JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 26 No. 9, pp. 97-104, September 2021
https://doi.org/10.9708/jksci.2021.26.09.097

A Study on the Standardization of On-Board Training System Software for Naval Ship Engineering Control System

Seung-Chul Kwak*

*Engineer, Naval R&D Center, Hanwha Systems, Gumi, Korea

[Abstract]

Since 1993, Successfully localized naval combat System has made steady development on various domestic and foreign ships. On the other hand, Engineering Control System(ECS) is dependent on foreign companies. Therefore, there is a lot of interest and research in the localization of ECS in the navy defense industry. As one of various studies, a preliminary study of domestic ECS software that can be commonly applied to naval ships is in progress. This paper propose Ecs Obts Scalable Platform(EOSPA) as the standard architecture of ECS On-Board Training System(OBTS) software by applying object-oriented programming and standardization. And this introduces EOSPA's structure, function, and features of each component. Furthermore, high reusability and maintainability are expected in the development of ECS OBTS software applying EOSPA in various naval ships.

▸ **Key words**: ECS(Engineering Control System), OBTS(On-Board Training System), Simulation, CMS(Combat Management System), OOP(Object-Oriented Programming), Component Platform, Standard Architecture

[요 약]

1993년 이후 국산화에 성공한 해군 전투체계는 다양한 국내외 함정에 탑재되어 다방면에서 꾸준한 발전을 이루어왔다. 반면 함정 통합기관제어체계(Engineering Control System, ECS) 소프트웨어는 아직도 해외 업체에 의존적임으로 해군 방위산업에서도 ECS의 국산화에 많은 관심과 연구들이 진행되고 있다. 다양한 연구 중 하나로 해군 함정에 공통적으로 적용 가능한 국산 ECS 소프트웨어 선행연구가 진행되고 있다. 본 논문에서는 함정 전투체계(Combat Management System, CMS)의 국산화를 하면서 개선하고 발전하였던 객체지향 프로그래밍과 표준화를 ECS의 함상훈련계통(On-Board Training System, OBTS) 소프트웨어에 적용하여 ECS OBTS 소프트웨어의 표준 아키텍쳐로 Ecs Obts Scalable Platform Architecture(EOSPA)를 제시하고 각 컴포넌트의 구조와 기능 및 특징을 소개한다. 더 나아가 다양한 함정에서 EOSPA를 적용한 ECS OBTS 소프트웨어 개발에 있어 높은 재사용성, 유지보수성을 기대한다.

▸ **주제어**: 함정 통합기관제어체계, 함상 훈련계통, 시뮬레이션, 전투체계, 객체지향 프로그래밍, 컴포넌트 플랫폼, 표준 아키텍처

• First Author: Seung-Chul Kwak, Corresponding Author: Seung-Chul Kwak
*Seung-Chul Kwak (sc11.kwak@hanwha.com), Naval R&D Center, Hanwha Systems
• Received: 2021. 07. 22, Revised: 2021. 09. 11, Accepted: 2021. 09. 12.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

1993년 국산화에 성공한 함정전투체계는 이후 국내외 다양한 함정에 탑재되며 많은 발전을 이루어왔다. 반면 함정 통합기관제어(Engineering Control System, ECS) 소프트웨어는 30년 이상 해외 직구매 또는 기술협력생산을 통해 확보/운용되고 있으며 관련 핵심기술에 대한 직접 개발 경험이 전무하여 독자적인 개발 능력이 없는 상태이다. 또한 개선 사항을 반영할 수 없고 문제점이 발생하였을 경우에도 조치하는 데에 적어도 수개월이 걸리는 불편함이 존재하였다. 그래서 최근 해군 방위산업에서도 향후 건조 함정의 체계통합, 운용 함정의 성능개선, 함정 추진체계 분야 건조제원 및 운용 관련 정보의 국외 유출 방지, 함정 수명주기 간 기술진부화 방지 및 주기적 성능개선, 작전 운용성 보장과 후속군수지원을 보장하기 위하여 공통 적용 가능한 함정통합기관제어 소프트웨어의 국산화를 위한 선행연구가 진행되고 있다[1]. 종래의 함정 통합기관제어 소프트웨어의 경우에 전량을 해외모델로 탑재하고 있어 이를 국산화하여 개발할 경우에 4,000억 원 이상의 수입 대체가 가능하며 최초 개발 시부터 함정전투체계에 적용되었던 객체지향 프로그래밍과 표준화를 함정 통합기관제어 함상훈련계통 소프트웨어에 적용하여 유지보수 및 신규 개발에도 높은 효율을 낼 수 있을 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 배경지식인 종래의 함정 통합기관제어 함상훈련계통 관련 연구 및 객체 지향 프로그래밍의 특징을 소개한다. 3장에서는 함정 통합기관제어 함상훈련계통 소프트웨어의 표준 아키텍쳐로서 EOSPA(Engineering control system On-board training system Scalable Platform Architecture)를 제시하고 각 컴포넌트들의 상세구조 및 동작에 대해 설명한다. 4장에서는 제안된 EOSPA의 특징 비교 및 성능을 검증하며 마지막으로 5장에서는 결론 및 향후연구방향을 제시한다.

