

e-Navigation 운영시스템을 위한 RESTful 이종 데이터 서비스 시스템 아키텍처 설계

장 원 석*, 이 우 진**

Design of The RESTful Heterogeneous Data Service Architecture for Korean e-Navigation Operation System

Wonseok Jang*, Woojin Lee**

요 약 국제해사기구(IMO)는 해상을 항해하는 선박에 대해, 해상사고의 인적요인이 발생하지 않도록 e-Navigation이라는 새로운 해상안전 지원체계를 개발하고 도입하려 하고 있다. 한국은 e-Navigation 개발에 적극적으로 참여하여 e-Navigation 에 자체적으로 개발한 개념을 더해 한국형 e-Navigation인 '차세대 해양안전종합관리체계 기술'을 개발하고 있으며 2019년 도입을 목표로 하고 있다. 한국형 e-Navigation은 해양안전을 위한 다양한 기능을 제공할 수 있도록 설계되고 있는 만큼 각 기능별로 필요한 데이터의 종류가 공간 데이터와 일반 데이터, 파일, 기상 그리드등 다양하다. 따라서 이들 데이터를 eNavigation의 각 기능 서비스 시스템에 적절히 제공할 수 있는 시스템이 필요하게 되었다. 이에 본 논문에서는 한국형 e-Navigation에서 필요로 하는 데이터를 분석하고 이러한 이종 데이터를 제공할 수 있도록 REST API를 이용한 이종 데이터 서비스 시스템의 아키텍처를 설계하였다.

Abstract The International Maritime Organization is developing a maritime safety system called eNavigation in order to effectively respond to accidents occurring on board vessels. Korea is actively participating in the development of eNavigation and is developing Korean eNavigation by adding its own concept to eNavigation of the IMO. eNavigation is designed to provide various functions for marine safety. The data required for each function is various such as spatial data, relational data, file, weather grid. Therefore, there is a need for a system that can appropriately provide heterogeneous data suitable for eNavigation to each eNavigation's service system. In this paper, we analyzed the kinds of data needed for e-Navigation and designed the architecture of heterogeneous data service system that could provide these data properly.

Key Words : e-Navigation, REST, heterogeneous database, DSP

1. 서론

해상에서 일어나는 선박사고는 발생할 경우 인명 및 재산 피해의 규모가 막대한 경우가 대부분이다. 이와 같은 해상사고는 사람의 실수로 인해 발생하는 경우가 대부분으로, 사고발생의 인적 요인에 대해 적절한 지원 및 예방을 수행할 수 있는 체계가 있다면

해상 선박사고는 크게 저감될 수 있다. 국제해사기구(International Maritime Organization, IMO)는 점점 커져가는 해상에서의 선박사고에 대해 위와같은 해양 안전 지원을 할 수 있는 체계를 개발하고 있다[1]. 2005년부터 시작된 해양안전지원체계는 e-Navigation이라는 이름이 지어졌고, IMO를 중심으로 세계 각국에서 기존의 해양안전체계를 통합하

*Naval R&D Center, Hanwha Systems/Computer Science and Engineering, Kyunpook University(cws0714@hanwha.com)

**Corresponding Author : School of Computer Science and Engineering, Kyungpook University(woojin@knu.ac.kr)

Received October 17, 2018

Revised November 10, 2018

Accepted January 17, 2019

고 새로운 기술을 도입하며 활발한 연구가 진행되고 있다[2-3]. 국토의 3면이 바다와 접촉해 있고 해상 교통량이 많은 한국은 e-Navigation 개발의 초기단계부터 참여하였고, 2015년부터 정부 주도로 한국에 적합한 e-Navigation 체계를 연구 개발하고 있으며 2019년 도입을 목표로 하고 있다. 개발 초기인 2015년에 한국 해양수산 개발원에서는 e-Navigation이 지원해야 할 기능을 도출하고자 실제 e-Navigation을 사용하게 될 선박 운항자 및 일반인들을 대상으로 필요 기능을 설문조사하였고, 그 결과 다음과 같이 조사되었다[4].

