한국항해항만학회지 제38권 제4호
*J. Navig. Port Res.* Vol. 38, No. 4 : 385-390, August 2014 (ISSN:1598-5725(Print)/ISSN:2093-8470(Online))
DOI : http://dx.doi.org/10.5394/KINPR.2014.38.4.385

# 알고리즘 적용이 용이한 시뮬레이터 개발 지원 도구에 관한 연구

† 이영주 · 김아영* · 박세길** · 오재용*** · 김정수****

† ,*, ** , *** 한국해양과학기술원 부설 선박해양플랜트연구소 해양안전연구부
**** 삼성탈레스주식회사 해양/시스템연구소

# A Study on the Support Tool for Simulator Algorithm Development

† *Yeong-Ju Lee · Ah-Young Kim* · Se-Kil Park** · Jae-Yong Oh*** · Jeong-Soo Kim****

† *, *, **, *** Maritime Safety Research Division, Korea Research Institute of Ships & Ocean Engineering, Daejeon, Korea*
**** Naval/System R&D Center, Samsung Thales, Gumi, 730-030, South Korea*

**요 약** : 시뮬레이터는 다양한 형태와 기능을 갖춘 여러 장치들로 구성되어 있으며, 이 장치들이 네트워크로 연동된 복잡한 구조를 이루고 있다. 이러한 이유로 시뮬레이터 개발 및 유지보수 과정에 많은 시간과 비용이 소요된다. 시뮬레이터의 성공적인 개발을 위해서는 관련 전문가들이 협력하고 업무를 분담하여 병렬적으로 수행하는 것이 이상적이지만 업무의 상호 의존성이 이를 어렵게 한다. 본 논문에서는 시뮬레이터 개발 업무를 알고리즘 구현과 이를 제외한 시스템 구현으로 나누어 두 업무의 상호 의존성을 낮추고 전문가를 지원하는 방안을 검토하였다. 특히 알고리즘 구현을 담당하는 도메인 전문가의 요구 사항을 분석함으로써 시뮬레이터 개발 지원 도구를 설계하고, 이를 활용한 시뮬레이터 개발 절차를 제안하였다. 또한 도메인 전문가의 알고리즘 개발 지원과 유연한 데이터 관리를 위해 데이터셋 개념을 도입하였고, 시뮬레이터 장비들이 유연하게 재구성될 수 있도록 네트워크 구조를 설계 하였다. 시뮬레이터 개발 지원 도구를 통해서 도메인 전문가는 알고리즘 개발에 전념할 수 있고, 효율적인 협업이 가능할 것으로 기대된다. 또한 개발 절차가 체계화 되고 더 명확해지기 때문에 개발 계획 및 관리가 용이해질 것으로 예상 된다.

**핵심용어** : 시뮬레이터 개발, 알고리즘 개발, 도메인 전문가, 개발 지원 도구, 시뮬레이터 개발 절차

**Abstract** : *Simulator is composed of several devices that have a variety of forms and functions. These devices are connected to each other by a network intricately. For this reason, simulator development and maintenance process require a lot of time and money. In order to successfully develop the simulator, it is ideal that related professionals share the work and work together in parallel. However, development is carried out inefficiently, because task interdependence makes it difficult to work in parallel. In this paper, the developments of the simulator were classified into algorithm development and system development, and it was discussed how to lower the interdependence of these two tasks and support professionals. In particular, based on the requirements analysis of the domain experts responsible for the development of the algorithm, we designed the support tool for simulator development and proposed development process using this tool. We also introduced the concept of a DataSet in order to support algorithm development of domain experts and manage data flexibly. And we designed network architecture to enable flexible reconfiguration of simulator equipment. By using the tools to support the simulator development, domain experts are able to concentrate on algorithm development and it is expected to be effective collaboration. In addition, the development plan and management are expected to be easy because the development process is systematic and clearer.*

**Key words** : *simulator development, algorithm development, domain expert, development support tool, simulator development process*

## 1. 서 론

시뮬레이터는 활용 용도와 규모에 따라 차이는 있지만 일반적으로 Fig. 1과 같이 다양한 장비들로 구성되어 있다. Fig. 1은 선박운항 시뮬레이터의 구성도로 Instructor를 중심으로 여러 장비들이 구성되어 있으며 이들 장비 간에 TCP/IP, UDP, RS-232 등의 네트워크를 통해 연동되어 시뮬레이션을 수행하게 된다. 이와 같이 시뮬레이터 구성이 복잡하기 때문에 이를 개발하고 유지하는데 많은 비용, 인력, 시간의 투자가 필요할 뿐만 아니라 목표한 수준의 성능을 갖춘 완성도 높은 시뮬레이터를 개발하기에 어려움이 있다.

