creative commons

COMMONS DEED

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

dCollection

공학석사학위논문

체계공학 및 인터페이스 관리를 이용한 초기단계 함정설계방법 연구

Early-phase Design of Naval ship using Systems Engineering and Interface Management

2016년 8월

서울대학교 대학원
조선해양공학과
곽 해 민

체계공학 및 인터페이스 관리를 이용한 초기단계 함정설계방법 연구

지도교수 신 종 계

이 논문을 공학석사학위논문으로 제출함

2016년 7월

서울대학교 대학원

조선해양공학과

곽 해 민

곽해민의 석사학위논문을 인준함

2016년 7월

위 원 장 장 범 선 (인)

부 위 원 장 신 종 계 (인)

위 원 양 영 순 (인)

초 록

함정은 기술발전에 따라 함정시스템이 복잡화 되어 일반적인 선박과 달리 다학제적 특성을 보유하며, 복잡한 상호작용이 수반됨에 따라 기존의 나선형설계에서 각각의 분야 설계결과가 동시에 다른 분야에 영향을 주는 동시설계로 전환되고 있다. 또한 함정은 개발구상부터 실전배치 및 운용에 이르기까지 일반적으로 10년 이상의 기간이 필요하며, 이에 따라 설계 진행간 탑재되는 구성품의 변경이 요구되기도 한다.

이러한 특성에 따라 함정은 상선대비 20배 이상의 설계변경 소요가 발생하며, 설계변경 발생 시 후속설계변경 미실시로 인한 설계오류가 발생할 가능성이 높다.

이에 따라 본 논문에서는 함정 설계 시 발생하는 설계변경에 의한 주요 설계오류 발생을 최소화하기 위하여 설계변경 감소를 위한 초기설계 단계의 체계공학 적용 방안 및 인터페이스 관리를 통한 설계오류 감소방법을 연구하였다.

설계변경 감소를 위하여 요구조건 분석, 기능분석 및 할당, 설계조합 등 체계공학 절차를 수정, 보완하여 함정의 초기설계에 적용 가능한 활동을 연구하였다. 또한 체계공학 절차인 기능분석을 통해 장비간의 인터페이스를 식별하고, 이를 DSM으로 표현하여 인터페이스를 관리함으로써 설계변경 전파경로를 식별하는 데 사용 가능한 도구로 사용하기 위한 방법을 연구하였으며, 이를 함정 설계에 적용하였다.

주요어 : 체계공학, 기능분석, 설계변경 관리, 설계변경 전파, Interface 관리

학 번 : 2014-20652

논문 목차

1. 개　요 ·····1
1.1 연구배경 ·····1
1.1.1 함정설계의 특징 ·····1
1.1.2 함절설계 진행 중 많은 설계변경 발생 ·····5
1.1.3 설계변경 발생에 따른 후속설계변경 미실시로 인한 설계오류 ·····7
1.2 연구목적 및 방법 ·····7
1.3 논문구성 ·····8

2. 설계변경 감소를 위한 초기설계 단계의 체계공학 적용 방안 연구 ·····9
2.1 설계변경 발생의 주요 이유 ·····9
2.2 체계공학(Systems Engineering) ·····10
2.2.1 SE 일반 ·····10
2.2.2 MBSE 일반 ·····15
2.3 초기단계 함정 시스템 설계 방법 ·····24

3. 기능분석을 이용한 인터페이스 식별 및 DSM을 이용한 인터페이스 관리를 통한 설계오류 감소방법 연구 ···27
3.1 설계변경의 후속설계변경 미실시로 인한 설계오류 발생 이유 ·····27

3.2 후속 설계변경 미실시로 인한 설계오류 감소를 위한 인터페이스 및 DSM 일반사항 ·····29
3.2.1 인터페이스 일반 ·····29
3.2.2 DSM 일반 ·····32
3.3 기능분석을 이용한 인터페이스 식별 및 DSM을 이용한 인터페이스 관리방법 ·····35

4. 초기설계 단계의 체계공학 설계 방법의 적용 ·····37
4.1 체계공학 방법을 이용한 함정 개념설계 ·····37
4.2 적용효과 ·····46

5. 기능분석을 이용한 인터페이스 식별/관리방법 적용 47
5.1 추진체계에 대한 인터페이스 식별/관리방법 적용 ···47
5.2 적용효과 ·····55

6. 결 론 ·····56

참고문헌 ·····57

Abstract ·····60

표 목차

표 1 함정의 대형 복합체계 특성 ·····3

표 2 Interface Matrix 예 ·····36

그림 목차

그림 1 함정기술 발전추세 ····· 1
그림 2 나선형 설계 ····· 2
그림 3 동시설계 ····· 2
그림 4 함정 설계프로세스의 복잡성 예 ····· 4
그림 5 함정 연구개발 세부절차 ····· 5
그림 6 함정과 일반상선의 설계변경 소요 ····· 6
그림 7 함정과 일반상선의 설계/건조 단계별 설계변경 소요 ····· 6
그림 8 VEE Diagram ····· 13
그림 9 EIA/IS-632 Process ····· 14
그림 10 기능분석 및 할당 ····· 15
그림 11 MBSE 개념 ····· 16
그림 12 Concept of Onion Model ····· 16
그림 13 요구사항 다이어그램 예 ····· 18
그림 14 EFFBD 예 ····· 20
그림 15 EFFBD 제어구문 ····· 20
그림 16 IDEF0 예 ····· 21
그림 17 IDEF0 구조 ····· 21
그림 18 N2 Chart 예 ····· 22
그림 19 거동 할당 ····· 23
그림 20. 체계공학 프로세스(SE Process) ····· 24
그림 21. 요구조건 분석 활동 ····· 25

그림 22. 기능분석 및 할당 활동 ·····25
그림 23. 설계조합 활동 ·····26
그림 24 Electrical Interface 예 ·····29
그림 25 Software Interface 예 ·····31
그림 26 Interface 분해 ·····32
그림 27 시스템 구성요소 간의 관계 ·····33
그림 28 Binary DSM의 예 ·····33
그림 29 Numerical DSM의 예 ·····33
그림 30 구성요소간 관계성의 종류 ·····34
그림 31 기능분석 및 할당 ·····35
그림 32 DSM 예 ·····36
그림 33 요구사항 분석 결과 ·····37
그림 34 요구사항 분석을 통한 요구조건 검증 ·····38
그림 35 최상위 기능거동 ·····39
그림 36 함정 운용프로파일 정의 ·····39
그림 37 기능분석 결과 ·····40
그림 38 요구조건-기능 Matrix ·····40
그림 39 물리구조 분석 결과 ·····41
그림 40 그림 40 기능-물리 Matrix ·····41
그림 41 함정 종합효과도체계(OMOE) ·····42
그림 42 가중치 산정 결과 ·····43
그림 43 모든 설계대안 도출(486개) ·····44

그림 44 조선공학적 제한사항 만족하는 설계대안(233개) ·····45
그림 45 일반적인 CODLAG 시스템의 구성 ·····47
그림 46 System Context ·····48
그림 47 추진체계 기능구성 ·····49
그림 48 추진계통 기능 구성 ·····50
그림 49 전기계통 기능 구성 ·····50
그림 50 보기계통 기능 구성 ·····51
그림 51 추진체계 구조 ·····52
그림 52 추진계통 구조 ·····52
그림 53 전기계통 구조 ·····53
그림 54 보기계통 구조 ·····53
그림 55 기능할당 ·····54
그림 56 Interface Matrix ·····54
그림 57 DSM으로 표기된 인터페이스 ·····55

약어

CODLAG	COmbined Diesel eLectric And Gas turbine
CTE	Critical Technology Element
DSM	Design Structure Matrix
EFFBD	Enhanced Functional Flow Block Diagram
IDEF0	Integration Definition for Function Modeling
MBSE	Model-Based Systems Engineering
OMOE	Overall Measure of Effectiveness
KPP	Key Performance Parameter
SE	Systems Engineering

1. 개요

1.1 연구배경

1.1.1 함정설계의 특징

함정은 네트워크, 센서, 무장, 플랫폼의 통합된 복합시스템(System of Systems)으로 날이갈수록 복잡성, 통합성, 정교성이 증대되고 있다. <그림 1>에서 보는바와 같이 추진동력(화석연료에서 핵연료), 무장(함포에서 미사일), 네트워크(Task Force에서 FORCENet) 등 함정에 적용되는 기술이 발전해가고 있다.

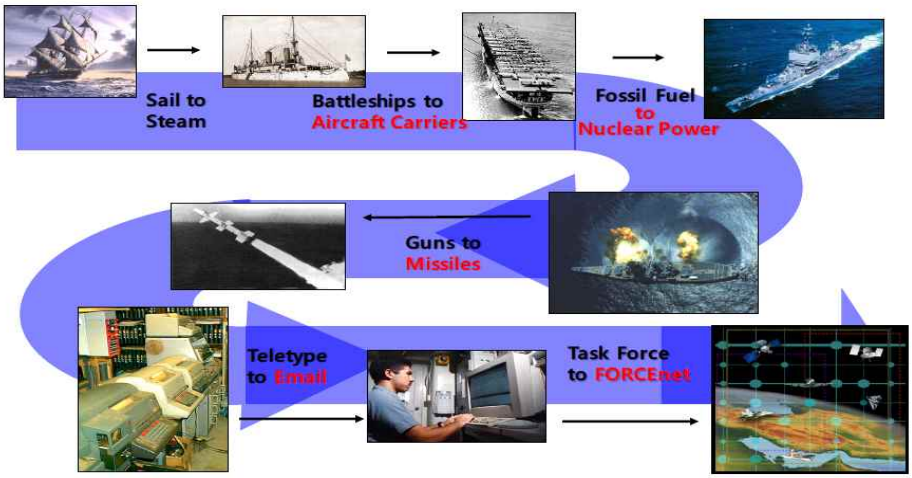

그림 1 함정기술 발전추세 [5]

이러한 함정 시스템의 복잡화로 일반적인 선박과 달리 다학제적 특성을 보유하며, 복잡한 상호작용이 수반된다. 또한 분야별 설계결과가 동시, 상호연관됨에 따라 주요제원 및 마력추정, 선형 설계 등의 순서로 차례대로 설계하는 나선형설계(<그림 2>)에서 각각의 분야 설계결과가 동시에 다른 분야에 영향을 끼치는 동시설계(<그림 3>)로 전환되고 있다.

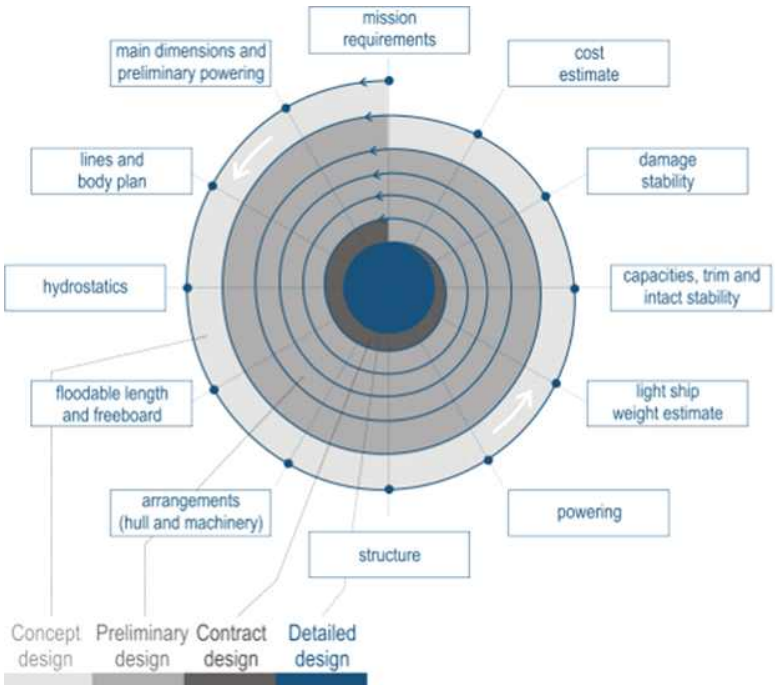

그림 2 나선형 설계 [5]