II. Preliminaries

1. Background

1.1 Engineering Control System

함정 통합기관제어는 함정을 구성하는 추진계통, 전력계통, 손상통제계통, 보기계통을 기반으로 구성된다. 이와 함께 운용자의 업무능력 향상, 업무부하 감소 및 수명주기 비용(Life Cycle Cost)의 절감효과를 위해 함상훈련계통도 포함된다[2]. 함정 통합기관제어는 별도로 각각 제어하던 시스템들을 네트워크 기반으로 상호 통합하여 제어/감시할 수 있는 장치로서, 함정의 운용성과 생존성, 안정성을 효율적으로 보장하기 위한 핵심장비이다. 현재 함정 통합기관 제어 시뮬레이션 환경은 Fig. 1과 같이 구성되며 함정의 제원에 따라 다양한 구성이 가능하다. 추진계통은 추진 HMI(Human Machine Interface, 운용자화면), 추진장비 제어, 추진계통 장비 연동으로 구성되며 전력계통은 전력 HMI, 발전기 시동/정지 등 원격제어, 발전기 상태 감시 및 경보, 함 전원 및 육상전원 교대, 전력계통 장비 연동으로 구성된다. 보기계통은 보기 HMI, 추진보기 제어/감시, 일반보기 제어/감시, 보기계통 장비 연동으로 구성되며 손상계통은 손상 HMI, 특수 손상 통제, 일반 손상 제어, 손상계통 장비 연동으로 구성된다. 마지막으로 함상훈련계통은 훈련 HMI(추진계통, 전력계통, 보기계통, 손상통제계통 HMI 모두 포함), 훈련 시나리오 생성 및 선정, 훈련 기록 및 재생, 계통 시뮬레이션으로 구성된다[3].

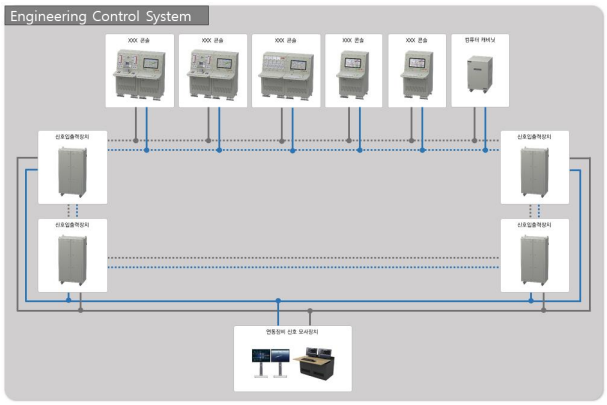

Fig. 1. Engineering Control System

1.1.1 Engineering Control System On-Board Training System

함정 통합기관제어를 구성하는 계통 중 함상훈련계통은 Fig.1의 컴퓨터 캐비닛에 하나의 PC에 설치가 된다. 컴퓨터 캐비닛 내 함상훈련계통 컴퓨터는 Fig. 2와 같다. 각 콘솔에서 훈련생이 가상머신으로 접속이 가능하며 접속한 훈련생들은 실장비의 구동 없이 시뮬레이션을 통하여 실 환경과 유사하게 훈련이 가능하다. 함상훈련계통의 시뮬레이터는 함정 통합기관제어의 모든 장비에 대하여 시뮬레이션이 가능해야 한다. 이러한 함정 통합기관제어는 각 장비의 다양한 조합으로 1.1에서 설명한 추진, 전력, 손상통제, 보기계통의 화면의 시뮬레이터를 구성한다. 실 함정과 동일하게 장비들의 구성을 통하여 만들어진 시뮬레이터는 각 가상머신 위에 구동되는 가상의 콘솔화면에서 훈련생의 입력에 대하여 실장비와 유사한 출력을 제공한다.

Fig. 2. ECS OBTS

1.2 Object-Oriented Programming

객체지향 프로그래밍은 컴퓨터 프로그래밍의 패러다임 중 하나이다. 객체 지향 프로그래밍은 컴퓨터 프로그램을 명령어의 목록으로 보는 시각에서 벗어나 여러 개의 독립된 단위, 즉 “객체”들의 모임으로 파악하고자 하는 것이다. 각각의 객체는 메시지를 주고받고, 데이터를 처리할 수 있다[4].

객체지향 프로그래밍은 프로그램을 유연하고 변경이 용이하게 만들기 때문에 대규모 소프트웨어 개발에 많이 사용된다. 또한 프로그래밍을 더 배우기 쉽게 하고 소프트웨어 개발과 보수를 간편하게 하며, 보다 직관적인 코드 분석을 가능하게 하는 장점을 갖고 있다. 객체지향 프로그래밍은 3요소, 5원칙이 존재한다. 3요소에는 캡슐화(Encapsulation), 다형성(Polymorphism), 상속(Inheritance)이 있다. 5원칙에는 Robert C. Martin이 명명한 객체지향 프로그래밍 설계의 다섯 가지 기본 원칙으로 S.O.L.I.D가 있다[5,6]. 3요소와 5원칙을 적용하면 프로그래머가 시간이 지나도 유지 보수와 확장이 용이하다.

