



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

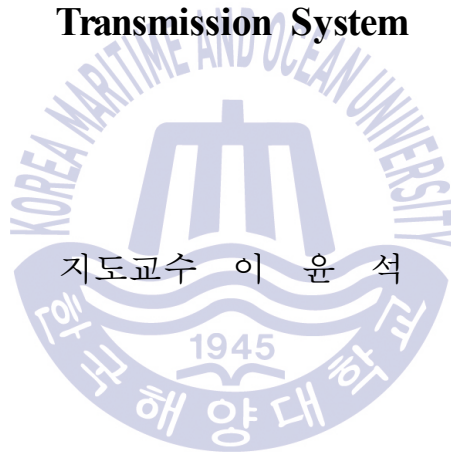
이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

工學碩士 學位論文

운항 정보전송 시스템 개발에 대한 연구

A Study on the Development of Ship Operation Data Transmission System



2018년 6월

한국해양대학교 대학원

해양경찰학과

김 종 필

본 논문을 김종필의 공학석사 학위논문으로 인준함

위원장 이 은 방 (인)

위 원 국 승 기 (인)

위 원 이 윤 석 (인)

2018년 6월

한국해양대학교 대학원

목 차

List of Tables	iii
List of Figures	iv
Abstract	vi
Nomenclature	x
1. 서 론	
1.1 연구의 배경 및 목적	1
1.2 연구의 범위와 방법	4
1.3 논문의 구성	4
1.4 연구의 절차	5
2. 기술동향 분석	
2.1 e-Navigation	6
2.1.1 e-Navigation 정의	6
2.1.2 e-Navigation 아키텍처	7
2.1.3 MSP(Marine Service Portfolios)	8
2.2 자율운항선박의 육상 관제	9
2.2.1 자율운항선박의 정의	9
2.2.2 육상관제 센터	9
2.3 선박 데이터 수집	11

2.3.1 NMEA0183 Sentence	13
2.3.2 Modbus	18
2.4 Kafka 메시징 시스템	25
2.4.1 MOM- Kafka 개념	26
2.4.2 MOM - Kafka 특징	28
2.5 다중이용자에 의한 메세지 전달 방법	30
3. 운항 정보전송 시스템 설계	
3.1 개발 방향	33
3.2 성능 구현	35
3.3 데이터 수집과 시스템	36
3.3.1 하드웨어 데이터 수집 장비 개발	36
3.3.2 소프트웨어 데이터 수집 장비 개발	39
3.4 데이터 프로세싱 및 Web UI 구현	47
4. 운항 정보전송 시스템 구현	
4.1 단위 및 통합 성능 시험	52
4.2 실험 및 결과	56
4.2.1 단위 기능 실험 및 평가	56
4.2.2 통합 기능 및 성능 실험	60
5. 결론	63
참고문헌	66

List of Tables

Table 2-1 Structure of NMEA Sentence	13
Table 2-2 Navigation NMEA data Sentence	16
Table 2-3 Frame format of Modbus RTU	22
Table 2-4 Modbus RTU Object & Function Codes	23
Table 2-5 Interface design for Read Discrete Inputs	24
Table 2-6 Messaging process speed of distribution application service program	25
Table 3-1 An example of Navigation equipment Output Data Boudrate	37
Table 3-2 Data collection communication type(RS 422)	37
Table 3-3 An example of Definition of Modbus for AMS Channel List	38
Table 3-4 An example of Definition of Modbus for AMS Channel List	40
Table 3-5 Interface description of data collection system	41
Table 4-1 H/W environment for ship's experiment	54
Table 4-2 Experiment environment for communication test	55
Table 3-5 Sample of single unit experiment	56
Table 3-5 Sample of integrated experiment	60

List of Figures

Fig. 1-1 Key element technology of Shipping 4.0	2
Fig. 1-2 Development of autonomous navigation technology	3
Fig. 1-3 Flow Chart	5
Fig. 2-1 e-Navigation architecture	7
Fig. 2-2 Example of shore control system	10
Fig. 2-3 Concept of shore control system	11
Fig. 2-4 Ship's network for Kongsberg	12
Fig. 2-5 An example of NMEA Sentence	14
Fig. 2-6 Navigation network for JRC	15
Fig. 2-7 Diagram for navigation NMEA data collection	15
Fig. 2-8 Navigation NMEA data collection cables and collectors	17
Fig. 2-9 Diagram of data collection	18
Fig. 2-10 Engine Room network for Kongsberg	19
Fig. 2-11 Hierarchincal struction of Modbus	20
Fig. 2-12 Cofiguration of Modbus network	21
Fig. 2-13 Kafka composition element	26
Fig. 2-14 Kafka Cluster	27
Fig. 2-15 TPS of Kafka producer/consumer	29
Fig. 2-16 Composition of WebSocket	31
Fig. 2-17 Waiting time comparison of Polling and WebSocket	32

Fig. 3-1 System definition	33
Fig. 3-2 System architecture	34
Fig. 3-3 An example of Output Data Sentences	36
Fig. 3-4 An example of Definition of Modbus for AMS Channel	
List	38
Fig. 3-5 NMEA Parsing	39
Fig. 3-6 An example of database AIS(Other Ship) Data storage	47
Fig. 3-7 UI for navigation data monitoring	49
Fig. 3-8 Monitoring based on electronic chart	50
Fig. 3-9 M/E, G/E Monitoring	51
Fig. 4-1 Experiment preparation procedure	53
Fig. 4-2 System Experiment environment setup	54
Fig. 4-3 Results of data transmission test	61

A Study on the Development of Ship Operation Data Transmission System

Kim Jong Pil

Department of Coast Guard Studies graduate School of

Korea Maritime and Ocean University



Abstract

Autonomous vessels are becoming a hot topic in the global shipping industry. In the EU, Autonomous ship technology is rapidly developing. Autonomous vessels are expected to become a new paradigm to change and accelerate the digitization, platform connectivity and intelligence of the shipping industry. Over the past few years, the marine industry has maintained its competitive edge by securing operating competitiveness by reducing operating costs such

as vessel size and oil costs. However, as with other industries, ICBM (Industry 4.0) IOT, Cloud, Big Data, Mobile), cyber security, simulation, etc., and to create new business by maximizing the unique capabilities of the shipping industry as defined in Shipping 4.0. Until recently, autonomous vessels were named as various types such as Smart Ship, Digital Ship, Connected Ship, Remote Ship, Unmanned Ship and Autonomous Ship. They were defined as MASS (Maritime Autonomous Surface Ship) by International Maritime Organization (IMO) This is a comprehensive product and service for the unmanned ship, autonomous transportation and efficient transportation in the stepwise upgrade for safety, reliability, and efficiency in ship operation.

In this paper, the research background and research scope is described in chapter 1, discussion about autonomous ship related contents are in chapter 2. The design and implementation of a large scale ship data transmission system including method of ship communication, data protocol and messaging processing technology to transmit and display large amount of data are describe in Section 3. The configuration and results of the experimental environment are in Section 4. Then finally the conclusion and further studies are described.

Chapter 2 - Autonomous vessels are that incorporate ship control systems and communication technologies that enable wireless monitoring and control, including improved decision support systems and remote control and autonomous navigation capabilities. The control function for the autonomous vessels is implemented in the Shore Control Center (SCC). In order to accurately recognize the situation of the ship at the remote land control center, the existing systems such as AIS and RADAR are equipped with thermal cameras, LIDAR an advanced sensor module is added to the vessel to replace the existing sailing company's watch keeping operations with the

landlord and system. The land control center simultaneously monitors the safe operation of autonomous vessels, manages the voyage plan update and autonomous navigation system, and manages the maintenance plan for autonomous vessels.

Chapter 3 - In this paper, we constructed and implemented a large capacity ship data acquisition and transmission system that can collect, process, and transmit data generated from ships. The required physical configuration is largely divided into onboard and onshore, and by design, it can have n ships and n land. On the ship side, data are collected from navigation equipment and AMS (Alarm Monitoring System) such as RADAR, Speed Log, Doppler, Gyro Compass and GPS, processed and stored for use on land and onboard service applications. The land consists of a server composed of modules capable of receiving data from various ships and transmitting them to a required place after processing, and servers for service application. The user is connected to the web browser on the PC and configured to use services such as flight information monitoring.

Chapter 4 - Experiments were conducted on the large scale ship data acquisition and transmission system implemented in this paper by constructing an environment in the T/V Hanbada Lab. For each unit function, the experiment proceeded while the Kongsberg simulator and the training line were at anchor, and the integrated test for the stability and function of the data was conducted during the long voyage.

Chapter 5 - In this study, the research was carried out in order to process the ship's navigation data generated from various equipment for several number of vessels in terms of ship operation data and remote control center technology for this purpose. For the research on application system

implementation and verify evaluation, an analysis of a data collection and process on ship and shoreside and design of data transmission has been studied. Also for this research, the basic research knowledge were achieved together with research results on ship data processing and transmission. The newly developed web based system can used to monitor ship' s data in real-time on the shoreside.



Nomenclature

AAWA	: The Advanced Autonomous Waterborne Application Initiative
AIS	: Automatic Identification System
AMQP	: Advanced Message Queuing Protocol
AMS	: Alarm and Monitoring System
API	: Application Program Interface
ASICII	: American Standard Code for Information Interchange
AWS	: AMAZON Web Service
CRC	: Cyclic redundancy check
ECDIS	: Electronic Chart Display and Information System
ENC	: Electronic Navigation Chart
EU	: European Union
GPS	: Global Position System
HDT	: True Heading
HTD	: Heading/Track Control Data
HTML	: Hypertext Markup Language
HTTP	: Hypertext Transfer Protocol
IaaS	: Infrastructure as a Service
IAS	: Intergrated Alarm System
ICBM	: IoT, Cloud, Big Data, Mobile

ICT	: Information and Communication Technology
IEC	: International Electrotechnical Commission
IoT	: Internet of Things
ISO	: International Organization for Standardization
JMS	: Java Message Service
LRC	: longitudinal redundancy check
LTE	: Long Term Evolution
MASS	: Maritime Autonomous Surface Ship
MOM	: Message Oriented Middleware
NMEA	: The National Marine Electronics Association
OSI	: open systems interconnection
RTU	: Remote Terminal Unit
RADAR	: Radio detecting and ranging
RPM	: Revolutions per minute
SCADA	: Supervisory Control And Data Acquisition
TCP/IP	: Transmission Control Protocol/Internet Protocol
TPS	: Throughput Per Second
UDT	: User Datagram Protocol,
UI	: User Interface
WAS	: Web Application Server
VSAT	: Very Small Aperture Terminal

제 1 장 서 론

1.1 연구의 배경 및 목적

자율운항선박(MASS, Maritime Autonomous Surface Ship)이 글로벌 해운업계에 화두로 떠오른 가운데, EU 지역에선 관련 움직임이 점차 속도를 내고 있다. 자율운항선박은 해운산업을 변화시키는 새로운 패러다임이 될 것이며 해운 산업의 디지털화, 플랫폼 연결화 및 지능화를 가속화 시킬 것으로 예측하고 있다 (Lavander 2017).

지난 몇 년간 해운산업은 선박의 초 대형화 및 유류비 등 선사의 운영비용 절감을 통한 영업 경쟁력을 확보하여 저 시황에서의 생존을 지속해 왔지만, 타 산업과 마찬가지로 4차 산업혁명(Industry 4.0)을 통해 ICBM (IoT, Cloud, Big Data, Mobile), 사이버 보안, 시뮬레이션 등 다양한 기술과 해운 산업의 고유의 역량을 극대화 시켜 새로운 비즈니스를 창출하려는 변화가 고유의 업무에 대한 체질 변화를 시도하고 있으며 이를 Fig. 1-1과 같이 Shipping 4.0으로 정의하고 있다. Shipping 4.0은 4차 산업혁명의 요소 기술과 굉장히 유사하지만, 선박 내 특유의 거친 환경에서의 통신망 구축, 선박 육상 간 통신, e-Navigation 시스템의 한 분야로서의 선박의 역할 수행 등 ‘Internet of Things at Sea’와 ‘Internet of Services at Sea, 즉 해양/해운 산업에 특화된 요소 기술이 다수 포함되어 있으며, 이를 통합적으로 제품으로 표현하면 자율운항선박이라고 볼 수 있다 (Rødseth *et al.*, 2016).

자율운항선박은 최근까지 Smart Ship, Digital Ship, Connected Ship, Remote Ship, Unmanned Ship, Autonomous Ship 등 다양한 형태로 명명되다가 국제해사기구(IMO)에 의해 MASS(Maritime Autonomous Surface Ship)으로 정의되었으며, 단순히 자율운항 할 수 있는 선박을 만드는 것이 아니라, 선박 운영에서의 안전성(Safety), 신뢰성(Reliability), 효율성(Efficiency)을 위한 단계적 고도화 안에서 선박의 무인화, 자율화 및 운송의 효율화를 위한 종합적인 제품 및 서비스라고 볼 수 있다 (Adams, 2014; Oltedal *et al.*, 2018).



Fig. 1-1 Key element technology of Shipping 4.0

자율운항 선박 시대는 완전한 자율 운항 선박만으로 한정하진 않는다. 규제와 통신환경과 같은 도전적인 과제가 산재하기 때문에, Rolls-Royce가 중심이 되는 AAWA(The Advanced Autonomous Waterborne Application Initiative) 프로젝트를 비롯하여 자율운항을 준비하는 프로젝트에서는 상용화(확산) 단계를 Remotely operated local vessel → Remote controlled unmanned coastal vessel → Autonomous unmanned ocean-going ship으로 설정하고 각 단계 선박 별로 개발 및 검증을 진행하고 있다 (Lolyds Register, 2016; NFAS, 2017).

자율운항선박의 운항 레벨은 매뉴얼 운항 레벨(Level 0)부터 완전 자율운항 레벨(Level 6)으로 구분된다 (Lolyds Register, 2016).

이러한 단계적 운항 레벨을 달성하기 위해 이와 관련된 핵심 기술에 대한 연구가 필요한 시점이다. 자율운항 선박에는 Fig. 1-2 와 같이 선박 내의 기술

도 필요하지만 선박에서 데이터를 수집하여 육상에서 분석 처리하는 원격관제 기술이 요구되고 이러한 데이터를 바탕으로 선박의 운영 최적화(Operation Optimization), 성능 관리(Condition Management) 등이 가능하도록 정의하고 있다 (Rolls-Royce, 2016; Burmeister *et al.*, 2014)

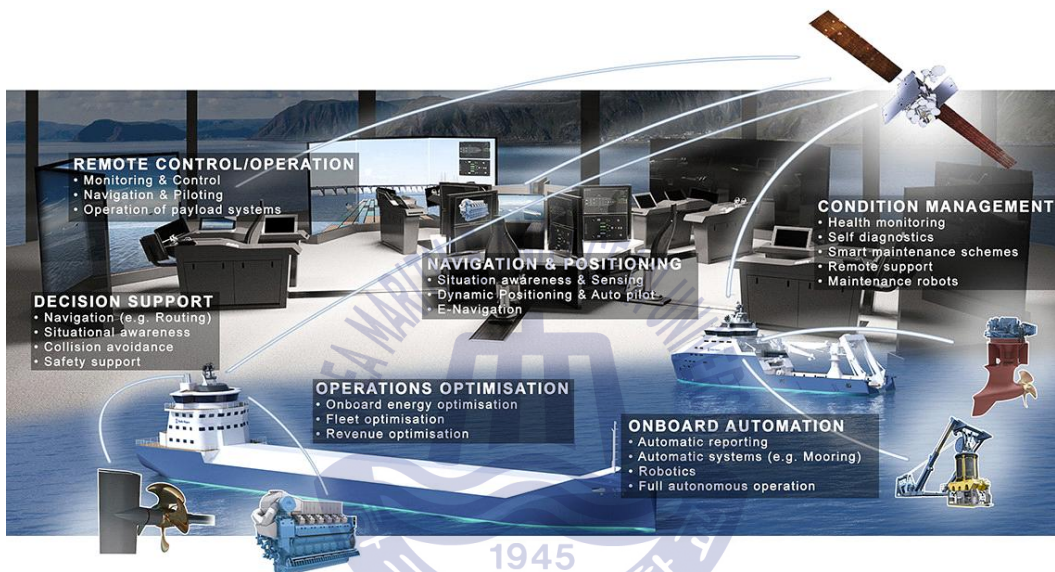


Fig. 1-2 Development of autonomous ship technology

원격 관제(Remote Operation/Control)를 하기 위해서는 선박 내 데이터 수집 및 처리, 수집된 데이터의 육상 전송, 육상에서의 데이터 공유 등이 최적화 되어 처리되어야 한다. 그동안의 연구는 선박 데이터 통합 및 처리, 선박 자동화 시스템 등 단일 선박에서의 데이터 처리 및 활용 관점에서 진행되어 왔으나, 선박을 원격에서 관제하기 위해서는 선박의 운항 데이터를 본선에서 수집/처리 하고 육상에서는 수신한 다수의 선박 데이터를 최적으로 처리할 수 있는 시스템과 이를 활용할 수 있는 서비스 시스템에 대한 연구가 필요하다.

1.2 연구의 범위와 방법

원격 관제를 위한 시스템은 데이터를 수집/처리하는 선박과 이러한 데이터를 취급하는 육상시스템으로 나뉘며, 각 환경에 따라 특성이 존재한다. 따라서 본 연구에서는 먼저 선박과 육상에서의 데이터 수집 및 처리 환경을 분석하고, 분석한 결과를 바탕으로 원격에서 관제 가능한 IoT (Internet of Thing) 및 ICT (Information and Communication Technology) 기술의 적용 여부를 확인한 후, 시스템에 적용하였다.

선박 운항 정보 수집 및 전송 시스템은, 먼저 NMEA0183과 Modbus를 활용하여 데이터가 수집되도록 설계 하였고, 그 응용(Application)으로 다량의 메시지를 처리하기 위해 Kafka와 WebSocket을 사용하는 것으로 구현하였다. 또한, 선박과 육상 간 Web Service를 구현하여 Web Browser로 모니터링 할 수 있도록 하였다.

따라서 본 논문에서는 선박 및 육상에서의 데이터 수집 및 처리에 대한 분석, 데이터 전송에 관한 설계, 이를 이용한 응용으로 요약 할 수 있다.

1.3 논문의 구성

본 논문은 1장에서 연구 배경 및 연구 범위를 알아보고, 2장에서는 자율운항 선박 관련 내용, 선박 내 통신 및 데이터 프로토콜, 운항 데이터를 전송 및 전시(Display)하도록 하는 메시징 처리 기술에 대하여 알아보았다. 3장에서는 선박 운항정보 시스템에 대한 설계 및 구현에 대하여 설명하였으며, 4장에서는 실험 환경 구성 및 결과를 나타내었고, 5장에서는 본 연구의 결론을 도출하였다.

연구의 절차는 Fig. 1-3과 같은 절차 순으로 진행을 하였고 진행 과정 중에 문제점에 대해서는 수정 및 보안을 통해서 진행하였다.