개발 시간 단축, 인력의 효율적 활용을 고려할 때 시뮬레이

† Corresponding author : 연회원, yjlee@kriso.re.kr 042)866-3667
* 연회원, aykim@kriso.re.kr 042)866-3660
** 연회원, skpark@kriso.re.kr 042)866-3647
*** 연회원, ojyong@kriso.re.kr 042)866-3648
**** 연회원, jeongsoo93.kim@samsung.com 054)460-8719

터 개발이 이상적으로 이루어지기 위해서는 업무 분담을 통해 작업이 병렬적으로 수행되어야 하지만 업무간의 상호 의존성으로 인해 병렬 수행이 어려워진다. 개발 초기에 설계 및 개발 계획의 완성도에 따라 상당 부분 극복이 가능하지만 시스템 전문가에 비해 알고리즘을 구현하는 도메인 전문가는 이러한 개발 프로세스에 익숙지 않아 초기 설계와 다른 방향으로 개발이 진행될 수 있다. 또한 구현된 알고리즘을 시뮬레이터에 적용할 때마다 시스템 전문가가 깊이 관여해야하기 때문에 효율이 떨어진다. 알고리즘 구현에 시스템 전문가가 관여하지 않고 도메인 전문가가 알고리즘을 자유롭게 개발, 수정, 테스트가 가능하다면 업무의 의존성을 낮추는 것이 가능하게 된다.

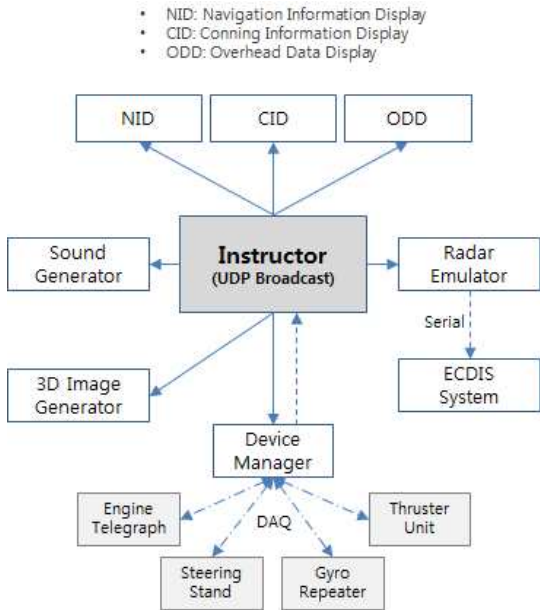


Fig. 1 Configuration of ship handling simulator

본 논문에서는 시뮬레이터 시스템을 Fig. 1과 같이 소프트웨어 및 하드웨어로 구성된 2개 이상의 장치들이 네트워크를 통해 연동되도록 구성된 시스템으로 제한하였으나, 선박, 항공 등 특정 용도의 시뮬레이터 시스템으로 제한하지는 않았다.

개발에 참여하는 전문가들을 크게 두 가지로 분류하였는데, 시뮬레이션 대상 또는 현상에 대해 연구하고 알고리즘을 구현하는 전문가를 도메인 전문가, 그 밖의 하드웨어 및 소프트웨어 개발과 관련된 전문가를 시스템 전문가라 정하고, 도메인 전문가는 알고리즘을 구현하기 위한 최소한의 프로그래밍 능력을 갖추었다고 전제하였다.

본 연구에서는 도메인 전문가가 알고리즘 구현에 집중할 수 있도록 지원하는 방안에 대해 연구하고, 그 결과물로 시뮬레이터 개발 지원 도구의 설계 및 구현을 목표로 하고 있다.

## 2. 시뮬레이터 개발 지원 도구의 개념

오늘날 업계, 학계 및 군에서 교육, 연구, 군사훈련, 엔터테인먼트 등의 다양한 목적으로 시뮬레이터를 적극 도입하여 활용 폭을 넓히고 있다. 이에 따라 시뮬레이터에 대한 수요가 증가하면서 시뮬레이터 개발에 대한 연구와 개발 지원 도구에 대한 관심이 높아지고 있다.