표 1. e-Navigation의 기능 설문조사 결과[4]

Table 1. Survey result of e-Navigation's service item

Service Group	eNavUser	Partk	Public	Partk
Real-time ENC update and navigation safety information	40.8%	1	41.3%	1
High-speed wireless communication within 100km at the sea	19.4%	2	14.0%	3
Provide VTS and maritime traffic information	16.0%	3	10.3%	5
Emergency situation support	12.6%	4	13.7%	4
Development the various application for the ship	8.7%	5	15.0%	2
Provide the weather information	2.4%	6	5.7%	6

이 조사결과를 기반으로 한국의 e-Navigation은 항해위험도 평가, 최적안전항로 제공, 선내 상황 원격 모니터링, S-101 전자해도 스트리밍과 같은 대표적인 서비스항목을 도출하고 이를 제공할 수 있도록 설계, 개발되고 있다. 도출된 다양한 기능을 제공하기 위해서는 공간 데이터, 일반 데이터, 파일, 기상 데이터 등 다양한 형태의 데이터들이 필요하므로 e-Navigation 시스템의 데이터 서비스를 담당하는 모듈은 이에 적합하게 설계될 필요가 있으나, 아직 e-Navigation이라는 개념은 국제해사기구에서 개념을 만들고 개발중인 단계로 구체적인 설계 연구 사례가 없다.

따라서, e-Navigation 시스템의 데이터 서비스 시스템을 설계하기 위해 본 논문에서는 먼저 e-Navigation 시스템의 세부 기능 모듈의 구조와

필요 데이터를 분석하였다. 그리고 이를 기반으로 e-Navigation 시스템에서 이종 데이터를 적절히 저장하고 제공할 수 있는 데이터 서비스 시스템의 구조를 설계하였다. 이와 함께, 데이터를 적절하고 간편하게 전달할 수 있도록 REST API를 이용한 기능 모듈과 데이터 서비스 모듈간 통신 구조를 제안한다.

2. e-Navigation 시스템

한국의 e-Navigation은 최적안전항로 제공등의 서비스를 제공하기 위해 다음과 같은 구조로 시스템을 설계하고 있다.

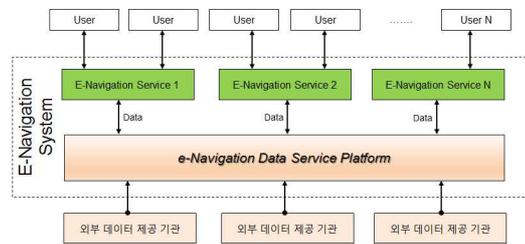


그림 1. 한국형 e-Navigation 운영시스템의 구성
Fig. 1. Korean e-Navigation operation system's structure

그림 1과 같이, e-Navigation 시스템은 다수의 서비스 시스템과 외부 데이터를 수집하고 기능 모듈에 필요한 데이터를 제공하는 데이터 서비스 모듈로 구성되어 있다. e-Navigation 시스템은 데이터를 각 모듈에서 별도로 갖지 않고 하나의 데이터 서비스 모듈에 집중시킴으로써 데이터의 보안성, 관리의 효율성 등을 재고하고 있다.

기능 모듈 간 통신 및 외부데이터수집 모듈과 데이터 서비스 모듈간의 통신은 DDS(Data Distribution Service) 를 이용하여 빠르고 간편한 데이터 메시지 전달이 이루어지도록 하였다[5]. 각 기능 모듈 간에 교환하는 데이터 메시지는 그 크기가 크지 않으며, 해양안전종합정보시스템(General Information Center on Maritime Safety and Security, GICOMS)과 같은 외부 시스템으로부터 전송되는 데이터 메시지도 1 패킷당 100바이트를 넘지 않으므로 DDS 통신을 이용한 빠른 통신이 적

합하다.

그러나 DDS 통신은 작은 크기의 데이터 메시지 교환에는 적합하나 큰 크기의 데이터 전송에는 부적절하다. 경우에 따라 대용량 데이터를 전달해야 하는 기능모듈과 데이터 서비스 모듈간의 데이터 통신에는 DDS가 적합하지 않다. 데이터 서비스 모듈에 제공해야 하는 공간데이터나 선박의 정적 데이터 및 S101전자해도 데이터는 보통 그 크기가 크므로 따라서 기능 모듈들과 데이터 서비스 모듈간의 통신은 DDS가 아닌 다른 방법의 통신방식으로 설계되어야 한다.