1.4 연구의 절차

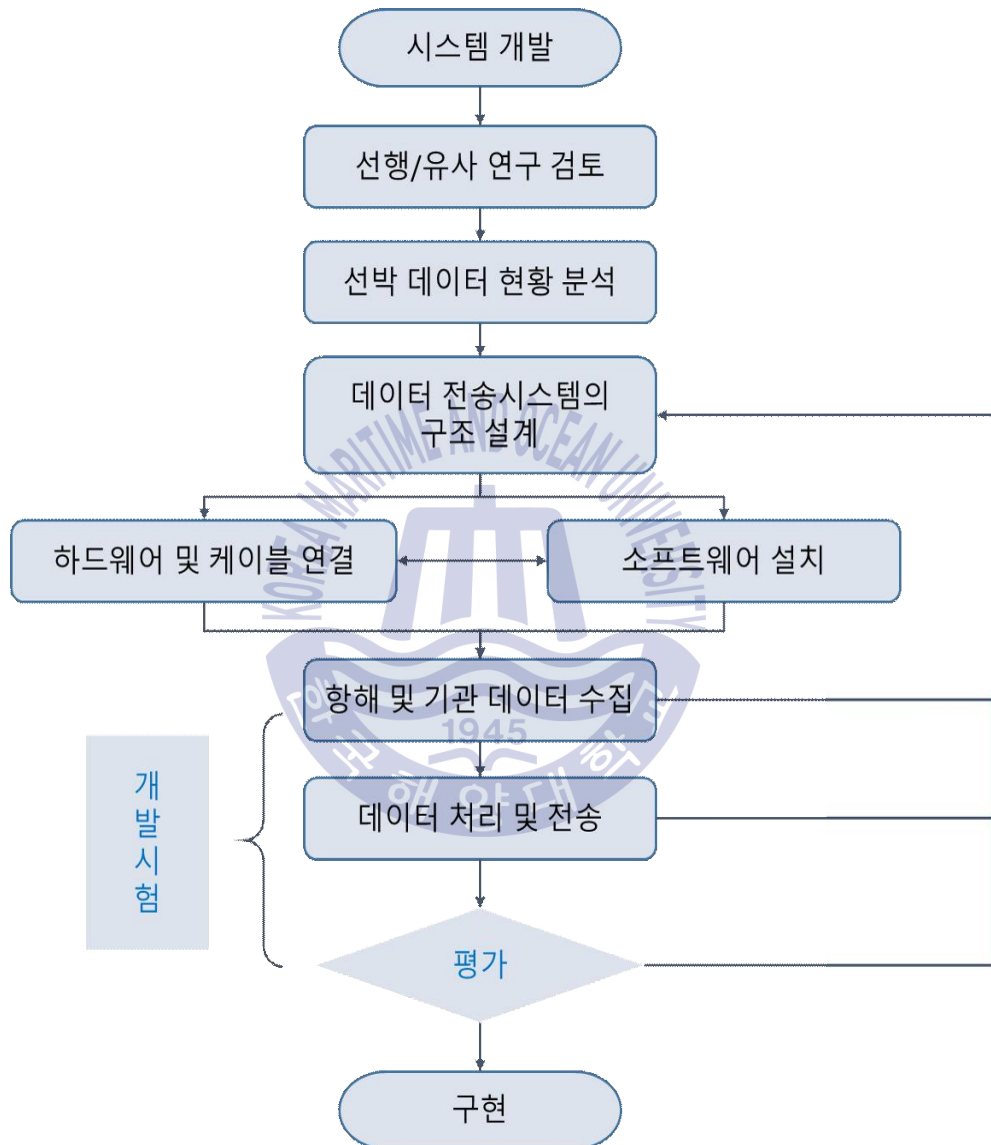


Fig. 1-3 Flow Chart

제 2 장 기술 동향 분석

2.1 자율운항선박의 육상 관제

2.1 e-Navigation

e-Navigation은 선박 운항 관리 체계에 ICT(Information and Communication Technology) 융복합 기술을 접목한 디지털 해상 교통안전 관리 체계로 육상관제센터로의 보고, 상황 인식에 대한 효과적인 표출, 각 이해관계자간의 데이터 교환 등 다양한 관점에서 활용되고 연계될 수 있다.

2.1.1 e-Navigation 정의

e-Navigation이란 “e-Navigation is the harmonized collection, integration, exchange, presentation and analysis of maritime information onboard and shore by electronic means to enhance berth to berth navigation and related services, for safety and security at sea and protection of the marine environment” 이다. NAV 53차 13번 의제(2007년 4월 20일) e-navigation 전략에서 e-Navigation의 목적은 안전항해, 선박의 보안 및 해양환경 보호로 정의되었으며, 이를 위해 선박이 부두와 부두를 항해할 때 항해와 항해에 관련한 각종 서비스를 증강시켜야 한다. 또한 이러한 서비스를 증강시키기 위하여 필요한 방법으로써, 육상과 선박의 해상정보를 수집하고, 통합하고, 변환하고, 표현하고, 분석하는 일치된 방법이 필요하다. 선박이 항해하면서 발생하는 데이터는 시간적으로 수시로 변하는 실시간 정보이며 선박 또는 육상 서비스 기관에서 알고 있는 정보는 서로 일치되지 않음으로써 발생하는 문제는 상당한 파급을 가져올 수 있다. 따라서 모든 정보는 일관성을 가지고 있어야 한다 (IMO, 2017; Rødseth *et al.*, 2014; Rødseth 2014)

또한 서비스를 제공하기 위하여 필요한 장비는 정보를 수집하는 장비(각종 항해에 필요한 센서와 시스템), 데이터를 변환하는 장치, 표현하는 시스템 및

정보를 분석하는 시스템 등으로 나누어 볼 수 있다. 이제까지의 정보 분석은 전문적인 항해사 및 기관사 등 사람에 의해 이루어 졌으나 앞으로는 정보화됨으로서 IT와 융합하여 많은 정보를 활용하여 보다 과학적이고 통계적으로 분석하여 항해사와 기관사가 의사 결정하는데 보조역할을 하는 장비가 필요하다.

2.1.2 e-Navigation 아키텍처

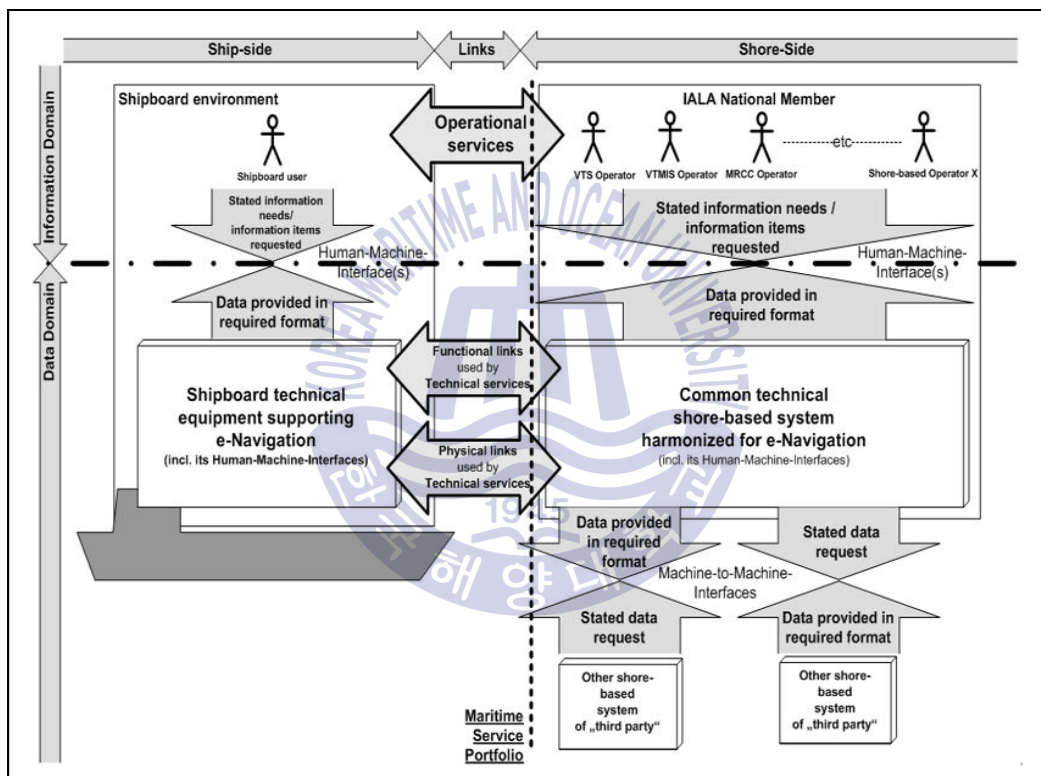


Fig. 2-1 e-Navigation architecture

포괄적인 e-Navigation 구조의 표현은 위 Fig. 2-1과 같이 선상 엔티티, 물리적인 링크와 육상 엔티티로 구성된다. Fig. 2-1에서 왼쪽은 선박의 환경이며 오른쪽은 육상의 환경이다. 선박과 육상의 환경은 e-Navigation 시스템 구조에서의 서비스 수준, 기술적 서비스에 의해 사용되는 기능적 링크, 기술적 서비스에 의해 사용되는 물리적 링크를 통해 연결된다.

e-Navigation 개념에서 선박은 미래에 그들에게 데이터를 공급하거나 수신할 수 있는 선상의 전자장비와 관련된 환경을 구축하고, 육상에서는 e-Navigation 사용자에게 필요로 하는 지원을 하는 서비스로 VTS 등 선박에 지원할 수 있는 어플리케이션을 통해 서비스를 제공한다. 그리고 육상의 서비스를 물리적 링크로 통해 해상에 제공한다. 육상과 선박의 서비스는 물리적 링크를 통해 육상 어플리케이션과 해상의 어플리케이션이 상호적으로 연결되어 제공한다.

2.1.3 MSP(Marine Service Portfolios)

e-Navigation 이행 정책에서는 항만 (Harbour operations), 연근해 및 혼잡 또는 제한 수역, 대양 횡단 항해 (Trans ocean voyages), 해양 (Offshore operations), 극지방 및 원양지역과 관련한 MSP (Marine Service Portfolios)를 2018년까지 개발하도록 계획하였다. MSP는 주어진 해역에서 제공되는 서비스 수준별 집합을 의미하는 것으로 서로 다른 운영과 서로 다른 지역에서의 서비스 요건이 사용자별로 달라질 수 있음에 대응하여 위해 고안된 개념이다. MSP는 정의된 운영 지역에서 정보 및 통신 서비스의 필요성과 함께 전 세계 해상 서비스와 나란히 하기 위한 서비스이다.

현재 선박에 공급되는 다양한 IT 장비들 덕분에 현대화된 선박운항 환경이 제공되는 반면 선박운항 인원이 감소되는 이유이기도 하여, 과중한 업무 부담과 함께 다양하고 복잡한 장비 사용에 대한 운항 자들의 어려움을 유발하고 있다. 그로 인해 운항에 따른 단순 반복 업무 및 입출항 시의 항만 보고업무 등의 자동화가 필요하며 이는 e-Navigation이 항해/안전을 위한 새로운 장비도입뿐 아니라 관련 서비스 또한 중요하다.

e-Navigation의 국제 표준화는 현재 진행형이며, MSP 성격상 공급자보다는 사용자 위주의 서비스로서, 조선 기자재 업체 입장에서는 선박에 공급되는 장비 및 시스템들에 해당 서비스 구현이 가능하도록 표준화된 프로토콜을 사용하는 것이 주된 변화요인이 될 것이다. 그리고 MSP 서비스를 실현하려는데 있어, 선박 내 다양한 정보의 통합은 매우 중요하며, 이를 이용해 사용자의 다양한 요구에 부응하는 새로운 부가서비스 시장을 창출할 수 있을 것으로 기대한다.

2.2.1 자율운항선박 정의

자율운항선박은 모델화된 제어 시스템과 무선 모니터링 및 제어를 가능하게 하는 통신 기술을 탑재하고, 개선된 결정 지원 시스템과 원격 조종 및 자율운항 능력을 포함하는 선박을 의미한다.

자율운항선박에 대한 관제 기능은 육상 관제 센터 (SCC, Shore Control Center)에서 실행되며 원격에 위치한 육상 관제 센터에서 정확하게 선박의 상황을 인지하기 위해 AIS, RADAR와 같은 기존 시스템에 열화상 카메라, 라이다 (LIDAR) 등 기존 항해사의 견시(Watch keeping) 업무를 육상 운영자와 시스템에서 대신 할 수 있도록 Advanced Sensor Module이 선박에 추가되어 활용된다. 육상 관제 센터에서는 자율운항 선박의 안전 운항을 동시에 모니터링하며 항해 계획 업데이트 및 자율운항 시스템을 관리하고, 자율운항 선박에 대한 유지 보수 계획을 담당한다 (Burmeister *et al.*, 2014; Rødseth, 2014).

2.2.2 육상 관제 센터

육상 관제 센터에서 항해에 관련된 전반적인 상황을 제어 및 모니터링 (Autonomous Navigation System)하고 기관에 관련된 전반적인 상황을 제어 및 모니터링 (Autonomous Engine Monitoring and Control)한다. 본선에서의 센서 데이터와 사전에 프로그래밍된 항해 매개 변수와의 비교를 기반으로 자율운항 시스템은 항해를 결정하고 해당 항해 및 기관 데이터를 육상 관제센터로 전송한다. 육상관제센터는 Fig. 2-2와 Fig. 2-3과 같이 자율운항선박과 관련된 정보를 파악하고, 이해 관계자에게 공유한다 (Rødseth, 2014; Rolls-Royce, 2016)

또한 기상 예보, 교통 밀집도 등 자율운항선박의 효과적인 항해를 위해 요구되는 정보를 육상관제시스템에서 수집 및 처리하여 자율운항선박과 이해관계자에 전달 및 공유한다. 이를 이용하여 선박 안전 항해에 적용할 수 있다. 육상에서는 선박의 효율적인 운항을 위하여 육상 운항사와 전문 기술 요원이 운항을 원활히 할 수 있는 능력을 갖추기 위해서는 전문적인 교육을 받고 능력을 갖추어야 한다.



Fig. 2-2 Example of shore control system

육상 관제 센터는 여러 선박을 관리할 수 있고, 법적으로 추가 검토가 필요하겠지만 하나의 선박에 대해 목적에 따라 여러 육상 관제 센터가 대상 선박을 관제할 수 있다. 또한 육상관제센터가 반드시 육상에 있을 필요는 없다 (Blanke, Henriquez and Bang, 2017).

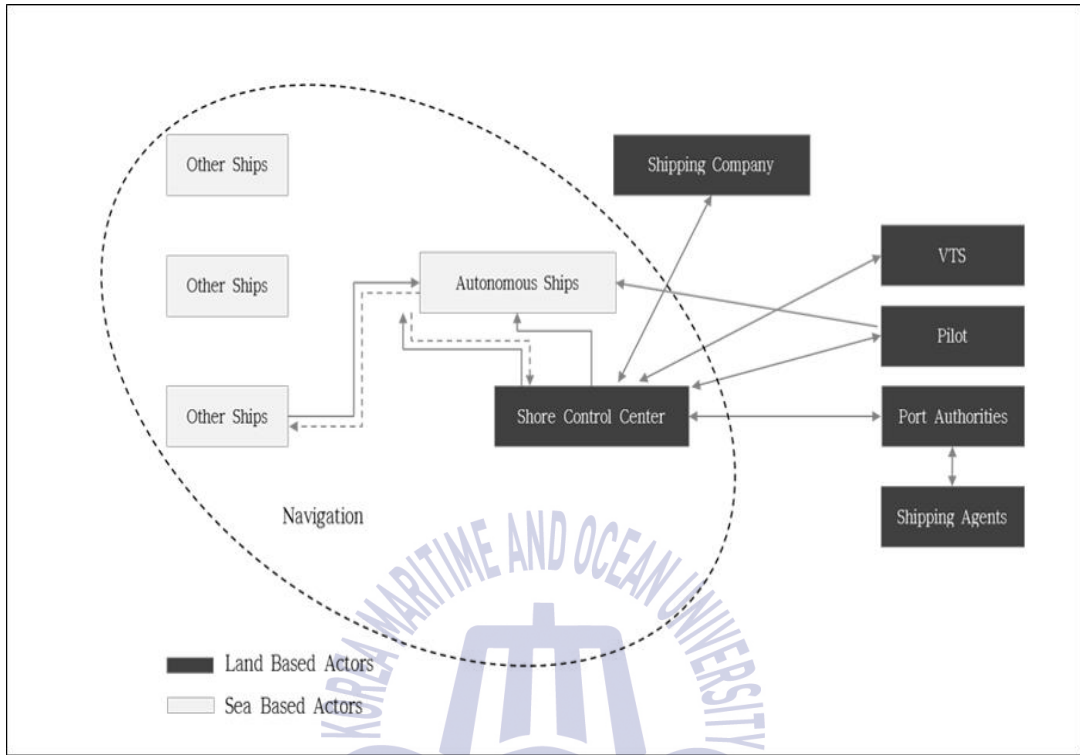


Fig. 2-3 Concept of shore control system

2.3 선박 데이터 수집

자율운항선박은 Fig. 2-4와 같이 선박은 네트워크로 연결되어 항해, 기관, 선체, 화물 등의 다양한 장비로부터 다양한 형태로 데이터를 공유하고 수집할 수 있으나 항해 장비는 표준에 의해 NMEA 0183 (IEC61162-1, 2)로 수집되고, 다양한 기관 장비로부터는 아날로그 신호를 디지털 신호로 전환하여 Modbus로 전환하거나 AMS(Alarm Monitoring System), IAS(Integrated Automation System)과 같은 기기 정보를 통합해서 관리하는 장비로부터 수집할 수 있다. 이들 장비는 분석 결과 일반적으로 Alarm List를 Tag로 관리하고 Modbus를 활용하여 인터페이스를 할 수 있고, 항통장비에 비해 기관 장비의 데이터 인터페이스는 비표

준이지만, 시장에서 주요 공급사가 항통장비에 비해 소수이기 때문에 연구적인 측면에서는 현재 사용 중인 프로토콜을 분석하여 설계에 반영하였다. 또한 Kongsberg와 같은 일부 제조사는 NMEA를 통해 Alarm, RPM 등 주요 정보를 인터페이스 할 수 있도록 지원한다. 하지만 AMS와 IAS는 전송해야 하는 데이터 종류가 수백 종이기 때문에 NMEA 사용하게 되면 NMEA 전송 특성 상 전체 데이터를 초 단위로 전송하는 것은 제한적이다.