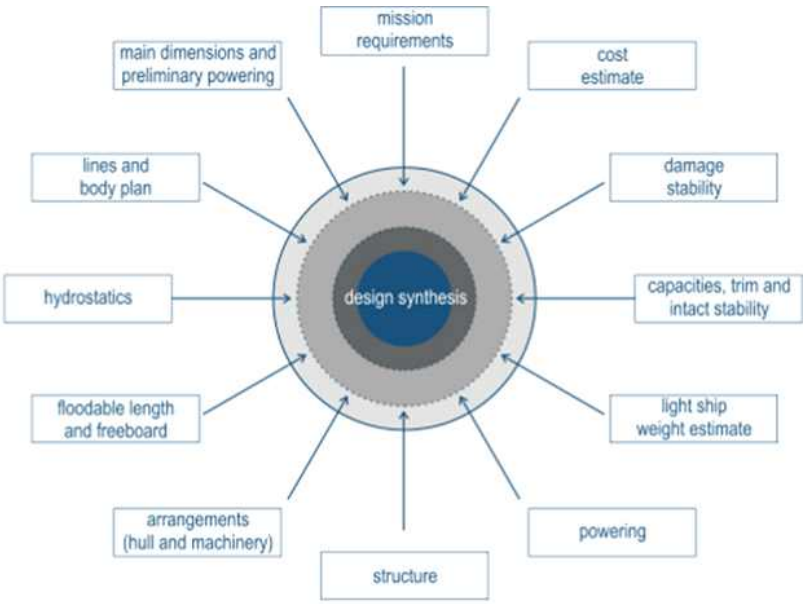

그림 3 동시설계 [5]

대형 복합체계로서의 함정의 복잡성은 지상 무기체계의 대표격인 전차와 대형 항공기 체계인 보잉 777 여객기와 비교할 경우 더욱 명확히 나타난다. 아래의 <표 1>를 통해 원자력추진 잠수함의 구성 시스템의 개수 및 소요부품에서 다른 시스템과의 현격한 차이를 볼 수 있으며, 이에 따라 함정 설계 시 고려해야 하는 사항이 급격히 증가하며 복잡한 설계프로세스를 거쳐야 한다.(<그림 4>)

구 분	M-1 전차	Boeing 777 항공기	SEAWOLF급 핵잠수함
Weight(T)	65	250	9,137
Length(Ft.)	25	200	353
Number of Systems	25	40	230
Crew Size	4	10(2 pilots)	116
Patrol Duration(Hr.)	24	8~14	~2,100
Number of Parts to Assemble	14,000	100,000	1,100,000
Assembly Man-hours / Unit	5,500	50,000	23,000,000
Production Time(Mo.)	7.5	14	72
Production Rate(Units/Yr.)	600	72	0.16

표 1 함정의 대형 복합체계 특성 [8]

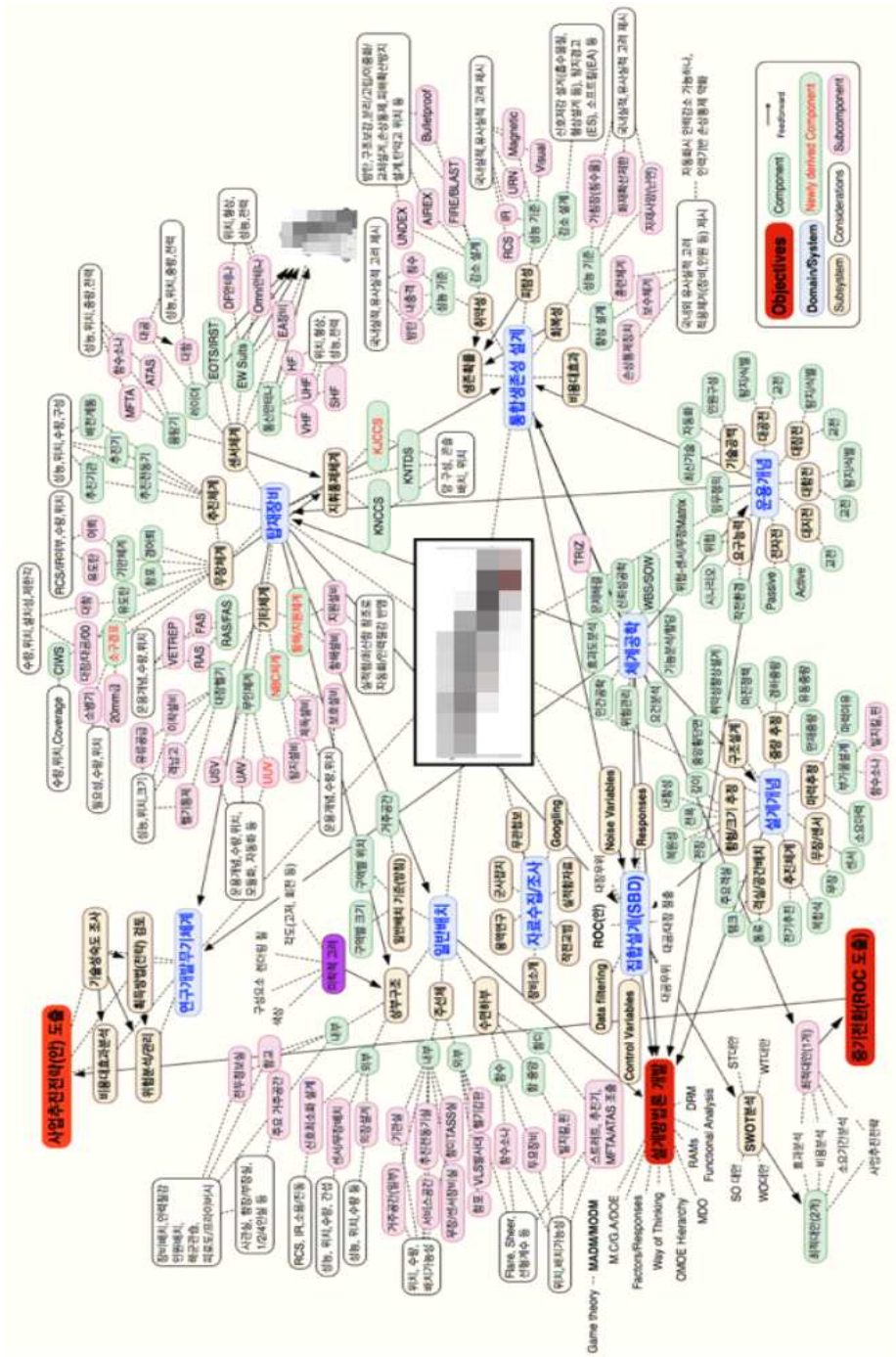

그림 4 함정 설계프로세스의 복잡성 예 [5]

또한 함정은 신규 무기체계의 경우 개발구상 단계에서부터 연구발 및 제작생산, 실전배치 운용단계에 이르기까지 일반적으로 10년에 가까운 긴 기간과 막대한 비용이 투입된다.(<그림 5>) 함정을 구성하는 선박체계(Platform System)는 일반적으로 30년 이상의 유효수명을 갖는 반면, 무장과 전자장비로 구성되는 전투체계(Combat System)는 군사과학기술의 급격한 발전으로 인해 유효수명이 수년 미만으로 극히 짧으며, 이에 따라 설계진행중에 탑재되는 장비 및 설비의 변경이 요구되기도 한다.

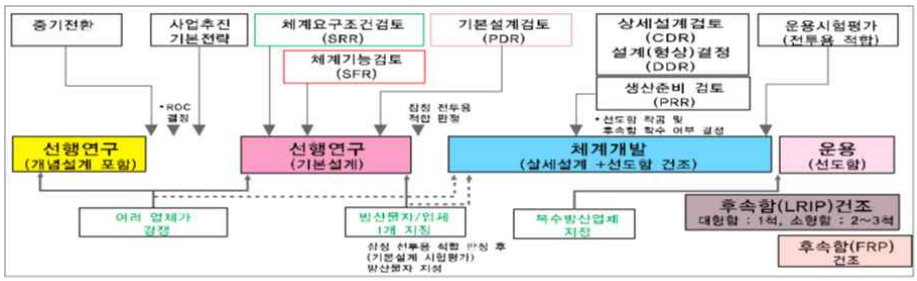

그림 5 함정 연구개발 세부절차 [9]

1.1.2 함정설계 진행 중 많은 설계변경 발생

함정은 상선대비 20배 이상의 훨씬 많은 설계변경 소요가 발생하고 있으며, 초도함 뿐만 아니라 후속함에서도 많은 설계변경 소요가 발생한다.(<그림 6>) 특히 함정의 경우 설계단계 뿐만 아니라 조립, 의장, 시운전단계에서도 60% 이상의 설계변경이 발생하는 특징이 있다.(<그림 7>)

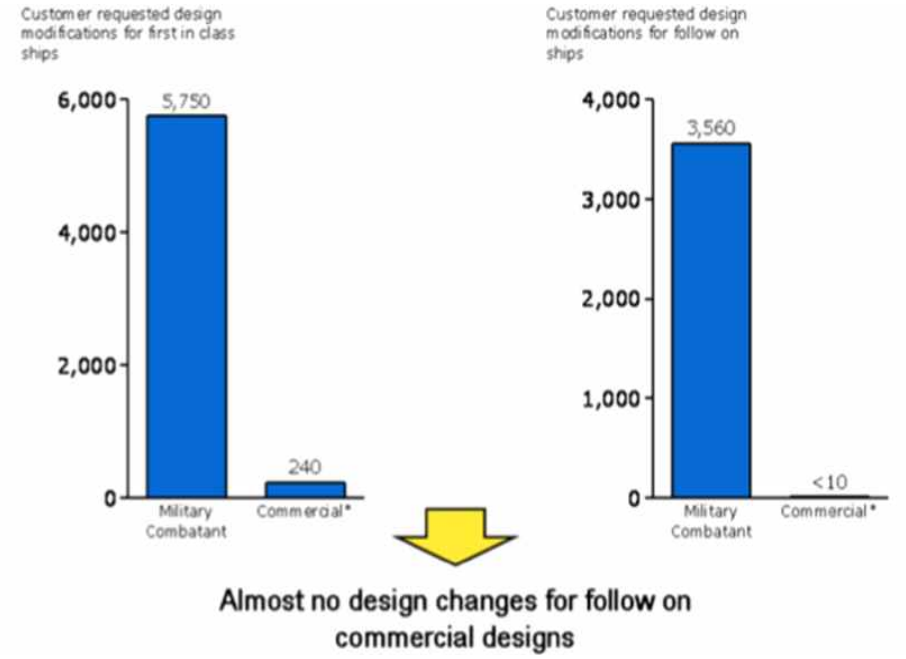

그림 6 함정과 일반상선의 설계변경 소요 [12]

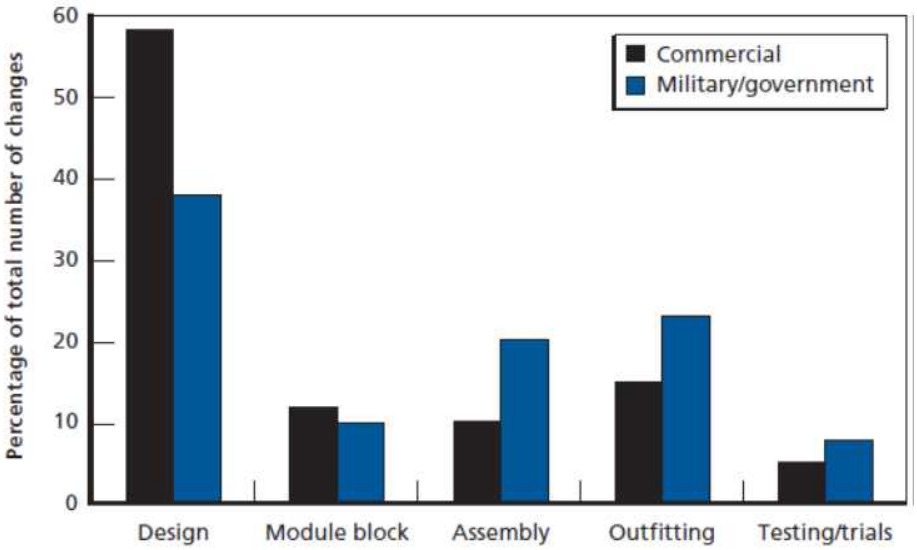

그림 7 함정과 일반상선의 설계/건조 단계별 설계변경 소요 [21]

1.1.3 설계변경 발생에 따른 후속설계변경 미실시로 인한 설계오류

신규함정 설계 시 설계 기준이 설정되어 있고 숙련된 설계자에 의해 설계가 수행되나, 설계과정 중 설계변경 발생 시 해당내용이 연관된 분야로 전파되지 않아 설계오류가 발생하는 경우가 발생한다. 특히 함정에 설치되는 장비/설비는 발주에서 납기까지 오래걸리는 종류가 많음을 고려하면 설계오류는 건조기간 지연 등 큰 문제가 발생할 가능성이 크다.

호주의 콜린스급 잠수함 설계 시 발생했던 설계 중 형상변경으로 인한 소음발생 문제, 함내 전원 케이블교체로 인한 14개월의 사업 지연 발생[11] 등의 사례를 보았을 때, 설계오류 발생에 따른 영향이 크게 발생함을 알 수 있다.

