 지정된 고유의 관리자에게 삐삐호출을 한다.

3. 메시지 포맷 설계

그림 4는 에이전트를 이용하여 메시지 송수신을 하기 위한 메시지 포맷을 나타낸다. 육상에서 선박으로 메시지를 전송하는 경우 수신자 주소는 "inmarsat@saraco.co.kr"로 작성하고 메시지 body 부분은 그림 4와 같이 첫째 줄에 <UserID>, 둘째 줄에 <Passwd>를 기입한다. 그 다음 줄부터 "MSG:" 구분자(delimiter) 전까지 메시지를 전송할 선박의 대양코드와 단말기 ID를 기입하면 되며, 여러 줄에 걸쳐 단말기 ID를 작성할 경우 여러 선박에 동보 전송하게 된다. 실제 메시지 내용은 "MSG:" 구분자 다음 줄부터 작성하면 된다.

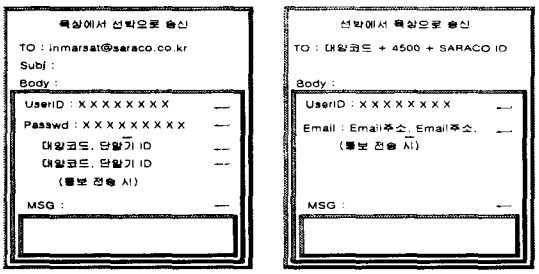


그림 4. 메시지 포맷
Fig. 4. Message format

선박에서 육상으로 메시지를 전송할 경우를 살펴보면 다음과 같다. 선박에서 메시지를 전송할 경우에는 받는 쪽의 주소를 기입한다. 즉 <대양코드> 3자리와 데이터 통신을 위한 <4500> 그 다음에 데이터통신을 위해 부여받은 <고유의 X.25 ID>를 기입한다. Body 부분에 들어가서 첫 번째 줄에 <UserID>, 두 번째 줄에 <Email 주소>를 기입하고, 동보 전송을 원할 경우 여러 <Email 주소>를 ","로 구분하면 된다. 실제 메시지는 "MSG:" 구분자 후에 작성하면 된다.

4. 데이터베이스 설계

가 입 자				
Key	ID	Data Type	Null	Format
P.K	user_ID	char(8)		
	user_name	char(20)		
	passwd	char(9)		*********
	telephone	varchar(20)		
	fax	varchar(20)	V	
	manager	varchar(20)		
	m_telephone	varchar(20)		
	m_e-mail	varchar(40)		
	m_pager	varchar(13)		
	address	varchar(50)		+우편번호

그림 5. 가입자 정보 테이블
Fig. 5. Member information table

그림 5, 6은 메시지를 송수신하는데 있어서 가입자와 선박에 관한 정보를 효율적으로 관리하기 위하여 최소한의 테이터베이스를 설계한 것이다. 이 중에서도 제일 중요한 것은 메시지 송수신 시 사용자 인증을 위한 <user_ID>, <passwd> 필드이다.

그림 7, 8은 송수신한 메시지에 대해서 송신한 메시지를 에이전트 시스템에 저장할 때의 일련의 번호, 경유한 해안지구국, 송신 시간, 메시지 길이, 중계소로부터 받는 메시지 실행 번호, 송수신 선박, 송수신 육상 사용자, 대양코드 등을 관리하는 데이터베이스이다.

5. 웹 서버 설계

웹을 사용할 경우는 에이전트의 구현을 용이하게 하기 위하여 웹에서 작성된 메시지를 서버에서 자동적으로 "inmarsat" 메일박스로 메시지를 전송하게 함으로써 메일을 이용할 경우의 기능을 그대로 사용하도록 한다. 이를 위하여 ASP(Active Server Page) 기법을 이용하여 웹에서 메일 전송을 할 수 있도록 구현한다.

선 박				
Key	ID	Data Type	Null	Format
P.K	ship_ID	char(9)		inmarsat_ number
	ship_name	char(20)		
F.K	user_ID	char(8)		

그림 6. 선박 정보 테이블
Fig. 6. Ship information table

송신 메시지				
Key	ID	Data Type	Null	Format
P.K	send_msg_id	int		9801*****
	LES_num	int		금산,외국
	date_time	datetime		
	msg_length	smallint		최대32K
	msg_ref	int		6자리
	ship_ID	char(9)		
F.K	user_ID	char(8)		
	ocean_code	char(3)		
	send_text	text		