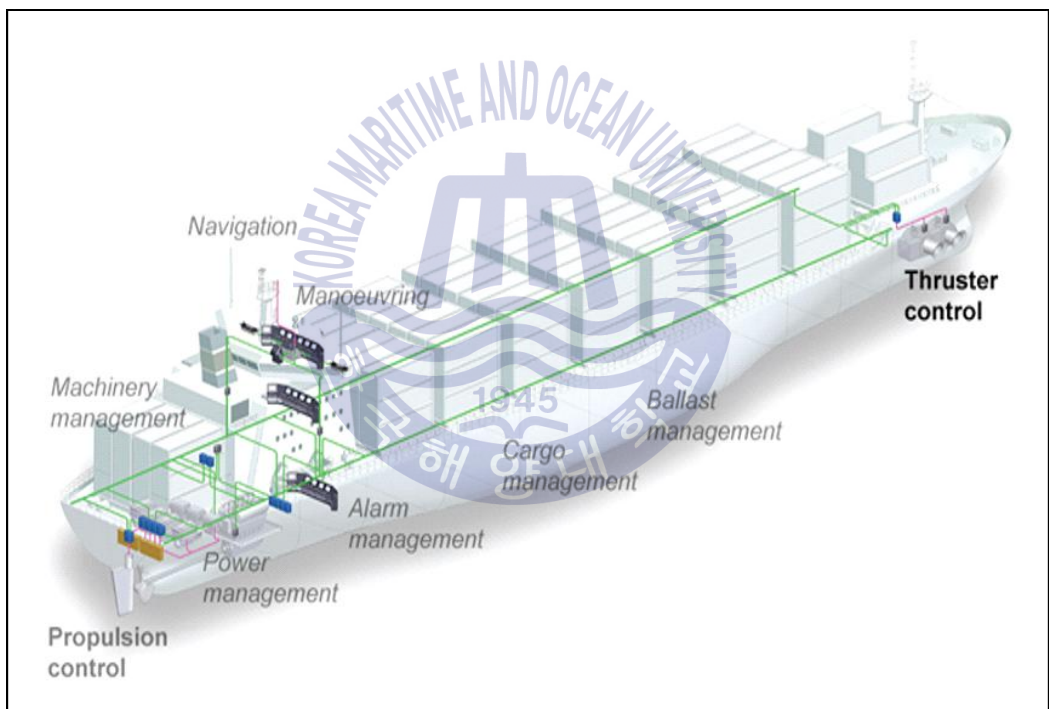


Fig. 2-4 Ship's network

2.3.1 NMEA0183 Sentence

NMEA0183 프로토콜은 1983년 3월에 처음 출시되었으며, 선박의 전자장비는 프로토콜에 의해 하나의 송신기(Talker) 또는 수신기(Listener)로 지정된다. 이는 시간, 위치, 방위 등의 정보를 전송하기 위한 선박용 규격으로써 미국의 The National Marine Electronics Association에서 정의했다 (NMEA 0183, 2002)

ECDIS, GYRO Compass, Speed Log, Doppler Log 등 항해 장비는 NMEA0183 표준에 의해 데이터를 송신 및 수신한다.

NMEA 0183은 물리 계층, 데이터링크 계층, 애플리케이션 계층 레이어로 구성되어 있다. 물리 계층은 RS-232, RS-422등의 전기적인 전송 규격을 뜻하며, 데이터링크 계층은 Baud Rate, Data Bit, Parity Bit, Stop Bit 등을 정해 놓는다.

애플리케이션 레이어는 데이터를 전송하는 Sentence에 대한 규약이다.

NMEA 구조는 Table 2-1 과 같이 Sentence를 구성한다. Sentence의 시작과 끝을 지정하고 송신 문장의 형식, 즉 수신자(Listener)가 ROT(Rate of turn)으로 해석하는지, HDT(Heading, true)로 해석해야 하는지를 알려주고 데이터 필드에 각 문장의 형식에 맞게 값을 넣어서 송신한다.

Table 2-1 Structure of NMEA Sentence

필드	설명
“\$” 또는 “!”	Sentence 시작
<주소 필드>	Talker ID, Sentence 형식
[“,”<데이터 필드>] - [“,”<데이터 필드>]	0개 이상의 데이터 필드
“*”<체크섬 필드>	Check Sum
<CR><LF>	Sentence 종료

NMEA Sentence 구조에 따라 ECDIS가 Heading/Track Control Data를 수신자에게 송신하는 HTD Sentence 구조이며 Fig. 2-5와 같이 HTD Sentence의 데이터 필드에는 조타 모드와 커맨드 정보들을 담아 송신한다.

본 연구에서는 실험 대상으로부터 환경으로부터 Fig. 2-6과 Fig. 2-7같이 항해/통신 장비로부터 데이터 연계 대상을 정의하고, Table 2-2와 같이 수신할 NMEA Sentence를 각 장비별로 정의하였다.

실험 대상이 자율운항 선박이 아닌 현재 장비를 기준으로 설계를 한 것이기 때문에, 육상관제에서 요구되는 모든 정보를 연계할 수는 없지만 NMEA Sentence 대상이 추가 되었을 때 확장 가능하도록 DB Table을 설계하였다. 또한 데이터 중복 처리를 위해 여러 장비에서 동일한 Sentence는 주었을 때 수신자는 Talker ID와 상관없이 NMEA 형식(Formatter)을 기준으로 수집하고 처리하도록 설계하였다.

HTD - Heading/Track control data																
\$**HTD,A,x.x,a,a,x.x,x.x,x.x,x.x,x.x,x.x,x.x,a,A,A,A,x.x*hh<CR><LF>																
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1. Override, A = in use, V = not in use 2. Commanded rudder angle, degrees 3. Commanded rudder direction, L/R = port/starboard 4. Selected steering mode 5. Turn mode R = radius controlled T = turn rate controlled N = turn is not controlled 6. Commanded rudder limit, degrees (unsigned) 7. Commanded off-heading limit, degrees (unsigned) 8. Commanded radius of turn for heading changes, n.miles 9. Commanded rate of turn to heading changes, deg/min 10. Commanded heading-to-steer, degrees 11. Commanded off-track limit, n.miles (unsigned) 12. Commanded track, degrees 13. Heading reference in use, T/M 14. Rudder status (A = within limits, V = limit reached or exceeded) 15. Off-heading status (A = within limits, V = limit reached or exceeded) 16. Off-track status (A = within limits, V = limit reached or exceeded) 17. Vessel heading, degrees																

Fig. 2-5 An example of NMEA Sentence

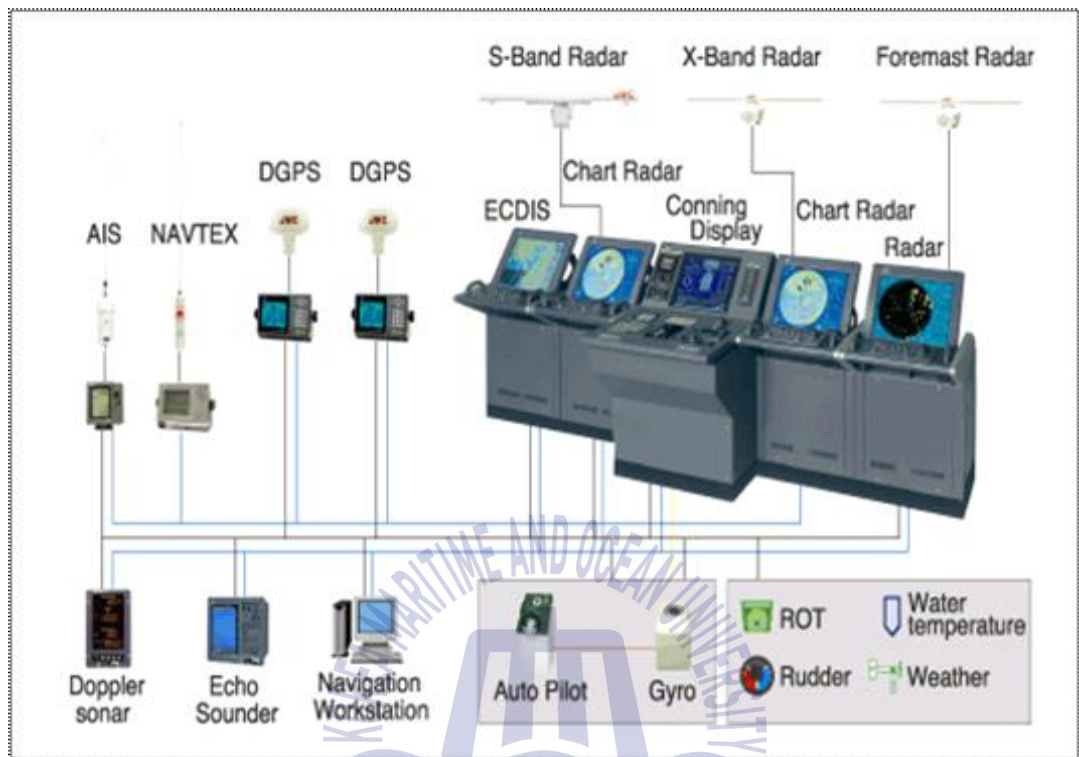


Fig. 2-6 Navigation network for JRC

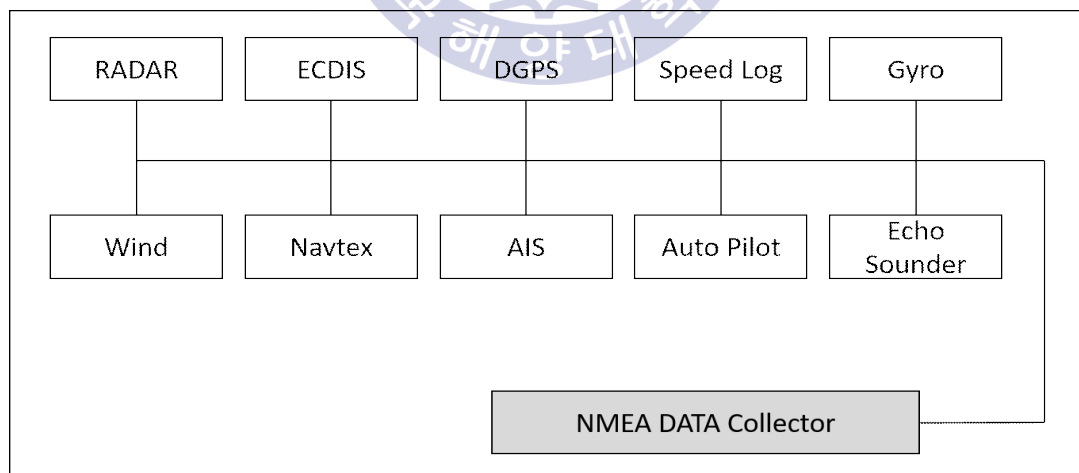


Fig. 2-7 Diagram for navigation NMEA data collection

Table 2-2 Navigation NMEA data sentence

항해 장비	형식	Sentence
Gyro	HDT	\$--HDT, x.x, T*hh<CR><LF>
	ROT	\$--ROT, x.x, A*hh<CR><LF>
DGPS	GGA	\$--GGA, hhmmss.ss, llll.ll, a, yyyyy.yy, a, x, xx, x.x, x.x, M, x.x, M, x.x, xxxx*hh<CR><LF>
	DTM	\$--DTM,ccc,a,x.x,a,x.x,a, x.x,ccc*hh<CR><LF>
	RMC	\$--DTM,ccc,a,x.x,a,x.x,a, x.x,ccc*hh<CR><LF>
	ZDA	\$--ZDA, hhmmss.ss, xx, xx, xxxx, xx, xx*hh<CR><LF>
Auto Pilot	HTD	\$--HTD,A,x.x,a,a,a,x.x,x.x,x.x,x.x,x.x,x.x,a,A, A, A,x.x,*hh<CR><LF>
	HTC	\$--HTC,A,x.x,a,a,a,x.x,x.x,x.x,x.x,x.x,x.x, x.x,a*hh<CR><LF>
	RSA	\$--RSA, x.x, A, x.x, A*hh<CR><LF>
Speed Log	VBW	\$--VBW, x.x, x.x, A, x.x, x.x, A, x.x, A, x.x, A*hh<CR><LF>
NAVTEX	NRX	\$--NRX, xxx, xxx, xx, aaxx, x, hhmmss,ss, xx, xx, xxxx,x.x,x.x,A, c--c*hh<CR><LF>
RADAR	TTM	\$--TTM, xx, x.x, x.x, a, x.x, x.x, a, x.x, x.x, a, c--c, a, a, hhmmss.ss, a*hh<CR><LF>
ECDIS	RTE	\$--RTE, x.x, x.x, a, c--c, c--c, c--c*hh<CR><LF>
	WPL	\$--WPL, llll.ll, a, yyyyy.yy, a, c--c*hh<CR><LF>
Wind	MWV	\$--MWV, x.x, a, x.x, a, A *hh<CR><LF>
Magnetic	HDG	\$--HDG, x.x, x.x, a, x.x, a*hh<CR><LF>

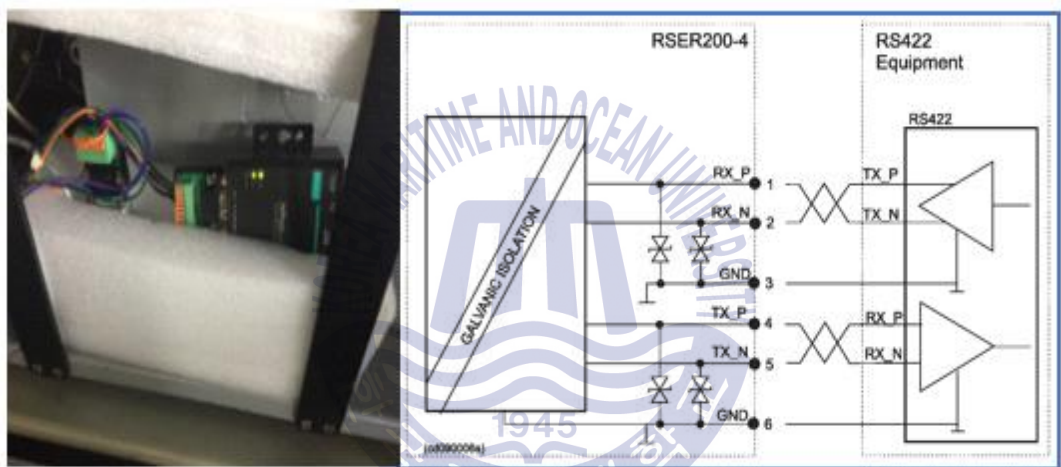


Fig. 2-8 Navigation NMEA data collection cables and collectors

2.3.2 Modbus

Modbus는 1979년 미국 Modicon에서 네트워크상에 연결된 장비들 간의 클라이언트/서버 통신을 위한 응용 계층 메시지 전달 프로토콜로 Serial, UDP 등 모든 형식의 네트워크 구조 내에서 손쉽게 데이터 통신을 지원할 수 있는 프로토콜로 개발하였다 (Modbus-IDA, 2004).

Modbus는 다양한 데이터 통신을 지원하며 사양이 공개된 무료 개방형 Ethernet으로 정보의 조작성이 용이한 단순한 구조로써, 현재 자동화 시스템 및 SCADA(Supervisory Control And Data Acquisition, 원격 감시제어) 시스템 네트워크 Fig. 2-9와 Fig. 2-10에서 대표적인 표준 프로토콜로 인식되고 있으며, ISO15745-4, IEC/PAS 62030로 인증된 국제 표준이다. 선박에서는 NMEA0183과 같이 표준은 아니지만 AMS, IAS와 같은 모니터링 및 제어 시스템에서는 일반적으로 많이 사용하고 있다.

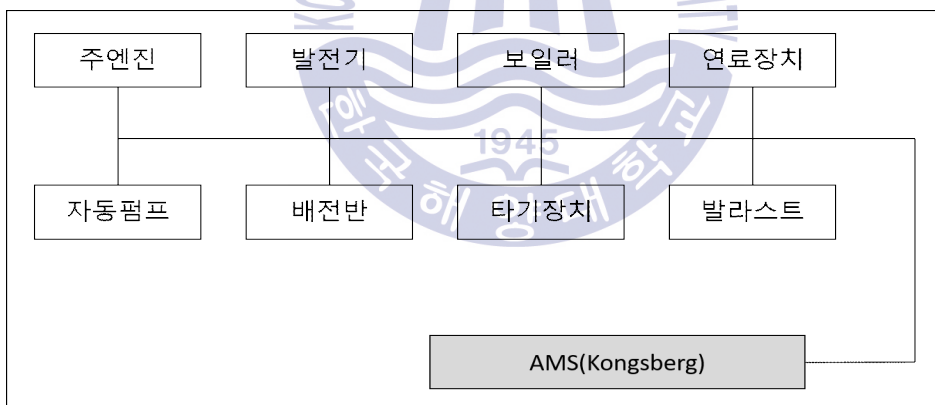


Fig. 2-9 Diagram of data collection

통신의 구성 체계를 이해하는데 가장 일반적으로 OSI 7 Layer 구성도를 사용한다. Modbus는 OSI 7 Layer 중 Application Level에 속한 프로토콜이며, Fig. 2-11과같이 크게 Modbus Serial, Modbus Plus, Modbus TCP/IP로 구분되어진 계층 구조를 가지고 있다 (MODBUS ORG., 2012(a); Gjersest, 2017).