1.2 연구목적 및 방법

본 논문은 초기단계 함정 설계시 설계변경의 발생빈도를 감소시키기 위해 MBSE(Model-Based Systems Engineering) 도구를 이용한 체계공학 절차 적용 방법에 대해 연구하고, 후속설계변경 미실시에 의한 설계오류를 감소하기 위한 기능분석을 이용한 인터페이스 식별 및 DSM을 이용한 관리방법을 검토 및 제시함으로써 함정 설계시 발생하는 설계변경에 의한 설계오류 발생을 최소화하고자한다.

1.3 논문구성

본 논문은 전체적으로 6장으로 구성되어 있다.

1장에서는 연구배경, 목적을 제시하여 전체적인 연구 방향을 제시하였다.

2장에서는 설계변경 감소를 위한 방법으로써 초기설계 단계의 체계공학 적용 방안을 제시하며, 체계공학 방법론을 함정의 초기설계 단계에서 적용할 프로세스를 제시하였다.

3장에서는 설계변경에 의한 설계오류를 줄이기 위한 방법으로써 기능분석을 이용한 인터페이스 식별/관리방법을 제시하며, 함정의 초기설계 단계에서 체계공학 방법론 중 기능분석을 이용한 인터페이스를 식별 및 관리방안을 제시하였다.

4장에서는 2장에서 제시한 방법론을 함정의 초기설계 단계인 개념설계에 적용하고 그 결과를 제시하였다.

5장에서는 3장에서 제시한 방법론을 함정의 추진체계 설계에 적용하고 그 결과를 제시하였다.

6장에서는 논문을 정리하며, 본 연구를 통해 기대되는 효과에 대해 제시하였다.

2. 설계변경 감소를 위한 초기설계 단계의 체계공학 적용 방안 연구

2.1 설계변경 발생의 주요 이유

함정설계에 있어 주요 설계변경 발생 이유를 검토한 결과 아래와 같은 이유를 식별하였다.

먼저 사용자에 의한 설계변경 발생이 발생하였다. 설계자는 설계단계에 의해 설계를 수행하였으나, 최초 운용자의 요구사항이 만족되지 않거나 계약도면과 함정사양서간 불일치가 발생하였을 때 설계변경이 발생하였다. 또한 장기적인 획득기간 중 발생하는 최신기술 반영을 요구하거나, 설계중간에 운용자에 의해 요구되는 개선사항이 발생할 경우 설계변경이 발생하였다.

또한 서로다른분야의 설계가자 참여함에 따라, 하위시스템간 상호요구조건을 만족시키지 못하여, 연관된 장비(설비)의 사양이 변경됨에 따라 발생하는 설계자에 의한 설계변경이 발생하였다.

이러한 설계변경 발생 사유 중 계약도면과 함정사양서간 불일치 수정 등 설계결과가 요구사항과 다를 때 발생하는 설계변경과, 연관된 장비(설비) 사양 변경에 따른 개선이 발생하는 설계변경은 설계 초기단계부터 요구조건 분석 및 연관분야 의견 검토를 통해 설계변경 발생 빈도를 감소시켜야 하며, 이를 체계공학(Systems Engineering) 절차 적용을 통해 이를 달성하고자 한다.

2.2 체계공학(Systems Engineering)

2.2.1 SE 일반

2.2.1.1 시스템의 정의 및 특징

시스템에 대한 어원은 「조직화된 전체」라는 의미의 그리스어 “SYSTEMA” 에서 시작되었다. 시스템에 대한 정의는 시대적 환경에 따라 변화되어왔으며, 아래와 같이 여러 가지로 정의된다.

- ISO/IEC 15288(2002) : 부여된 목적을 달성하기 위하여 조직된 상호작용하는 요소들의 결합체 - IEEE-1220(2005) : 제품 수명주기 유지를 위해 사용되는 요소들의 집합 또는 배열 - MIL-STD-499C(2005) : 단일개채만으로는 이룰 수 없는 목적을 달성하기 위해서 상호연관된 여러 개체의 집합 - INCOSE SE Handbook(2006) : 하나 이상의 기술된 목적을 달성하기 위하여 상호작용하는 구성요소들의 결합

일반적으로 시스템(System)은 정의된 요구(needs)와 목적(objectives)을 충족시키기 위해 능력을 제공하는 사람(people), 제품(products), 프로세스(processes)의 결합체이며, 광의의 시스템은 정의된 목적을 이루기 위한 요소들의 결합체를 의미한다. [3]

이때 시스템의 정의는 각 엔지니어링 분야에서 다르게 나타나며, 주로 요소(elements)와 시스템이 혼돈되는 경우가 많으며, 이는 시스템의 목적이나 기능을 어떻게 정의하느냐에 따라 달라진다. 예를 들면 비행기나 자동차 등은 어느 한 레벨에서 시스템으로 분류될 수 있으나, 다른 레벨에서는 단지 하나의 요소나 하부시스템으로 구분된다. 이들은 운송시스템의 주요요소로서 개발 대상의 최종제품이

된다. 또한 주어진 기능을 수행하기 위한 연료, 전력, 정비 등과 같은 다른 지원요소(지원제품) 또한 시스템의 일부로 구성된다. 즉, 시스템은 정의된 목적을 이루기 위해 함께 작용하는 요소들의 결합체로, 하드웨어, 소프트웨어, 펌웨어, 인원, 정보, 기술, 설비 및 기타 지원요소를 포함한다. [3]

시스템의 속성은 구체적으로 아래와 같다

- 시스템은 목적(purpose)을 가지고 있어야 한다
- 시스템은 반드시 정해진 기능과 절차에 따라 동작되어야 한다
- 시스템은 필요한 기능을 수행하기 위하여 여러 개의 요소로 구성된다.
- 시스템은 계층구조(hierarchy)형태로 이루어지며 계층구조 내의 시스템 요소는 상호 유기적인 협조관계를 유지해야 한다.

시스템이 복잡해지고 다양한 분야의 전문가들이 함께 공동작업을 하기 위해서는 명확한 의사소통과 시스템의 정의, 요소, 관련 프로세스 등을 분명히 제시하여야 하며, 특히 시스템 계층구조(system hierarchy) 레벨을 정의하는 것이 중요하다. 이러한 시스템 계층구조는 시스템의 종류 및 목적에 따라 달라질 수 있며, 일반적인 레벨 관점에서 정의하면 아래와 같다

- 시스템(system) : 하나 이상의 기술된 목적을 달성하기 위하여 상호 작용하는 구성요소들의 결합
- 요소(element) : 시스템의 주요 제품, 서비스 또는 설비
- 하부시스템(subsystem) : 명확하게 분리된 기능을 수행하는 조립품(assembly)의 통합체
- 조립품(assembly) : 하부시스템을 구상하는 하부조립품 또는 구성품의 통합체
- 하부조립품(subassembly) : 잘 정의된 조립품의 일부를 구성하는 구성품, 부품의 통합체
- 구성품(component) : 여러 부품들로 구성되며 분명하게 식별된 품목(item)
- 부품(part) : 개별적으로 식별 가능한 품목의 최하위 레벨

2.2.1.2 SE 기본개념

시스템엔지니어링(SE, Systems Engineering)이란 고객의 요구를 충족시키는 통합된 수명주기를 고려한 균형잡힌 다학제간 접근방법을 말하며, 시스템의 정의의 변화와 함께 아래와 같이 여러 가지로 정의된다.

- IEEE-1200(1995) : 고객의 기대를 만족시키고 공공의 수락을 얻는 생명주기를 고려할 때 균형잡힌 시스템 해결방안을 유도, 전개 및 검증하기 위한 학제간 협력 접근법 - EIA/IS-632(1994) : 고객요구(Customer Needs)를 충족시기키 위하여 통합되고 수명주기적으로 균형된 시스템 제품과 프로세스 솔루션을 개발하고 검증하기 위한 다분야 학문적인 상호 연관된 접근방법 - INCOSE SE Handbook(2006) : 성공적인 시스템의 실현을 가능하게 하는 다분야 융합적 접근방법 - NASA SE Handbook(2007) : 시스템을 설계, 구현 및 운용하기 위한 강건한 접근법

시스템엔지니어링의 기본개념은 아래와 같이 설명할 수 있다.

1) 전체적인 관점에서 시스템 하향식(top-down) 분석과 상향식(bottom-up) 통합 접근방법
2) 시스템 설계, 개발, 생산/건설, 분배, 운용, 유지 및 지원, 폐기 및 처분 등 모든 단계에 걸친 요구사항을 촉에 도출하고 획득하고 운용하는 수명주기(life-cycle) 관점
3) 초기 시스템요구사항을 특정 설계목표와 적합한 설계기준을 제시하고 설계프로세스에 보다 빨리 효과적인 방법으로 의사를 결정하고 분석할 수 있도록 미리 식별
4) 효율적인 시스템설계 및 개발 프로세스는 다분야 학문분야(interdisciplinary)에 종사하는 전문가들의 협력활동을 통하여 요구시스템의 설계목표를 달성 가능

2.2.1.3 SE 설계프로세스

시스템엔지니어링관점에서 전체적인 개발 프로세스(Development Process)는 VEE Diagram으로 대표되며, <그림 8>에서 보는바와 같이 설계를 분해, 구체화 시켜나가고, 이를 토대로 시스템을 통합, 결합하면서 각 단계에서 검증을 통해 최종적인 시스템을 검증해 나가는 과정으로 정의한다.

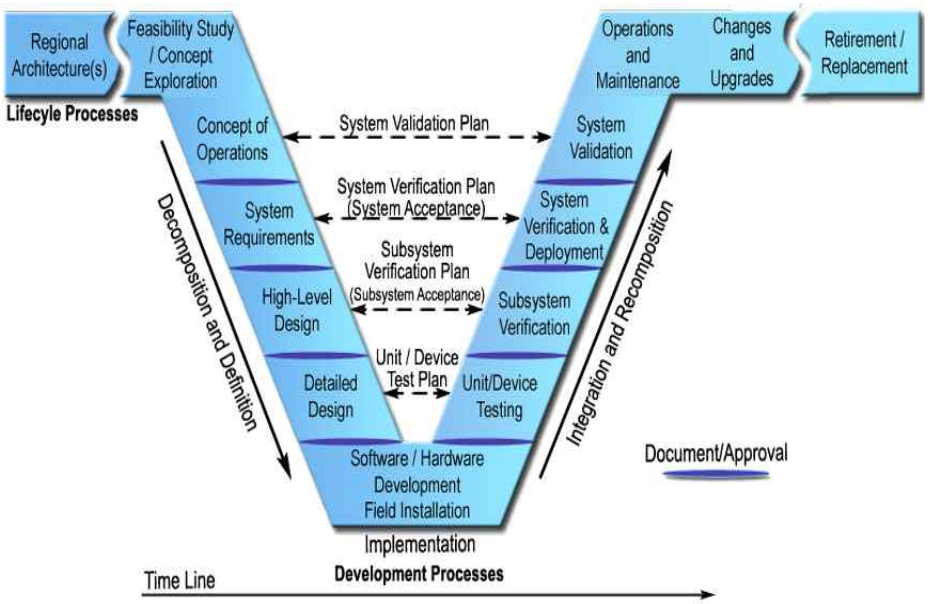

그림 8 VEE Diagram [21]

이러한 각 단계에서의 개별적인 설계 프로세스(Design Process)는 가장 기본적이고 폭넓게 사용되고 있는 EIA/IS-632 시스템엔지니어링 프로세스를 중심으로 설명할 수 있다.

EIA/IS-632 관점에서 시스템엔지니어링 프로세스는 <그림 9>에서 볼 수 있듯이 모든 개발단계 동안 적용되는 하향식(top-down)이고 포괄적이며 반복적인 순환적인 문제 해결과정으로 다음과 같은 목적을 가지고 있다.