HLA(High Level Architecture)는 미 국방성에서 제안한 컴퓨터 시뮬레이션 간의 연동을 위한 구조이다. HLA 프레임워크 내에서 서로 다른 시뮬레이션들이 설계 규칙 및 인터페이스를 준수함으로써 상호 운용성 및 재사용성을 높일 수 있다(IEEE, 2010). HLA 인터페이스 명세를 준수하도록 구현된 RTI(Runtime Infrastructure)는 시뮬레이터 간의 연동을 위한 인터페이스를 제공한다. Fig. 2는 HLA/RTI를 통해서 서로 다른 시뮬레이터 간의 상호 연동을 보여주는 예이다.

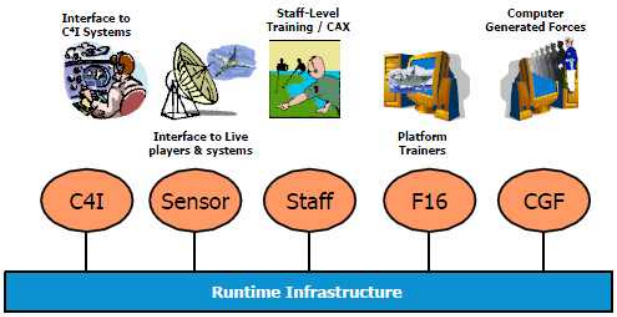


Fig. 2 HLA - Sample Federation (Möller and Lutz, 2013)

CIGI(Common Image Generator Interface)는 Fig. 3과 같이 시뮬레이터에서 영상 생성기(Image Generator, IG)와 연동하는 호스트 장치간의 표준 통신 방법을 설계한 인터페이스이다. 영상 생성기 개발자와 호스트 개발자가 CIGI의 표준을 준수함으로써 이들 장치간의 통합이 쉬워지고 코드 재사용이 가능해 진다(CIGI, 2014; Durham, 2006).

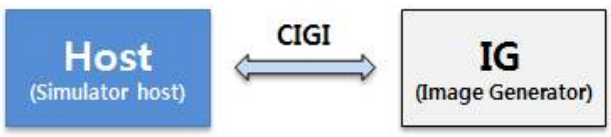


Fig. 3 CIGI - Interface between Host and IG

VWF(Virtual World Framework)는 가상세계를 구현하기 위한 웹 기반 도구이다. 웹 기반 도구를 통해서 구현된 웹 애플리케이션은 하나의 소스 코드를 통해서 다양한 장비와 플랫폼에서 실행될 수 있고 업데이트가 편리하며 사용자 접근성이 용이하다. 시뮬레이터 개발에 VWF 기술을 활용하는 것은 이러한 웹 기반 도구의 장점을 시뮬레이터 개발에 적용하기 위

한 시도이다(Chadwick and Easter, 2013).

시뮬레이터 개발에 특화된 도구 이외에도 게임 개발 도구를 시뮬레이터 개발에 활용하기도 한다. 게임 개발 도구는 렌더링, 사운드, 네트워크, 스크립트 등 다양한 기능을 지원하고 사용이 편리하다는 장점이 있다. 하지만 다양한 기능을 익히고 개발하는데 많은 시간이 소요된다.

시뮬레이터 개발 지원 도구들은 기능이 많고 자유도가 높을수록 도구를 배우고 개발하는데 많은 시간이 소요된다. 개발이 비교적 쉽고 빠른 도구들은 자유도가 떨어지는 경향이 있어 구현할 수 있는 기능이 제한된다. 본 연구는 개발 자유도를 높이면서도 시뮬레이터를 손쉽게 개발할 수 있는 도구 개발을 목표로 한다.

HLA, CIGI 등의 시뮬레이터 개발 지원 도구들은 시뮬레이터 시스템 간의 연동, 시뮬레이터 시스템을 구성하는 장치들 간의 연동을 위한 인터페이스를 통해서 재사용성 및 상호 운용성을 높임으로써 개발 효율을 높이고자 하였다. 본 논문에서는 접근 방법을 달리하여 개발 과정에서 도메인 전문가와 시스템 전문가간의 독립적인 업무 수행이 가능하도록 지원하는 도구와 이를 활용한 개발 절차를 제시하여 개발 효율을 높이고자 하였다.

## 3. 요구사항 분석

시뮬레이터의 개발과 유지보수 과정에서 도메인 전문가가 알고리즘 구현에 집중하기 위해서는 다음과 같은 요구사항들이 충족되어야 한다.