그림 7. 송신 메시지 테이블
Fig. 7. Transmission message table

수신 메시지				
Key	ID	Data Type	Null	Format
P.K	get_msg_id	int		9801*****
	LES_num	int		금산,외국
	date_time	datetime		
	msg_length	smallint		최대32K
	msg_ref	int		6자리
	ship_ID	char(9)		
F.K	user_ID	char(8)		
	Email	varchar(150)		
	get_text	text		

그림 8. 수신 메시지 테이블
Fig. 8. Receiving message table

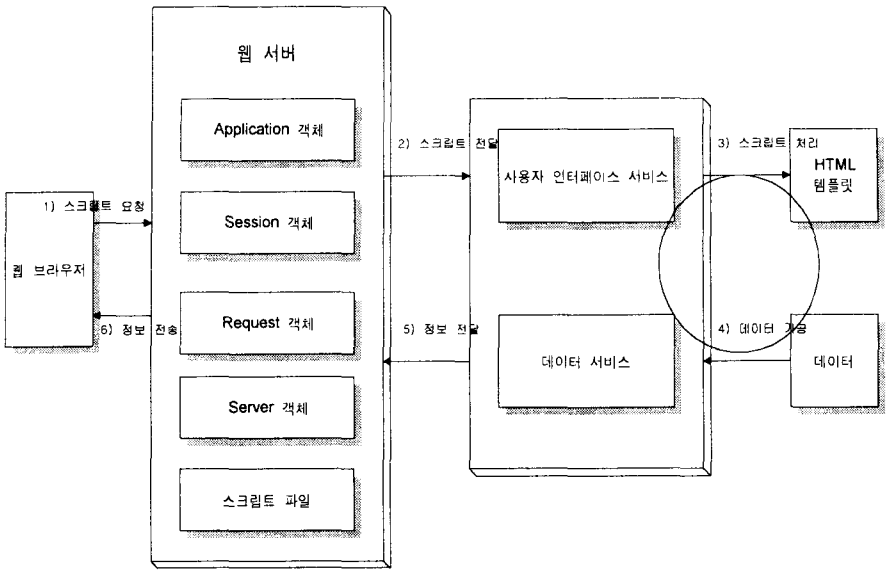


그림 9. ASP 순서도
Fig. 9. ASP flow

ASP는 서버측 스크립트로서 서버에서 실행될 페이지에 스크립트를 추가해서 ASP 어플리케이션을 생성하도록 한다. 그 후에는 <.ASP> 확장자를 가진 HTML(HyperText Markup Language) 문서 파일을 만든다. 왜냐하면, 액티브 서버는 <.ASP> 확장자를 가진 문서만을 파싱할 수 있기 때문이다. 그렇지 않으면 서버는 문서를 파싱하지 않고 단순히 정적인 문서만을 서버로 보낸다. 클라이언트측 어플리케이션은 스크립팅 언어가 COM(Component Object Model) 인터페이스를 지원하기만 하면 어떤 스크립팅 언어를 사용하더라도 생성될 수 있는 것과 유사하게 액티브 서버 페이지도 어떤 스크립팅 언어를 사용하더라도 생성될 수 있다. 그러므로 서버가 <.ASP> 문서에 대한 요청을 받는다면 <.ASP> 파일이 ASP 서버로 전달될 수 있다. 서버에서는 명시되어 있는 포함되어야 할 파일을 추가하고 서버측 스크립트를 실행하고 해당 객체를 생성할 수 있도록 한다. 그 다음에는 스크립트가 실행되고 결과로 생기는 HTML이 표준 HTML과 문자로 출력 버퍼에 쌓이고 일반적인 방법으로 브라우저로 보내지게 된다[6 - 8].

Ⅲ. 메시지 전송 에이전트의 구현

에이전트 시스템을 구현하는데 있어서의 구현 환경은 표 1과 같다.

표 1. 구현 환경
Table 1. Implementation environment

운영체제	MS Windows NT Server 4.0
메일 서버	MS Exchange Server 5.0
웹 서버	MS Internet Information Server 4.0
데이터베이스 서버	MS SQL Server 6.5
개발 툴	Visual Basic, VB script, Java script, ASP

1. Email 송수신 모듈

Email을 POP과 SMTP 프로토콜을 이용하여 송수신하는 과정은 다음과 같다.