- 요구(needs)와 요구사항(requirements)을 시스템 제품과 프로세스 설명서로 변환(개발단계가 진행됨에 따라 질이 높아지고 보다 구체화 됨)
- 의사 결정자에게 필요한 정보 생성
- 다음 개발 단계에 필요한 입력 자료 제공

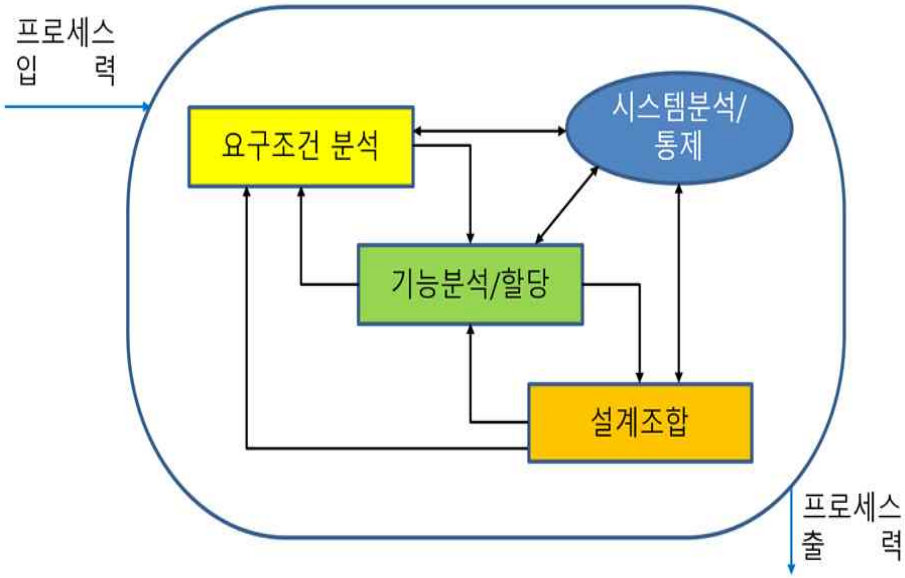

그림 9 EIA/IS-632 Process

EIA/IS-632 프로세스는 시스템 분석 및 통제로 불리어지는 균형적인 관점에서 총체적인 기술과 도구를 통해 이루어지는 요구사항분석, 기능분석 및 할당, 설계조합으로 구성된다. 시스템엔지니어링 통제는 의사결정과 요구사항을 추적하고 베이스라인을 유지하며, 인터페이스와 위험을 관리하는 활동이며, 비용, 일정 및 기술성능을 추적하고 요구사항의 충족여부를 검증하며, 진행과정에 대해 검토 및 감사활동을 수행한다.

기능구조는 할당된 기능과 성능 요구사항을 식별하고 구조화하며, 물리구조는 시스템이 하부시스템과 구성품으로 어떻게 분해되어지는지를 보여줌으로써 시스템 제품의 구성을 나타낸다.

이러한 요구사항 분석으로부터 기능구조, 물리구조 분석 및 할당의 과정을 통하여 최종 구성품이 가져야 하는 기능을 구체적으로 제시하여, 요구사항을 만족하는 효율적인 시스템을 구성을 할 수 있다. <그림 10>

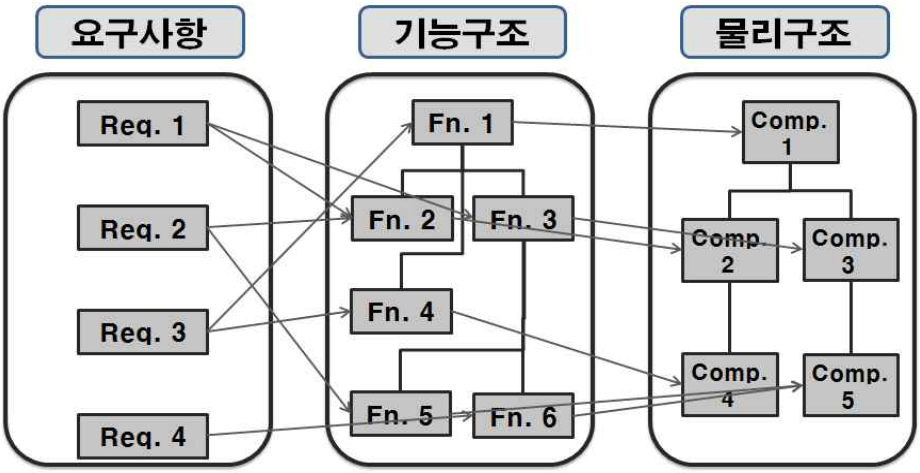

그림 10 기능분석 및 할당

2.2.2 MBSE 일반

모델기반 시스템엔지니어링(MBSE, Model-Based Systems Engineering)은 초기 개념설계단계부터 시작하여 전반적인 개발단계 그리고 추후 라이프 사이클 단계까지 전 수명주기에 걸쳐 시스템 요구사항, 설계, 분석, 확인 및 검증을 지원하는 전산환경 기반의 시스템 모델링 방법(<그림 11>)이며, 이전단계의 결과물이 다음단계 입력사항으로 들어가며, 점진적으로 시스템을 구체화 시켜 나가기 때문에 'Onion Model' 으로 불리기도 한다.(<그림 12>)

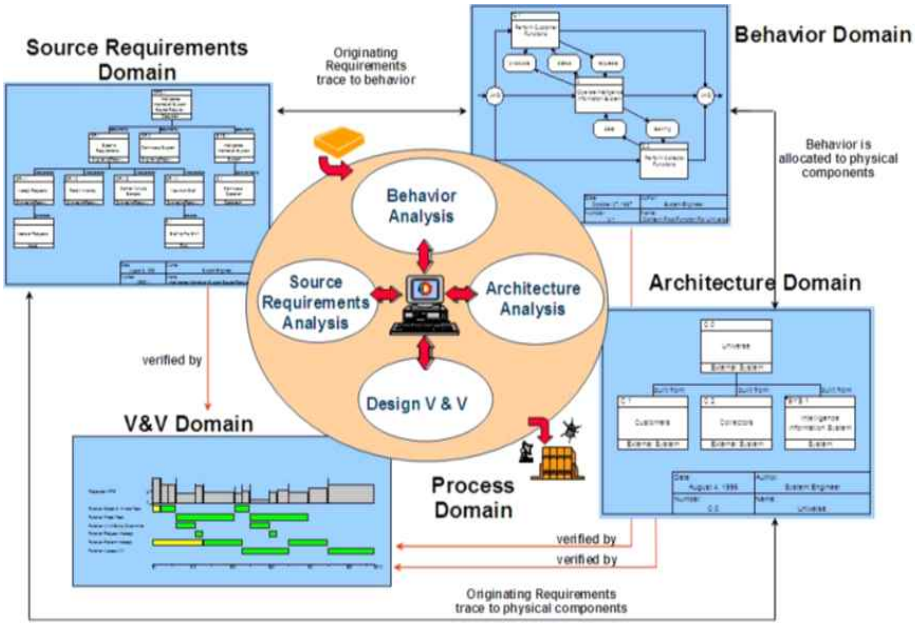

그림 11 MBSE 개념 [16]

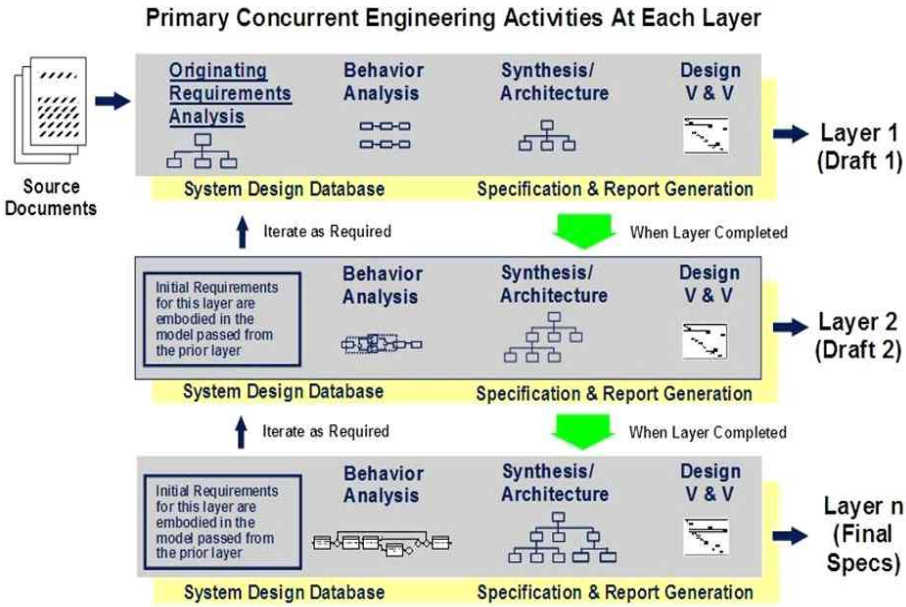

그림 12 Concept of Onion Model [16]

MBSE는 시스템의 복잡도가 증가함에 따라 설계 정보의 급격한 증가, 언어의 다의성과 서로 다른 경험으로 인한 잘못된 의사소통 및 추적성 관리의 어려움 등 전통적인 문서기반의 시스템엔지니어링의 문제점을 해결하고자 등장하였다. MBSE는 시스템 엔지니어링의 장점을 더 효과적으로 얻을 수 있는 전산환경 기반의 방법론으로 아래와 같은 장점이 있다.

MBSE는 단일 모델을 활용하여 여러 가지 다이어그램을 생성, 시스템의 전반적인 구조를 가시적으로 정의한다. 이는 단일모델을 통해 원하는 관점(view)에서의 다이어그램을 생성할 수 있으므로, 전체를 통합하는 다이어그램뿐만 아니라 세부적인 다이어그램을 생성 가능하게 하여 연관된 시스템의 확인을 용이하게 하고, 결과적으로 설계자의 이해를 도와 세부 설계를 지원한다.

MBSE는 시스템의 구조를 Requirement diagram, Functional Flow Block Diagram, N2 Chart 등의 가시적인 다이어그램으로 보여줌으로써 설계자와 이해관계자 사이의 이해와 설득을 원활하게 한다. 이를 통해 주요 의사결정점에서의 신속한 의사소통 및 의사결정을 돕는다.

덧붙여 MBSE는 최초 설계 요구사항, 제한사항에서부터 검증 요구사항에 이르기까지의 효율적인 요구조건 추적성을 보장하며, 동시에 모델의 변화이력까지 체계적으로 관리가 가능하다. 이를 통해 요구조건 도출에서부터 설계조합에 이르는 모든 함정 체계공학 활동을 단일 전산모델로 관리함으로써 설계행위 외 문서작성 등의 부수적인 부담을 감소시킬 수 있다.

MBSE는 요구조건 분석, 기능거동 분석, 물리구조 분석, 거동할당, 검증 단계로 수행하며, 아래와 같은 절차를 통해 수행한다.

2.2.2.1 요구조건(Requirement) 분석

첫 번째 단계는 설계요구사항으로 제시된 요구조건을 식별하여 가시적으로 분석하는 단계이다. <그림 13>과 같이 문서를 통해 초기 요구사항(Originating)을 분석하며, 이를 구체화하여 요구사항을 세밀히 구분(Refines)한다. 요구사항 분석 중 문제 발생 시(요구사항 모순 발생, 요구사항 불명확 등) 이해관계자의 협의를 통해 이를 구체화하여야 하며, 이를 통해 추가적인 요구사항을 도출하여(Derived) 하부 기능 및 성능 요구조건을 구체화한다.

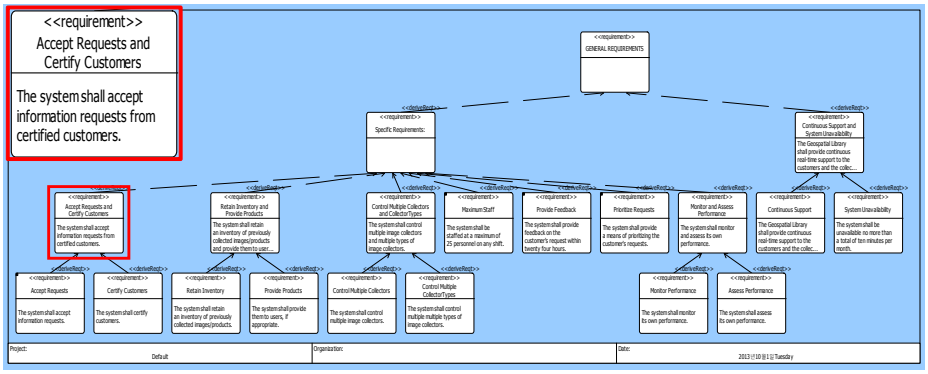

그림 13 요구사항 다이어그램 예 [22]

2.2.2.2 기능(Function) 거동(Behavior) 분석

시스템 거동은 시스템이 '어떻게' 작동하는지를 나타내며, 이러한 기능 거동분석을 통해 설계하는 시스템이 수행해야하는 기능을 구체화할 수 있다. 시스템 거동은 제어(기능과 구문) 모델과 인터페이스(입력과 출력) 모델이 통합된 단일 모델이다.

시스템이 가져야 하는 기능을 하향식으로 분해(decomposed)하여 각 기능의 흐름을 제어 구문(Control Constructs)을 통해 구체

화 하며, 동시에 데이터 트리거와 아이템을 통해 기능 간 정보교환 구체화 및 기능의 순차성이 구체화(제어)된다. 기능분해시 기능은 더 이상 구체화되지 않는 단계까지 분해하여야 한다.