3장에서 제시되는 ECS OBTS 소프트웨어의 표준 아키텍쳐(Ecs Obts Scalable Platform Architecture, EOSPA)의 클래스들은 모두 위의 3요소 5원칙을 최대한 반영하여 설계되었다.

2. Related works

함정 통합기관제어의 국산화는 1단계 기능별 핵심기술 확보, 2단계 성능검증을 위한 시험 개발, 3단계 함정 탑재를 위한 체계 개발의 단계를 거쳐 완성될 수 있을 것이며 현재는 핵심 기술 확보인 1단계가 진행 중이며 함정 통합기관제어 공통 적용 가능한 소프트웨어 개발(2020~2023/진행)을 위한 핵심기술 응용 연구 과제가 진행 중이다[1].

함정 통합기관제어 특성상 장비와 연동을 위하여 운항제어 솔루션(SSAS Master)을 통하여 소프트웨어를 개발하기 때문에 객체지향 프로그래밍을 적용하기가 불가능하다. 적용 가능한 부분은 함상훈련계통의 시뮬레이션 기능이며 함상훈련계통 표준화에 적용할 수 있는 설계를 제시한 사례는 없었다. 하지만 함정 통합기관제어의 계통 중 함정 추진체계에 대한 동적 시뮬레이션 소프트웨어 개발과 함정 통합기관제어와 유사한 함정전투체계에서는 표준화를 위한 다양한 연구들이 진행되었다.

2.1 A Study of the Naval Ship Propulsion System Dynamic Simulation Software Development

함정 통합기관제어를 국산화하기 위한 연구 중 함정 추진체계의 동적 시뮬레이션 소프트웨어개발 연구[7]가 진행되었다. 국산화 단계 중 1단계인 기능별 핵심기술 확보의 일환으로 진행되어 함정 통합기관제어의 여러 계통 중 추진계통의 동적 시뮬레이션을 위한 소프트웨어 개발에 대한 연구이다. 동적 시뮬레이션 소프트웨어 연구를 위하여 사전에 CODOG(COmbined Diesel Or Gas-turbine) 함정 추진체계 시뮬레이터 개발에 대한 연구[8], CODLOG(COmbined Diesel-eLectric Or Gas-turbine) 함정 추진제어시스템을 위한 시뮬레이터 개발에 대한 연구[9]와 추진체계 동적 시뮬레이션을 위한 수학적 모델을 개발하기 위한 연구[10]도 진행되었다. 하지만 함정 통합기관제어 추진체계에 국한되어 함정 통합기관제어 전반에 적용할 수 있는 설계를 제시하지는 못하였고 객체지향 프로그래밍을 고려하지 못한 HILS(Hardware-In-the-Loop Simulation)으로 개발되었다.

2.2 Design of the Scalable Naval Combat System Software using Abstraction and Design Pattern

디자인 패턴을 이용한 함정 전투체계 교전 소프트웨어 수정 개발에 대한 연구[11]는 진행되었다. 객체지향 기반의 설계로 재사용성, 유지보수성을 만족하지만 전투체계의 기능의 일부분인 교전만을 고려하고 있다. 하지만 클래스 구조 변경 및 다양한 디자인 패턴을 적용한 소프트웨어 표준화를 통하여 함정 통합기관제어에서의 표준화 설계에도 도움이 되었다.

2.3 Research of OSD Standardization in Naval Combat System

함정 통합기관제어에서도 함정전투체계와 유사한 부분이 많이 존재한다. 그 중 전투체계의 함기준센서(Own Ship Data, OSD) 연동단의 외부와 전투체계 데이터 버스(Combat System Data Bus, CSDB)와 관급장비(Government Furnished Equipment, GFE)와의 통신은 함정 통합기관제어 함상훈련계통의 외부통신과 유사하다. 재사용성 및 범용성을 향상 시킬 수 있는 함기준센서 연동단 표준화 방안에 대한 연구[12]도 진행되었지만 모듈 내 객체지향 설계가 고려되지 않아 장비변경 등 환경변화에 따른 소프트웨어 변경 폭이 컸다.

III. The Proposed Scheme

함정 통합기관제어체계 함상훈련계통 소프트웨어는 함정의 임무에 따라 추진계통, 전력계통, 손상통제계통, 보기계통으로 나뉘며 이 네 분야의 계통들을 훈련하기 위한 함상훈련계통으로 이루어져있다. 기존의 체계는 장비의 추가 및 기능 변경에 영향을 받아 유지보수에 많은 시간과 비용이 투입되고 있다. 추진계통, 전력계통, 손상통제계통, 보기계통의 경우 운항제어 솔루션(SSAS Master)을 통하여 변경을 하여야 하므로 함정 통합기관제어체계의 함상훈련계통을 확장성 있는 소프트웨어로 개발하는 대상으로 선정하였다. 그리하여 3장에서는 함정 통합기관제어체계 표준 소프트웨어 아키텍쳐로서 Ecs Obts Scalable Platform Architecture(EOSPA)를 제시한다.