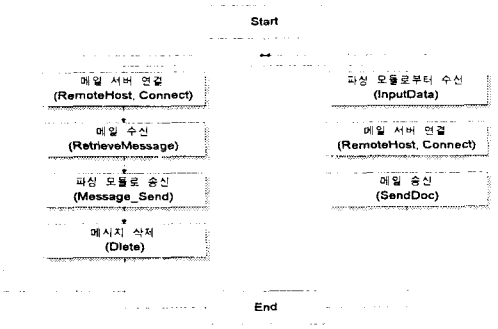


그림 10. Email 송수신 순서도
Fig. 10. Email transmission & receiving flow

에이전트가 구동하게 되면 <saraco.co.kr>의 메일 서버로부터 메일이 도착하였는지 계속해서 확인을 하게된다. 수신된 Email은 parsing 모듈로 전달되고, 전달이 끝난 Email를 삭제한 후, 또 다른 Email이 도착하였는지 확인을 한다. 에이전트를 중단 시키면 연결을 해제하고, 에이전트 구동을 종료한다. Email 송수신 모듈 구현을 위해 ActiveX 컨트롤을 이용한다.

표 2. POP/SMTP 프로퍼티
Table 2. POP/SMTP property

ActiveX Control	프로퍼티	기 능
POP	Busy	명령이 현재 처리중이면 True
	MessageCount	현재 POP 서버에 있는 전자우편의 개수를 저장
	RemoteHost	연결할 POP 서버의 호스트 이름이나 IP 주소
	RemotePort	POP 서버에 연결할 때 사용할 포트 번호
	UserID	POP 서버에 로그인할 때 사용할 사용자 이름
	Password	POP 서버에 로그인하는 데 필요한 패스워드를 저장
SMTP	Busy	명령이 현재 처리중이면 True
	DocInput	데이터 전송과정에 관한 정보를 갖는다
	RemoteHost	연결할 SMTP 서버의 호스트 이름이나 IP 주소
	RemotePort	SMTP 서버에 연결할 때 사용할 포트 번호
	State	SMTP 접속의 현재 상태를 알리는 데 사용

표 2는 POP과 SMTP ActiveX 컨트롤의 주요 프로퍼티로서 사용자의 요구에 맞게 설정하기 위해서 사용된다. 특히 IP 주소와 포트를 필요로 하는데 지정된 포트는 다른 응용 프로그램에 의해 사용되어서는 안된다. 많은 포트들이 이미 표준 포트로 정의되어 있기 때문에 아무 것이나 사용할 수는 없다. 사용이 자유로운 1000번 이상의 포트번호를 사용한다[9, 10].

표 3. POP/SMTP 메소드
Table 3. POP/SMTP method

ActiveX Control	메 소 드	기 능
POP	Authenticate	사용자의 권한을 확인
	Cancel	대기중이던 요청을 중지
	Connect	POP 서버로 접속 요청
	Delete	POP 서버상의 지정된 전자우편 삭제
	GetDoc	URL로 지정된 문서를 읽어오기 위한 요청을 한다
	MessageSize	다음 메시지의 크기를 요구
	Quit	접속을 끊고 Quit 이벤트를 발생시킨다
	Retrieve -Message	인지로 넘겨진 번호에 해당하는 전자우편을 읽어온다
SMTP	Cancel	대기중이던 요청을 중지
	SendDoc	전자우편을 전송

표 3에서 POP ActiveX 컨트롤의 메소드에서 가장 중요한 부분은 <RetrieveMessage>이다. 이 메소드는 POP 서버로부터 Email을 읽어오는데 사용된다. POP ActiveX 컨트롤이 서버로부터 Email을 받는 방법에는 두 가지가 있다. POP 컨트롤의 내장 <DocOutput> 객체를 사용하거나 <Retrieve-Message> 메소드에 <DocOutput> 객체를 인자로 지정하면 된다. SMTP ActiveX 컨트롤의 메소드에서는 <SendDoc>을 사용하는데 있어서 <URL>, <Headers>, <InputData>, <InputFile> 등의 인자를 사용한다.

표 4. POP/SMTP 이벤트
Table 4. POP/SMTP event

ActiveX Control	이 벤 트	기 능
POP	Authenticate	Authenticate 메소드가 호출되면 발생
	Cancel	Cancel 메소드가 실행되면 발생
	DocOutput	컨트롤에 데이터가 도착하면 발생
	Quit	Quit 메소드를 호출한 뒤에 발생
SMTP	Cancel	Cancel 메소드가 실행되면 발생
	DocInput	컨트롤이 데이터를 송신하면 발생
	Error	에러가 생기면 발생
	TimeOut	지정된 시간이 TimeOut 프로퍼티에 설정한 시간내에 일어나지 않으면 발생

표 4와 같이 POP과 SMTP ActiveX 컨트롤은 프로그램이 특정 행동을 취할 수 있도록 하기 위해서 이벤트를 제공한다.