3. REST API와 AMQP의 적용

3.1 REST API를 이용한 데이터 통신

REST(REpresentational State Transfer)는 로이 필딩이 2000년 그의 박사논문에서 제안한 기술로, 네트워크 상에 존재하는 특정 자원을 정의하고, 그 자원에 접근할 수 있는 주소를 지정할 수 있도록 한다[6]. 본 논문에서 제안하는 DSP는 각 기능모듈간의 인터페이스가 복잡하고 계속해서 기능이 확장될 가능성이 커 인터페이스 방법은 높은 범용성을 가져야 한다. 또한 각 기능모듈과 DSP는 데이터를 사용하는 클라이언트와 제공하는 서버의 구조로 설계되므로, 인터페이스는 이런 구조에서 사용하기 적합해야 한다. 그리고 빠르고 간단하게 기능 모듈에서 필요로 하는 데이터나 자원에 대한 접근, 검색, 제공이 이루어져야 한다. REST방식은 위와 같은 필요 조건을 충족시키면서 그 사용이 간편한 이점이 있어

DSP의 적용에 적합하였다. 그림 2는 REST 방식을 적용한 e-Navigation DSP의 구조를 나타낸다.

아래의 URL은 DSP에서 데이터 자원의 주소를 지정하는 예이다.

URL : *http://test.dns.xxx:0000/PrtAtCodeFInfo*

REST API로 DSP의 데이터 자원에 접근할때는 위와같이 URL을 이용해 접근할 수 있다. 여기서, PrtAtCodeFInfo 는 데이터베이스에 존재하는 테이블로 이 테이블의 특정 데이터에 접근하기 위해서는 아래와 같은 조건을 추가할 수 있다.

조건 : *"prtAtCode=pa9"*

이렇게 URL을 특정하여 REST API를 이용해 GET 명령을 수행하게 되면 아래와 같은 결과를 JSON으로 반환받게 된다.

```
{"success":true,"msg":null,"data":{"prtAtCode":
"pa9", "prtAtKorNm": "항만
9","prtAtEngNm":"prt9"}}
```

공간 데이터의 경우 데이터 베이스의 공간 데이터를 DSP의 공간 DB 컨트롤 모듈에서 POJO 또는 GML Text 형태로 변환하여 각 기능 모듈로 전달한다. 공간 데이터의 경우 점이나 선, 면과 같은 지오메트리의 형태를 표현하는 경우가 많으므로 전달되는 GML Text에는 데이터의 형태, 좌표가 태그되게 되며 그 형태는 아래의 예와 같다.

```
<gml:exterior><gml:LinearRing><gml:posList>
128.9953424 35.0633913 129.0008546
35.0647029 129.0024044 35.0583517
128.9976923 35.0573973 128.9953424
35.0633913 </gml:posList>
</gml:LinearRing></gml:exterior>
</gml:PolygonPatch></gml:patches>
</gml:Surface>
```

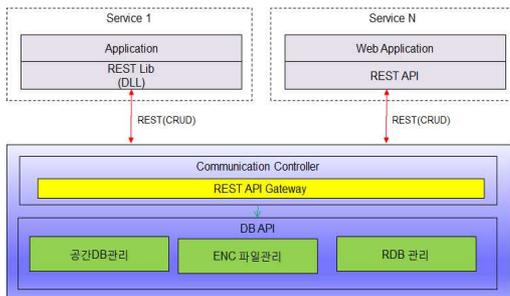


그림 2. REST를 이용한 eNavigation DSP 구조
Fig. 2. eNavigation DSP structure using REST

위와 같이 DSP는 REST API를 이용하여 관계형 데이터나 공간데이터를 기능 모듈로 제공할수 있다. 기능모듈은 실제 데이터의 종류나 구조에 관계없이 정해진 API를 이용하여 간단히 데이터를 사용할 수 있다. 또한 기능 모듈 뿐 아니라 REST API를 이용하여 Web에서도 e-Navigation의 데이터에 접근이 가능하므로 향후 Web 기반의 기능확장 용이성도 고려하여 설계하였다.