1. Class Design & Design Pattern Applying

Table 1. Development Process of EOSPA

Step	Description
Requirement Analysis	Feature of the EOSPA
Design	Class Diagrams Design of EOSPA
	Application of Design Patterns
Implementation	Reuse ECS Algorithm
Test	Unit Test of EOSPA

EOSPA 개발과정은 객체지향의 기본개념을 적용하여 세분화하고 객체별 클래스에 알맞은 설계 디자인 패턴을 적용하여 확장성을 확보하였다. 개발 단계는 Table 1의 4단계 과정으로 진행하였다.

1.1 Step 1 : Requirement Analysis

요구사항 분석 단계에서는 먼저 함정 통합기관제어체계 함상훈련계통의 요구사항에 따른 고유의 기능과 장비를 식별하였다. 각 장비별로 특성들을 확인하여 공통적인 부분과 차이점을 구분하였다. 또한 각 함정별로 공통적으로 적용할 수 있는 요구사항을 나누었고 장비에 영향을 받아 수정이 일어날 수 있는 특화된 요구사항은 별도로 분리하여 공통적인 특성을 식별하기 쉽도록 하였다.

1.2 Step 2 : Design

1.2.1 Class Abstraction

유사한 기능의 클래스가 중복 구현되지 않도록 EOSPA 클래스의 구조를 추상화 과정을 통하여 객체 지향적으로 설계하였다.

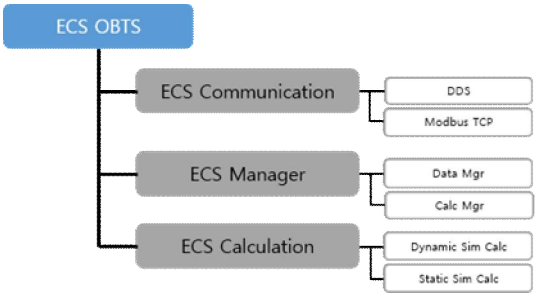

Fig. 3. EOSPA Architecture

클래스 작성과정에서는 아래의 클래스 작성 5대 원칙인 S.O.L.I.D 원칙을 적용하고자 하여 가장 적합한 디자인 패턴을 적용하고자 하였다.[5]

단일 책임 원칙을 적용하여 한 클래스 안의 응집도를 높이고 다른 클래스와의 결합도를 낮추어 기능별로 세분화하였다. 이로써 단일 기능을 가지도록 하여 재사용성 및 유지보수에 유리하도록 독립성을 높일 수 있었다.

개방 폐쇄 원칙을 적용하여 각 장비의 확장성을 고려한 설계가 되도록 하였다. 장비 추가 시 최소한의 변경으로 기능이 동작하도록 공통적인 부분을 일반화 시키고 그 일반화 된 속성 값(장비 명칭, modbus port, Tag, modbus 주소, modbus Type, bit Value, 16bit Value, 장비의 값 변경여부, 장비의 로직상태)을 파일로 읽어 들이고 실행될 때 그 값에 따라 설정되도록 하여 파일 수정만으로 장비를 추가 또는 삭제 할 수 있도록 하였다. 하지만 각 장비 간의 연관성 들을 통하여 Dynamic 시뮬레이션, Static 시뮬레이션을 계산하는 부분에서는 일반화 할 수 없었기에 해당 부분에서는 재사용성 및 유지보수성이 다소 떨어진다. 아래의 Table 2는 EOSPA의 3개 컴포넌트별 주요 클래스의 세부 역할을 정리한 표이다.

Table 2. EOSPA Component

Name	Major Function
ECS Comm	External Communication
ECS Manager	Data Classification & Distribution, Calc Management
ECS Calc	Dynamic Simulation Calculation, Static Simulation Calculation

Fig.4는 작성된 클래스 설계를 바탕으로 만들어진 EOSPA의 구성을 개략적으로 보여준다.

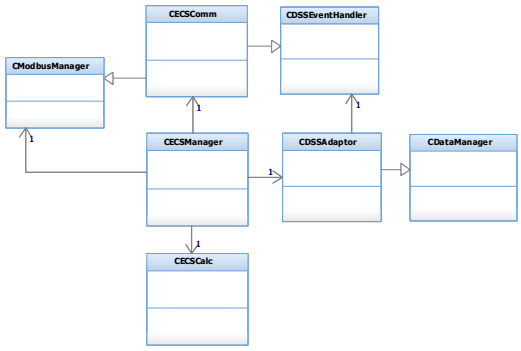

Fig. 4. Ecs Obts Scalable Platform Architecture

EOSPA는 크게 3개의 컴포넌트로 구성되어 있다. ECS Comm 컴포넌트는 외부와 통신하기 위한 어댑터 역할을 담당하도록 설계되었다. ECS Manager 컴포넌트는 ECS Comm에서 외부로부터 수신 받은 데이터를 분류 및 분배를 담당하고 ECS Calc 컴포넌트를 관리한다. ECS Calc 컴포넌트는 동적 시뮬레이션과 정적 시뮬레이션을 관리한다.