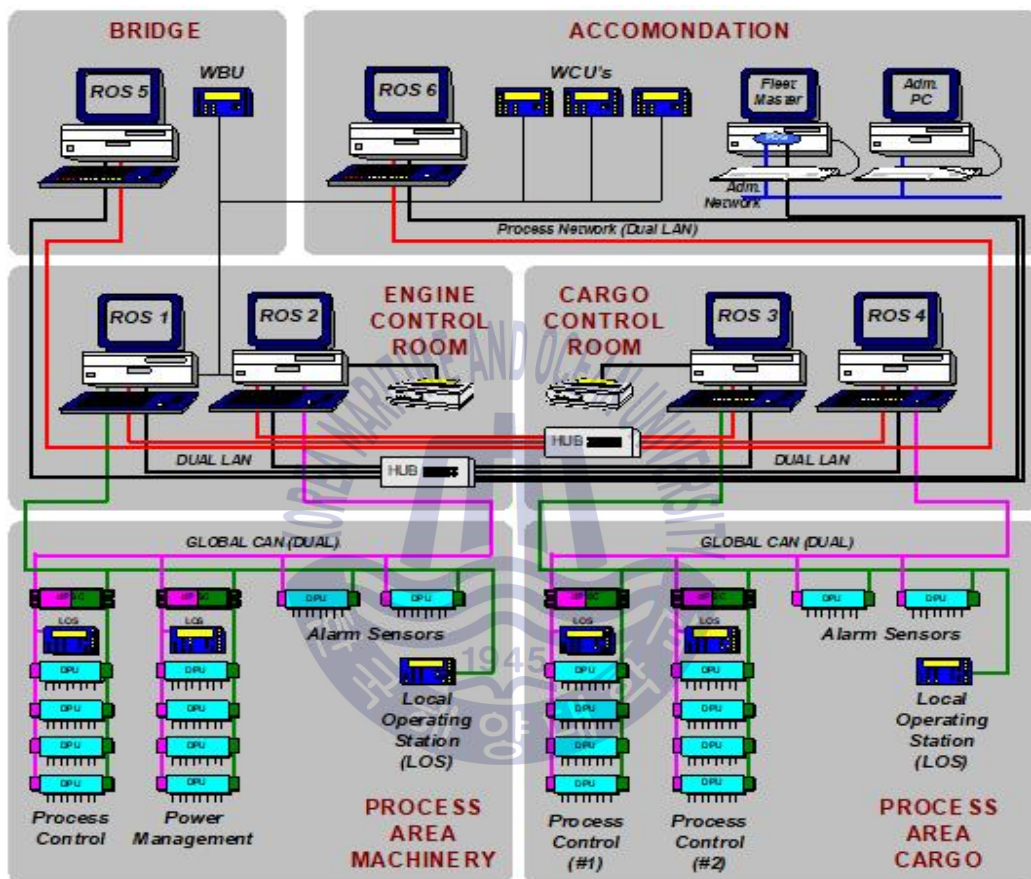


Fig. 2-10 Engine Room network for Kongsberg

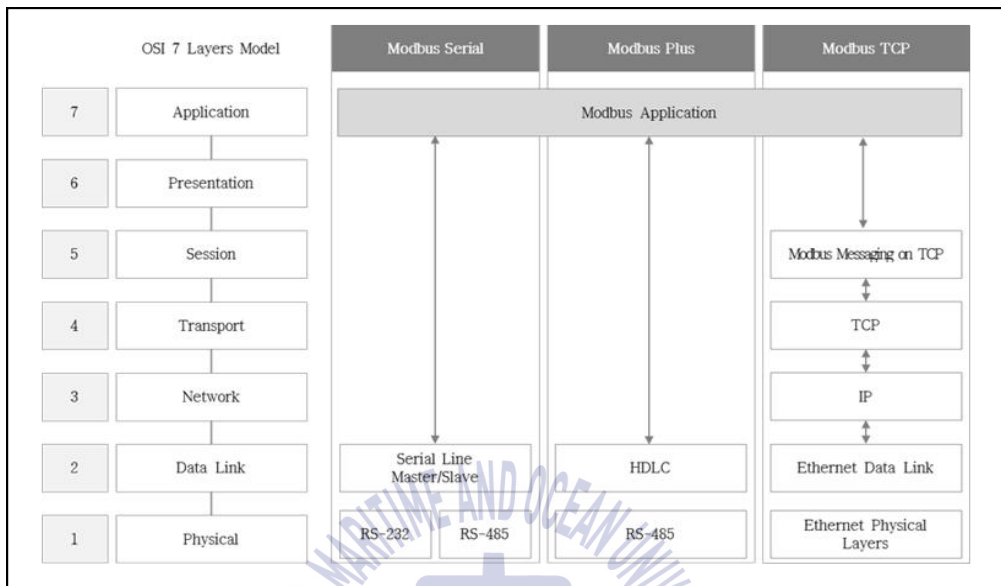


Fig. 2-11 Hierarchical structure of Modbus

Modbus Serial과 Modbus Plus는 Physical, Data Link, Application 계층으로 구성되어 있다. Modbus TCP/IP는 위의 3개 계층 사이에 Network, Transport, Session 계층으로 복잡한 계층 구조를 가지고 있다 (MODBUS ORG., 2012(b); Gjerset, 2017).

Modbus Serial은 물리 계층(Physical)은 RS-232, RS-422 기반의 Modbus 그리고 RS-485 기반의 Modbus로 세분화 되어 있다. 데이터 링크는 마스터/슬레이브 (Client/Server) Fig. 2-12인 아키텍처로서, Client가 서버에 데이터 몇 가지의 기능을 요청 할 수 있다. 통신은 1:1 통신 및 다자간 통신이 가능하게 설계되어 있다. Modbus 1:1 통신은 두 통신 모듈 간 통신을 하며 마스터에서는 리퀘스트를 요청하면 슬레이브에서는 응답하는 방식으로 통신을 하게 되어있다. 두 모듈 간에는 미리 정해진 function code 및 address를 사용하여 필요한 데이터를 송·수신 하는 방식이다.

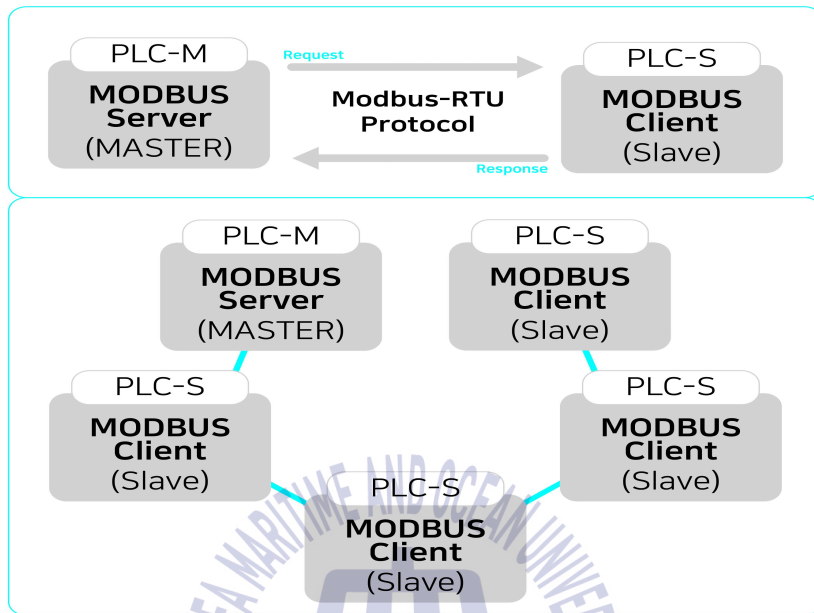


Fig. 2-12 Configuration of Modbus network

다자간 통신은 각 모듈마다 번호를 정하여 마스터에서 필요한 통신 모듈에 데이터를 요청하는 방식으로 데이터를 송·수신하는 방식이다. 여기에는 하나의 마스터만 존재하고 다른 모듈은 슬레이브로 동작해야 한다.

Client와 Server 사이에 주고받는 메시지를 프레임이라고 한다. 전송 방식에 따라 RTU(Remote Terminal Unit)와 ASCII로 나뉘며, 두 가지 전송 방식은 프레임의 전송 내용에 차이가 있다. RTU의 경우 Bit 단위로 전송하며 ASCII의 경우 문자 단위로 전송한다. 또한 통신상의 에러를 검출하기 위해 RTU는 CRC 에러 체크를 사용하고, ASCII는 LRC 에러 체크를 사용한다.

본 연구에서 기관 데이터를 추출하기 위해 사용한 프레임 포맷은 Modbus RTU이며 이에 대한 정보는 Table 2-3과 같다.

Table 2-3 Frame format of Modbus-RTU

이름	길이	기능
Start	28	최소 3.5 Character의 프레임으로 시작
Address	8	Station Address
Function	8	Function Code
Data	n*8	Data+길이는 개발 메시지 Type에 따라 변동
CRC	16	Error 체크
End	28	최소 3.5 Character의 전(前)의 프레임과 다음 프레임 사이

Modbus 프로토콜의 프레임은 Table 2-4 과 같이 5가지의 데이터 포맷을 가진다. Discrete Input은 디지털 입력으로 비트 단위의 데이터를 입력할 수 있으며, Coil은 디지털 출력으로 비트 단위의 데이터를 슬레이브에 출력할 수 있다. Input Registers는 2 Byte 단위로 아날로그 입력 및 내부 메모리로 사용되어지며, Holding Register는 2 Byte 단위로 메모리와 같은 용도로 사용되어진다. 본 연구에서는 Modbus는 데이터의 Type 별로 Object를 분기하여 처리하였다. 예를 들어 메인엔진, 발전기의 On/Off 상태에 대한 데이터를 받을 때는 Input Discrete(Read Function Code: 0x02)을 사용하였고, RPM과 같은 수치 데이터 값을 받을 때는 Holding Register(Read Function Code: 0x03)로 처리하였다. Coil과 Input Register를 사용 가능하나 제어권이 필요가 없거나 가질 필요가 없는 시스템일 경우 Read Type의 Object와 Function Code를 사용하게 되고, 실질적으로 제조사에서도 Read-Write에 대한 Function Code를 사용할 수 없게끔 Address를 지정한다.

Table 2-4 Modbus RTU Object & Function Codes

Object	Object type	Type of	사용 Function Code			Address
			Read	Single Write	Multiple Write	
Coil	Single bit	Read-Write	0x01	0x05	0x0F	0xxxx
Input Discrete	Single bit	Read-Only	0x02			1xxxx
Holding Register	16-bit word	Read-Write	0x03	0x06	0x10	4xxxx
Input Register	16-bit word	Read-Only	0x04			3xxxx
Exception Status	Eight bits	Read-Only	0x07			

본 연구의 프로세스는 Table 2-5와 Modbus 프로토콜에 맞춰서 설계하였다. 즉, 해당 송신은 본 연구에서 개발하는 시스템이고, 수신은 AMS가 된다. 송신자는 정의된 Function Code 0x02와 0x03으로 데이터를 요청하고 수신자는 해당 Address에 요청한 Code에 맞춰 데이터를 보낸다.

Table 2-5 Interface design for Read Discrete Inputs

프로세스 명	Read Discrete Inputs		
설명	특정 Input Bits(1~2000의 연속된)의 정보를 읽음		
Message Type	hexadecimal (16진수) byte		
전송방식	Asynchronous		
송신	[데이터항목] Slave Address Function Code (0x02) Starting Address (0x0~0xFFFF) Input Starting Address Hi Input Starting Address Lo Quantity of Inputs (0x1~0x7D0) No. of Points Hi No. of Points Lo Error Check (CRC)	[데이터타입] Byte(1) Byte(1) Byte(2) Byte(1) Byte(1) Byte(2) Byte(1) Byte(1) Byte(2)	[예시] 11 02 00C4 00 C4 0016 00 16 XXXX
수신	[데이터항목] Slave Address Function Code (0x02) Byte Count *N) Input Status Error Check (CRC)	[데이터타입] Byte(1) Byte(1) Byte(1) Byte(가변) Byte(2)	[예시] 11 02 03 ACDB35 XXXX
수신 Exception	[데이터항목] Slave Address Function Code (0x82) Data *Exception code Error Check (CRC)	[데이터타입] Byte(1) Byte(1) Byte(1) Byte(1) Byte(2)	[예시] 11 82 01 01 XXXX

2.4 Kafka 메시징 시스템

효율적으로 데이터를 수집하기 위하여 다양한 방법의 중간 과정이 요구된다. 일반적인 기능을 운영 체제로부터 제공 받지만, 서비스 어플리케이션으로 유연하고 확장성 있는 기능을 제공해 줄 수 있는 소프트웨어가 요구된다. 이는 독립적으로 개발되어 서로 다른 네트워크 플랫폼에서 실행되는 소프트웨어 구성 요소 간 상호 작용을 가능하게 함으로써 통합과정에서 불필요한 프로세스를 줄여 줄 수 있다. 자율운항선박의 육상관제 센터를 위한 시스템들은 복잡한 선박 시스템, 선박과 육상간의 메시지 교환, 육상에서의 이해 관계자와의 메시지 교환, 다양한 서비스 시스템의 구축 등을 효과적으로 하기 위해 MOM(Message Oriented Middleware)와 같은 메시지 처리 S/W가 요구된다. MOM은 분산 응용 서비스 프로그램 간에 메시지를 전송 및 수신하여 데이터 통신과 교환을 가능하도록 지원한다 (Banavar *et al.*, 1999; Tran, Greenfield and Gorton, 2002).

MOM은 통합 대상 혹은 Client 간 Broker(중재)의 역할을 담당한다. Broker는 Client 간 연결, 메시지 라우팅, 보안, 메시지 저장, 모니터링 등의 기능을 수행한다. 이러한 MOM은 메시징 시스템이로도 정의되며 대표적으로 ActiveMQ, ZeroMQ, RabbitMQ, Kafka 등이 있다 (Magnoni, 2015).

Fig. 2-6과 같이 Kafka의 메시지 처리 속도, 신뢰성, 확장성 등이 메시징 시스템으로써의 장점과 다른 메시징 시스템에 비해 성능이 우수하다고 판단되어, 본 연구의 시스템에서 아키텍처에 반영하여 적용하였다.

Table 2-6 Messaging process speed of distribution application service program

Type	Message Size/sec
 ACTIVEMQ	5,000
 RabbitMQ	10,000
 ØMQ	40,000
 kafka®	3,000,000

2.4.1 MOM - Kafka 개념

Kafka는 발행-구독(Publish-subscribe) 모델을 기반으로 동작하며 Producer, Consumer, Broker로 구성되며, 기본적인 개념은 Fig. 2-13과 같이 각 Producer에서 받은 데이터를 Broker가 Consumer에게 주는 중재자 역할을 수행하며, Consumer에게 배당하는 것이 아닌 Consumer가 원하는 것을 소비할 만큼 가져가는 구조이다(Varga, 2016; Apache kafka 0.8.1 Documentation).

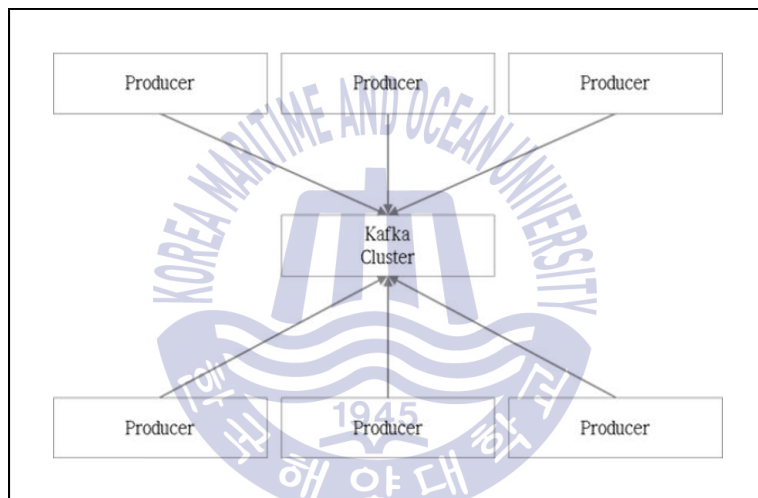


Fig. 2-13 Kafka composition element

Kafaka 메시징 시스템의 Broker는 토픽(Topic)을 기준으로 메시지를 관리한다. Producer는 특정 토픽의 메시지를 생성한 해당 메시지를 Broker에게 전달한다. Broker가 전달 받은 메시지를 토픽별로 분류하여 쌓아 두면, 해당 토픽을 구독하는 Consumer 들이 메시지를 가져가서 처리한다.

Kafka의 토픽은 파티션(Partition)이라는 단위로 쪼개어져 클러스터의 각 서버들에 분산되어 저장되고, 고 가용성을 위해 복제 설정을 할 경우 파티션 단위로 복구(Fail Over)가 수행된다. Kafka는 확장성(Scale-out)과 고가용성(High

Availability)을 위하여 Broker들이 Cluster로 구성되어 동작하도록 설계되어 있다. 심지어 Broker가 Single이어도 Cluster로 동작한다. Cluster 내의 Broker에 대한 분산 처리는 Fig. 2-14과 같이 Zookeeper가 담당한다(Kreps, Narkhede and Rao, 2011).

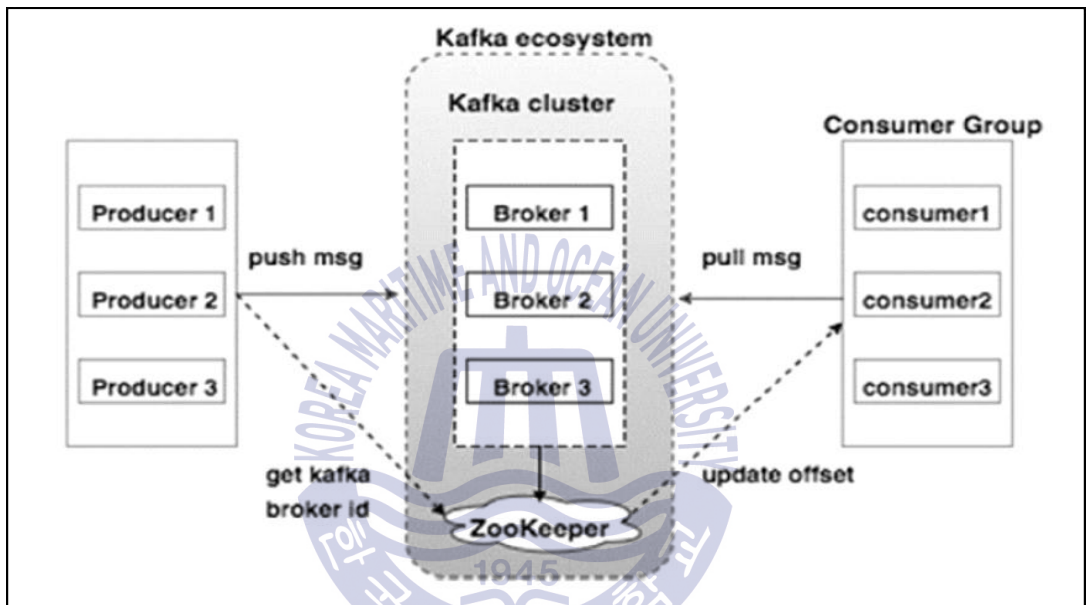


Fig. 2-14 Kafka Cluster

2.4.2 MOM - Kafka 특징

Kafka는 대용량 메시지의 실시간 로그 처리에 특화된 아키텍처 설계, 즉 기존의 메시징 시스템에서는 Broker가 Consumer에게 메시지를 Push해주는 방식인데 비해 Kafka는 Consumer가 Broker로부터 직접 메시지를 가지고 가능 Pull 방식으로 동작하기 때문에 Consumer는 자신의 처리 능력만큼의 메시지만 Broker로부터 가져온다. AMQP 프로토콜이나 JMS API를 사용하지 않고 단순한 메시지 헤더를 지닌 TCP 기반의 프로토콜을 사용하여 프로토콜에 의한 오버헤드를 감소시킨다. 또한 Producer가 Broker에게 다수의 메시지를 전송할 때 각 메시지를 개별적으로 전송해야 하는 기존 메시징 시스템과는 달리, 다수의 메시지를 Batch 형태로 Broker에게 한번에 전달 할 수 있어 TCP/IP 라운드 트립 횟수를 줄일 수 있다. 따라서 Fig. 2-15 과 같이 기존 메시징 시스템 보다 우수한 TPS(Throughput Per Second)를 보여 줄 수 있다. 붉은색은 메시지를 한번에 50개씩 전송한 결과이고, 연두색은 한 번에 하나씩 전송한 결과이다. 왼쪽 그래프와 같이 분산 배치 처리한 경우 더 우수한 성능을 보요 준다는 것을 확인할 수 있고, ActiveMQ와 같이 타 메시징 시스템에 비해 Kafka가 보다 나은 성능을 보여 준다는 것을 확인 할 수 있다 (Kreps, Narkhede and Rao, 2011).