첫째, 알고리즘 개발 시 변수 및 함수명을 자유롭게 사용할 수 있어야 한다. 보통은 소프트웨어 개발 초기에 이름 규칙(Naming rule)을 결정하여 이에 따라 구현을 시작하게 되지만, 도메인 전문가들은 해당분야에서 사용하는 전문용어 또는 개인이 선호하는 용어들을 사용하려고 한다. 이는 동일한 데이터에 대해서 도메인 전문가마다 다른 이름으로 접근하여 사용할 수 있어야 함을 의미한다.

둘째, 다른 기술적인 면을 배제하고 알고리즘 구현에 집중할 수 있어야 한다. 예를 들면 알고리즘 구현에 필요한 데이터가 서버, 데이터베이스, 파일 등에 있을 경우 이를 읽어오는 코드를 도메인 전문가가 개발할 필요가 없어야 한다. 이를 위해서는 관련 기능을 프레임워크에서 지원하고 코드를 자동 생성하되 도메인 전문가에게는 상세 구현이 노출되지 않도록 하여 알고리즘 구현에만 전념하도록 유도하여야 한다.

셋째, 테스트가 용이해야 한다. 알고리즘별로 단위 테스트가 가능하고 조합된 알고리즘들의 테스트가 가능해야 한다. 도메인 전문가는 알고리즘의 수정과 검증 과정을 빈번하게 수행하면서 구현을 완성해 나가기 때문에 테스트 기능은 개발 효율에 큰 영향을 미친다.

본 연구에서는 상기의 요구사항들을 충족시키기 위해 DataSetManager라는 이름의 시뮬레이터 개발 지원 도구를 개발하고 이를 활용한 시뮬레이터 개발 방법을 제안한다.

## 4. DataSetManager를 이용한 시뮬레이터 개발

도메인 전문가의 알고리즘 개발 지원과 유연한 데이터 관리를 위해 UnionSet, AppSet, LocalSet이라는 데이터셋 개념을 도입하였다. UnionSet은 전체 시뮬레이터에서 공유되는 데이터 집합이고 AppSet은 응용 프로그램 단위로 관리되는 데이터 집합이다. LocalSet은 함수 내에서 사용하는 데이터 집합으로 도메인 전문가가 알고리즘 개발에 사용하는 변수이다. 이들 데이터셋 간에는 맵핑이라는 방식으로 서로 연결되어 데이터를 교환한다.

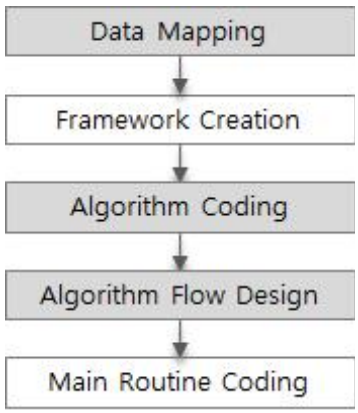


Fig. 4 Simulator development procedures using DataSetManager

DataSetManager는 UnionSet, AppSet, LocalSet의 데이터들을 생성하고, 맵핑하고, 알고리즘 흐름 설계를 가능하게 하는 도구이다. Fig. 4와 같이 DataSetManager를 활용한 개발은 데이터 맵핑(Mapping), 프레임워크 생성, 알고리즘 코딩, 알고리즘 흐름 설계, 메인 루틴 코딩의 순으로 진행된다.

데이터 맵핑은 시뮬레이터에서 사용할 데이터 구조를 설정하는 과정이다. 도메인 전문가는 DataSetManager를 이용하여 사용할 데이터들을 생성하고 맵핑 과정을 통해서 UnionSet의 데이터들을 AppSet으로 연결 후 선호하는 이름 형태인 LocalSet으로 연결하여 알고리즘 구현을 위한 데이터 집합을 설정한다.

프레임워크 생성 단계는 DataSetManager를 통해 시뮬레이터 개발을 위한 프로그램 소스 코드를 생성하는 단계로 데이터 맵핑 단계에서 설정한 데이터 구조를 반영한다. UnionSet 관리를 위한 서버 프로그램용 소스 코드, AppSet, LocalSet 및 함수가 포함된 각 응용프로그램용 소스 코드를 생성한다. 또한 데이터셋 간의 맵핑 코드와 서버와 응용프로그램 사이에 통신을 위한 코드도 생성된다.