1.2.1.1 ECS Comm

Fig. 5와 Fig.6는 ECS Comm 컴포넌트의 클래스 구성을 나타낸다.

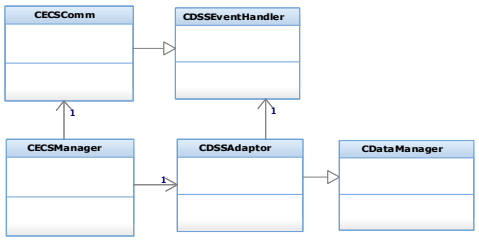

Fig. 5. ECS Comm – Data Distribution Service

EOSPA의 ECS Comm 컴포넌트의 역할은 외부와 통신을 수행하기 위함이다. 일반적으로 OBTS 시뮬레이터와 OBTS 내부 다른 장비와의 통신을 위해 사용되는 데이터 분산 서비스(Data Distribution Service, DDS)와 가상머신에 올라가는 각 계통별 HMI와의 PLC 통신을 위해서 사용되는 modbus TCP 통신을 모두 수행하여야 하므로 Fig. 4와 같은 DDS용 ECS Comm 인스턴스와 Fig. 5와 같은 modbus TCP용 ECS Comm 인스턴스를 모두 가진다.

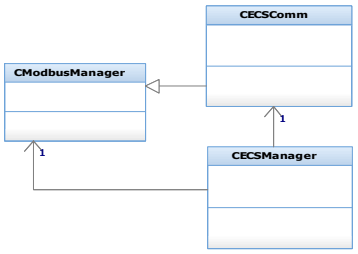

Fig. 6. ECS Comm – Modbus TCP

OBTS 내부에 사용되는 DDS 통신의 경우 함형별로 동일하게 적용 가능하며 변경될 가능성이 매우 적으므로 표준화 하여 사용 가능하다. 하지만 문제점은 modbus TCP를 사용하는 통신의 경우 실제 장비와 연동하는 부분이다. 실제 장비는 함형별로 다르기 때문에 각 장비의 추가 및 modbus TCP 주소, 각 장비별 로직정보 등의 변경이 잦은 항목에 대해서는 ECS Manager 컴포넌트에서 설정파일을 읽어 ECS Comm 컴포넌트에 반영함으로 다양한 함정에서 코드의 수정 없이 적용되도록 설계하였다.

1.2.1.2 ECS Manager

Fig. 7는 ECS Manager 컴포넌트의 클래스 구성을 나타낸다.

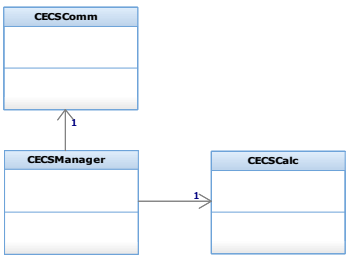

Fig. 7. Ecs Manager

ECS Manager 컴포넌트의 역할은 크게 두 가지이다. 첫 번째로 데이터의 관리이다. ECS Manager에는 ECS의 외부로부터 들어온 데이터를 EOSPA 표준화된 데이터로 변환하여 저장한다. 이렇게 저장된 데이터를 내부에서 계산을 하거나 외부로 전달을 할 때 모두 사용된다. 또한 프로그램 시작할 때 설정 파일을 읽어 필요한 정보를 처리한다. 설정 파일에는 장비 명칭, modbus port, Tag,

modbus 주소, modbus Type, bit Value, 16bit Value, 장비의 값 변경여부, 장비의 로직상태가 있으므로 함형이 달라 장비의 구성이 달라지는 경우에도 해당 설정 파일만 업데이트하면 ECS Manager의 수정 없이 사용 가능하다. 두 번째로 표준화된 데이터를 ECS Calc 컴포넌트로 전달하여 계산된 결과를 수신 받아 표준화된 데이터에 반영하여 ECS Comm에서 사용할 수 있도록 한다.

1.2.1.3 ECS Calc

ECS Calc 컴포넌트는 추진계통, 전력계통. 손상통제계통, 보기계통에 구성된 장비의 시뮬레이션을 담당한다. 시뮬레이션에는 Dynamic 시뮬레이션과 Static 시뮬레이션이 구동된다. Dynamic 시뮬레이션은 추진계통에 한하여 적용이 되며 HMI 입력에 의해서 시간에 따른 물리적 변화량 등을 동적으로 모사한다. Static 시뮬레이션은 HMI 입력에 따른 장비 값의 변경을 즉시 모사한다.