3.2 Message Broker

e-Navigation 시스템을 위한 DSP는 GICOMS, V-PASS등과 같은 외부 시스템과의 연동을 통해 해상을 항해하고 있는 선박의 정도등을 실시간으로 전달받아야 한다. 이러한 데이터는 약 7만척에 달하는 한국의 선박들에 대한 데이터를 수신하게 되므로 고빈도, 고용량의 데이터가 될 가능성이 높다. 외부의 데이터는 e-Navigation 시스템 내에서 DDS 메시지 형태로 Publish 되게 되므로 각 서비스 시스템들이 직접사용할 수 있으나 시스템 과부하와 같은 이유로 데이터가 누락될 수 있다.

이러한 환경적 특성에 따라 본 논문에서는 DSP에 데이터의 버퍼 역할을 하면서 DSP 내부에서 각 패키지 간 데이터의 중계 역할을 수행할 수 있는 메시지 브로커를 두어 데이터의 전달을 보장할 수 있도록 설계하였다. 여기서 메시지 브로커는 AMQP(Advanced Message Queuing Protocol)를 이용하였다. AMQP는 느슨하게 결합된 시스템간의 통신에서 상호 정보교환이 가능한 메시지 큐 프로토콜을 의미한다. 그림 3은 AMQP의 동작 구조를 나

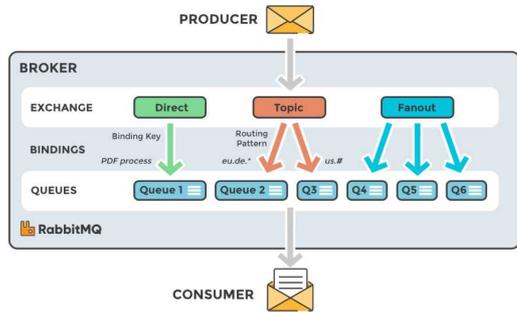


그림 3. AMQP의 동작 개념[7]
Fig. 3. Concept of AMQP

타낸다. 본 논문에서는 AMQP 프로토콜을 구현한 오픈소스 메시지 브로커인 RabbitMQ를 사용하여 설계를 검증하고 테스트하였다.

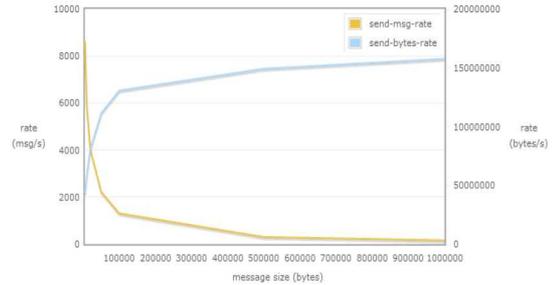


그림 4. RabbitMQ의 성능 분석[8]
Fig. 4. Performance analysis of RabbitMQ

RabbitMQ는 그림 4와 같이, 100,000byte 크기의 데이터 메시지를 초당 1,000개 이상 처리할 수 있는 고성능의 메시지 큐로, 고빈도, 고용량 데이터 메시지 처리가 필요한 eNavigation DSP의 데이터 메시지 큐로 적당하다.

4. e-Navigation Data Service Platform 아키텍처 설계

이 논문에서는 e-Navigation을 위한 데이터 서비스 시스템의 아키텍처를 그림 5와 같이 설계하고, Data Service Platform(DSP) 로 명명하였다. e-Navigation 시스템 내에서 DSP는 각 기능모듈과 상호작용하여 데이터를 저장, 전달하는 것이 가장 큰 목적이다. 또한 DSP는 e-Navigation 시스템과 연결되어 데이터를 전송하는 외부 시스템의 데이터를 받아 이 데이터를 e-Navigation 시스템에 적합하게 가공하고 저장하는 기능도 수행하게 된다. 이러한 목적에서, 본 논문의 DSP는 크게 Database의 기능을 담당하는 부분과 데이터 통신을 담당하는 부분으로 구분하였다.