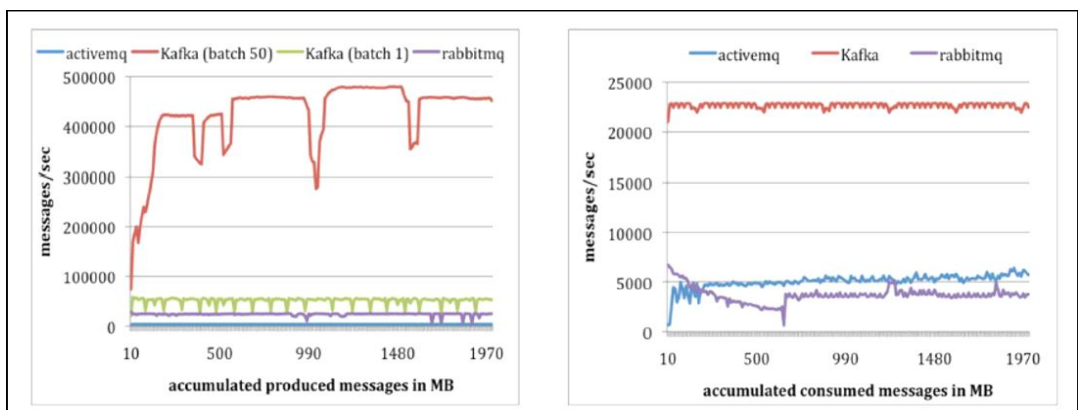


Fig. 2-15 TPS of Kafka producer/consumer

분산 시스템을 기본으로 설계 되어있기 때문에, 기존 메시징 시스템에 비해 분산 및 복제 구성을 손쉽게 할 수 있어 확장성이 뛰어나다.

Kafka는 기본적으로 메모리에 저장하는 기존 메시징 시스템과는 달리 메시지를 파일 시스템에 저장한다. 따라서 데이터의 영속성(Durability)이 보장되며, 기존 메시징 시스템에서는 처리되지 않고 남아 있는 메시지의 수가 많을수록 시스템의 성능이 크게 감소하였으나 Kafka에서는 많이 쌓아두어도 성능이 크게 감소하지 않는다. 또한 많은 메시지를 파일 시스템에 쌓아 둘 수 있기 때문에 주기적인 Batch 작업에 사용할 데이터를 쌓아두는 용도로도 사용할 수 있다. Consumer에 의해 처리된 메시지를 곧바로 삭제하는 메시징 시스템과는 달리 처리된 메시지를 삭제하지 않고 파일 시스템에 그대로 두었다가 설정된 수명이 지나면 삭제할 수 있기 때문에, 메시지 처리 도중 문제가 발생하였거나 처리 로직이 변경되었을 경우 Consumer가 메시지를 처음부터 재처리하도록 할 수 있는 장점이 있다.



2.5 다중이용자에 의한 메시지 전달 방법

본 연구에서는 선박에서 다양한 데이터를 초단위로 수집하고 처리한 정보를 어플리케이션에서 어떻게 보여줄 것인지에 대해 분석하였다. 일반적으로 웹 어플리케이션에서 Server와 Client간의 통신은 대부분 HTTP를 통해 이루어 졌으며 HTTP는 Request/Response 기반의 Stateless Protocol(=단방향 통신)입니다. 즉, Server와 Client 간의 Socket Connection 같은 영구적인 연결이 되어 있지 않고 Client(ex. Web Browser)에서 필요할 때 Request를 할 때만 서버가 응답하는 방식으로 통신이 진행되는 프로토콜이다. 따라서 Server 쪽 데이터가 업데이트 되더라도 Client에서 화면 새로 고침을 하지 않는 한 바뀐 데이터가 업데이트 되지 않는다. 물론 Long Polling, Ajax를 사용해서 해결이 가능하지만 데이터의 빠른 업데이트가 중요한 요소 중에 하나인 서비스일 경우 서버에 부담을 줄 수 있다.

본 연구에서는 항해 및 기관데이터를 초단위로 수집하고, 수집된 데이터를 사용자에게 제공 서비스를 해야 하므로 지속적인 업데이트를 위하여 Stateful Protocol(=양방향 통신)인 WebSocket을 분석 하였다.

WebSocket은 원격 호스트간의 양방향 통신을 위하여 웹 페이지에서 사용할 수 있는 API이다. HTML5 이전 버전에서는 서버와의 비동기적인 데이터 통신을 위해서 Ajax-Polling, Comet 등과 같은 방법을 사용하였지만 진정한 의미의 실시간 양방향 통신이라고 할 수 없다. Polling 방식은 동시에 양방향으로 통신을 할 수 없으며, Client에서 요청하면 서버의 응답이 있을 때까지 대기하고 응답을 수신한 후 다시 요청을 보내는 단방향 통신이다. 반면 WebSocket는 Fig. 2-16 웹페이지에서 하나의 소켓으로 실시간 양방향 통신을 지원한다 (WebSocket.org).

양방향 통신이 가능해지면 Client의 요청이 없어도 서비스는 데이터를 Client로 전송할 수 있다.

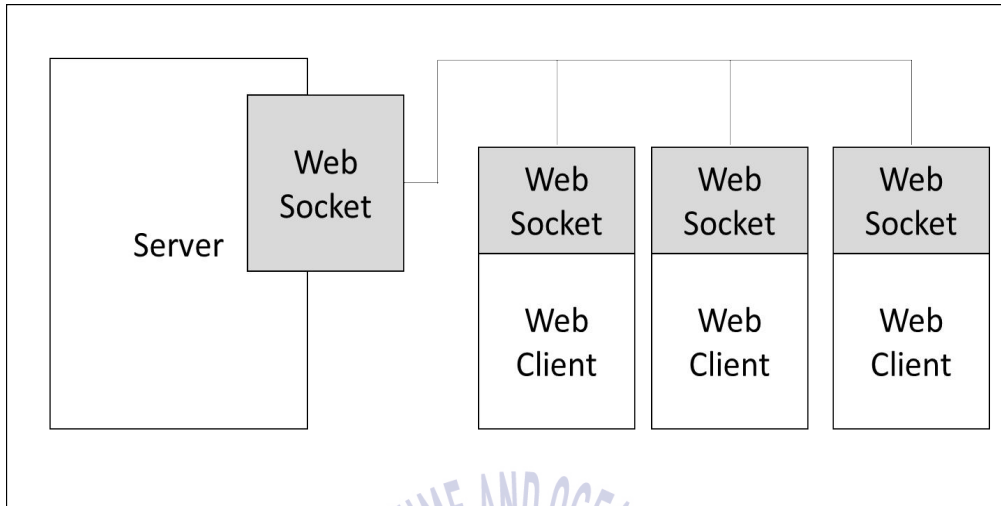


Fig. 2-16 Composition of WebSocket

WebSocket은 브라우저의 네이티브 소켓을 제공하기 때문에 대기시간과 HTTP의 요청 및 응답 헤더를 최소화하고 복잡성을 줄일 수 있다. 또한 Polling 방식은 통신을 할 때마다 HTTP 헤더가 포함되는 반면 WebSocket은 WebSocket 핸드셰이크 시에만 요청 및 응답 헤더가 있고 데이터를 송수신 할 때는 2 Byte의 프레임만을 사용하기 때문에, HTTP 헤더와 비교하였을 경우 1/500, 1/1000가량 데이터를 감소할 수 있다.

WebSocket은 네트워크 트래픽의 감소뿐만 아니라 응답 대기시간도 줄일 수 있다. Polling 방식은 서버로부터 데이터를 받기 위해 클라이언트가 지속적으로 데이터를 요청하지만, WebSocket은 클라이언트와 서버가 연결된 후 클라이언트의 별도 요청이 없어도 서버는 Client에게 데이터를 전송할 수 있다. Fig. 2-17은 서버에서 클라이언트로 데이터를 전송하기 위해 필요한 시간이 50ms가 필요한 어플리케이션의 경우, Polling과 WebSocket의 처리시간을 비교한 것이다. WebSocket은 클라이언트의 요청이라는 단계를 거치지 않기 때문에 같은 시간의 자원으로 보다 많은 데이터의 처리가 가능하다.

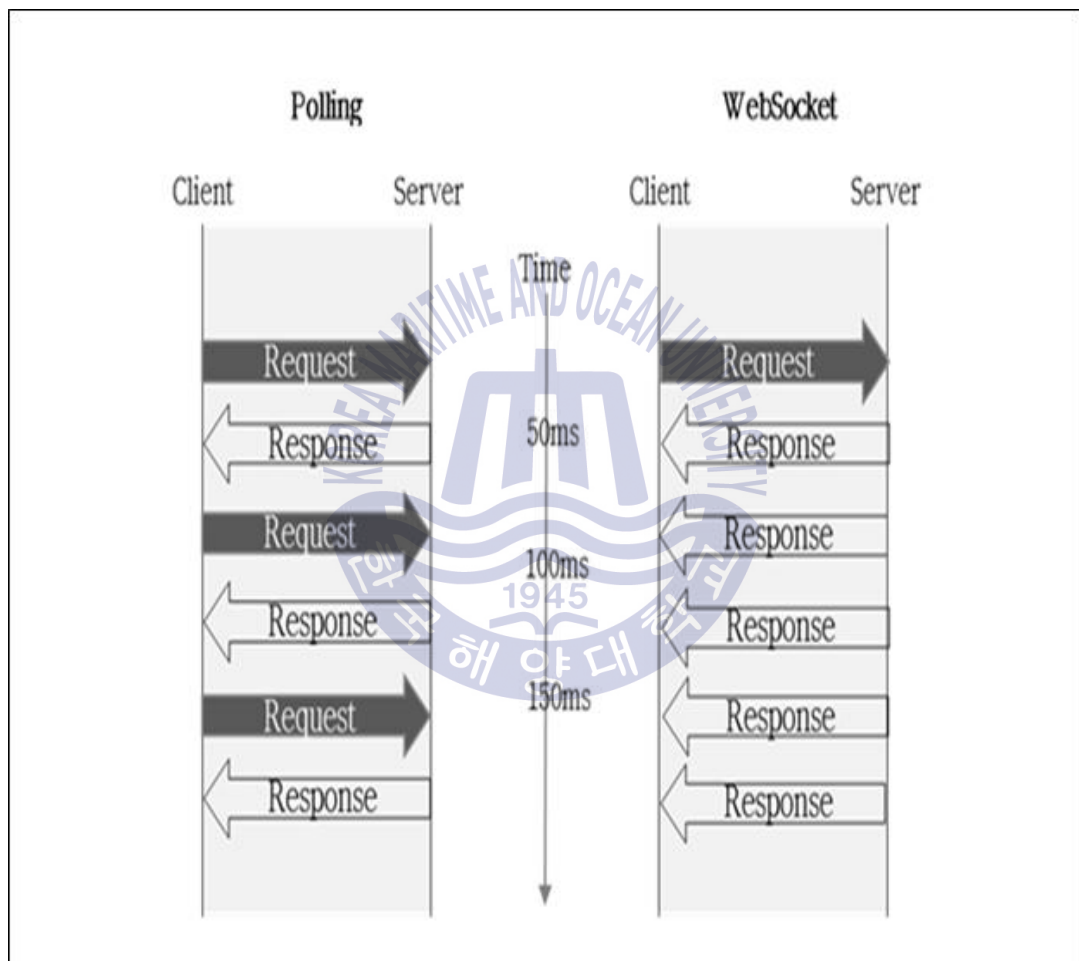


Fig. 2-17 Waiting time comparison of Polling and WebSocket

제 3 장 운항 정보전송 시스템 개발 방향 설계

3.1 개발 방향

본 논문에서 선박에서 발생하는 데이터를 수집/처리/전송할 수 있는 선박 운항 정보 수집 및 전송 시스템에 대해 Fig. 3-1 과 같이 구성하고 구현 하였다. 요구하는 물리적 구성은 크게 선박(Onboard)과 육상으로 구분되고, 설계상으로는 n개의 선박과 n개의 육상을 가질 수 있다. 선박 측에는 RADAR, Speed Log, Doppler, Gyro Compass, GPS 등 항해 장비와 AMS로부터 데이터를 수집하고, 데이터를 육상 및 선박 내 서비스 어플리케이션에게 사용할 수 있도록 처리 및 저장하고, 요구되는 곳에 전송할 수 있는 모듈로 구성된 Server가 구성된다. 육상은 여러 선박으로부터 데이터를 수신(=구독)하여 처리 후 요구되는 곳에 전송할 수 있는 모듈로 구성된 Server와 서비스 어플리케이션 용 서버들로 구성된다. 사용자는 PC에서 웹브라우저로 접속하여 운항 정보 모니터링과 같은 서비스를 사용할 수 있도록 구성하였다.

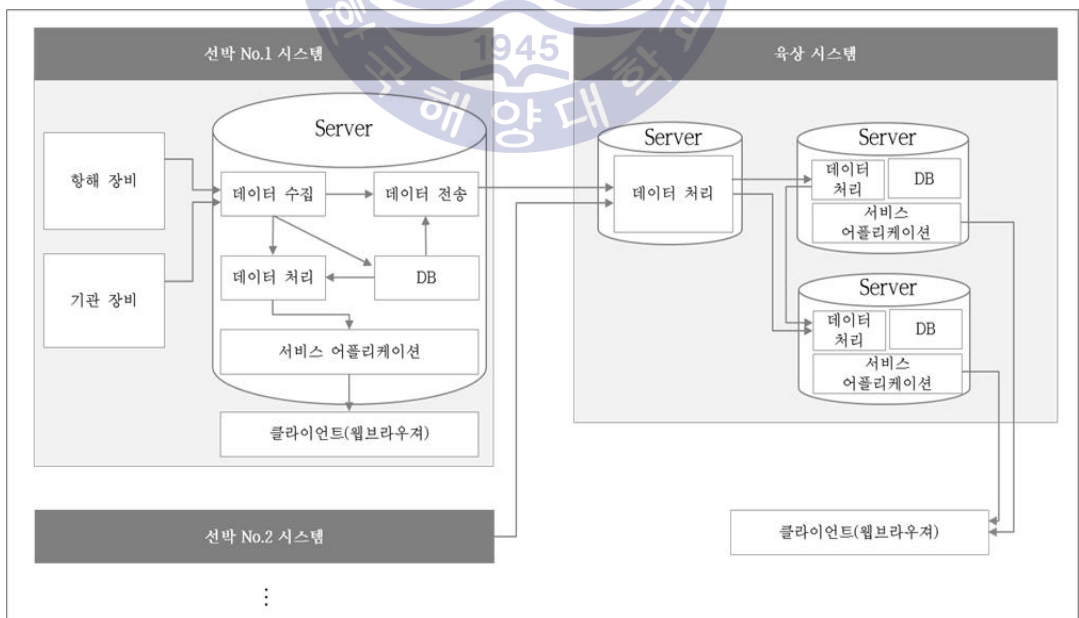


Fig. 3-1 System definition

구성된 운항 정보 수집 및 전송 시스템의 아키텍처는 Fig. 3-2 과 같다. 각 선박에서는 항해장비로 부터는 NMEA 프로토콜을 사용하여 데이터를 수집하고 기관 정보는 AMS로부터 Modbus Serial로 수집한다. 항해데이터의 경우 각 장비에서 0.2 ~ 1초 간격으로 수집되고 기관 장비는 10초 간격으로 데이터가 수집된다. Server 간 데이터 전달은 Kafka 메시징 시스템을 적용하여 수집된 데이터는 선박과 육상 Consumer가 구독하여 데이터를 사용할 수 있도록 하였다. 데이터 수집 장치가 만든 NEMA Topic, 기관 정보 Topic 등을 구독한 각 웹 서버는 DB에 저장하여 재가공 하여 쓸 수 있도록 하고, WebSocket을 통한 웹 브라우저 사용자에게 2초 단위로 업데이트 되는 모니터링 서비스를 제공하도록 한다.

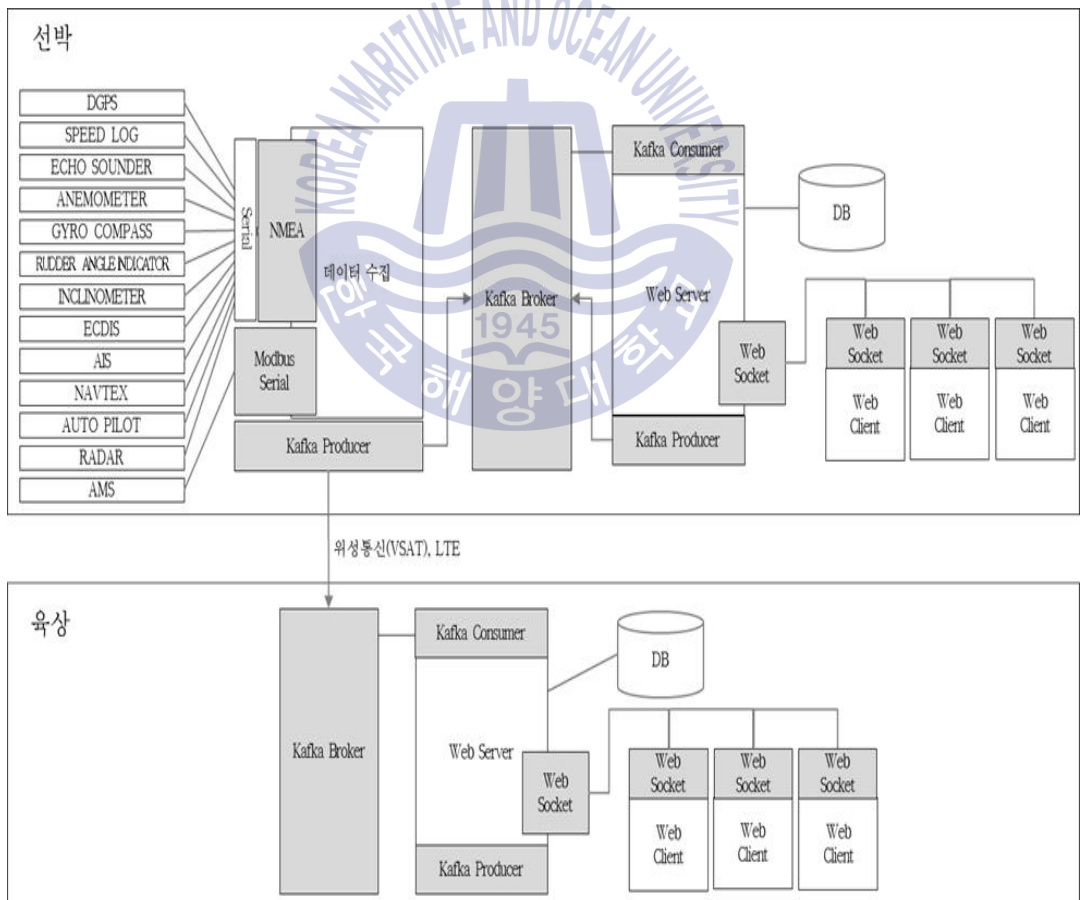


Fig. 3-2 System architecture

3.2 성능 구현

본 논문에서 구현한 선박 운항 정보 수집 및 전송 시스템은 데이터 수집과 육상용 서버는 Linux, 선박의 웹서버는 Window기반 운영 체제에서 구현하였다. WAS (Web Application Server)는 Apache Tomcat 8.5.11을 사용하였다. 구현을 위한 언어는 JAVA(JDK 1.8.0.121)을 사용하였고, POSTGRES로 DB를 구성하였다. 메시징 시스템은 Apache Kafka 2.1.0-0.10.2.0을 사용하였다. 구현환경에서 데이터 인터페이스는 Kongsberg 항해 시뮬레이터와 연계하여 구현되었고, 실 데이터를 활용한 구현은 한국해양대학교 실습선 한바다호 장비와의 연계를 통해 구현하였다.