ECS Calc 컴포넌트는 EOSPA에 포함되어 있지만 각 함형별로 로직이 달라 로직 계산 방식도 모두 다르다. 그러므로 각 함형별로 변경을 해주어야 하는 측면에서 2장의 1.2에서 말한 객체지향 프로그래밍의 3요소 5원칙 기법과는 거리가 있다. 예를 들면 각 함정별로 연료탱크의 개수도 다르고 연결되는 밸브의 개수와 위치가 다르므로 표준화에 어려움이 따른다.

1.2.2 Design Patten Applying

본 논문에서는 소프트웨어 설계과정에서 사용할 수 있는 여러 디자인 패턴 중 본 과제에 적합한 디자인패턴을 분석하고 식별하여 적용 가능한 디자인 패턴을 선정하여 클래스 설계 시 반영하려고 노력하였다.

첫 번째 적용된 디자인 패턴은 추상팩토리(Abstract Factory) 패턴이다. 추상팩토리 패턴의 적용은 ECS Comm 클래스와 ECS Manager 클래스에 적용하였다. 각 함정별 구성되는 장비의 개수, 로직상태 등이 클래스를 변경시키는 요인이 되며 EOSPA의 변경을 최소화하기 위해 ECS Comm 클래스와 ECS Manager 클래스에 추상팩토리 패턴을 적용함으로써 다양한 함정에서 EOSPA를 사용하였을 때 변경을 최소화 하도록 설계를 하였다.

외부와의 통신하는 ECS COMM 클래스처럼 외부에서 들어오는 데이터를 다른 클래스에서 사용하여야 하는 오직 하나만 존재하도록 보장되어야 하는 클래스이므로 싱글턴(Singleton) 디자인 패턴을 적용하였다.

1.2.3 Scalable EOSPA Design

제안된 EOSPA는 추상화를 통한 클래스 식별과 디자인 패턴 적용을 통하여 클래스를 세분화하고 서로 의존성을 때어 냄으로써 신규 장비가 추가되더라도 속성 및 개수 등의 정보를 파일로 읽어 들이는 것으로 소프트웨어의 직접적인 수정이 일어나지 않도록 하였다. 가변적인 장비의 속성 값을 파일로 읽도록 하여 공통요소를 뽑아낸 특성을 클래스 객체화함으로써 신규 장비가 추가되어도 실행시 읽어 들이는 파일만 수정되도록 하였다. 이렇게 함으로써 확장성은 높이고 유지보수는 더욱 쉽도록 구현할 수 있었다. 다만 장비를 모사해주는 시뮬레이션 부분에서는 장비가 새로 추가되었을 경우 해당 로직의 계산을 수정해주어야 하는 부분이 존재하였다. 아래의 Fig. 8 파일은 제안된 EOSPA의 읽어 들이는 각 장비의 속성이 쓰인 파일이다.

ECS_DEVICE_SET.ini

```
Device,Port,TagNum,address,address-bit,type,bit-value,16bit-value,Changed,Connected,
SW-PP1,502,SW-PP1-01,10001,0,DI,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,ASCS_01_01,
SW-PP1,502,SW-PP1-02,10002,0,DI,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,ASCS_01_01,
SW-PP1,502,SW-PP1-03,10003,0,DI,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,ASCS_01_01,
SW-PP2,502,SW-PP2-01,10004,0,DI,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,ASCS_01_02,
SW-PP2,502,SW-PP2-02,10005,0,DI,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,ASCS_01_02,
SW-PP2,502,SW-PP2-03,10006,0,DI,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,ASCS_01_02,
GT,103,GT-2001,32001,0,DI,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,MCAS_01_03,
GT,103,GT-2002,32001,1,DI,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,MCAS_01_03,
GT,103,GT-2003,32001,2,DI,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,MCAS_01_03,
GT,103,GT-2004,32001,3,DI,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,MCAS_01_03,
GT,103,GT-2005,32001,4,DI,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,MCAS_01_03,
GT,103,GT-2006,32001,5,DI,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,MCAS_01_03,
GT,103,GT-2007,32001,6,DI,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,MCAS_01_03,
```

Fig. 8. Properties File of EOSPA

1.3 Step 3 : Implementation

구현단계에서는 속성파일로 각 장비 로직의 객체를 생성한다고 하여도 해당 장비와의 로직계산을 통하여 변경되는 값들을 관리하기 위해서 함정전투체계에서 사용하는 자료구조 들을 식별하여 효율성을 분석하여 EOSPA의 구조와 사용에 맞으면서도 안정적으로 사용할 수 있도록 모든 장비의 속성들을 List를 사용하여 모든 정보를 저장하고 변경되는 정보도 해당 List에 저장될 수 있도록 하였다.

여러 클래스와 외부 입력에 의하여 List에 접근하여 값을 바뀌는 것을 방지하기 위하여 외부 입력을 관리하는 List와 내부 계산을 위한 List를 구분하여 관리하였으며 계산이 끝난 List와 기존에 입력된 List의 변경을 확인하여 결과 값을 외부로 전달해주도록 구현하였다.