데이터를 담당하는 부분은 실제 공간 DB, 관계형 DB, 파일서버등의 시스템과 연동하여 데이터를 저장, 삭제, 수정, 검색, 읽기등을 수행하는 패키지와

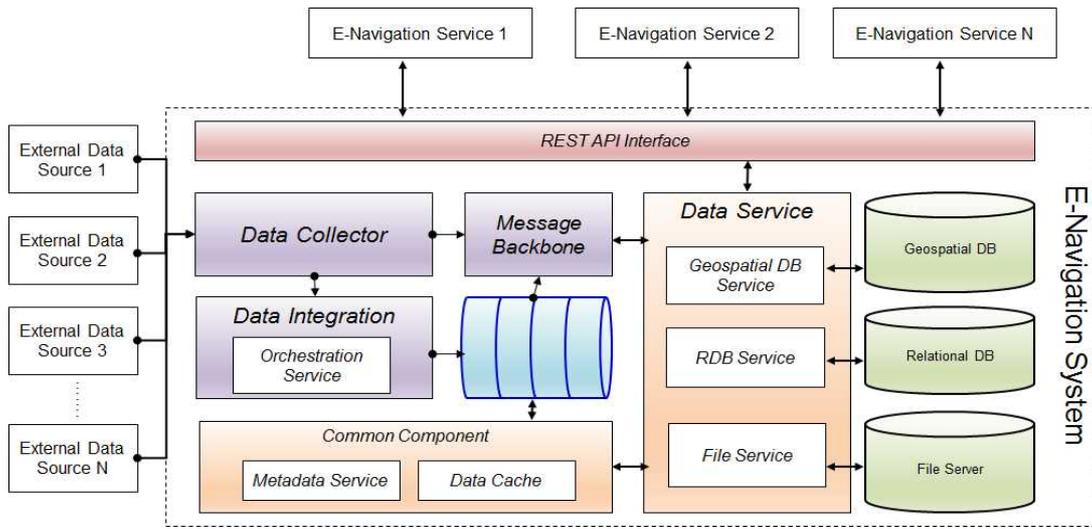


그림 5. e-Navigation DSP의 아키텍처
Fig. 5. Architecture of e-Navigation DSP

외부 기관에서 전송받은 데이터를 결합, 수정하는 패키지로 구성하였다. 데이터 통신을 담당하는 부분은 REST API를 이용하여 기능 모듈들과의 통신하는 기능을 담당하는 패키지와 외부기관의 데이터를 수

표 2. DSP의 패키지
Table 2. Data Service Platform's Packages

Package	Desc
API Interface	Service interface using REST
Data Collector	Data Collection interface for external data
Message Backbone	Networking Backbone for DSP's packages. Send the Received data from Data Collector Package to Data Integration Package.
Data Service	Provide the Data Service function. Return the retrieve result to requested service via API interface package.
Data Integration	Save the collected data into internal data storage using Data Service Package's function
Common Component	Provide the cache, metadata service

집하는 패키지로 구분하였다. 그리고 이 두 핵심 기능이 원활히 작동할 수 있도록 메시지 백본과 일반적인 기능을 제공하는 패키지를 두었다. 표 2는 이들 패키지를 정리한 것으로, DSP를 구성하는 패키지의 기능을 설명한다.

그림 6은 DSP의 패키지들의 관계를 나타낸 패키지 다이어그램이다.

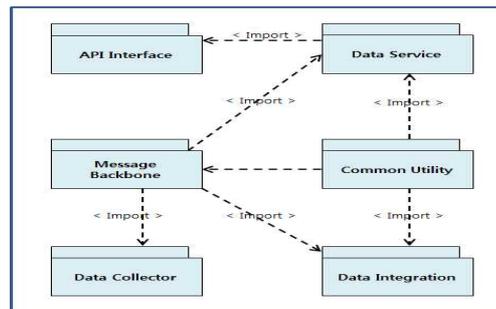


그림 6. DSP의 패키지 다이어그램
Fig. 6. DSP's Package Diagram

Data Service 패키지는 표 1에서 기술한 것처럼 DSP의 핵심 기능을 담당한다. 그림 7과 같이, Data Service 패키지는 메타 데이터 서비스와 캐시 매니저를 통해 데이터를 검색, 제공, 저장하는 컴포넌트

인 Data Service Controller와 데이터의 전달을 담당하는 Communication Controller, 초기 기동시 초기값 설정, 연결설정등을 담당하는 Data Service Initializer, 그리고 DSP 내에 저장된 데이터를 조회하기 위한 Data Viewer Controller로 구성하였다.