3.3. 데이터 수집

3.3.1 하드웨어 장비 데이터 수집

데이터를 수집하기 위해서는 Table 3-1과 Fig. 3-2와 같이 각 장비별로 데이터를 수신 받을 수 있는지를 분석해야 한다. 각 제조사마다 장비 모델에 대한 매뉴얼을 제공하며, 매뉴얼에 일반적으로 Fig. 3-3과 같이 ‘Interface’에 Output Data Sentences를 제공한다(FURUNO, 2009).

항해 장비의 경우 제조사/모델 별 차이가 있을 수 있다. 본 연구에서는 항해 장비의 경우 JRC, FURUNO 제품을 분석하여 수집 대상 데이터 및 연계를 정의하였다.

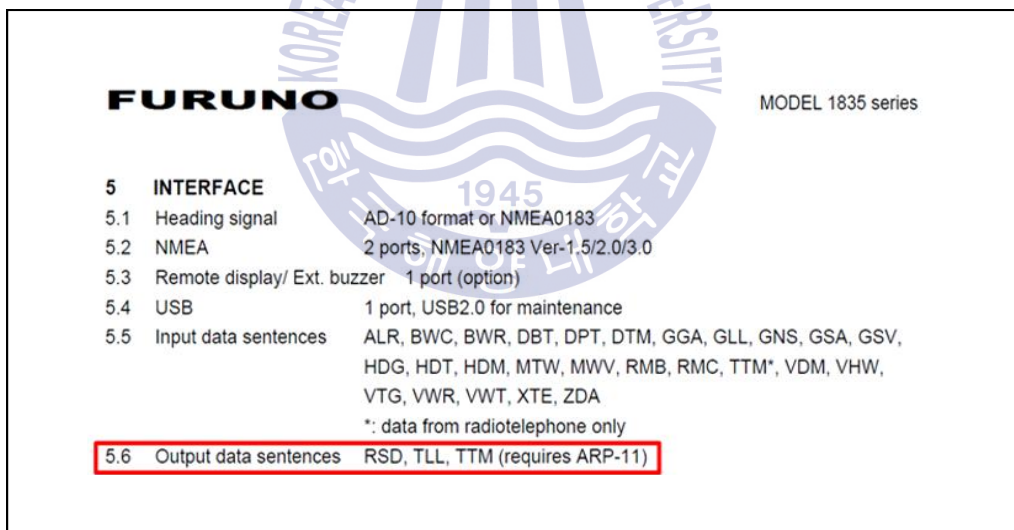


Fig. 3-3 An example of Output Data Sentences

Table 3-1 An example of Navigation equipment Output Data Boudrate

장비	Com Port	BOUDRATE	STOPBIT	PARITY	Interval
AIS	1	38,400	0	0	1,000 ms
RADAR	2	4,800	0	0	1,000 ms
ECDIS	3	4,800	0	0	1,000 ms
DGPS	4	4,800	0	0	1,000 ms
GYRP	5	4,800	0	0	1,000 ms
WIND	6	4,800	0	0	1,000 ms
NAVTEX	7	4,800	0	0	1,000 ms
Speed Log	8	4,800	0	0	1,000 ms
Echo Sounder	9	4,800	0	0	1,000 ms
Auto Pilot	10	4,800	0	0	1,000 ms

Table 3-2 Data collection communication type (RS 422)

Specification	RS 232	RS 422	RS 485
동작 모드	Single -Ended	Differential	Differential
최대 통달 거리	약 15 meter	약 1.2 km	약 1.2 km
최고 통신 속도	20 kb/s	10 Mb/s	10 Mb/s
지원 전송 방식	Full Duplex	Full Duplex	Full Duplex
최대 추력 전압압	+ - 15 VDC	-7 V to 7 V	-7 V to 7 V

기관 장비에 탑재되는 대부분의 센서는 아날로그 신호를 가지고 있다. 따라서 선박에서 데이터 연계를 효과적으로 하기 위해서는 AMS와 같은 아날로그 신호를 디지털로 변환하여 사용자 및 타 장비에게 정보를 제공하는 시스템을 통해 연계하는 것이 바람직하다. 일반적으로 AMS는 Modbus Serial(RTU)를 이용하여 데이터를 제공해 준다. 다만 NMEA0183 Sentence처럼 매뉴얼에 데이터 연

계에 관한 정보를 제공해 주지 않는다. 또한 AMS는 선박마다 장비의 체계 및 구성이 다르므로 제공되는 정보가 일치하지 않는다. Table 3-3과 같이 제조사의 엔지니어로부터 AMS Channel List를 제공받아 분석을 통해 Address 정의, 데이터의 Scale과 Range에 따른 Function Code를 정의해야 한다.

```
## DAT-file for calctest
# Param1 Param2 Param3 Tagname
#-----
## TAGNAME          COMPORT    BAUDRATE    STOPBIT    0=1,1=1.5,2=2  4-8    0=NO,1=ODD,2=EVEN,3=MARK,4=SPACE
# SYS300            2          9600        0          8          8          0

# COIL_OFFSET        00001      // For command 1
# DIGITAL_OFFSET     00001      // For command 2
# ANALOG_OFFSET      00001      // For command 3 and 4
# DIGITALCMD_OFFSET  00001      // For command 5 and 15
# ANALOGCMD_OFFSET   00001      // For command 6 and 16
# MAXADDR            3000      // number of tags in dat file

# NO_OF_PLC          1        // Number of PLCs
# ADDR_GAP           100      // Max gap between address in one query
```

Fig. 3-4 An example of Definition of Modbus for AMS Channel List

Table 3-3 An example of Definition of Modbus for AMS Channel List

No.	Tag Name	DESCRIPT	UNIT	ENG LOW	ENG HIGH	Fun. Code	Address	Data Type
1	MC067	SHIP PERFORMANCE SYSTEM RPM	RPM	0	200	0x03	40000	SINGED INTEGER
2	MC066	SHIP PERFORMANCE SYSTEM POWER	KW	0	6600	0x03	40001	SINGED INTEGER
3	MC068	SHIP PERFORMANCE SYSTEM TORQUE	KNM	0	250	0x03	40002	SINGED INTEGER
4	MF003	M/E F.O FLOW	L/H	10	3	0x03	40003	SINGED INTEGER
5	MF006	M/E F.O IN PRESS	BAR	0	16	0x03	40004	SINGED INTEGER
6	MF007	M/E F.O IN TEMP	DEG.C	0	200	0x03	40005	SINGED INTEGER
7	MA017	M/E SCAV AIR RECEIVER PRESS	BAR	0	4	0x03	40006	SINGED INTEGER
8	MA018	M/E SCAV AIR RECEIVER TEMP	DEG.C	0	200	0x03	40007	SINGED INTEGER
9	MS061	NO.1 MAIN AIR RESERVOIR PRESS	BAR	0	40	0x03	40008	SINGED INTEGER
10	MS069	NO.2 MAIN AIR RESERVOIR PRESS	BAR	0	40	0x03	40009	SINGED INTEGER

각 장비(송신 시스템)으로부터 정의된 NMEA Sentence와 Modbus를 통해 데이터를 Table 3-4의 리스트로 수신하면 시스템에서는 일련의 문자열을 갖게 된다. Fig. 3-5과 같이 Parsing Processor에 의해 의미 있는 Token(어휘 분석의 단위)으로 분해하고 분석하여 문법적 관계를 시스템에서 해석하여 Table3-5와 같은 데이터 항목으로 처리한다. Parsing 된 데이터는 Kafka Producer에 의해 메시징 처리 된다.

```

04:20:10.802 DEBUG 2385 --- [Thread-9] n.r.c.s.c.SerialNmea0183ProtocolDecoder Found NMEA End(F) Delimiter index : 61
04:20:10.802 DEBUG 2385 --- [Thread-9] n.r.c.s.c.SerialNmea0183ProtocolDecoder Found NMEA Start Delimiter index : 0
04:20:10.802 INFO 2385 --- [Thread-9] c.l.n.r.common.serial.io.SerialHandler messageReceived(id=1)
04:20:10.802 DEBUG 2385 --- [Thread-9] c.l.n.r.common.serial.io.SerialHandler Received data length : 62
04:20:10.802 INFO 2385 --- [Thread-9] c.l.n.r.n.n.com.MavSerialTxKtThread eventListener() Navigation received. eventType=LISTENER_EVENT_DATA RECEIVED, len=62
04:20:10.802 INFO 2385 --- [Thread-9] c.l.n.r.parser.nmea.SentenceValidator Checksum ok [received = 2222222222222222,0,417]
04:20:10.802 TRACE 2385 --- [Thread-7] c.l.n.r.n.h.s.NaviReceiverDataProcess L(S) : /AIWVO,2,1,0,A,1:5HRT4q015JfWvdbbc38B0501,0,66
04:20:10.802 INFO 2385 --- [Thread-7] c.l.n.r.parser.nmea.SentenceValidator Checksum ok [received = 17 calculated = 17]
04:20:10.803 INFO 2385 --- [Thread-11] com.lqmc.ndc.message.KafkaSender send message topic = 'LIVE_NMEA'
04:20:10.862 DEBUG 2385 --- [Thread-9] n.r.c.s.c.SerialNmea0183ProtocolDecoder Found NMEA End(F) Delimiter index : 52
04:20:10.862 DEBUG 2385 --- [Thread-9] n.r.c.s.c.SerialNmea0183ProtocolDecoder Found NMEA Start Delimiter index : 0
04:20:10.862 INFO 2385 --- [Thread-9] c.l.n.r.common.serial.io.SerialHandler messageReceived(id=1)
04:20:10.862 DEBUG 2385 --- [Thread-9] c.l.n.r.common.serial.io.SerialHandler Received data length : 53
04:20:10.862 INFO 2385 --- [Thread-9] c.l.n.r.n.n.com.MavSerialTxKtThread eventListener() Navigation received. eventType=LISTENER_EVENT_DATA RECEIVED, len=53
04:20:10.862 DEBUG 2385 --- [Thread-9] c.l.n.r.n.n.com.MavSerialTxKtThread >> /AIWVO,2,2,0,A,1:00570Wf3SQEnCqR88888888882,2+2
04:20:10.863 TRACE 2385 --- [Thread-7] c.l.n.r.n.h.s.NaviReceiverDataProcess L(S) : /AIWVO,2,2,0,A,1:00570Wf3SQEnCqR88888888882,2+2
04:20:10.863 INFO 2385 --- [Thread-7] c.l.n.r.parser.nmea.SentenceValidator Checksum ok [received = 42 calculated = 42]
04:20:10.863 INFO 2385 --- [Thread-11] com.lqmc.ndc.message.KafkaSender send message topic = 'LIVE_NMEA'
04:20:11.535 DEBUG 2385 --- [Thread-9] n.r.c.s.c.SerialNmea0183ProtocolDecoder Found NMEA End(F) Delimiter index : 48
04:20:11.535 INFO 2385 --- [Thread-9] n.r.c.s.c.SerialNmea0183ProtocolDecoder Found NMEA Start Delimiter index : 0
04:20:11.536 INFO 2385 --- [Thread-9] c.l.n.r.common.serial.io.SerialHandler messageReceived(id=1)
04:20:11.536 DEBUG 2385 --- [Thread-9] c.l.n.r.common.serial.io.SerialHandler Received data length : 49
04:20:11.536 INFO 2385 --- [Thread-9] c.l.n.r.n.n.com.MavSerialTxKtThread eventListener() Navigation received. eventType=LISTENER_EVENT_DATA RECEIVED, len=49
04:20:11.536 DEBUG 2385 --- [Thread-9] c.l.n.r.n.n.com.MavSerialTxKtThread >> /AIWVO,1,1,A,1:NR4q015JfWvdbbc38B0501,0,66
04:20:11.536 TRACE 2385 --- [Thread-7] c.l.n.r.n.h.s.NaviReceiverDataProcess L(S) : /AIWVO,1,1,A,1:NR4q015JfWvdbbc38B0501,0,66
04:20:11.537 DEBUG 2385 --- [Thread-7] c.l.n.r.parser.nmea.SentenceValidator Checksum ok [received = 66 calculated = 66]
04:20:11.537 INFO 2385 --- [Thread-11] com.lqmc.ndc.message.KafkaSender send message topic = 'LIVE_NMEA'
04:20:12.543 DEBUG 2385 --- [Thread-9] n.r.c.s.c.SerialNmea0183ProtocolDecoder Found NMEA End(F) Delimiter index : 48
04:20:12.543 INFO 2385 --- [Thread-9] n.r.c.s.c.SerialNmea0183ProtocolDecoder Found NMEA Start Delimiter index : 0
04:20:12.543 INFO 2385 --- [Thread-9] c.l.n.r.common.serial.io.SerialHandler messageReceived(id=1)
04:20:12.543 DEBUG 2385 --- [Thread-9] c.l.n.r.common.serial.io.SerialHandler Received data length : 49
04:20:12.543 INFO 2385 --- [Thread-9] c.l.n.r.n.n.com.MavSerialTxKtThread eventListener() Navigation received. eventType=LISTENER_EVENT_DATA RECEIVED, len=49
04:20:12.544 TRACE 2385 --- [Thread-7] c.l.n.r.n.h.s.NaviReceiverDataProcess L(S) : /AIWVO,1,1,A,1:NR4q015JfWvdbbc38B0501,0,66
04:20:12.544 INFO 2385 --- [Thread-7] c.l.n.r.parser.nmea.SentenceValidator Checksum ok [received = 28 calculated = 28]
04:20:12.544 INFO 2385 --- [Thread-11] com.lqmc.ndc.message.KafkaSender send message topic = 'LIVE_NMEA'

```

- 39 -

Table 3-4 An example of Definition of Modbus for AMS Channel List

NAME	TAG명	Modbus Function Code	Modbus Address	Scale	Range	Data Unit
M/E RPM	MC056	0x03	0	*100	-150~150	RPM
SHAFT POWER	ML100	0x03	202 (Dword)	*100	0~15000	KW
SHAFT TORQUE	ML097	0x03	204 (Dword)	*100	-1400~1400	KNM
SHAFT RPM	ML099	0x03	1	*100	-120~120	RPM
M/E F.O MASS FLOW	MF001	0x03	200 (Dword)	*100		KG/H
M/E F.O IN PRESS	MF005	0x03	2	*100	0~16	KG/CM2
M/E F.O IN TEMP	MF006	0x03	3		0~200	DEG.C
M/E SCAV AIR RECEIVER PRESS	MC005	0x03	4	*100	0~4	KG/CM2
M/E SCAV AIR RECEIVER TEMP	MA012	0x03	5		0~200	DEG.C
NO.1 MAIN AIR RESERVOIR PRESS	MC006	0x03	6	*100	0~40	KG/CM2
NO.2 MAIN AIR RESERVOIR PRESS	MC007	0x03	7	*100	0~40	KG/CM2
S/T BEARING(AFT) TEMP	ML080	0x03	8		0~150	DEG.C
S/T BEARING(FWD) TEMP	ML081	0x03	9		0~150	DEG.C
NO.1 INTER SHAFT BRG TEMP	ML082	0x03	10		0~150	DEG.C
M/E THRUST BRG SEG(AHEAD) TEMP	ML056	0x03	11		0~200	DEG.C
M/E LO SUMP TK LEVEL	ML072				0.16~1.484	M
M/E JACKET CPW IN PRESS	MW008	0x03	12		0~6	KG/CM2
M/E NO.1 AIR CLR CPW OUT TEMP	MW010	0x03	13		0~200	DEG.C
M/E NO.2 AIR CLR CPW OUT TEMP	MW011	0x03	14		0~200	DEG.C
M/E NO.1 T/C RPM	MC018	0x03	15		0~25000	RPM
M/E NO.2 T/C RPM	MC019	0x03	16		0~25000	RPM
M/E NO.1 T/C EXH GAS IN T H	MA009	0x03	17		0~600	DEG.C
M/E NO.2 T/C EXH GAS IN T H	MA011	0x03	18		0~600	DEG.C
M/E NO.1 T/C EXH GAS OUT T H	MA008	0x03	19		0~600	DEG.C
M/E NO.2 T/C EXH GAS OUT T H	MA010	0x03	20		0~600	DEG.C
M/E NO.1 T/C LO OUT TEMP	ML005				0~200	DEG.C
M/E NO.2 T/C LO OUT TEMP	ML006				0~200	DEG.C
M/E NO.1 CYL EXH GAS OUT TEMP	MA001	0x03	21		0~600	DEG.C
M/E NO.2 CYL EXH GAS OUT TEMP	MA002	0x03	22		0~600	DEG.C
M/E NO.3 CYL EXH GAS OUT TEMP	MA003	0x03	23		0~600	DEG.C
M/E NO.4 CYL EXH GAS OUT TEMP	MA004	0x03	24		0~600	DEG.C
M/E NO.5 CYL EXH GAS OUT TEMP	MA005	0x03	25		0~600	DEG.C
M/E NO.6 CYL EXH GAS OUT TEMP	MA006	0x03	26		0~600	DEG.C
M/E NO.7 CYL EXH GAS OUT TEMP	MA007	0x03	27		0~600	DEG.C

Table 3-5 Interface description of data collection system

송신자	항목 구분	항목명	단위	데이터 타입	Max Size	데이터 포맷
DGPS	GP	Latitude		string	16	llll.ll, a
		Longitude		string	16	yyyyy.yy, a
		Positioning system indicator		string	1	A,D,E,M,S,N, V
		UTC of position		string	10	hhmmss.ss
		GPS quality indicator		string	1	x
		Number of satellites in use		string	2	xx
		Horizontal dilution of precision		string	8	x.x
		Antenna altitude	meter	string	8	x.x
		Geoidal separation	meter	string	8	x.x
		Age of the differential GPS data		string	8	x.x
		Differential reference station ID		string	4	xxxx
		Local datum		string	4	xxx
		Local datum sub code		string	4	x
		Latitude offset		string	12	xx.xxxxx, a
		Longitude offset		string	12	xx.xxxxx, a
		Altitude offset	meter	string	8	
		Reference datum		string	4	
		UTC		string	10	hhmmss.ss
		Day		string	2	xx
		Month		string	2	xx
		Year		string	4	xxxx
		Local zone hours		string	4	xx
		Local zone minutes		string	4	xx
		NMEA sentence		string		
Gyro compass	HE	Rate of turn	degree/ min	string	8	x.x
		True heading	degree	string	8	x.x

		mode indicator		string	1	A,E,M,S,V
		True course over ground	degree	string	8	
		NMEA sentence		string		
Magnetic Compass	HE	Magnetic sensor heading	degree	string	8	x.x
		Magnetic deviation	degree	string	8	
		Magnetic variation	degree	string	8	
		Magnetic course over ground	degree	string	8	
		NMEA sentence		string		
Echo Sounder	DE	Water depth below keel	meter	string	8	x.x
		Water depth below surface	meter	string	8	x.x
		Water depth below transducer	meter	string	8	x.x
		Offset from transducer	meter	string	8	x.x
		Maximum range scale in use	meter	string	8	x.x
		NMEA sentence		string		
Speed Log	VE	Longitudinal ground speed	knot	string	8	x.x
		Transverse ground speed	knot	string	8	x.x
		Stern transverse ground speed	knot	string	8	x.x
		Longitudinal water speed	knot	string	8	x.x
		Transverse water speed	knot	string	8	x.x
		Stern transverse water speed	knot	string	8	x.x
		NMEA sentence		string		
Anemometer	WI	Relative wind direction	degree	string	8	x.x
		Relative wind speed	knot	string	8	x.x
		True wind direction	degree	string	8	x.x
		True wind speed	knot	string	8	x.x
		NMEA sentence		string		
Inclinometer	IN	Actual heel angle	degree	string	8	x.x
		Actual pitch angle	degree	string	8	x.x
		Roll period	second	string	8	x.x
		Roll amplitude port side	degree	string	8	x.x
		Roll amplitude starboard side	degree	string	8	x.x
		NMEA sentence		string		