1.4 Step 4 : Test

제안된 EOSPA에 각 계통의 HMI 시뮬레이터를 통하여 운용자 입력 값을 주고 EOSPA의 출력 값을 확인하는 테스트를 진행하였다. 현재는 각 계통별 HMI가 개발이 진행중 이므로 각 계통의 HMI의 기능을 단순화하여 관련 기능을 확인 및 정보가 전시되도록 구현된 별도의 HMI 시뮬레이터 프로그램을 개발하여 사용하였다.

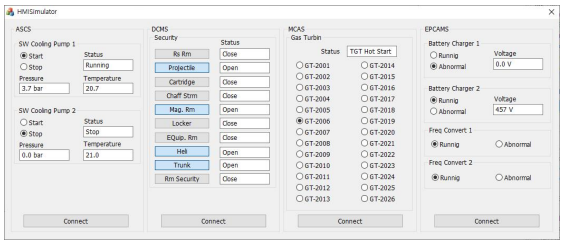

Fig. 9. HMI 시뮬레이터

HMI 시뮬레이터와 EOSPA가 적용된 OBTS 시뮬레이터 간의 Modbus TCP 통신을 통하여 정보를 주고 받으며 결과는 DDS로 OBTS 내부로 공유한다. 그 정보를 DDS 수집기를 별도로 개발하여 확인하였다.

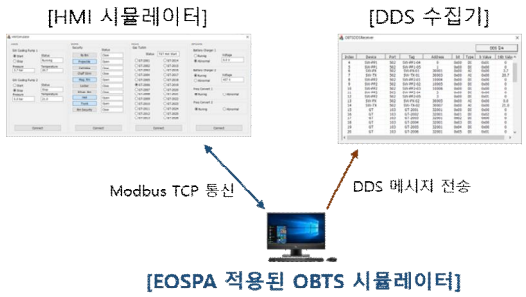

Fig. 10. Test Environment

IV. Performance Analysis

1. Proposed EOSPA OBTS Analysis

제안된 EOSPA가 적용된 OBTS와 기존의 해외에서 도입된 OBTS를 동일한 조건에서 비교하는 것은 불가하였다. 해외에서 도입된 OBTS의 경우 특정 장비에 실행프로그램이 설치되어 확보된 상태이기 때문에 코드의 총 라인 수, 코드 복잡도 분석, 코드 확장성, CPU 사용률 등을 확인 할 수 없었고 신규 장비 연동 시 수정이 필요한 항목에 대해서 비교하였고, 실행과 해당 프로그램의 운용 방식으로 비교 확인을 하였다.

1.1 Scalable and maintenance

제안된 OBTS의 경우, 장비가 신규로 추가되면 설정 파일 하나로 연동 및 기본 설정은 모두 가능하다. 코드 수정은 해당 장비의 로직을 계산하는 과정에서 필요하다. 기존의 OBTS는 장비가 신규로 추가될 경우 장비의 Input/Output 설정, 로직 계산 등을 모두 수정을 해주어야 한다. 제안된 OBTS는 장비의 초기 설정 등을 설정 파일 수정만으로 가능함으로 유지 보수에 효율적이다. 또한 해당 프로그램을 다른 함정에 설치 할 때 동일한 함정이 아닌 경우에는 기존의 OBTS는 장비 설정, 전체 로직을 모두 수정해야 한다. 하지만 제안된 OBTS는 변경되는 장비는 모두 설정 파일에서 수정하여 기본 설정을 코드 수정 없이 가능하고 로직 부분만 코드 수정이 필요하다.

1.2 Constraints

먼저 기존의 OBTS와 제안된 EOSPA가 적용된 OBTS의 운용적 제약사항에 대해서 비교를 하였다.

Table 3. OBTS Operational constraints

SW	User Num	Communication Method
OBTS	4	OPC
proposed OBTS	No limit	modbus TCP, DDS

위의 표와 같이 기존의 OBTS는 운용자가 최대 4명까지만 접속할 수 있었다. 제안된 OBTS의 경우에는 최초 목적부터 다양한 함정에 적용이 가능하도록 설계를 하였기 때문에 컴퓨터 사양이 허용하는 한 최대 접속 가능한 운용자를 제한하지 않도록 설계를 하였다. 이는 다양한 크기의 함정에 적용할 수 있는 이점이 존재한다.

기존의 OBTS의 외부와 통신은 OPC(OLE for Process Control) 통신을 이용하였다. 제안된 OBTS는 외부와의 통신은 모두 modbus TCP 통신으로 변경을 하였고 OBTS 내부의 UI와 서로 정보를 주고받기 위해서 DDS가 추가되었다. DDS 통신을 추가함으로 향후 함정전투체계와의 연동이 가능한 부분이다.