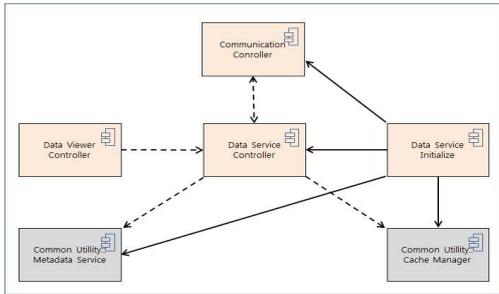


그림 7. Data Service 패키지의 클래스 다이어그램
Fig. 7. Data Service package's Class Diagram

Data Service Controller는 그림 8과 같이 설계하였다. Data Service Initializer 클래스는 Data Service 패키지 내 Component 및 Class들의 초기화를 담당한다. Main Data Service Provider는 내부 저장소 유형에 따라 RDB, File, Geospatial Service Provider를 사용하여 서비스 요청을 처리하고, 데이터에 대한 결합 또는 처리와 같은 오케스트레이션이 필요할 경우 Orchestration Service를 사용한다. Main Data Service Provider는 하위 Service Provider들이 제공한 API 목록을 API Entry Class로 관리한다

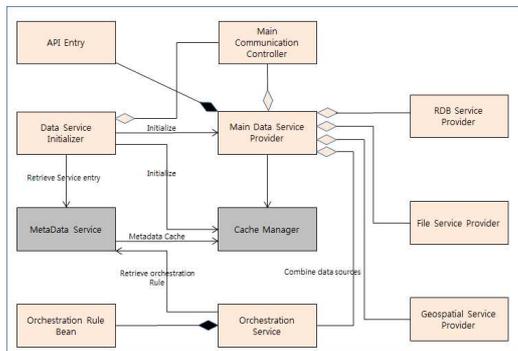


그림 8. Data Service Controller의 클래스 다이어그램
Fig. 8. Data Service Controller's class diagram

Communication Controller는 REST, DDS, 그리고 AMQP와 연결되는 메시지 백본을 제어하고 관리한다. 그림 9는 Communication Controller의 구조를 나타낸다. DDS Communicator는 DDS 인터페이스를 통해 DDS 패킷을 처리하며, REST Communicator는 서비스 시스템이 전송하는 REST 메시지를 처리한다. 메시지 백본은 AMQP와 연결되어 큐의 데이터를 처리하거나 큐의 상태를 감시하도록 설계하였다.

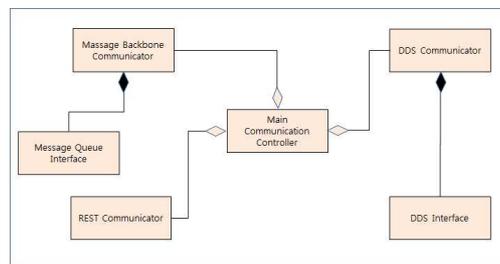


그림 9. Communication Controller의 클래스 다이어그램
Fig. 9. Communication Controller's class diagram

5. 데이터 전송 시험

e-Navigation 시스템의 실제 서비스 시스템들과 외부 시스템들을 DSP에 연결하는 대신 Mockup 소프트웨어를 제작하여 DSP에 연결하고 동작을 시험하였다. 시험에 사용된 구성요소는 그림 10과 같다.

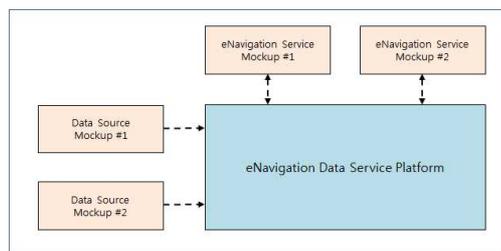


그림 10. DSP 테스트 환경
Fig. 10. DSP Testing Environment

서비스시스템 mockup은 DSP로 REST API를 이용해 데이터를 요청하고 그 결과를 확인하도록 하였다. 테스트는 e-Navigation 시스템에서 사용되는 공간데



그림 11. eNavigation Service system Mockup
 Fig. 11. eNavigation Service system Mockup

이터, 관계형 데이터, 파일, 기상 그리드 데이터 중 공간 데이터, 관계형 데이터의 정상적인 전달 여부 시험을 수행하였다. 파일, 기상 그리드 데이터는 REST 방식이 아닌 별도의 방식으로 제공되므로 테스트에서 제외하였다. 그림 11은 서비스 시스템 mock업(Mock Up) SW로, 공간데이터 및 관계형 데이터 등의 REST 요청기능과 수신된 데이터 확인기능등을 구현하여, DSP의 정상동작을 확인할 수 있도록 하였다.