기능 거동을 시각적으로 나타내는 다이어그램으로는 EFFBD (Enhanced Functional Flow Block Diagram), IDEF0 (Integration Definition for Function Modeling), N2 Chart 등이 있다.

EFFBD는 시간흐름에 따라 시스템 기능 흐름 및 시스템 기능간의 상호작용을 나타내는 다이어그램으로 시스템의 기능흐름을 나타내기 위해 주로 사용된다.(<그림 14>) EFFBD 다이어그램은 <그림 15>의 제어구문을 통해 구성되며, 기능(a)과 기능간의 인터페이스의 아이템(b)을 sequence(c), select(d), concurrency(e), multiple-exit(f)의 제어구문을 통해 제어된다. 이때 아이템(b)가 해당 기능의 시작에 영향을 주지 않는 경우(g)와 아이템이 전달되어야 기능이 시작되는 경우(트리거, h)를 이용하여 기능의 구성을 보다 구체적으로 작성할 수 있다.

IDEF0는 분석, 개발, 리엔지니어링 및 소프트웨어 엔지니어링 분석을 위한 일반적인 모델링 기법으로 EFFBD가 제품의 기능흐름을 나타내기 위해 사용되는 반면 데이터흐름, 시스템통제 및 수명주기 프로세스의 기능흐름에 사용된다.[3](<그림 16>) IDEF0 다이어그램은 <그림 17>과 같이 각 기능별 입력(Input), 출력(Output), 통제(Control), 지원메커니즘(Mechanism)으로 구성된다.

N2 Chart는 매트릭스 형식으로 시스템 요소 간의 기능적 또는 물리적 인터페이스를 나타내는 다이어그램으로 식별, 정의, 표로, 디자인·기능 및 물리적 인터페이스를 체계적으로 분석하는데 이용한다.(<그림 18>)

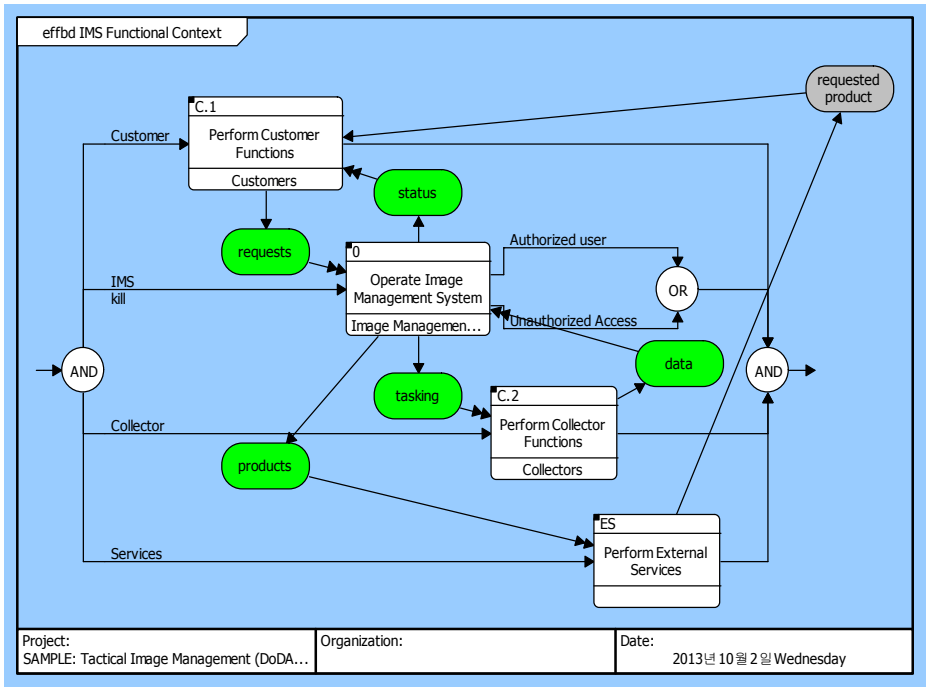

그림 14 EFFBD 예 [24]

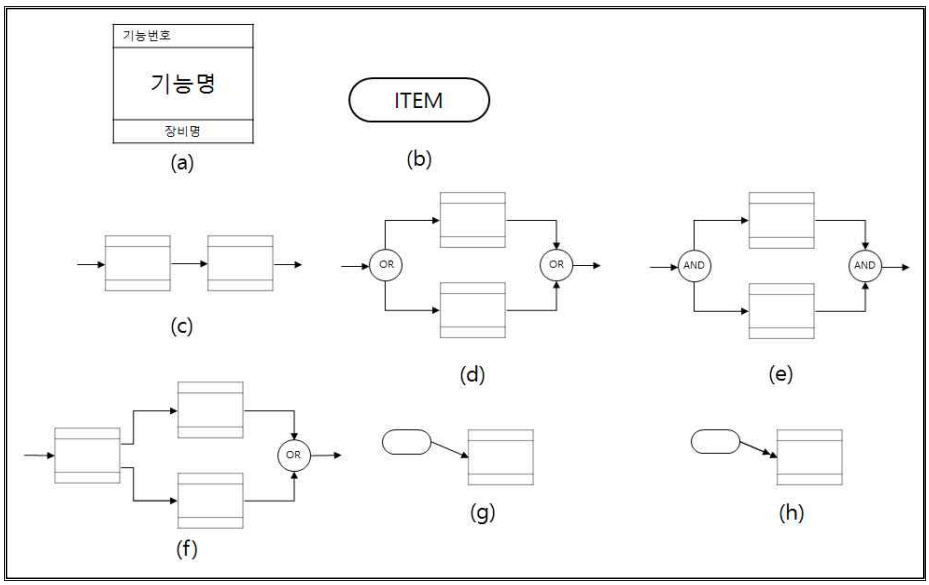

그림 15 EFFBD 제어구문

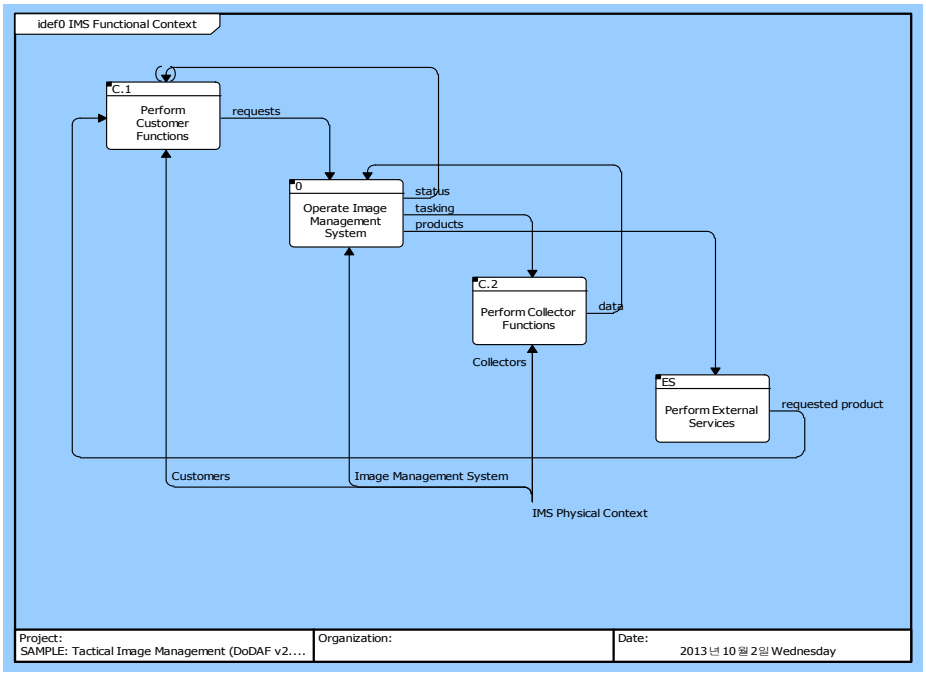

그림 16 IDEF0 예 [24]

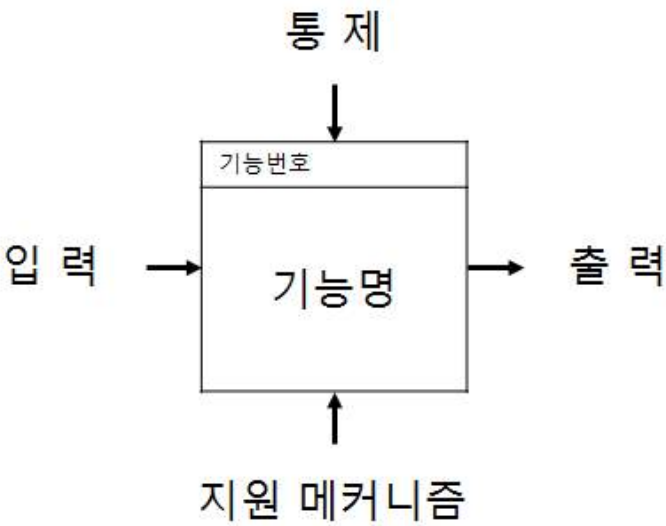

그림 17 IDEF0 구조

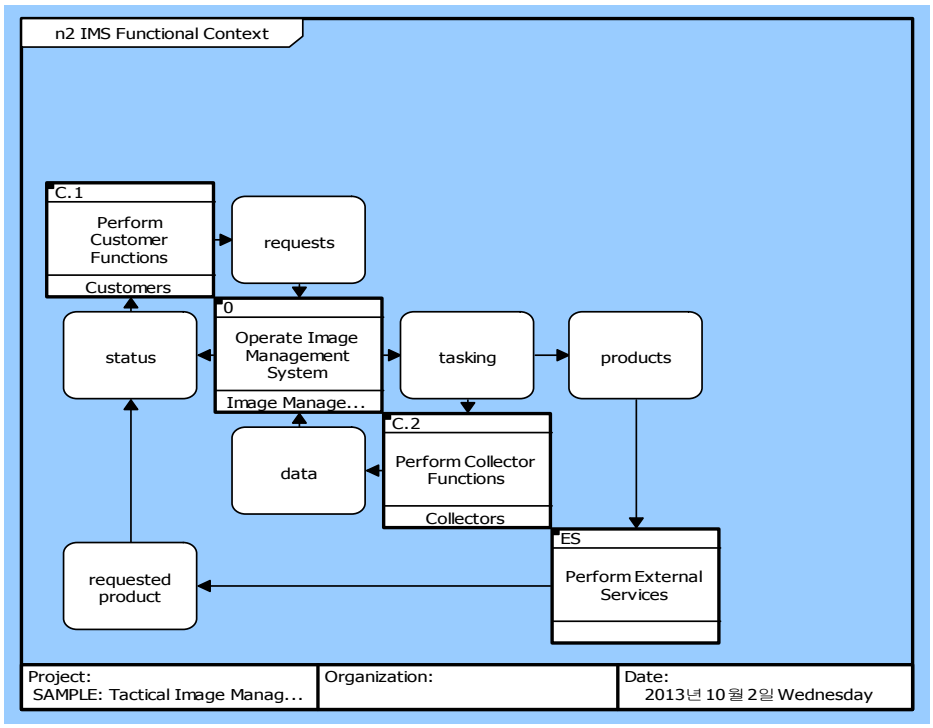

그림 18 N2 Chart 예 [24]

<그림 14, 16, 18>의 다이어그램은 동일한 시스템에 대한 표현으로, 주어진 상황과 여건에 따라 적절히 판단하여 시각적으로 유리한 다이어그램을 선택할 수 있다. MBSE는 이런 다양한 다이어그램을 단일 전산모델로부터 별도의 작업 없이 얻을 수 있다는 장점이 있다.

2.2.2.3 물리구조(Physical Architecture) 분석

이 단계에서는 시스템이 실제로 구동되어야 하는 물리적인 구성품(장비, 설비 등)들을 계층구조로 분해하여(built in) 분석한다. 레벨이 높아질수록 물리구조는 구체화될 것이며 최종적으로 시스템에 탑재되어야 하는 구성품의 목록을 확정한다.

2.2.2.4 거동 할당

통합된 하나의 시스템 모델을 구성하기 위하여 <그림 19>와 같이 요구사항(Requirements), 기능(Function), 물리(Physical) 구조간의 상관관계 설정이 필요하다. 이를 통해 요구사항-기능거동-물리구조간 추적성(Traceability)을 가지며, 요구사항 변경이 시스템에 미치는 영향을 쉽게 추적할 수 있다.

거동 할당을 통하여 구성된 기능거동-물리구조의 부족하고 불필요한 부분 및 오류 유무를 검토하고 필요시 모델 설계에 반영한다.