2. 파싱 및 변환 모듈

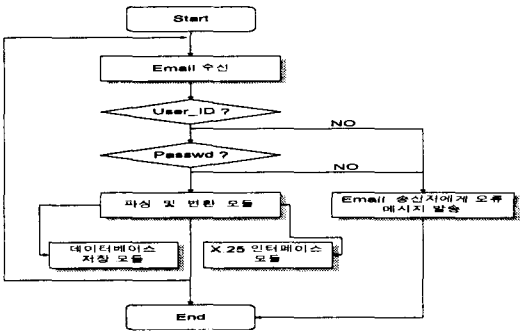


그림 11. 인증 순서도
Fig. 11. Identification flow

Email 송수신 모듈에서 전달된 메시지를 파싱 및 변환 모듈에서는 Email을 분석한다. 그림 4와 같은 포맷으로 사용자가 메시지를 전송하였을 경우, 수신된 메시지를 파싱하게 되는데, 먼저 <User_ID>, <Passwd>를 확인하고, <대양코드>, <선박이동국 번호>, <텍스트> 등으로 파싱하여 데이터베이스에 저장을 한다. 그 후 X.25 인터페이스 모듈로 전달된다.

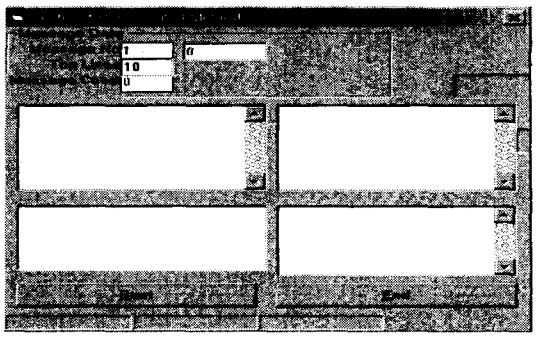


그림 12. 파싱 및 변환 모듈 GUI
Fig. 12. Parsing and conversion GUI

그림 12는 Email 송수신 모듈과 파싱 및 변환 모듈을 비주얼 베이직을 이용하여 구현한 GUI 형태의 결과를 나타내고 있다.

3. 데이터베이스 저장 모듈

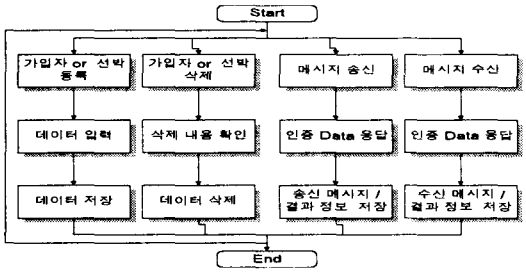


그림 13. 데이터베이스 저장 순서도
Fig. 13. Database store flow

데이터베이스 저장 모듈은 육상의 사용자와 선박에 관련된 각종 데이터 및 육상과 선박 간의 송수신 메시지에 대한 정보를 저장, 관리한다. 육상의 사용자가 에이전트 시스템을 사용하기 위해서는 가입 신청을 해야한다. 그 후 에이전트 시스템 관리자는 육상의 사용자에게 고유의 <User_ID>, <Passwd>를 부여 한다. 또한 육상의 사용자는 가입 시 데이터를 정확히 기입하여야 한다. 그리고 육상

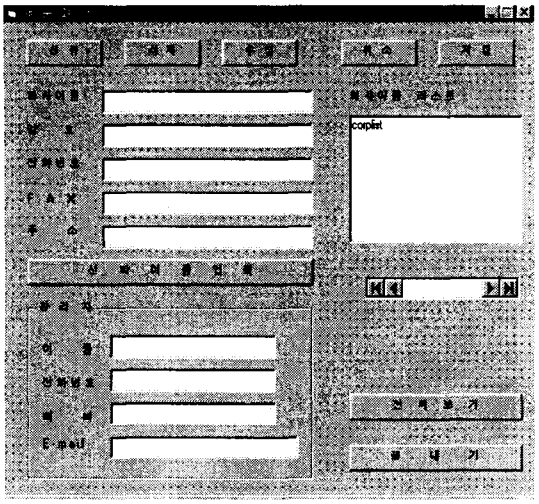


그림 14. 데이터베이스 저장 모듈 GUI
Fig. 14. Database store module GUI

과 선박 간에 메시지를 송수신하는 경우는 사용자에 대한 인증에 필요한 데이터 응답과 송수신된 메시지, 결과에 대한 정보를 일정기간 동안 저장, 관리한다.

하나의 필드라도 적합하지 않으면 에러 메시지를 나타낸다.