공간 데이터의 경우, e-Navigation에서 사용되는 공간 데이터는 S101 표준을 따르는 S101 전자해도의 데이터를 미리 공간DB에 적재하고 테스트하였고 테스트에 사용된 공간DBMS는 PostgreSQL과 GIS Extension인 PostGIS이다.

관계형 데이터는 e-Navigation 시스템에서 주로 사용되는 선박 정적, 동적 데이터의 정상적인 조회 및 전달여부를 테스트하였다.

6. 시험결과 및 토론

관계형 DB의 선박 정적정보 데이터 검색에는 아래와 같은 시험용 REST API가 사용되었다.

GET / URL/mrn=SHIPMrn1

여기서 mrn은 선박의 번호로, 선박정적 정보 Table 의 PK가 되는 필드이다. 위와 같은 명령을 서비스 시스템 mock업을 통해 DSP로 전달하였을 때 아래와 같이 정상적으로 데이터가 수신됨을 확인하였다.

```
{"success":true,"msg":null,"data":{"shipMrn":"SHIPMrn1","mmsi":"123456789","dateTime":"","shipName":"","callSign":"","shipType":0,"dima":0,"dimb":0,"dimc":0}}
```

공간DB는 경도 129.001, 위도 35.06인 지점에서 1000NM의 범위에 있는 Object를 검색하면 REST 명령을 DSP로 전송하여 정상 동작 여부 시험을 수행하였다. 결과는 GMLText 형식으로 표현되며, 아래와 같은 결과가 정상적으로 반환되었다. GMLText의 경우 긴 데이터가 반환되므로, 검색된 오브젝트의 좌표 자료만을 제시하였다.

```
....<gml:posList>128.9953424 35.0633913  
129.0008546 35.0647029 129.0024044  
35.0583517 128.9976923 35.0573973  
128.9953424 35.0633913</gml:posList>...
```

상기의 결과와 같이, 공간 데이터와 관계형 데이터와 같은 데이터의 종류와 상관없이 데이터를 정상적으로 전달받음을 확인할 수 있었다.

본 논문에서 수행한 테스트는 DSP의 아키텍처가 정상적으로 동작하는지의 여부 검사로 한정하였다. 데이터 검색 및 전송에 대한 속도 검사와 데이터 부하 테스트는 이후 DSP를 구성하는 각 컴포넌트를 상세히 설계하고 구현이 완료된 후 수행할 예정이다.

본 논문에서 제안하는 e-Navigation 시스템의 데이터 서비스 플랫폼(DSP) 아키텍처의 목적은 데이터에 접근하는 서비스 시스템에 대해 각 서비스가 필요한 데이터만을 접근할 수 있도록 하며 나머지 데이터나 구조에 대해서는 캡슐화하여 보안성을 높이는 것이다. 그리고 각 서비스 시스템을 개발하는 개발사에서는 REST API를 이용하여 DB 구조를 알 필요없이 간편하게 DB를 이용할 수 있도록 한다. 동시에 WEB 과 응용 소프트웨어 어디든 e-Navigation의 데이터에 쉽게 접근할 수 있도록 확장성을 높이는 것이다. 본 논문의 DSP 아키텍처는 이러한 목적에 부합하면서 REST API를 이용하여 쉽게 데이터를 취득할 수 있도록 함을 실험을 통

해 확인하였다.

그러나, 공간 DB의 경우 검색된 오브젝트의 양이 클 경우 데이터의 전송시간이 늘어날것으로 예상되므로 이에 대한 고려를 포함하여 지속적으로 DSP 기능과 성능을 개선할 필요가 있다.