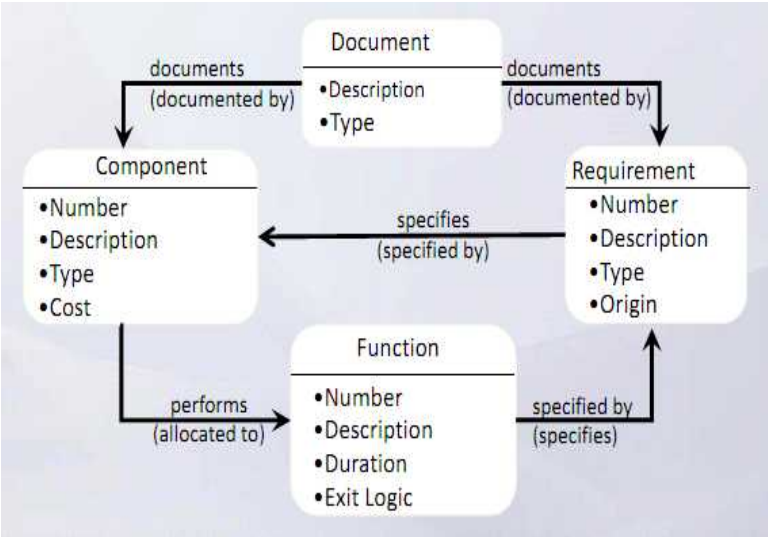

그림 19 거동 할당 [24]

2.3 초기단계 함정 시스템 설계 방법

체계공학을 이용한 초기단계 함정 시스템 설계 방법으로써 체계공학 프로세스인 요구조건 분석, 기능분석 및 할당, 설계조합 프로세스별 설계활동을 <그림 20>과 같이 정의하였다.

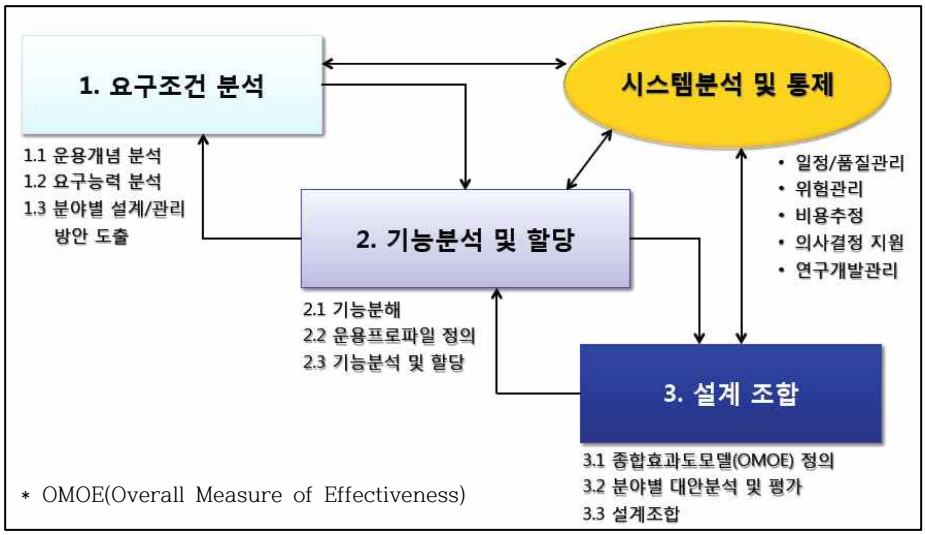

그림 20. 체계공학 프로세스(SE Process)

개념설계 수행 중 첫 번째 활동인 '요구조건 분석' 활동은 <그림 21>과 같이 수행한다. 본 활동은 올바른 시스템(Right system) 개발을 위한 올바른 요구조건 식별과 정의를 위해 중요하게 수행되어야 하는 프로세스이다.

기능분석 및 할당 활동은 <그림 22>와 같이 위 요구조건 분석 활동에서 재정의 및 정제된 요구조건을 토대로 요구되는 기능요구조건 및 성능요구조건을 식별 및 정의하는 요구조건-기능의 분해 및 할당 활동이다.

설계조합 활동은 <그림 23>과 같이 분석된 요구조건과 할당된 기능을 토대로 분야별 설계 대안을 도출하고 조합(Synthesis)하는 활동으로 설계에 영향을 미치는 인자를 식별하여 대안을 도출하고 대

안별 효과도 및 핵심 요구조건인 경하중량간의 비교분석을 통해 최적의 대안을 도출하는 과정이다.

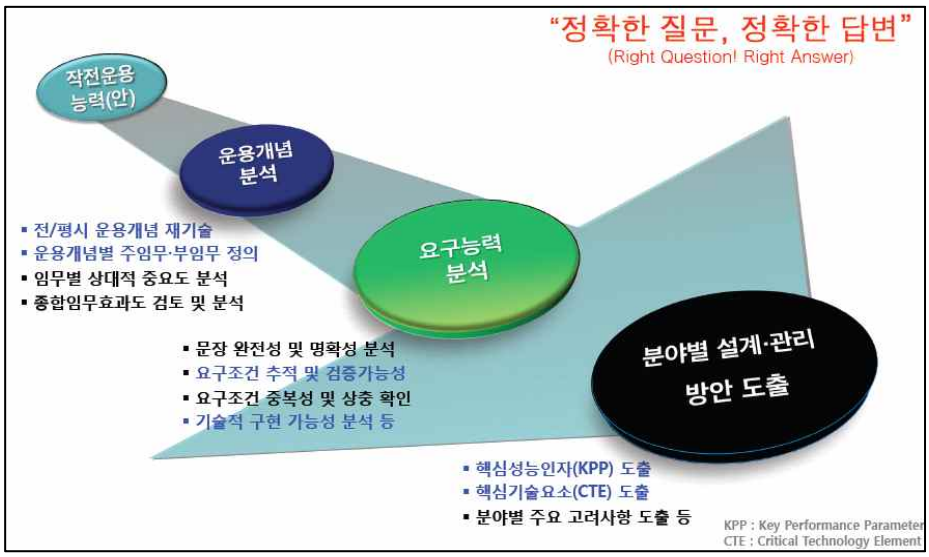

그림 21. 요구조건 분석 활동 [5]

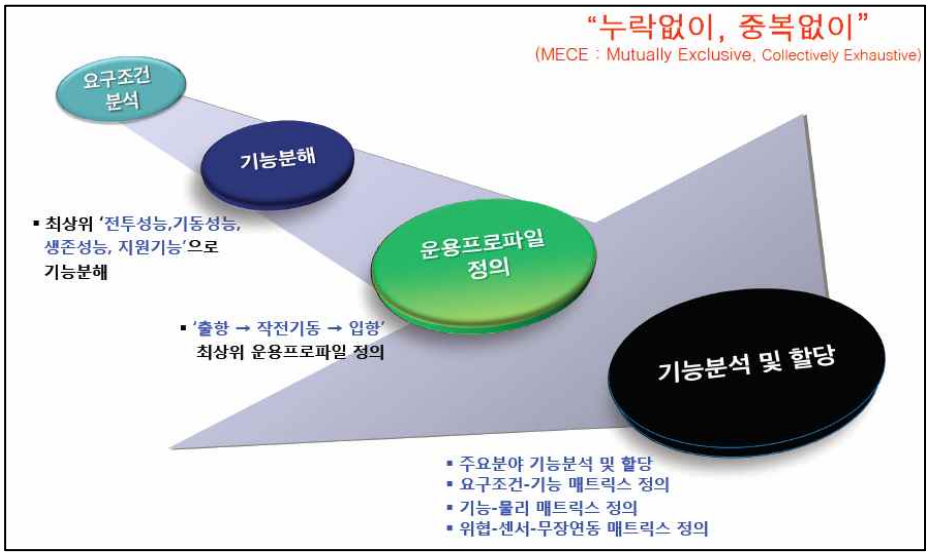

그림 22. 기능분석 및 할당 활동 [5]

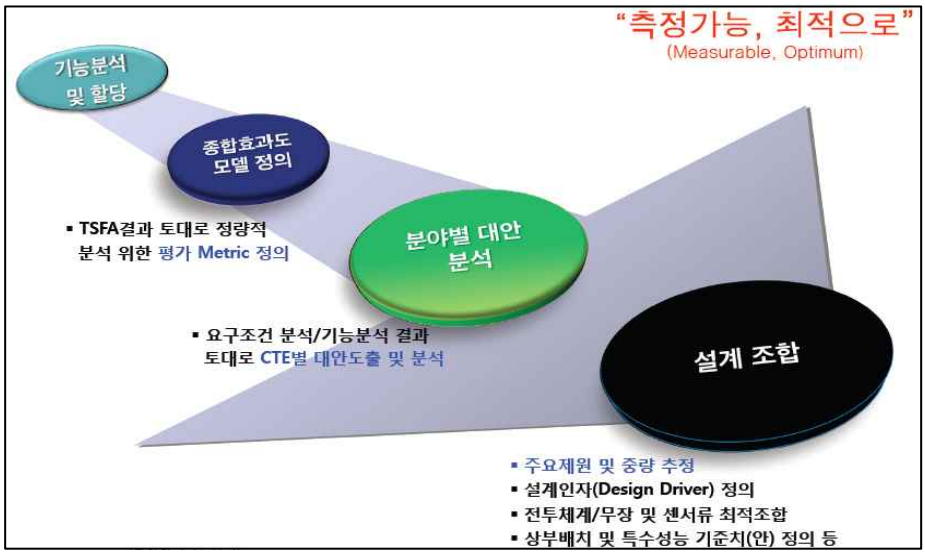

그림 23. 설계조합 활동 [5]

이 장에서 제시한 초기단계 함정설계 방법에 대한 사례는 4장에서 다루도록 한다.

3. 기능분석을 이용한 인터페이스 식별 및 DSM을 이용한 인터페이스 관리를 통한 설계오류를 감소방법 연구

3.1 설계변경의 후속설계변경 미실시로 인한 설계오류 발생 이유

함정 설계중에 주요 설계변경 사항이 연관된 분야로 전파가 되지 않을 경우, 연관된 분야에서 설계오류가 발생할 수 있으며, 이러한 설계오류가 제때 식별되지 않으면 건조단계에서 요구장비 변경, 건조기간 증가 등의 심각한 문제가 발생 할 수 있다. 따라서 설계과정 중 설계변경이 전파되지 않아 후속 설계변경이 미실시되는 중요한 이유를 아래와 같이 검토하였다.

함정 설계는 앞서 말한 바와 같이 다양한 분야의 설계자가 참여하고 있다. 함정시스템이 복잡, 다양해지면서 과거와 달리 IT, 제어 등 다양한 분야의 장비가 탑재, 연동되고 있으며, 이에 따라 조선소(각 분야), 국방과학연구소, 장비업체 등 다양한 기관의 설계자가 동시에 설계에 참여하게 된다. 따라서 설계자간의 물리적 거리 이격에 의한 실시간 상황전파가 제한되고, 각 분야에서 정의하는 용어의 차이로 인한 커뮤니케이션 제한이 발생한다.

또한 함정은 단기간에 상선의 20배 이상에 달하는 과다한 설계변경사항이 발생하며, 이에따라 각 설계자의 설계부담이 가중된다. 이러한 상황에서 별도로 책임있는 인터페이스 관리인력이 배치되지 않기 때문에, 주요 설계변경사항 발생 시 설계자가 주도적으로 연관된 분야의 설계자에게 해당사항을 전파하지 않는 경우, 연관된 분야에서 설계변경사항을 인지하지 못하여 결과적으로 설계오류가 발생

하는 원인이 된다.

따라서 1.2장에서 제시한 바와 같이 각 분야별 인터페이스 식별, 정의 및 관리를 통해 설계변경사항이 관련된 분야로의 시기적절한 전파로, 설계변경에 의한 설계오류를 최소화하기 위한 인터페이스 관리 방법을 연구, 적용을 통해 이를 달성하고자 한다.

3.2 후속 설계변경 미실시로 인한 설계오류 감소를 위한 인터페이스 및 DSM 일반사항

3.2.1 인터페이스 일반

인터페이스는 NASA의 정의에 따르면 다른 장비의 설계 특징에 영향을 주는 장비의 설계 특징을 말하며, Electrical/Functional, Mechanical/Physical, Software, Supplied Services 4가지 종류로 분류할 수 있다.

Electrical/Functional Interface는 전기적, 기능적 신호를 특정 장비에서 다른 장비로 이동시키는 인터페이스로 아래와 같은 형식으로 정의한다.(<그림24>)

- Function name and symbol
- Impedance characteristics
- Shielding and grounding
- Signal characteristics
- Cable characteristics
- Data definition
- Circuit logic characteristics
- Data transmission losses, Etc.