그림 14는 데이터베이스 저장 모듈을 비주얼 베이직을 이용하여 구현한 GUI 형태의 결과를 나타내고 있다.

4. X.25 인터페이스 모듈

X.25 인터페이스 모듈은 두 가지 경우로 나누어 볼 수 있다. 첫 번째는 메시지를 육상에서 선박지구국으로 전송할 때 이다. 이 경우는 파싱 및 변환 모듈에서 전달된 메시지를 받아서 금산지구국에 연결을 한 후, 메시지를 전송한다. 그 후 금산지구국으로부터 받아들인 송신 메시지에 대한 결과 정보를 다시 파싱 및 변환 모듈에 전송하고, 금산지구국과 연결을 해제한다. 두 번째는 선박지구국에서 육상으로 전송할 때 이다. 이 경우는 일정한 시간 간격을 두고 금산지구국에 연결을 하고 메시지가 도착하는지 대기 상태로 있는다. 대기 상태에서 메시지가 도착하면 파싱 및 변환 모듈에 수신 메시지

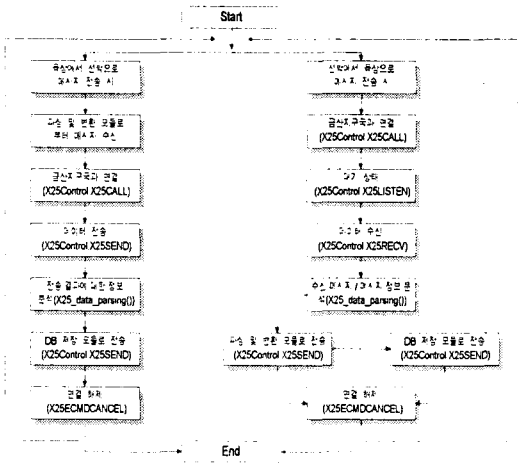


그림 15. X.25 인터페이스 순서도
Fig. 15. X.25 interface flow

와 수신 메시지에 대한 결과 정보를 파싱 및 변환 모듈에 전달하고, 연결을 해제하고, 또 다시 대기 상태로 돌아간다.

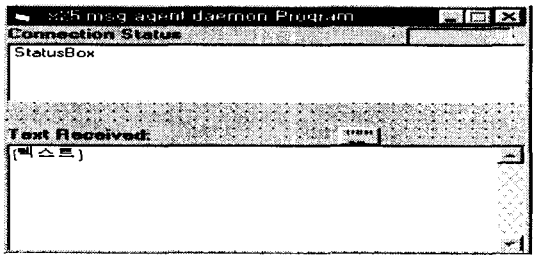
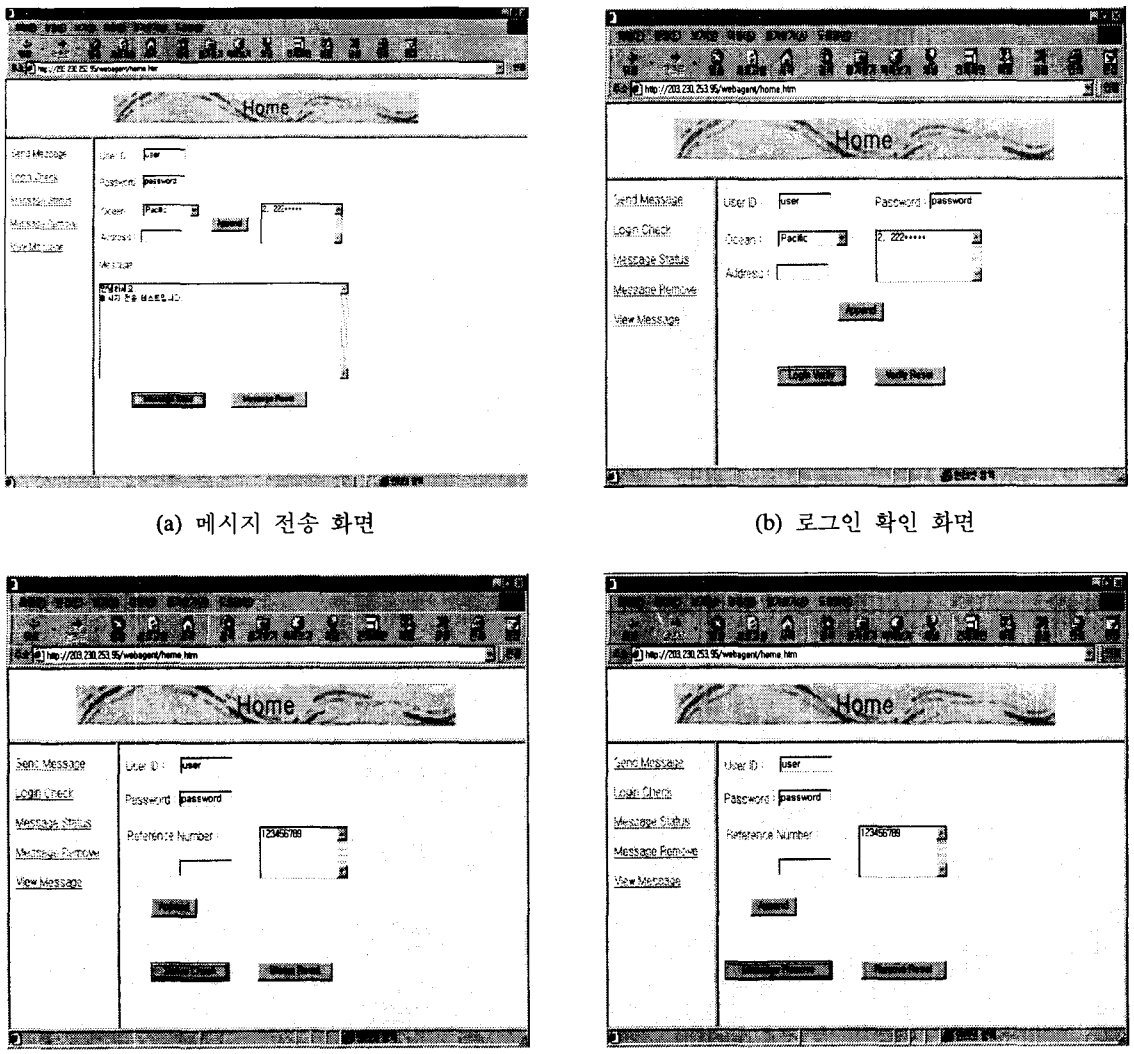