Navtex	NA	Message identifier		string		xx
		Message Code		string		aaxx
		Current line		string		xxx
		Total lines		string		xxx
		freq		string		x
		UTC of receipt of message		string		hhmmss.ss
		day		string		xx
		month		string		xx
		year		string		xxxx
		Message characters number		string		x.x
		Bad characters number		string		x.x
		Message		string		n--n
		NMEA sentence		string		
AIS	VDO	Type of ship and cargo		string		x.x
		Static draft	meter	string		x.x
		Persons on board		string		x.x
		Destination		string		c--c
		Estimated UTC of arrival	UTC	string		hhmmss.ss
		Estimated day of arrival	UTC	string		xx
		Estimated month of arrival	UTC	string		xx
		Navigational status		string		x.x
		Regional application flags		string		x.x
		NMEA sentence		string		
ECDIS	EC	NMEA sentence		string		
RADAR	RAX	Target number		string	2	xx
		Target latitude		string	16	llll.ll, a
		Target longitude		string	16	yyyyyy.yy, a
		Target name		string		
		UTC of data		string	10	hhmmss.ss
		Target status		string	1	a
		Reference target		string	1	a
		Target distance		string		x.x
		Bearing from own ship		string		x.x
		Degrees true/relative (T/R)		string		a
		Target speed		string		x.x
		Target course		string		x.x

		Degrees true/relative (T/R)		string		a
		Distance of CPA (closest point of approach)		string		x.x
		TCPA		string		x.x
		Type of acquisition		string		a
		NMEA sentence		string		
Auto Pilot	RU	Commanded heading true	degree	string		x.x
		Commanded heading magnetic	degree	string		x.x
		Override		string		A or V
		Commanded rudder angle	degree	string		x.x
		Commanded rudder direction		string		L or R
		Selected steering mode		string		M,S,H,T,R
		Turn mode		string		R, T, N
		NMEA sentence		string		
Engine Telegraph	TE	Number of engine		string		x
		Event time		string		hhmmss.ss
		Message type		string		X
		Position of engine telegraph		string		XX
		Position of sub telegraph		string		XX
		Operating location indicator		string		X
		NMEA sentence		string		
Engine Propulsion	TE	Lever demand position		string		x.x
		RPM demand value		string		x.x
		RPM mode indicator		string		X
		Pitch demand value		string		x.x
		Pitch mode indicator		string		X
		Operating location indicator		string		
		NMEA sentence		string		
Thruster	TH	Number of thruster		string		x
		RPM demand value		string		x.x
		RPM response value		string		x.x
		RPM mode indicator		string		X
		Pitch demand value		string		x.x
		Pitch response value		string		x.x
		Pitch mode indicator		string		X

		Azimuth demand		string		x.x
		Azimuth response		string		
		Operating location indicator		string		
		Sentence flag status		string		
		NMEA sentence		string		
Rudder Angle Indicator	RU	Starboard rudder sensor	degree	string		x.x
		Port rudder sensor	degree	string		x.x
		NMEA sentence		string		
AMS	ME	ME running status		string	1	
		ME FO flow Total	kg/hour	string	20	
		ME FO flow Each	kg/hour	string	20	
		ME DO use		string	1	
		ME RPM	RPM	string	10	
		ME power	KW	string	20	
		ME torque	KNM	string	20	
		ME FO inlet press	bar	string	10	
		ME FO inlet temperature	celsius	string	10	
		ME scavenge air inlet press	bar	string	10	
		ME scavenge air inlet temperature	celsius	string	10	
		ME stern tube after bearing temperature	celsius	string	10	
		ME stern tube forward bearing temperature	celsius	string	10	
		ME inter shaft bearing temperature	celsius	string	10	
		ME thrust bearing temperature	celsius	string	10	
		ME turbo charger RPM	RPM	string	10	
		ME turbo charger gas inlet temperature	celsius	string	10	
		ME turbo charger gas outlet temperature	celsius	string	10	
		ME turbo charger lo outlet temperature	celsius	string	10	
		ME cylinder exhaust outlet	celsius	string	10	

		temperature				
		ME cylinder PCO outlet temperature	celsius	string	10	
		ME cylinder JCFW outlet temperature	celsius	string	10	
GE		GE running status		string	1	
		GE FO flow total	kg/h	string	20	
		GE FO flow each	kg/h	string	20	
		GE DO use		string	1	
		GE RPM		string	10	
		GE power	kw	string	10	
		GE frequency	hz	string	10	
		GE voltage	V	string	10	
		GE current	A	string	10	
		GE FO inlet press	bar	string	10	
		GE FO inlet temperature	celsius	string	10	
		GE LO inlet press	bar	string	10	
		GE LO inlet temperature	celsius	string	10	
		GE cylinder exhaust outlet temperature	celsius	string	10	
		GE turbo charger RPM	RPM	string	10	
		GE turbo charger gas inlet temperature	celsius	string	10	
		GE turbo charger gas outlet temperature	celsius	string	10	
		GE turbo charger lo outlet temperature	celsius	string	10	
AB		AB running status		string	1	
		AB FO flow	kg/h	string	20	
		AB Stean press	bar	string	10	
		AB drum water level	meter	string	10	
ETC		Total FO flow	kg/h	string	20	
		Sea water temp	celsius	string	10	
		Fore perpendicular draft	meter	string	10	
		Mid port draft	meter	string	10	
		Mid starboard draft	meter	string	10	
		After perpendicular draft	meter	string	10	

3.4 데이터 프로세싱 및 Web UI 구현

선박의 데이터 수집 장치의 Kafka Producer로부터 구독하는 Consumer를 구성할 때, 구독한 선박과 육상의 서버(Kafka Consumer)Text Data는 500ms 이내의 성능으로 Fig. 3-6과 같이 DB에 ms 단위로 적재되고, WebSocket을 통해 5초 단위로 모니터링 할 수 있도록 UI에 표출 되도록 설계 및 구현하였다.

pgAdmin 4

File • Object • Tools • Help

Browser

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_othership_voyage_hist

tb_ra_nav_safe_msg

tb_ra_nav_safe_msg_hist

tb_ra_nav_target

tb_ra_nav_telegraph

tb_ra_nav_telegraph_hist

tb_ra_nav_thruster

tb_ra_nav_thruster_hist

tb_ra_nav_voyage

tb_ra_nav_voyage_hist

tb_ra_rmea

tb_ra_rmea_hist

tb_ra_scomp_mst

tb_ra_scomp_mst_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_seeems_usr_rnl_map

tb_ra_seeems_usr_rnl_map_hist

tb_ra_seeems_usr

tb_ra_seeems_usr_hist

tb_ra_nav

Columns

Constraints

Indexes

Rules

Triggers

tb_ra_nav_aron_position

tb_ra_nav_aron_position_hist

tb_ra_nav_gps

tb_ra_nav_gps_hist

tb_ra_nav_hist

tb_ra_nav_navex

tb_ra_nav_navex_hist

tb_ra_nav_othership_position

tb_ra_nav_othership_position_hist

Columns

Constraints

Indexes

Rules

Triggers

Fig. 3-6 An example of database AIS(Other Ship) Data storage

서버 시스템 내 인터페이스는 Kafka 메시징 시스템 아키텍처로 정해진 Consumer로 구독되도록 구성하였다. 즉 Producer를 통해 생성된 메시지는 Consumer가 구독하면 본선/육상의 어플리케이션에서 지리적 조건과 상관없이 데이터를 인터페이스 되도록 하였다.

어플리케이션은 실험을 위해 웹 기반으로 구성하였으며, 실시간/과거이력 데이터를 활용하여 표출되는 모니터링 항목으로 구성하였다. 본 논문의 2장에서 설명한 것과 같이 웹에서의 효과적인 실시간 표출을 위해 WebSocket을 적용하여 활용하였다.

Fig. 3-7은 각 항해 장비로부터 수집한 주요 정보를 표출하는 UI로 구성되어 있으며, 선박의 ID와 시간, 현재 위치, 진행 방향, 선속, RPM을 표출한다. 중앙 영역은 각 장비로부터 받은 데이터를 값 및 Gauge로 표시하고, 수심, 선속, RPM 등은 Trend를 함께 표출하도록 구현하였다. 침로, 선속과 같이 값으로 표현되는 것은 선박에서 데이터를 구독하여 2초 단위로 업데이트 한다. Trend Data의 경우에는 표출 직전 30분 데이터를 DB로부터 조회하고 마지막 실시간 데이터를 더해 표출하도록 구현하였다.



Fig. 3-7 UI for navigation data monitoring

ENC(Electronic Navigation Chart)를 Web에서 사용할 수 있도록 이미지 처리하여 Fig. 3-8과 같이 자선의 72시간 동안의 항해 이력을 모니터링하고 원격에서 AIS 및 RADAR로부터 수신 받은 데이터를 통해 타선의 위치, 침로, 선속 등을 모니터링 할 수 있도록 구현 하였다.



Fig. 3-8 Monitoring based on electronic chart

Fig. 3-9은 메인엔진과 발전기의 상태를 모니터링 할 수 있는 UI이다. 일반적으로 기관 데이터는 각 센서와 직접 통신하는 것 보다는 앞서 2장 및 앞 절에서 설명한 것과 같이 Engine Data Logger 또는 AMS로부터 데이터를 수집한다. 각 데이터 항목의 주기는 일반적으로 10초이며, 본 논문에서는 해당 데이터를 상세한 정보를 10초 단위로 수집 저장하며, 수집되는 데이터의 종류는 150종이 넘는다. 따라서 DB에는 수집 가능한 데이터는 향후 신규 서비스 구현에 대한 활용 목적으로 모두 저장하도록 설계하여 구축하고, UI는 메인엔진과 발전기의 주요 데이터만 구현하였다.



Fig. 3-9 M/E, G/E Monitoring

제 4 장 운항 정보전송 시스템 구현 및 실험

본 논문에서 구현된 선박 운항 정보 수집 및 전송 시스템 실험은 한국해양대학교 실습선 한바다호에 환경을 구축하여 실험을 진행하였다. 실험은 각 단위 기능에 대해서는 Kongsberg 시뮬레이터와 실습선이 정박 중일 때 진행을 하고 데이터에 대한 안정성 및 기능에 대한 통합 테스트는 원양운항 기간에 실험을 진행하여 결과를 도출하였다.

4.1 단위 및 통합 성능 시험

선박의 운항 데이터에 대한 처리 및 모니터링을 실험하기 위해, 실제 운용한 선박의 데이터를 획득하였고, 선박의 LTE, VSAT(위성통신)와 같은 통신을 사용하므로 실험 기회를 많이 확보할 수 없어, 본 논문에서는 시뮬레이션 환경, 실습선의 정박 환경, 항해 환경으로 나누고 적합한 실험을 위해 Fig. 4-1과 같이 실험 환경을 구축 하였다. 또한 Fig. 4-2와 같이 각 장비로부터 데이터 수집을 위해 우선 기관 장비의 경우 한바다호의 AMS를 선교에서 직접 Modbus 통신을 위한 Port를 따로 생성하여 Serial로 연결하고, 각 항해 장비의 경우 Serial Line을 NMEA Buffer와 Combiner를 활용하여 케이블링 하였다. 항해 장비의 경우가용 가능한 Port 수가 제한적이기 때문에 NMEA Buffer를 사용하여 실험을 위한 서버에 데이터를 확장하여 사용할 수 있도록 하고, 서버의 포트수의 확장보다는 NMEA Combiner를 통해 수신 받은 NMEA Sentence 들을 하나의 Port에서 수신 받을 수 있도록 구성하였다.

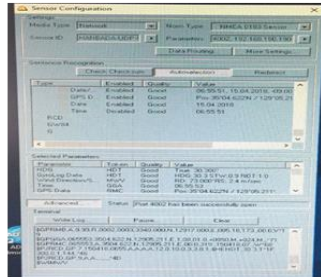
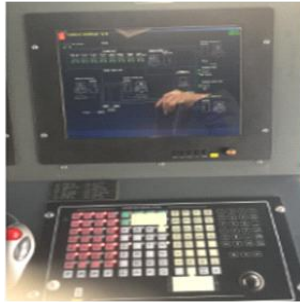


Fig. 4-1 Experiment preparation procedure



Fig. 4-2 System Experiment environment setup

또한 웹 서버, DB, 데이터 프로세서 역할을 할 수 있는 Table 4-1의 사양으로 2식을 구축하였다.

Table 4-1 H/W environment for ship's experiment

구분	사양
데이터 수집/처리	CPU: i3 4330 3.50 GHz, Memory: 8GB
웹서버	CPU: E5-2620 2.40 GHz, Memory: 16GB

육상에서는 데이터의 보안성, 접근성 및 구축의 효율성을 높이기 위해 클라우드 서비스인 AMAZON Web Service(Infrastructure as a Service: IaaS) EC2(가상서버) 프리티어에 구축하였다. AWS 프리티어는 OS를 구동하는 인스턴스 750시간 사용, 엘라스틱 로드 밸런싱 750시간 및 15GB 데이터 처리, EBS 30GB 및 스냅샷 스토리지 1GB, 모든 서비스를 합산하여 15GB의 데이터 전송 서비스를 제공 받을 수 있다. AWS의 경우 서비스의 지역 설정에 따라 요금제가 다르며, 본 연구에서는 미국 동부(버지니아 북부) Region을 적용하였다. 사용자 환경은 Chrome, Internet Explorer 11 브라우저를 통해 실험하였다.

실험에 사용할 주 데이터는 시뮬레이션 데이터와 실 데이터를 사용하였습니다. 시뮬레이션 데이터의 경우 Kongsberg 시뮬레이터에 있는 NMEA Generator를 사용하여 Gyro, DGPS, Doppler 등 항해 데이터를 발생시켜 기능 테스트를 진행하였고, Modbus 통신을 이용한 기관 데이터 처리는 진행하지 않았다. 실 데이터 수집 환경은 정박의 경우 RADAR, Doppler Log, 메인엔진 및 발전기는 구동이 안하는 상태로 테스트를 진행하였다. 즉, Table 4-2와 같이 개발 및 환경에 따라 단계별로 시뮬레이터 및 정박 중 실선 테스트는 단위 기능 위주의 실험을 진행하였고 통합 실험은 원양 항해 구간에서 진행하였다.

Table 4.2 Experiment environment of communication test

구분	통신	데이터 출처	실험 공간	실험 항목
단위 실험	Ethernet	시뮬레이터	KMOU 아산관	단위 기능 실험
	Ethernet	실 데이터	한바다호(Moored)	단위 기능 실험
통합 실험	LTE	실 데이터	한바다호(Sailing)	통합 기능/성능 실험
	VSAT	실 데이터	KMOU 아산관	통합 기능/성능 실험

4.2 실험 및 결과

4.2.1 단위 기능 실험 및 평가

각각의 분리된 시스템 기능에 대한 검증을 위해 각 항해/통신/기관 장비 별 데이터 수신과 데이터의 처리에 대한 단위 실험을 Table 4-3과 같은 항목으로 실시하여 결과를 도출하였다. 단위 실험에서는 다른 모듈과의 데이터 인터페이스 적합성, 모듈 내의 자료 구조상의 오류 유/무, 각종 오류들에 대한 처리 적절성 등을 실험하여 결과를 도출하였으며, 통합 실험 전에 보완하였다.

Table 4-3 Sample of single unit experiment result

실험 명	실험 항목	실험 조건/데이터	결과	확인
선속 데이터 수신	Serial I/F 연결 및 NMEA 수신	1. 선속(VBW) sentence를 출력 디바이스 연결 - Speed Log Sensor 연결 - NMEA emulator 사용 2. 연결된 양측 Serial 설정을 맞춤 3. 선속 데이터 송신	1. Serial port open 성공 2. 출력 측에서 송신한 NMEA string 그대로 System에서 수신 * 로그로 확인	OK
	재기동시 Serial I/F 자동 연결 및 NMEA 수신	1. 선속 NMEA sentence를 정상적으로 출력하여 System 에서 정상 수신 하고 있는 상 태 2. System 재기동을 하여 System 시스템 재 시작	1. 재기동 이후 Serial port open 성공 2. 출력 측에서 송신한 NMEA string 그대로 System에서 수신 * 로그로 확인	OK

선속 데이터 수신	NMEA 출력 기기 재 시작 시 Serial NMEA 수신	1. 선속 NMEA sentence를 정상적으로 출력하여 System에서 정상 수신 하고 있는 상태 2. NMEA출력 기기의 재시작 (Sensor, MEA emulator 재시작)	1. 출력 기기 재 기동 이 후에 출력 기기에서 출력 하는 NMEA string 그대로 System에서 수신 * 로그로 확인	OK
	수신 선속 데이터가 Valid 일 때 처리	1. NMEA emulator에서 송신 하는 선속 NMEA의 data status를 "A" 로 설정하여 송신	1. 수신한 선속 NMEA를 NDDS로 전송할 raw data의 선속 필드에 값을 저장 2. NDDS에 전송하기 전에 raw data가 최신 수신 값으로 계속 변경 -veLongiSogKn -veTransSogKn -veSternTransSogKn -veLongiStwKn -veTransStwKn -veSternTransStwKn	OK
	수신 선속 데이터의 단위 변환	1. NMEA emulator에서 송신 하는 선속 NMEA의 data status를 "A" 로 설정하여 송신 2. 데이터 값의 단위를 변경할 수 있으면 변경하여 송신	1. 수신한 수심 NMEA의 단위값을 NDDS 관리 단위 값으로 변환 (knot) * 로그로 확인	OK
	수신 선속 데이터가 Invalid일 때 처리	1. NMEA emulator에서 송신 하는 NMEA의 data status를 "V"로 설정하여 송신	1. 수신한 선속 NMEA의 data가 invalid일 경우 NDDS로 전송할 raw data의 선속 필드에 공백 값이 저장 * 로그로 확인	OK