운용 방식에서는 큰 차이점이 존재한다. 기존의 OBTS의 경우 교관의 모든 통제 하에 운용자들이 교육을 진행할 수 있다. 교관의 HMI에서 OPC 통신을 위한 기본 설정을 매번 부팅 시마다 적용이 필요하기 때문이다. 제안된 OBTS의 경우 부팅 시 modbus TCP 연결을 하여 통신 설정을 자동으로 진행하기 때문에 시나리오 훈련 등 특이사항을 제외하고는 운용자들의 자유훈련이 가능하다.

V. Conclusions

본 연구를 통해 제안된 함정 통합기관제어의 함상훈련 계통은 최초 개발부터 유지보수와 재사용성이 뛰어난 객체지향 프로그래밍을 설계의 기본으로 둔 EOSPA를 적용함으로써 현재 연구 중인 함상 통합기관제어 공통 적용 소

프트웨어에서도 개선 및 확장, 유지보수에 높은 효율을 낼 것으로 보이며 향후 다양한 함정에서 적용에도 유용하게 사용될 수 있을 것으로 예상된다.

EOSPA의 여러 컴포넌트 중 표준화를 달성한 부분도 존재하는 반면, ECS Calc 컴포넌트의 경우 함정별로 변경되는 부분이 상당히 많아지는 부분에서 표준화를 달성하지 못한 것이 아쉬운 부분이다. 향후, EOSPA의 ECS Calc 컴포넌트의 표준화를 위한 설계 고도화와 리팩터링을 통하여 본 논문을 통하여 표준화를 달성하지 못한 부분에 대하여 기존대비 개발편의성과 재사용성을 높일 방안에 대해서 연구가 필요하고 함정 통합기관제어 함상훈련계통 전반에 걸친 표준화에 대한 연구가 필요 할 것으로 예상된다.

ACKNOWLEDGEMENT

This research was supported by Defense Industry Technology Center of Korea(Development of Common SW for Naval Ship Engineering Control System)

REFERENCES

[1] Seung-Woo Shin, Sang-Hoon Lee, Dong-Jin Kim, ”A Study of localization of engineering control system for naval ships,“Naval Ship Technology & Weapon Systems seminar, Vol.1, No.1, pp.582-585, 2017.

[2] Sung-Wook Park, ”A Study On The Improvement of Engineering Control System Hardware,“The Institute of Electronics and Information Engineers, pp.621-622, 2015

[3] Jung-Sung Young, Hun-Seok Lee, Jin-Seok Oh, ”Development of an ECS Simulator for Warship Propulsion Systems,“ American Society of Naval Eng, Vol.132, No.2, pp.133-140, 2020.

[4] Peter Wegner, ”Concepts and paradigms of object-oriented programing,“ ACM SIGPLAN OOPS Messenger, Vol.1, No.1, pp.7-87, 1990.

[5] Robert C.Martin, ”Agile Software Development, Principles, Patterns, and Practices,“Prentice Hall, New Jersey, pp.95-145,2002.

[6] Robert C.Martin, ”Clean Code: A Handbook of Agile Software Craftsmanship,“ Prentice Hall, New Jersey, pp.138-140, 2008.

[7] Kim Mingon, Shim Jaesoon, Jung Sungyoung, Park Sungchan, ”A Study on the Naval Ship Propulsion System Dynamic Simulation Software Development ,Vol.1,No.1.pp.100-100, 2018.

[8] Kyoun-Wan Lee, Kwang-yul Yu, Sung-Chan Park, Jeong-Soo Kim, Min-Gon Kim, Moon-Chan Kim, ”Controllable Pitch propellers for the simulation of naval ship propulsion system dynamics,“ Journal of the Korean Society of Marine Engineering, Vol.43, No.9,pp.693-700, 2019.

[9] Jea-hee Jang, Seung-woo Shin, Min-gon Kim, Jin-seok Oh, ”Development of CODOG Propulsion System Simulator,“ Journal of the Korea Institute of Information and Communication Engineering, Vol. 21, No. 9, pp.1808-1817, 2017.

[10] Hunseok Lee, Jae-hee Jang, Nayoung Son, Young-min Kang, Min-wook Kim, Dong-jin Kim, Jin-Seok Oh, ”Simulator Development for the Control System of CODLOG Propulsion System,“ The Korean Society of Marine Engineering Fall Conference, Busan Port International Passenger Terminal, 2016.

[11] Ki-Tae Kwon, Ki-Pyo Kim, HwanJun Choi, ”Design of the Scalable Naval Combat System Software using Abstraction and Design Pattern,“ Journal of the Korea Society of Computer and Information, Vol. 24, No. 7, pp.101-108, 2019.

[12] Hun-Yong Shin, Joo-Yong Kim, ”Research of OSD standardization in Naval Combat System,“ Information and Control Symposium, pp.354-355, 2012.

Authors

Seung-Chul Kwak Received the B.S degree in Information and Communication Engineering from Yeungnam Univ., Korea., in 2014. He is currently an Engineer in Hanwha Systems Co., Ltd.

He is interested in Combat System, Engineering Control System, Standard Programming Model and so on.