그림 16. X.25 인터페이스 모듈 GUI
Fig. 16. X.25 interface module GUI

에이전트를 구동하면 금산 지구국으로부터 메시지가 전송되어 오는지를 계속적으로 체크하고, 이벤트(event)에 의해 이를 처리하게 된다. 그러므로 육상과 선박 간에 메시지를 전송하기 위해 공통적으로 사용되는 X.25 컨트롤 함수가 있다. 금산지구국에 연결을 하기 위한 컨트롤 함수는 <X25Control.X25 CALL>, 메시지를 전송하기 위한 함수는 <X25Control. SEND>, 전송한 메시지에 대한 각종 정보를 분석하는 함수는 <X25_data_parsing>, 메시지를 수신하기 위한 함수는 <X25Control.RECV>, 금산지구국과 연결 해제를 위한 함수는 <X25ECMDCANCEL> 이다.

5. 웹 서버 모듈



(a) 메시지 전송 화면

(b) 로그인 확인 화면

(c) 메시지 상태 검사 화면

(d) 메시지 취소 화면

그림 17. 웹 서버 모듈 GUI
Fig. 17. Web server module GUI

웹은 인터넷을 전송매체로 사용하는 응용프로그램으로서 그림 17의 (a)는 메시지를 전송하는 기능으로 <user id>, <password>, <ocean>, <address>, <append>, <message> 항목을 기입한 후 <message send> 버튼을 누르면 된다. (b)는 선박의 INMARSAT-C 단말기가 로그인 상태인지를 확인하는 기능이다. (c)는 사용자가 보낸 메시지가 목적지에 도착하였는지 여부를 검사하는 기능이다. (d)는 사용자가 보낸 메시지를 취소 할 수 있는 기능이다. 동작 과정을 좀 더 살펴보면, 웹에서 작성된 메시지를 전송하면 에

이전트 서버에서 자동적으로 "inmarsat" 메일박스로 메시지를 전송하게 함으로써 인터넷 메일을 이용할 경우의 기능을 그대로 사용하도록 한다. 그리고 사용자에게는 보다 동적인 GUI 형태를 제공한다.