수심 데이터 수신	Serial I/F 연결 및 NMEA 수신	1. 수심 (DBK, DBS, DBT, DPT) sentence를 출력 디바이스 연결 - Speed log sensor 연결 - NMEA emulator 프로그램 사용 2. 연결된 양측 Serial 설정을 맞춤 3. 수심 데이터 송신	1. Serial port open 성공 2. 출력 측에서 송신한 NMEA string 그대로 System에서 수신 * 로그로 확인	OK
	System 재가동시 Serial I/F 자동 연결 및 NMEA 수신	1. 수심 NMEA sentence를 정상적으로 출력하여 System에서 정상 수신 하고 있는 상태 2. System 재가동을 하여 System 시스템 재 시작	1. 재가동 이후 Serial port open 성공 2. 출력 측에서 송신한 NMEA string 그대로 System에서 수신 * 로그로 확인	OK
	NMEA 출력 기기 재시작시 Serial NMEA 수신	1. 수심 NMEA sentence를 정상적으로 출력하여 System에서 정상 수신 하고 있는 상태 2. NMEA 출력 기기의 재시작 (Sensor, NMEA emulator 재 시작)	1. 출력 기기 재가동 이후에 출력 기기에서 출력하는 NMEA string 그대로 System에서 수신 * 로그로 확인	OK
	Ethernet I/F 연결 및 450(NMEA) 수신	1. 수심 출력 장비와 450 Convertor 연결 - Speed Log sensor 연결 - NMEA Emulator 사용 2. System와 450 Convertor 연결 및 설정 3. 수심 데이터 송신	1. Serial port open 성공 2. 출력 측에서 송신한 NMEA string 그대로 System에서 수신 * 로그로 확인	OK

수심 데이터 수신	System 재기동시 Ethernet I/F 자동 연결 및 450(NMEA) 수신	1. 수심 NMEA sentence를 정상적 으로 출력하여 System에서 정 상 수신 하고 있는 상태 2. System 재기동을 하여 System 시스템 재시작	1. 재기동 이후 Serial port open 성공 2. 출력 측에서 송신한 NMEA string 그대로 System에서 수신 * 로그로 확인	OK
	NMEA 출력 기기 재시작시 UDP 450(NMEA) 수 신	1. 수심 NMEA sentence를 정상적으로 출력하여 System 에서 정상 수신 하고 있는 상 태 2. NMEA 출력 기기의 재시작 (Sensor, NMEA emulator 재 시작)	1. 출력 기기 재기동 이후 에 출력 기기에서 출력하 는 NMEA string 그대로 System에서 수신 * 로그로 확인	OK
	수신 수심 데이터 가 Valid일 때 처 리	1. NMEA emulator에서 송신 하는 수심 NMEA의 data status를 "A" 로 설정하여 송신	1. 수신한 수심 NMEA를 NDDS로 전송할 raw data 의 수심 필드에 값을 저장 2. NDDS에 전송하기 전에 raw data가 최신 수신 값으로 계속 변경 -deWaterDepthKeelM -deWaterDepthSurfaceM -deWaterDepthTransducer M -deOffsetFrmTransducer M -deMaxRngScaleM	OK
	수신 수심 데이터 의 단위 변환	1. NMEA emulator에서 송신 하는 수심NMEA의 data status를 "A" 로 설정하여 송신 2. 데이터 값의 단위를 변경할 수 있으면 변경하여 송신	1. 수신한 수심 NMEA의 단위값을 NDDS 관리 단위 값으로 변환 (meter) * 로그로 확인	OK

4.2.2 통합 기능/성능 실험

본 연구에 구성하는 시스템의 각 컴포넌트 간의 상호 작용을 검증하는 실험을 진행하였다. 하위 수준의 기능 검증을 배제하고, 서브시스템 간에 통합에 중점을 두고 실험을 수행하였으며 이를 통해 성능과 시스템의 안정성을 검증하였다. 이를 위해 Table 4-4과 같이 실험 결과를 도출하였다. 본 실험을 통해서 는 시스템으로부터 수신한 데이터가 정상적으로 DB에 적재되고, 기 구성된 웹 UI에 표출되는지를 확인하였다.

Table 4-4 Sample of integrated experiment result

실험 항목	화면/프로그램/DB	예상 결과	결과
LIVE 데이터 웹UI 표출	Monitoring Popup 화면에서 각 항목별 데이터 확인	정상 표출	OK 인터페이스 된 데이터에 대해 정상표출
LIVE 데이터 DB 적재	하기 테이블 데이터 적재확인	정상 적재	OK
	tb_ra_nav_voyage 항해정보		
	tb_ra_blr 발라스트		
	tb_ra_ge 발전기		
	tb_ra_ge_cyl 발전기 실린더		
	tb_ra_ge_tc 발전기 터보차저		
	tb_ra_mach 엔진룸 장비		
	tb_ra_mach_me 메인엔진		
	tb_ra_mach_me_cyl 메인엔진실린더		
	tb_ra_me_tc 메인엔진 터보차저		
	tb_ra_nav 항해장비		
	tb_ra_nav_gps 위치정보		
	tb_ra_nav_navtex 나바텍스		
	tb_ra_nav_telegraph 엔진 텔레그래프		
	tb_ra_nav_thruster 스러스터		

데이터가 각 서브시스템 간, 선박과 육상 간 전송 성능을 실험하기 위해 TPS(Transaction Per Second)를 측정하였다. 선박의 데이터 전송 실험을 Kafka의 설정을 통해 압축과 비압축으로 실험하여 결과를 도출하였다. Fig. 4-3과 같이 각 데이터 마다 0.5초~10초의 주기를 가지는 데이터를 수신한 결과 AIS 신호에 따른 총량이 다를 뿐 연속적으로 데이터를 수신하는 것을 확인하였다. 또한 선박과 육상의 통신 제약사항을 고려하여 5초간 데이터를 압축하여 전송한 결과 5초의 간격을 가지고 정확하게 전송하는 것을 확인하였다. Table 4-5는 압축과 비 압축의 메시지 전송량을 나타내고 있다. 비 압축은 실시간으로 선박 데이터를 VSAT를 통하여 전송하고 압축은 5초 단위로 전송하고 있다. 하루(24시간) 압축으로 하여 전송량은 300 Mbite이며 비 압축일시 915 Mbite 이다. 압축으로의 이점은 데이터를 5초 단위로 전송함으로 인하여 데이터 량을 더 많이 전송 할 수 있다. 데이터는 토픽으로 분류하여 실시간으로 전송해야 될 데이터와 5-10초 단위로 전송해야 될 것으로 분류하여 데이터를 수집한다. 이로 인하여 선박의 중요데이터를 어느 식으로 전송할지를 설계 단계에서 결정한다.

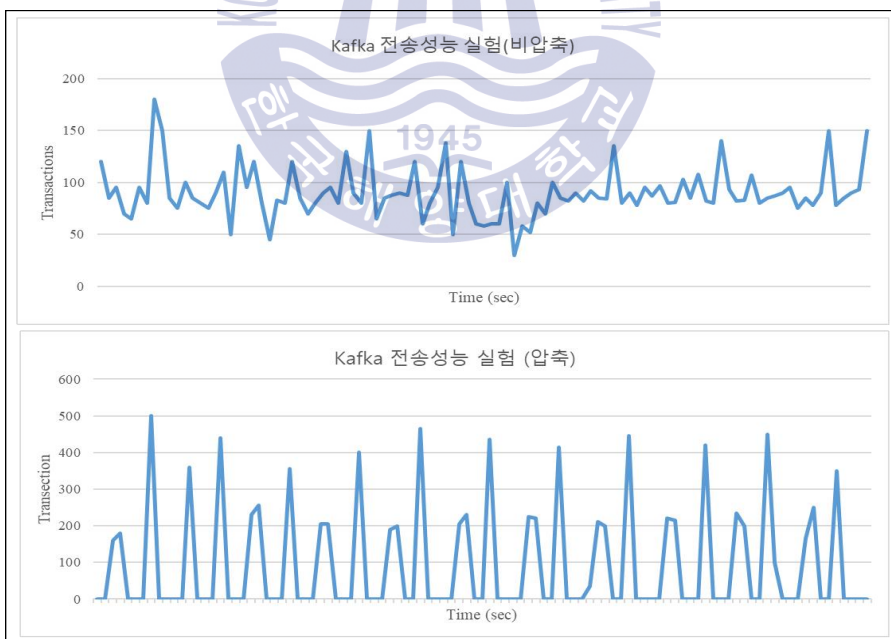


Fig. 4-3 Sample of integrated experiment result

Table 4-5 Comparison of data transmission experiment result

	비압축 (0.5 sec)	압축 (5 sec)
Message / sec	100	300
전송 용량	5.3 Kbite/ 0.5 sec	17.7 Kbite/ 5 sec
1일 전송 데이터	915 Mbite	300 Mbite
압축률	-	약 67%
장점	실시간 데이터 전송	대용량 데이터 전송
전송되는 데이터 채널	230개 채널	
VSAT Max 전송	0.5 Mbite / sec	

선박에서의 데이터 전송량은 선박 운항 비용에 직접적으로 영향을 미친다. 선박의 데이터 량을 전송 시 얼마 정도의 압축으로 하느냐는 앞으로 통신비용에 많은 영향을 미치게 된다. 현재 본 선박의 VSAT의 최대 전송 속도는 0.5 Mbite로서 여유는 있으나 현재는 모든 선박의 데이터를 전송하는 것이 아니라 일부 중요한 데이터에 한해서만 전송하게 되어있다. 현재 모든 데이터를 보낼 경우 더 많은 데이터의 전송 용량을 가질 수 있으나 통신 영상이라든지 다른 추가 데이터를 보내기에는 충분하지 않다. 앞으로 통신 기술의 발달로 인하여 많은 데이터 용량을 보낼 수 있을 거라 생각한다.

제 5 장 결론

본 연구에서는 선박 운항 데이터 및 이를 위한 원격에서 선박을 관제하기 위한 기술 확보 관점에서 선박의 다양한 장비로부터 발생하는 항해 및 기관 데이터를 각 단계별로 수집 및 처리를 할 수 있도록 연구를 진행하였다.

선박 운항 관제를 운용하기 위해서는 항해, 기관, 선체, 화물 등의 다양한 장비로부터 다양한 형태로 데이터를 수집하여야 하며, 항해 장비들은 인터페이스 표준 NMEA 0183 (IEC61162-1,2)을 통하여 수집되었고, 기관실의 정보는 AMS, IAS에 의하여 정보를 수집하였다. AMS 시스템은 기관 장비로부터 아날로그 신호를 디지털 신호로 전환하여 데이터를 가지고 있으며, 이런 선박 정보를 Modbus를 사용하여 장비로부터 추가 장비가 없이 정보를 수집할 수 있었다.

선박의 항해 장비로는 앞부분에 언급된 ECDIS, GYRO Compass, Speed Log, Doppler Log, AIS, WIND, RADAR, NAVTEX, Auto Pilot 등 육상에서 필요한 선박 정보를 수집하는 방법에 대하여 설계하였다. 이런 항해 장비의 정보를 수집하여 육상에 선박 관리 및 운용에 필요한 정보에 대해서 분석하고 수집하였다.

선박에서 기관 및 화물 데이터를 효과적으로 수집하기 위해서는 AMS에서 메인 엔진, 발전기, 보일러, 연료 탱크 및 장치, 각종 펌프, 배전반, 발라스트, 온도 센서, 압력 센서 등 많은 데이터를 수집할 수 있다. 각 선박의 종류나 탑재한 자동화 장비에 따라서 선박마다 운용 엔진, 발전기 등 장비들이 조금씩 다르며, 수집되는 정보도 조금씩 다르다. 따라서 AMS는 선박마다 장비의 체계 및 구성이 다르므로 제공되는 정보가 일치하지 않으므로 AMS 장비 업체로부터 AMS Channel List 및 통신 방식을 확정하여 MODBUS의 Master/Slave, Address, Coil, 데이터 Scale, Range, Function Code를 정하여 통신을 구현하였다.

MOM은 통합 대상 혹은 Client 간 Broker(중재)의 역할을 담당한다. Broker는 Client 간 연결, 메시지 라우팅, 보안, 메시지 저장, 모니터링 등의 기능을 수행하였고, 이러한 MOM은 메시징 시스템으로도 정의되며 대표적으로 ActiveMQ, ZeroMQ, RabbitMQ, Kafka 등이 있으나 Kafka를 사용하여 연구하였다.

Kafka의 메시지 처리 속도, 신뢰성, 확장성 등이 메시징 시스템으로써의 장점과 다른 메시징 시스템에 비해 성능이 우수하다고 판단되어, 본 연구의 시스템에서 아키텍처에 반영하여 적용하였다.

선박 및 육상에서의 데이터 수집 및 처리에 대한 분석, 데이터 전송에 관한 설계, 이를 이용한 응용 시스템 구현 및 검증에 관한 연구를 수행하여 이를 위한 기반 연구 지식을 확보하고 선박 데이터 처리와 전송에 대한 연구 성과를 얻었다. 또한 개념 연구 및 설계 뿐만 아니라, 본 연구를 통해 NMEA 0183, Modbus RTU로 수집되는 데이터를 수집 및 처리할 수 있는 모듈, 실시간으로 데이터를 처리하여 VSAT & LTE를 사용하여 데이터를 육상에 전송하여 메시징 시스템을 활용하여 Server System을 구현하였다. 선박의 데이터를 Web 기반으로 하여 실시간 육상에서 모니터링 할 수 있도록 구현하였다.

향후 자율운항선박 및 이 세부 항목인 원격 관제를 통한 선박 운용은 시스템 관점에서 선박과 육상간의 보다 복잡한 구조와 운항 선박의 데이터를 통한 다양한 연구를 진행할 만한 가치를 가지고 있다. 따라서 3가지 관점에서 향후 연구를 진행할 수 있다.

첫째, 선박 운항 및 관리 사용자 관점의 데이터 처리에 관한 연구를 진행할 수 있다. 사용자 어플리케이션은 데이터를 다양한 방식으로 재처리 및 사용하게 되고, 새로운 데이터를 생성할 것으로 예상 되므로, 본 연구의 아키텍처를 기반으로 향후 사용자 어플리케이션의 형태를 정의하고 이를 위한 데이터 처리 방법을 확장 설계할 수 있는 연구를 진행할 수 있다.

둘째, 본 연구에서는 선박의 데이터 업로드 기준으로 0.5Mbps의 통신 속도를 감안하여, 허용 가능한 수준의 데이터를 전송하였다. 본 연구에서는 230개의 선박 데이터를 안정적으로 전송할 수 있었으며 압축과 비 압축으로 분류하여 선박 운항 정보의 중요도에 토픽으로 나누어 전송할 수 있도록 구성하였다. 따라서 향후 통신의 발전 및 선박별 통신량이 증대 될 것을 고려하여 통신 트래픽에 따른 데이터 전송 규칙과 이를 위한 통신 트래픽 모니터링에 관한 연구를 진행할 수 있다.

셋째, 본 연구에서는 NMEA 0183, Modbus를 통해 데이터를 수집하였으나 통신 인터페이스 발달로 NMEA 2000 및 CAN 통신 등 새로운 인터페이스에 대한 준비가 필요하다. 자율운항선박에서는 보다 스마트한 센서와 영상 영상데이터와 같은 새로운 형식의 데이터 수집에 대한 정의와 연구가 요구되므로 이를 위한 연구를 진행할 수 있다.



참고문헌

- Adams, S.D., 2014. ReVolt - next generation short sea shipping.
<https://www.dnvgl.com/news/revolt-next-generation-short-sea-shipping-7279>
- Apache Kafka 0.8.1 Documentation. Available:
<http://kafka.apache.org/081/documentation.html>
- Banavar, G., Chandra, T., Strom, R. and Sturman, D., 1999, September. A case for message oriented middleware. In International Symposium on Distributed Computing (pp. 1-17). Springer, Berlin, Heidelberg.
- Blanke, M., Henriquez, M. and Bang, J., 2017. A pre-analysis on autonomous ships. *Danish Maritime Authority and the Technical University of Denmark*
- Burmeister, H.C., Bruhn, W.C., Rødseth, Ø.J. and Porathe, T., 2014. Can unmanned ships improve navigational safety?. In *Proceedings of the Transport Research Arena, TRA 2014, 14-17 April 2014, Paris*.
- Fette, I. and Melnikov, A., 2011. The websocket protocol (No. RFC 6455).
- FURUNO, 2009. Operator's Manual, Marine RADAR, Model 1835/1935/1945. Available :
http://www.furuno.com/files/Manual/34/upload/OME35790D1_MODE L1835_1935_1945.pdf
- Gjerset, A.Z., 2017. IoT and Modbus protocol development (Master's thesis, Høgskolen i Sørøst-Norge).
- IMO, N., 2007. 53/13. Development of an E-Navigation Strategy. Report of the Correspondence Group on enavigation, submitted by the United Kingdom. Sub-Committee on Safety of Navigation. International Maritime Organization, London, 20.

- Kreps, J., Narkhede, N. and Rao, J., 2011, June. Kafka: A distributed messaging system for log processing. In Proceedings of the NetDB (pp. 1-7).
- Levander, O., 2017. Autonomous ships on the high seas. IEEE Spectrum, 54(2), pp.26-31.
- Lloyds Register, 2016. Cyber-enabled ships: Deploying information and communications technology in shipping - Lloyds Register's approach to assurance. London: Lloyds Register.
- Magnoni, L., 2015. Modern messaging for distributed systems. In Journal of Physics: Conference Series (Vol. 608, No. 1, p. 012038). IOP Publishing.
- MODBUS ORG., 2012(a). MODBUS over Serial Line Specification & Implementation guide (ver 1.0), Available: <http://www.modbus.org>.
- MODBUS ORG., 2012(b). MODBUS Application Protocol Specification (ver 1.1b3), Available: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf
- Modbus, I.D.A., 2004. Modbus application protocol specification v1. 1a. North Grafton, Massachusetts (www.modbus.org/specs.php).
- National Marine Electronics Association, NMEA 0183 Standard for interfacing marine electronics devices, Ver. 3.01, January 2002.
- NFAS(Norwegian Forum for Autonomous Ships), 2017, Definitions for Autonomous Merchant Ship, <http://nfas.autonomous-ship.org/resources/autonom-defs.pdf>
- Olstedal, H.A. and Lützhöft, M., 2018. Autonomous ships, ICT and safety management. In *Managing Maritime Safety* (pp. 151-175). Routledge.

- Rødseth, Ø.J. and Tjora, Å., 2014, May. A system architecture for an unmanned ship. In *Proceedings of the 13th International Conference on Computer and IT Applications in the Maritime Industries (COMPIT)*.
- Rødseth, Ø.J., 2014. MUNIN Deliverable D4. 5: Architecture specification, August 1st 2013. Available from www.unmanned-ship.org. Accessed March
- Rødseth, Ø.J., Perera, L.P. and Mo, B., 2016. Big data in shipping-Challenges and opportunities.
- Rolls-Royce, 2016. Remote and autonomous ships the next steps. White paper.
- Tran P., Greenfield P., Gorton I., Behavior and Performance of Message-Oriented Middleware Systems. Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops, Vienna, 2002, pp. 645-650.
- Varga, E., 2016. Apache Kafka as a Messaging Hub. In *Creating Maintainable APIs* (pp. 187-201). Apress, Berkeley, CA.
- WebSocket.org, HTML5 Web Socket; A Quantum Leap in Scalability for the Web, <http://www.websocket.org/quantum.html>