7. 결론

점차 중요해 지고 있는 해상에서의 안전사고 예방 및 효과적인 대응을 위해 국제해사기구를 중심으로 개발하고 있는 e-Navigation이라는 체계는 삼면이 바다로 둘러싸인 한국에서는 특히 중요하다. AIS 가 장착된 선박만 7만척 이상에 달하고 해상 운송량도 상당한 한국은 일찍부터 e-Navigation을 도입하고자 노력하고 있다. 이에 한국의 산업계, 학계, 정부 부처의 공동노력으로 한국형 e-Navigation의 개발이 빠르게 진행되고 있다. 한국형 e-Navigation에서 제공될 서비스는 사고취약선박 모니터링, "사고취약선박 선내지원 모니터링", "최적안전항로 지원", "소형선박용 전자해도 스트리밍", "도선 및 예선 지원", "해양 안전정보 제공"이 대표적으로, 이들 서비스를 중심으로 개발되고 있다. 이러한 한국형 e-Navigation 시스템은 국제해사기구에서 제안된 e-Navigation이라는 개념을 최초로 실체화 하고 있음으로 인해 최종 개발 결과물은 국제적으로 중요한 참고물이 될 것으로 예상된다. 한국형 e-Navigation에서 제공하고자 하는 서비스는 새로운 전자해도 표준인 S101을 따르는 전자해도 파일과 실시간 선박 정보, 지형과 위치정보를 제공할 공간데이터등 다양한 종류의 데이터를 필요로 한다.

이에 본 논문에서는 이러한 다양한 종류의 데이터를 필요로 하는 서비스 시스템에 효과적으로 제공할 수 있으면서 사용이 간단하고 향후 확장성을 가질수 있는 데이터 서비스 플랫폼의 아키텍처를 설계하고 제안하였다. 손쉬운 접근을 위해 데이터 사용시 REST API를 이용하여 데이터의 CRUD를 수행할 수 있도록 설계하였다. 그리고 제안한 DSP 아키텍처를 기반으로 DSP의 프로토타입을 구현하고 데이터의 전달, CRUD 등 시험해 하여 정상적으로 기능

함을 확인하였다.

그러나 본 논문에서 제안한 아키텍처에서는 이중 데이터 서비스를 효과적으로 수행할 수 있는 보다 깊은 수준의 클래스 설계나 데이터 흐름 설계는 포함되지 않았다. 향후 우리는 본 논문에서 제안한 아키텍처를 보다 고도화하여 기능과 성능을 향상시키고 그 아키텍처를 기반으로 데이터 서비스 플랫폼의 개발을 진행할 것이다. 그리고 개발된 DSP를 실제 e-Navigation 시스템에 적용하고 그 성능과 안정성, 기능, 부하에 대한 강인성등을 테스트하고 검증할 예정이다.

REFERENCES

- [1] IMO MSC, Report of the Maritime Safety Committee on Its Eighty-Fifth Session, MSC 85/26/Add.1, Annex 20, Jan. 2009.
- [2] IMO NAV, Development of E-navigation Strategy, Report of the Working Group, NAV 54/WP.6, July 2008.
- [3] Dr. Sally Basker, e-Navigation: The way ahead for the maritime sector, 2005
- [4] SY Kim, HC Lee, KW Lee, HJ Lee, "A Study of the utilization plan for Korean e-Navigation", K MI Report, 2015.12.
- [5] WS Jang, BJ Kim, MS Kang, "Analysis of the Communication Middleware for Service Integration of Korean e-Navigation Operation Center", KIICE 2018 conference proceeding, pp274-276, 2018
- [6] Roy Thomas Fielding, "Architectural Styles and the Design of Network-based Software Architectures", Doctoral Dissertation, University of California, 2000
- [7] Lovisa Johansson, "RabbitMQ for beginners - What is RabbitMQ?", <https://www.cloudamqp.com/blog/2015-05-18-part1-rabbitmq-for-beginners-what-is-rabbitmq.html>, 2015
- [8] Adam Bloom, "How fast is a Rabbit? Basic RabbitMQ Performance Benchmarks", <https://blogs.vmware.com/vfabric/2013/04/how-fast-is-a-rabbit-basic-rabbitmq-performance-benchmarks.html>, 2013

저자약력

장 원 석(Wonseok Jang)

[정회원]



- 충남대학교 컴퓨터공학과 석사
- 07~16년 한국해양과학기술원 연구원
- 현 한화시스템 해양연구소 과장
- 현 경북대학교 컴퓨터공학과 박사과정

관심분야는 해양 무인체계, 해양 항법, 해양 안전 시스템.

이 우 진(Woojin Lee)

[정회원]



- KAIST 전산학과 박사
- 99~02년 ETRI 선임연구원
- 현 경북대학교 컴퓨터학부 교수

관심분야는 임베디드 소프트웨어 테스트, 임베디드 소프트웨어 개발환경