Ⅳ. 시험 및 고찰

1. 시험 결과

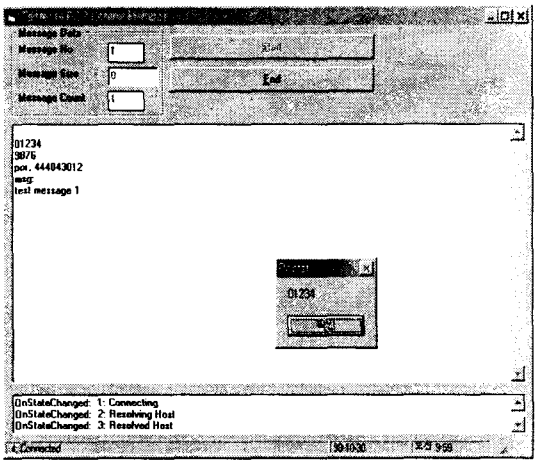


그림 18. 파싱 결과
Fig. 18. Parsing result

그림 18은 육상에서 선박으로 메시지를 전송한 결과를 나타내고 있다. 즉, 에이전트의 Email 송수신 모듈에서 POP을 사용하여 메일을 받아들인다. 그 후 파싱 및 변환 모듈에서 메시지 내의 내용을 INMARSAT 단말기 ID, 대양코드, 실제 데이터를 파싱하여 메시지 정보를 DB에 저장하고 X.25 데이터 형식으로 변환하여 X.25 인터페이스 모듈을 통해서 금산지구국으로 데이터 전송이 성공적으로 이루어졌음을 보이고 있다.

2. 고 찰

본 논문에서는 동작 시험을 위하여 육상 사용자 측면에서는 인터넷망과 PSTN망을 지원하는 통신망을 이용하고, 메시지 전송과 데이터베이스 관리를 위한 에이전트 서버와 MS SQL 서버를 탑재한 데이터베이스 서버를 구성하였다. Windows 95 운영체제를 사용하는 클라이언트에서는 인터넷과 전화접속 네트워킹을 이용하여 에이전트 서버에 Email을 전송함으로써 육상과 선박간에 메시지 전송을 하였다.

외국의 경우는 국제 해사 위성 기구에서 INMARSAT-C를 통해 Email 서비스를 제공하고 있다. 이 경우는 <Thrane & Thrane Capst>, <Trimble Galaxy>, <JRC> 장비를 지원하고 있으며[11, 12], 이를 정리하면 표 5와 같다.

이 경우에도 선박과 육상지구국간 인터넷 Email을 전송하기 위해서 데이터 통신에는 X.25를 이용하고, 일정한 Email 주소 포맷을 가지고 있다는 점에서 공통점이 있다. 그러나 이 시스템은 7bit를 제

표 5. 서비스 제공자
Table 5. Service provider

서 비 스 제 공 자	Shore-to-Ship Address "imn=Inmarsat Mobile Number	인터넷 주소 구 분 자	예
BT	imn@csat.bt.com or imn@satmail.bt.com	TO+	TO+ Maritime_Applications@inmarsat.org
COMSAT	imn@c-link.net	TO:	TO: Maritime_Applications@inmarsat.org
T-MOBIL	imn@inmarsat.de or imn@shamrock.de	TO:	TO: Maritime_Applications@inmarsat.org
STATION 12	imn@station-12.unisource.nl	TO:	TO: Maritime_Applications@inmarsat.org
STRATOS	imn@stratosmobile.net	INET:	INET: Maritime_Applications@inmarsat.org
TELSTRA	imn@telstra.ves.net	INET:	INET: Maritime_Applications@inmarsat.org

공하기 때문에 영문자만을 지원하고, 한글은 지원되지 않는다.

본 논문의 결과, 선박내 자동화 시스템을 웹에서 응용 프로그램을 구동시킬 수 있는 자바(java) 언어로 구현한다면 웹 브라우저를 이용한 선박 자동화 시스템의 제어가 가능할 것이다. 더불어서 선박과 육상지구국간의 Email 서비스를 위한 국제적인 일정한 표준이 마련되어야 할 것이다.

V. 결 론

산업이 발달하면서 해상 물동량이 증가하므로 해상에서의 선박 사고도 늘어나게 되었다. 국제 해사기구에서는 선박 사고에 대비한 더욱 효율적인 조난 안전 제도를 갖추고자 오랫동안 검토하여 전세계 해상 조난 안전 제도라는 새로운 해상 이동 통신 방식을 도입하게 되었다. 또한 육상에서는 정보통신의 발달로 TCP/IP 프로토콜을 기반으로 한 인터넷망의 기하급수적인 증가와 더불어 분산 컴퓨팅과 데이터의 공유 및 효율적인 관리가 대두되고 있다.

우리나라는 세계 조선 수주량 1, 2위의 위치를 점유하고 있으면서도, 해양 장비의 개발과 효율적인 사용이 이루어지지 못하고 있는 실정이다.