Pulse characteristics (see fig. A.2)	Guidance telemetry					
	Data 1	Data 2	Bit synchronization	Frame synchronization	Data 1 word synchronization	Data 2 word synchronization
Pulse duration	255 + 50 ns					
Minimum amplitude	9 ± 2 V (see V027)					
Maximum amplitude	15 V					
Rise time	75 ns maximum					
Fall time	125 ns maximum					
Undershoot	2.5 V maximum					
Reference level offset	0 to –4.5 V relative to no-transmission level					
Noise	1.4 V maximum peak to peak					
Receiver susceptibility	2.0 V minimum					
Test parameters:						
Test load	75 Ω±5% resistive					
Receiver susceptibility	2.0 V minimum					

그림 24 Electrical Interface 예 [18]

Mechanical/Physical Interface는 기계적 특징, 크기, 공차 등 다른 subsystem 설계에 영향을 주는 인터페이스로 아래와 같은 형식으로 정의된다.

- Optical characteristics - Parallelism and straightness - Size, shape, mass, mass distribution - Service ports - Surface finish - Hard points for handling - Thermal conductivity and expansion characteristics, Etc.

Software Interface는 정보 교환의 결과를 나타내며, 구성요소간 직접적인 electrical/mechanical 인터페이스가 없어도 존재 가능한 특징이 있다. Software Interface는 아래와 같은 형식으로 정의된다.(<그림25>)

- Interface communication protocol - Digital signal characteristics - Data transmission format, coding, timing, and updating requirements - Data and data element definition - Message structure and flow - Operational sequence of events - Error detection and recovery procedures, Etc.

Signal definition	Centaur sequence control unit switch number	Initiating event + time	Persistence	Function
Satellite vehicle (SV) pyro unshort (primary)	45	Main engine cutoff (MECO) 2 + 3±0.5 sec	3±0.5 sec	Unshorts SV pyro capacitor banks
SV latch valve arm (primary)	33	MECO2 + 10±0.5 sec	3±0.5 sec	Arms safety inhibit relay for SV main engines
SV pyro unshort (secondary)	89	MECO2 + 15±0.5 sec	3±0.5 sec	Provides redundant unshort of SV pyro capacitor banks
SV latch valve arm (secondary)	88	MECO2 + 17±0.5 sec	3±0.5 sec	Provides redundant arm of inhibit relay for SV main engines
Radiofrequency monopropellant driver backup enable	34	Titan IV/Centaur separation + 24±0.5 sec	3±0.5 sec	Services backup (redundant to SV ground support equipment command) enable of safety inhibit SV functions (radiofrequency sources and reaction control system thruster drivers)

그림 25 Software Interface 예 [18]

Supplied services Interface는 각 장비가 기능을 하기 위해 필요한 지원 요구사항으로 아래와 같은 카테고리로 분류된다.

❍ Electrical power interface
- Phase, Frequency, Voltage, Interrupt time, etc.

❍ Communication interface
- Type of communication required between equipment
- Location of communication stations, etc.

❍ Fluid interface
- Type of fluid required(gaseous, liquid)
- Fluid properties(pressure, temperature, flow rate, etc.)

❍ Environmental characteristic interface
- Ambient temperature
- Atmospheric pressure
- Humidity, etc.

인터페이스는 아래의 방법을 통해 정의되며, 각 시스템 레벨(system, subsystem, element 등)에 따라 분해해나가며 internal interface를 구체화해 나간다.(<그림26>)

- Functional description에 대한 Interface 식별 - 연관된 파라메터를 포함한 연결(connection)의 정의 - 필요할 경우 세부레벨로 확장 - Design reivew를 통해 확정

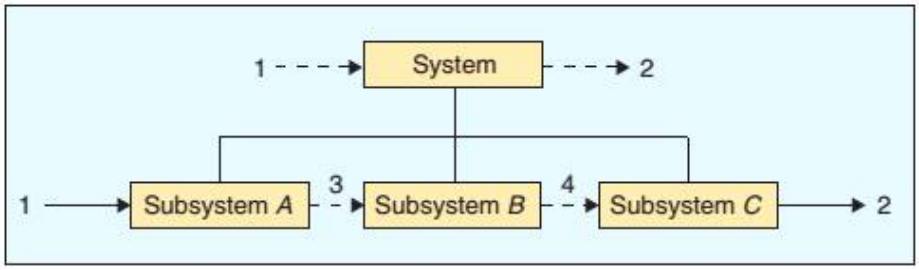

그림 26 Interface 분해 [18]

3.2.2 DSM 일반

DSM(Design Structure Matrix)은 시스템 분석을 위해 널리 쓰이는 모델링 기법으로써, 시스템의 분해(Decomposition)와 통합(Integration)에 적합한 도구이다. DSM은 구성요소 간의 관계를 나타내는 정사각행렬로, 시스템이나 프로젝트를 시각적이며 직관적으로 표현가능하다. DSM은 시스템 구성요소의 복잡한 영향력을 직관적으로 인식할 수 있는 행렬로 나타내며, 이 행렬은 다양한 분석 기법에서 활용이 가능하다.

DSM에서 각 구성요소의 의미가 각 행•열에 표시되며, 대각 성분을 제외한 부분에 구성요소 간 의존관계(Dependency)가 표시된다. DSM의 열은 영향을 주는 요소(Input source), 행은 영향을 받는 요소(Output source)를 의미한다.[6]

예를 들어, <그림 27>과 같은 구성요소 간 관계를 갖는 시스템은, 이것을 <그림 28>과 같이 DSM으로 표현 가능하다. 여기서 A행 B열은 구성요소 B 가 A 에 미치는 영향을 의미하며, B행 A열

은 구성요소 A가 B에 미치는 영향을 의미한다.

DSM은 표현 수준에 따라 Binary DSM(<그림 28>)과 Numerical DSM(<그림 29>)으로 분류한다. Binary DSM은 구성요소의 영향력 유무만을 표현하나, Numerical DSM은 영향력의 크기까지 함께 표현함으로써 Binary DSM보다 구체적 정보를 표현할 수 있다.

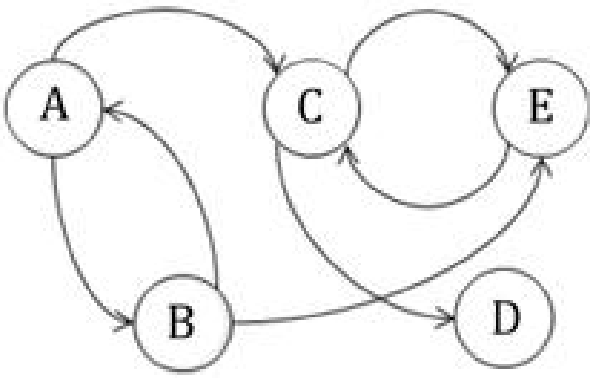

그림 27 시스템 구성요소 간의 관계 [25]

	A	B	C	D	E
A		1			
B	1				
C	1				1
D			1		
E		1	1		

그림 28 Binary DSM의 예 [25]

	A	B	C	D	E
A		0.1			
B	0.3				
C	0.1				0.2
D			0.3		
E		0.2	0.1		

그림 29 Numerical DSM의 예 [25]

시스템 구성요소 간의 관계는 병렬(Parallel), 순차(Sequential), 순환(Coupled)으로 분류할 수 있다.(<그림 30>) 병렬 관계는 구성

요소 간 아무런 영향을 주고 받지 않는 관계이며, 순차 관계는 일방적인 영향의 흐름을 의미한다. 순환 관계는 서로 영향을 주고 받는 관계로, 구성요소 간의 반복적인 변경을 발생시키는 관계이다. 따라서 순환 관계는 시스템을 분석할 때 가장 중요한 관계로 간주된다. 앞서 보인 예제(<그림 27>)에서 구성요소 A와 D는 병렬 관계, A와 C는 순차 관계, A와 B는 순환 관계를 갖고 있다.[6]

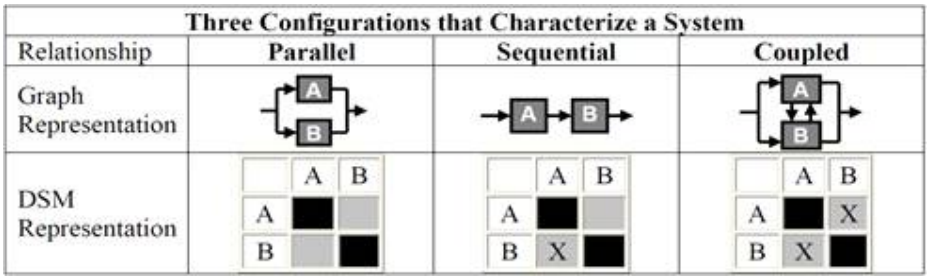

그림 30 구성요소간 관계성의 종류[25]

DSM을 이용하여 시스템 구성요소 간의 정보교환 형태 및 인터페이스 종류를 간단하게 표현 가능하며 특정 구성요소의 설계에 필요한 정보가 어느 구성요소의 설계결과로부터 제공되는지가 표현 가능하다. 이를 이용하여 각 구성요소의 설계순서대로 DSM을 구성하면, DSM 대각 상단의 Mark는 설계결과의 Feedback을 의미하며, 이를 이용하여 반복적으로 수행되는 설계를 미리 식별하여 설계 순서를 최적화 하는데 적용할 수 있다.

3.3 기능분석을 이용한 인터페이스 식별 및 DSM을 이용한 인터페이스 관리방법

3.2.1장에서와 같이 인터페이스의 정의를 이용하여, 이용하여 '다른 장비의 설계특징에 영향을 요소' 인 인터페이스를 식별, 관리함으로써, 설계변경의 전달을 돕고자 하였다.

인터페이스를 식별/관리하기 위한 방법을 검토하면서 가장 중점으로 둔 사항은 추가적인 업무가 최소화 되어야 한다는 점과, 인터페이스를 누락 없이 식별 가능하여야 한다는 점이었다. 이를 위해서 2장에서 사용하였던 MBSE(Model-Based Systems Engineering) Tool을 이용한 기능분석을 통하여 인터페이스를 식별하고, 3.2.2에서 제시한 DSM을 이용한 관리를 제시하였다.

첫 번째 단계는 기능분석 시 기능간에 EFFBD(Enhanced Functional Flow Block Diagram) 작성 간에 인터페이스를 NASA에서 정의한 4가지 분류에 따라 정의하고, 이를 물리구조로 할당함으로써, 물리적인 구성품간의 인터페이스를 정의한다.(<그림 31>)

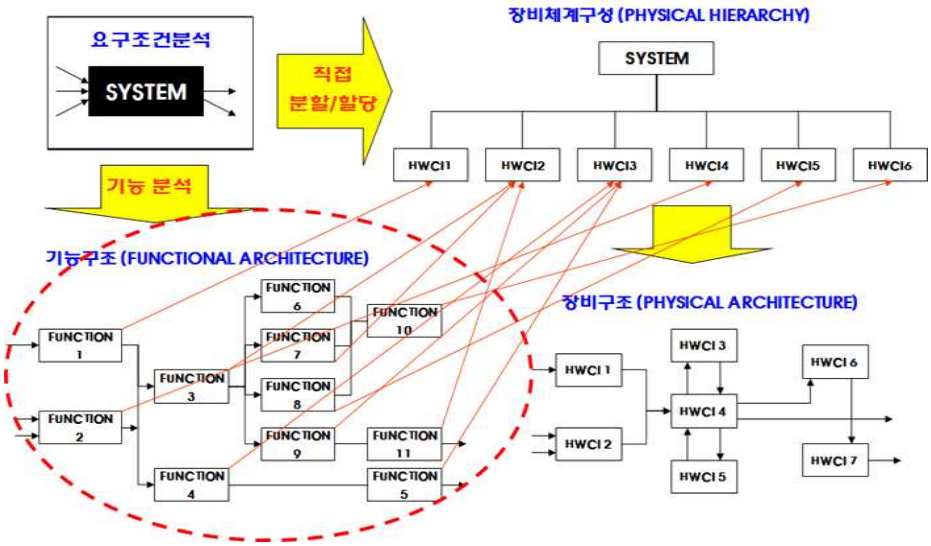

그림 31 기능분석 및 할당 [5]

정의된 인터페이스를 이용하여 인터페이스 Matrix를 작성하고, (<표 2>) 이를 토대로 인터페이스 DSM을 작성, 관리한다(<그림 32>). 이렇게 작성된 DSM을 이용하여, 각 구성품간의 인터페이스를 시각적, 직관적으로 식별할 수 있으며, 이를 이용하여 설계변경 전파경로를 식별할 수 있다.