알고리즘 코딩 단계에서는 도메인 전문가가 데이터 맵핑 시 생성된 LocalSet을 활용하여 함수 단위로 알고리즘을 구현

하고, 알고리즘 흐름 설계는 함수 단위로 구현된 알고리즘을 조합하고 순서를 지정한다. 이 때 테스트 프로그램을 통해서 알고리즘의 검증과 수정 과정을 반복한다.

메인 루틴 코딩은 구현된 알고리즘들을 통합하여 하나의 소프트웨어를 완성하는 단계이며 시스템 전문가에 의해서 이루어진다.

## 5. 시뮬레이터 개발 지원 도구 구현

시뮬레이터 개발 지원 도구는 네트워크, 데이터베이스, 파일 입출력 등의 기능을 지원해야 한다. 이를 위해 네트워크 프레임워크 개발, 데이터베이스 설치 및 환경 설정, 관련 라이브러리 구현들이 선행되어야 한다. DataSetManager는 시뮬레이터 개발의 핵심으로 이러한 지원 도구들을 기반으로 알고리즘 구현을 위한 프로그램 소스 코드를 생성한다.

### 5.1 네트워크 서버 및 라이브러리 구현

DataSetManager를 통해 설정된 데이터 맵핑을 서버와 클라이언트간의 데이터 통신으로 가능하게 하고 기존 연동 구조상의 단점을 극복하기 위하여 유연한 네트워크 구조를 설계하였다.

기존 시뮬레이터에서는 Fig. 1과 같이 시뮬레이션 환경 설정 및 제어를 담당하는 소프트웨어인 Instructor가 주기적으로 명령 패킷을 브로드캐스팅하고 다른 장치들은 이를 수신하여 기능을 수행하는 구조였다. 이러한 형태는 구조가 단순하고 기존 시스템에 영향을 미치지 않으면서도 새로운 장치를 추가하는 것이 쉬운 장점이 있다. 하지만 Instructor가 전달하는 패킷의 형태가 수정되면 모든 장비의 패킷 수신 코드가 변경되어야 하고 장비들 간의 직접적 정보 교환 및 제어 방법이 없어서 유연한 구조를 갖추지 못한다.

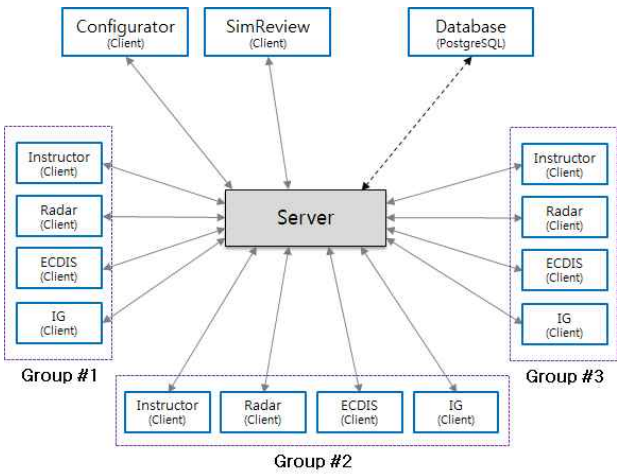


Fig. 5 Network topology of simulator

DataSetManager를 통해 생성되는 시뮬레이터를 위한 네트워크 시스템은 Fig. 5와 같이 서버를 중심으로 모든 접속 장비의 관리가 가능하고 장비들 간의 양방향 정보 교환이 가능하도록 하였다. 물리적으로는 단순한 클라이언트/서버 구조이지만 논리적인 계층인 시뮬레이터 그룹을 통해서 그룹별 독립된 시뮬레이션, 그룹 간의 연동 시뮬레이션을 가능하게 한다. 논리적인 계층을 통해 얻을 수 있는 또 다른 장점은 실행 시에 유연하게 시뮬레이터 그룹 및 장비들을 재구성할 수 있다는 점이다. 예를 들면 Fig. 5의 그룹1에서 레이더가 1대 더 필요한 경우에 그룹2에 있는 레이더를 그룹 1에 연동되도록 하는 것이 가능하다.