본 논문에서는 육상과 선박간 INMARSAT-C를 통한 인터넷 기반의 메시지 전송 에이전트를 구현하는데 있다. 이를 위하여 선박과 육상간 통신을 위한 전체적인 시스템 구성 및 데이터베이스 연동과 X.25 데이터 통신을 위한 인터페이스 모델을 제시하였다. 아울러 본 논문의 타당성 검토를 위하여 마이크로소프트 Windows NT 서버와 Windows 95 클라이언트의 연결을 이용자에게 보다 편리하고 저렴한 사용환경을 제공하기 위해서 인터넷망과 PSTN망을 이용할 수 있는 모델을 제시하고, 데이터 관리를 위한 데이터베이스 설계 및 구축, 인터넷 Email 기반의 메시지 전송 에이전트 개발을 통하여 송수신 메시지를 관리자에게 효율적으로 통보할 수 있다. 또한, 국내 INMARSAT-C 단말기 개발, 생산, 판매 촉진과 더불어 국내 금산 지구국의 사용 증대를 실현할 수 있게 하였다. 이 결과를 토대로 선박내 데이터베이스 접근과 더불어 웹 브라우저를 이용한 선박 자동화 시스템의 구현 가능성을 논하게 되었다.

추후 연구로는 선박과 육상간 통신 기능과 더불어 선박 자동화 기능을 제공할 수 있는 플랫폼의 개발과 더불어 웹 브라우저를 이용하여 원격으로 선박 자동화 시스템을 제어할 수 있는 방법이 추진되어야 할 것이다[13].

참고문헌

[1] 배정철, 선박용 구난 통신 시스템 개발에 관한 연구, 석사학위논문, 해양대, 1992.

[2] International Maritime Organization, GMDSS Handbook, 1992.

[3] Laurie Telteley & David Calcutt, Understanding GMDSS, pp. 178-189, 1994.

[4] 한국통신 마케팅본부, 인말세트 통신 서비스 업무처리 지침, 1997. 10.

[5] 한국통신 금산위성지구국, 해사통신안내, 1996. 6.

[6] 이원하 외7명, 한글윈도우 NT에서의 ASP 활용, 사이버출판사, pp. 19-22, 1999. 1.

[7] 김덕겸 역, VBScript로 웹 페이지 개발하기, 한빛미디어, pp. 637-649, 1998. 3.

[8] 이승준, 전주예 편저, 이것이 Visual Basic 6, 삼각형프레스, pp. 1211-1231, 1998. 10.

[9] 한기용 역, 특별한 사용자를 위한 ActiveX, 대림출판사, p251 - 273, 1997. 2.

[10] 주경민, 박성완, 정동길 공저, Visual Basic Programming Bible Ver 5, 영진출판사, p795 - p846, 1998. 1.

[11] International Maritime Satellite Organization, INMARSAT, 1998

[12] http://www.inmarsat.org/c-guide, INMARSAT-C를 통한 Email 서비스

[13] Marine Managent Systems, INC., ISIT Platform, 1997

朴 沇 植(Yeoun-Sik Park)

1971년 2월: 광운대학교 무선통신공학과 졸업(공학사)

1980년 2월: 건국대학교 행정대학원 통신행정학과 졸업(행정학석사)

1995년 8월: 경상대학교 대학원 전자계산학과 졸업(공학석사)

1996년 3월 ~ 현재: 한국해양대학교 대학원 전자통신공학과 박사과정

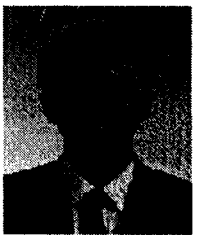
1979년 3월 ~ 현재: 경상대학교 해양과학대학 정보통신공학과 교수 / 해양산업연구소

李 泰 吾(Tae-Oh Lee)

1997년 2월: 한국해양대학교 전자통신공학과 졸업(공학사)

1999년 2월: 한국해양대학교 대학원 전자통신공학과 졸업(공학석사)

1999년 2월 ~ 현재: 한국해양대학교 대학원 전자통신공학과(박사과정)



林 宰 弘(Jae-Hong Yim)

1986년 2월: 서강대학교 전자공학과 졸업(공학사)

1988년 8월: 한양대학교 대학원 전자공학과 졸업(공학석사)

1995년 2월: 한양대학교 대학원 전자공학과 졸업(공학박사)

1995년 3월 ~ 1997년 2월: 한국해양대학교 전자통신공학과 전임강사

1997년 3월 ~ 현재: 한국해양대학교 전자통신공학과 조교수

* 관심분야: 컴퓨터네트워크, 분산 컴퓨팅, 그룹웨어