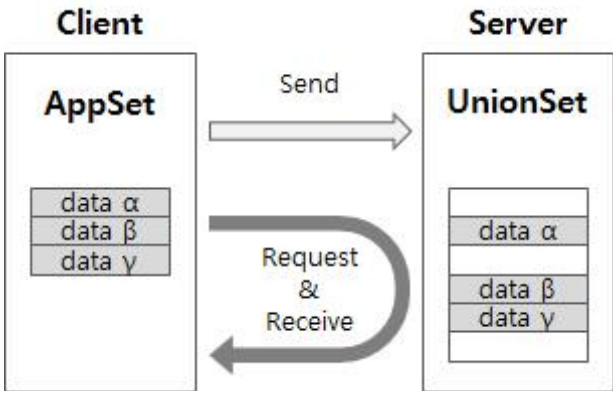


Fig. 6 Transmission and reception of data between client and server

서버에서는 UnionSet 데이터를 관리하고, 각 클라이언트들은 AppSet 데이터를 관리한다. DataSetManager를 통해 맵핑된 UnionSet과 AppSet의 데이터 교환은 Fig. 6과 같이 네트워크를 통한 데이터 송·수신으로 가능하다. AppSet의 변수에 값을 할당하면 서버로 데이터가 전송되어 맵핑된 UnionSet의 변수가 업데이트 된다. AppSet의 값을 사용하는 경우에는 맵핑된 UnionSet의 변수를 서버에 요청하고, 수신된 값을 AppSet의 변수에 할당하도록 구현된다.

### 5.2 데이터 구조 및 형식

```
<?xml version="1.0" encoding="UTF-8"?>
<DataSet>
    <Data type="DateTime" key="SimTime">2014-02-01 17:37:39.059</Data>
    <Data type="String" key="OwnshipName">10K_LNG</Data>
    <Data type="Double" key="posX">19809.2</Data>
    <Data type="Double" key="posY">3140.7</Data>
    <Array type="Int32" key="lightOn">
        <Item>1</Item>
        <Item>2</Item>
        <Item>3</Item>
        <Item>7</Item>
        <Item>8</Item>
    </Array>
</DataSet>
```

Fig. 7 XML format data example

기존에 네트워크를 통해 전송되거나 파일에 저장되는 형식은 데이터 종류, 크기, 순서 등이 약속되어 있는 형태였다. 이 때문에 단순히 데이터의 순서만 바꾸는 경우에도 시뮬레이터를 구성하는 거의 대부분의 프로그램들을 수정해야 했다. 이

러한 문제를 해결하고자 키와 값 형태의 자료구조로 데이터를 관리하도록 하여 기존 프로그램들에 영향을 끼치지 않으면서도 새로운 데이터의 추가가 가능하고 새로운 시뮬레이터 장치와 연동도 가능하게 하였다. 파일 저장 및 네트워크 전송 시에는 호환성 및 확장성을 고려하여 Fig. 7과 같이 XML 형식으로 데이터를 생성하거나 저장 공간을 절약하기 위해 바이너리 형식으로 생성이 가능하다. 이와 같이 유연한 데이터 구조를 통해서 DataSetManager에서 데이터셋 간의 자유로운 맵핑과 장치간의 연동이 가능하다.

### 5.3 DataSetManager

Fig. 8 DataSetManager user interface

데이터셋은 시뮬레이션을 위한 데이터의 집합 또는 변수 집합을 의미하며 앞서 언급한 Union Set, App Set, Local Set으로 구성된다. DataSetManager는 이러한 데이터셋들을 관리하는 역할을 수행하며, 시뮬레이터 개발 지원 도구의 핵심이다. 도메인 전문가는 Fig. 8과 같은 DataSetManager의 사용자 인터페이스를 통해서 알고리즘 구현을 위한 틀을 생성한다.

DataSetManager는 데이터셋 맵핑, 알고리즘 흐름 설계, 코드 생성 기능들을 지원한다.

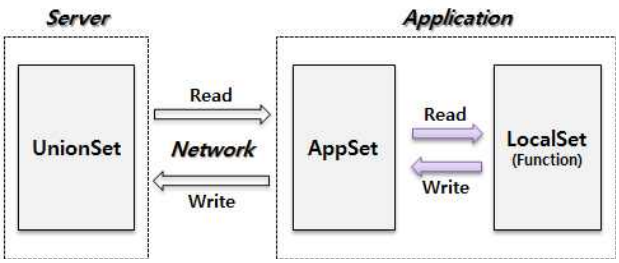


Fig. 9 Data mapping

데이터셋 맵핑은 UnionSet, AppSet, LocalSet을 서로 연결하는 작업으로 Fig. 9과 같은 구조를 이룬다. UnionSet은 하나의 시뮬레이터 그룹에서 공유되는 데이터셋이며, AppSet은 UnionSet과 연결되어 특정 애플리케이션에서 사용하는 데이터셋이다. AppSet의 데이터 중 함수 내에서 알고리즘 구현에 직접적으로 사용하는 데이터는 LocalSet으로 연결한다. UnionSet은 각각의 시뮬레이터 그룹에서 공유하기 위해 서버에 저장된다. 이 때문에 UnionSet과 AppSet의 맵핑은 네트워크를 통한 데이터 송·수신을 의미한다.

변수의 접근 제한 기능도 지원하는데 DataSetManager를 통해 읽기 전용의 변수를 생성하는 경우 Fig. 9의 Read에 해당하는 기능만 구현함으로써 쓰기 기능을 제한할 수 있다.

알고리즘 흐름 설계 기능은 관련 있는 알고리즘의 실행 순서를 설정하고 추가, 삭제, 수정 기능을 지원한다.

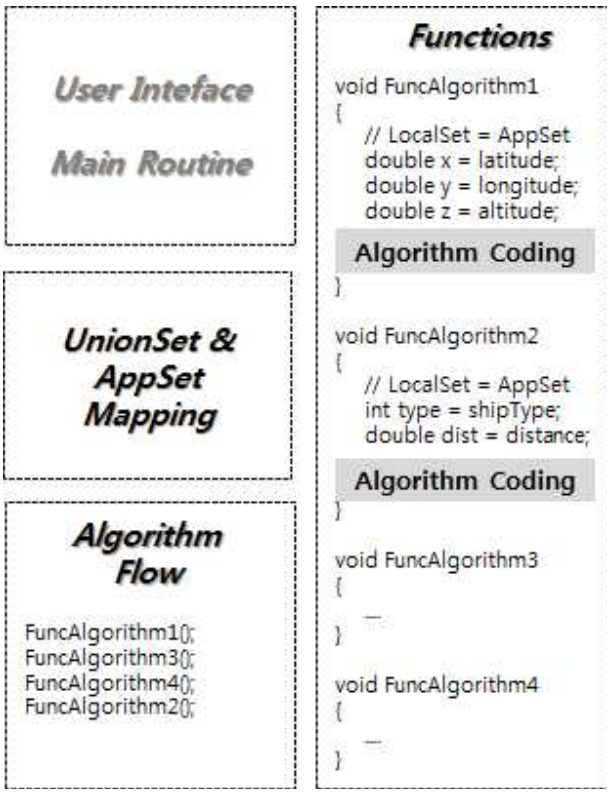


Fig. 10 A framework for algorithm implementation

DataSetManager을 통해서 데이터 맵핑과 알고리즘 흐름 설계가 완료되면 마지막으로 Fig. 10과 같은 프로그램 소스 코드를 생성할 수 있다. 이 소스 코드에는 데이터 맵핑과 알고리즘 흐름 코드 등과 함께 각각의 알고리즘을 구현할 수 있는 함수 형태의 틀이 생성된다. 도메인 전문가는 함수 내에서 맵핑된 LocalSet 변수들을 사용하여 알고리즘을 구현할 수 있다.

데이터 맵핑 및 알고리즘 흐름과 같이 DataSetManager를 통해 설정된 정보들은 별도로 저장하여 새로운 시뮬레이터를 개발하거나 기존 시뮬레이터를 개선하는데 재사용 가능하다.

### 5.4 시뮬레이터 구현

Fig. 11은 DataSetManager를 통해서 생성된 프로그램 소스 코드를 활용하여 간단한 선박운항 시뮬레이터의 구현 예를 보여준다.

SimServer는 시뮬레이터를 구성하는 장비인 Instructor과 MapView를 관리하고 데이터 교환의 중계 역할을 한다. Instructor는 시뮬레이션을 위한 제어 정보를 시뮬레이션 그룹에 속한 장비들에게 전달하고, MapView는 전달받은 데이터를 이용하여 데이터를 전시 및 관련 기능을 수행한다.

DataSetManager를 통해서 네트워크 통신 및 데이터 교환을 위한 틀을 신속하게 구현 가능하다. 이 후 프로그램의 역할에 맞게 데이터를 시각적으로 표현하고 사용자의 입력을 받을 수 있는 그래픽 사용자 인터페이스(GUI, Graphical User Interface), 각종 하드웨어 장비와 연동을 위한 디지털 및 아날로그 입출력 데이터 처리 등의 기능이 구현되어야 한다.

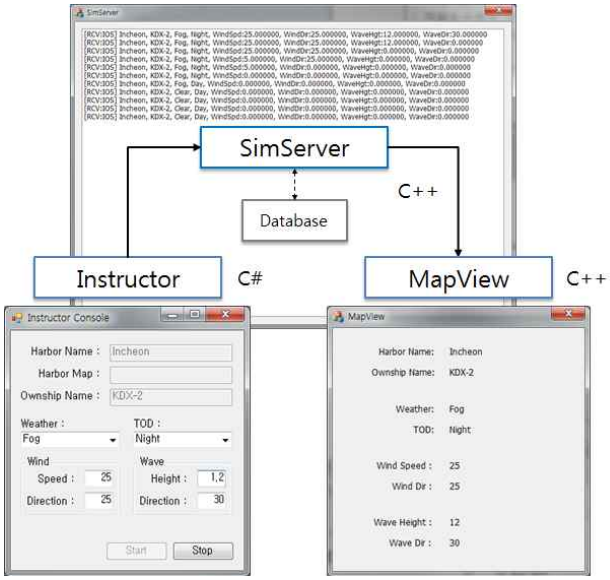


Fig. 11 Implementation examples using the Support Tool for Simulator Algorithm Development

## 6. 결 론

본 연구는 시뮬레이터 개발 구조에 대한 연구에서 시작하여 개발 효율성을 높이기 위한 방안을 모색하였다. 개발 효율을 높이기 위해 전문가간의 협력 및 업무 분담이 필요하지만 업무 의존성이 이를 어렵게 한다. 도메인 전문가가 알고리즘 구현에 집중 할 수 있는 방법 및 도구를 제안함으로써 도메인 전문가와 시스템 전문가 간의 상호 의존성을 낮추고 개발 효율성을 높이고자 하였다.

알고리즘 단위로 변수명을 맵핑하고 함수 단위의 알고리즘 구현 틀을 제공하여 도메인 전문가가 알고리즘 개발에 전념할 수 있도록 하였다. 이를 위해 UnionSet, AppSet, LocalSet이라는 데이터셋의 개념을 도입하고 유연한 데이터 구조를 사용하였다. UnionSet은 시뮬레이터 전체에서 공유할 수 있도록 서버에 저장되고 AppSet은 애플리케이션 단위로 저장된다. UnionSet과 AppSet의 맵핑은 네트워크를 통해 데이터 송·수신으로 이루어진다. LocalSet은 AppSet과 맵핑되어 알고리즘 구현에 직접 활용할 수 있도록 함수 내에 생성된다.

DataSetManager는 시뮬레이터 개발 지원 도구의 핵심으로 데이터 맵핑 및 알고리즘 흐름 설계를 가능하게하고 시뮬레이터 구현을 위한 프로그램 소스 코드를 생성한다.

시뮬레이터 개발 지원 도구를 통해 각 도메인 전문가는 알고리즘 개발에 집중 할 수 있으며 협업의 효율성을 꾀할 수 있을 것으로 기대된다. 또한 개발 절차가 체계화 되고 명확하게 되어 개발 계획 및 관리가 용이해질 것으로 예상 된다.

향후 DataSetManager가 실무에 적극 활용이 되도록 템플릿 유지 관리, 다양한 데이터형 및 자료 구조 지원, 스크립트 언어 지원 등의 기능을 추가하고 개선할 계획이다. 구현된 시뮬레이터 개발 지원 도구가 클라이언트/서버 구조를 이루고 있어 웹 기반의 시뮬레이터 구현에도 활용할 수 있을 것으로 예상되며 이에 대한 연구를 진행할 계획이다.

## 후 기

본 연구는 한국해양과학기술원 부설 선박해양플랜트연구소의 주요사업인 “제한수역에서의 선박 운항 시뮬레이션 기술 고도화” 과제의 지원을 통해 수행되었습니다.

## References

[1] CIGI(2014), CIGI Overview, http://cigi.sourceforge.net
[2] Chadwick R. and Easter D.(2013), “The Virtual World Framework: Implementing a Web Based Client Side Simulator”, I/ITSEC 2013, Paper No. 13057, pp 1-12.
[3] Durham L. W.(2006), Interface Control Document for the Common Image Generator Interface(CIGI) Version 3.2, p. 2.
[4] IEEE(2010), IEEE Standard for Modeling and Simulation (M&S) High Level Architecture(HLA) - Framework and Rules, IEEE Std 1516.
[5] Möller B. and Lutz B.(2013), Introduction to HLA, I/ITSEC 2013, Tutorials, p. 14.

원고접수일 : 2014년 4월 23일
심사완료일 : 2014년 8월 14일
원고채택일 : 2014년 8월 14일