Functions	Interface Items	Interfacing Elements
Fn.1.1.2 대함 표적 식별	→ 대함 표적정보	Fn.1.1.4 대함 무장통제 C.1.4 전투체계 Fn.1.1.6 대함 교전평가 C.1.4 전투체계
	← 미식별 대함 표적정보	Fn.1.1.1 대함 탐지 및 추적 C.1.1.1 다기능레이더

표 2 Interface Matrix 예

		C.1.1.1	C.1.1.2	C.1.1.3	C.1.1.4	C.1.2.1	C.1.2.2	C.1.2.3	C.1.2.4	C.1.3.1	C.2.1.1	C.2.1.2	C.2.1.3	C.2.1.4	C.3.1.1	C.3.1.2	C.3.2.1	C.3.2.2	C.3.2.3	C.3.3.1	C.3.3.2	C.3.3.3	C.3.4.1	C.3.4.2	C.4.1.1
		No.1 G/T	No.1 P/M	No.1 R/G	Port Shaft & CPP	No.2 G/T	No.2 P/M	No.2 R/G	STBD Shaft & CPP	Rudder	Diesel Generator	Main S/W	Prop. S/B	Freq. Converter	공기압축기	압축공기 탱크	F.O 이송펌프	F.O 정유기	F.O 공급펌프	L.O 이송펌프	L.O 정유기	L.O 공급펌프	Sea Chest	해수펌프	ECS Console
C.1.1.1	No.1 G/T															4			4						1
C.1.1.2	No.1 P/M												1											4	1
C.1.1.3	No.1 R/G																							4	1
C.1.1.4	Port Shaft & CPP		2	2																					1
C.1.2.1	No.2 G/T															4			4						1
C.1.2.2	No.2 P/M												1											4	1
C.1.2.3	No.2 R/G																							4	1
C.1.2.4	STBD Shaft & CPP						2	2																	1
C.1.3.1	Rudder											1													1
C.2.1.1	Diesel Generator																		4			4			1
C.2.1.2	Main S/W										1														1
C.2.1.3	Prop. S/B													1											1
C.2.1.4	Freq. Converter											1													1
C.3.1.1	공기압축기																								1
C.3.1.2	압축공기 탱크														4										1
C.3.2.1	F.O 이송펌프																								1
C.3.2.2	F.O 정유기																4								1
C.3.2.3	F.O 공급펌프																	4							1
C.3.3.1	L.O 이송펌프																								1
C.3.3.2	L.O 정유기																			4					1
C.3.3.3	L.O 공급펌프																				4				1
C.3.4.1	Sea Chest																								
C.3.4.2	해수펌프																						4		1
C.4.1.1	ECS Console	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	

그림 32 DSM 예

4. 초기설계 단계의 체계공학 설계 방법의 적용

4.1 체계공학 방법을 이용한 함정 개념설계

2장에서 제시한 초기단계 함정 시스템 설계방법을 적용하여 함정의 대표적 종류인 전투함의 개념설계를 진행하였다. 이때 대표적 MBSE(Model-Based Systems Engineering) 도구인 Core 프로그램을 이용하여 설계를 진행하였다. 다이어그램이 구체적인 내용을 포함할 경우 '기능 1'과 같은 일반용어로 사용하여 표시하였다.

최초 요구사항 문서인 작전운용성능을 토대로 최초 요구사항을 입력하였으며, 문장완전성 및 명확성 분석 등의 활동을 통하여 요구사항 분석을 실시하였으며, 최종 50여개의 요구사항을 도출하였다.(<그림 33>) 다이어그램의 제일 위가 요구사항 문서이며, 그 아래는 요구사항 문서에 기술된 최초 요구사항이다. 이 최초 요구사항 분석을 통하여 도출된 요구사항이 그 아래에 있으며, 이들이 최종적으로 관리하여야 하는 요구사항이다.

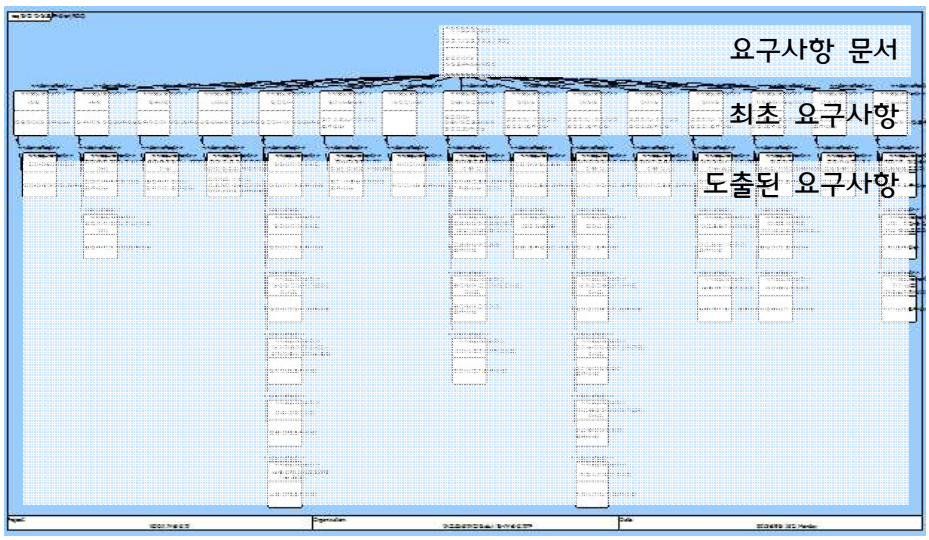

그림 33 요구사항 분석 결과 [1]

요구사항 분석 중 <그림 34>와 같이 최초 요구사항문서에서 탑재가 요구되었던 장비에 대한 요구사항 검토를 통해 불필요 장비로 판단하여 장비 미탑재로 요구사항을 변경 도출함으로써, 설계 중 설계변경 발생사항을 사전에 식별할 수 있었다.

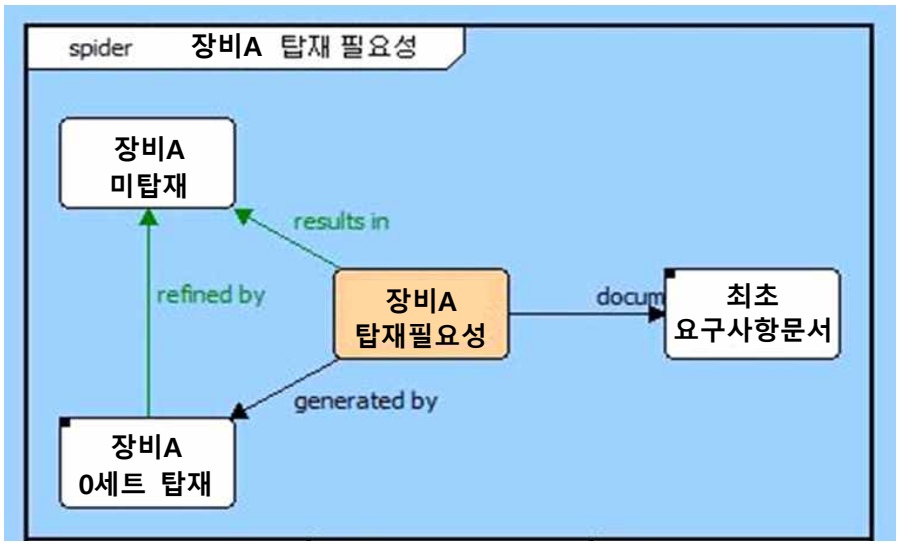

그림 34 요구사항 분석을 통한 요구조건 검증

요구사항 분석 결과를 토대로, 요구사항을 만족할 수 있는 기능분석을 실시하였다. 최상위 기능거동으로 전투성능, 기동성능, 생존성능, 지원성능 4가지로 구분하여 기능분해하였으며(<그림 35>), 대상함정의 운용 프로파일을 <그림 36>과 같이 정의하였다.

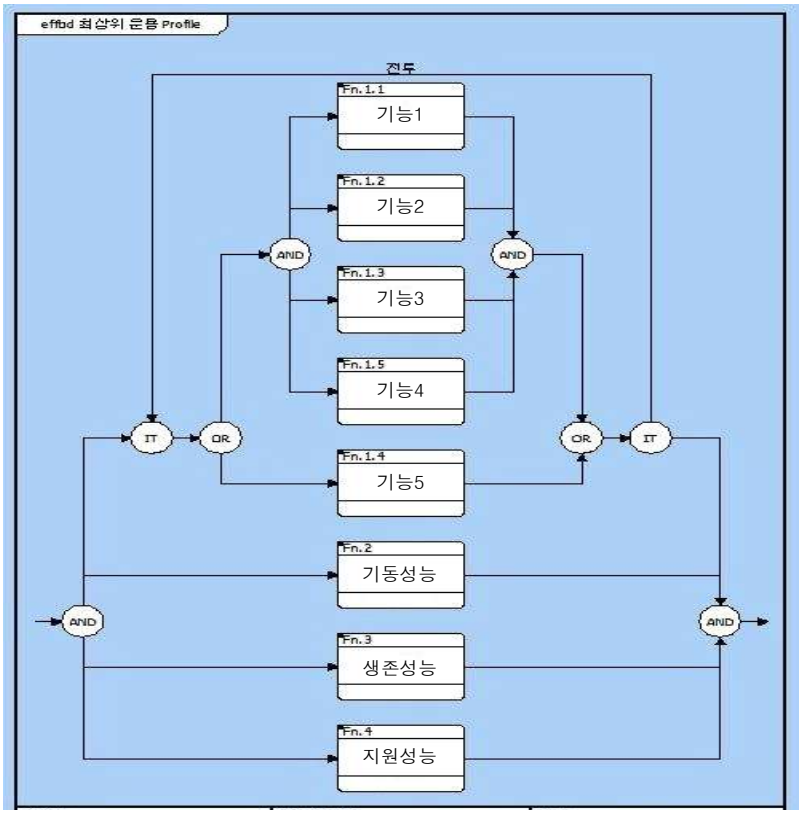

그림 35 최상위 기능거동 [1]

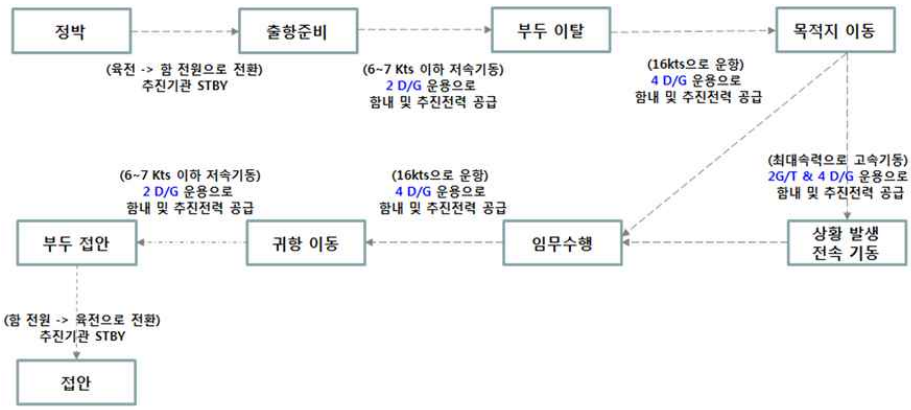

그림 36 함정 운용프로파일 정의

각각 분해된 기능을 <그림 37>과 같이 현재 설계단계를 고려하여 3수준까지 기능을 분석하였으며, 이를 통해 최종 70여개 기능으로 분해하였다. 기능분석을 통하여 대상함정의 요구조건을 만족할 수 있는 기능을 구성하였으며, 각 단계에서 다양한 기관의 설계관계자들과의 회의 시 다이어그램을 이용함으로써 공통된 인식을 갖고 설계를 진행 할 수 있었다.

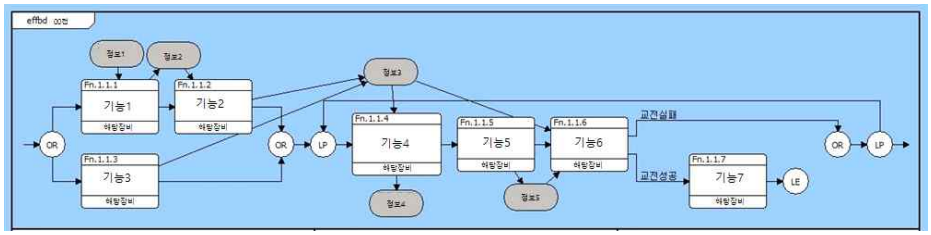

그림 37 기능분석 결과 [1]

이러한 기능분석 결과를 검토하기 위하여 <그림 38>과 같이 요구조건-기능 Matrix를 작성하였으며, 이를 이용하여 설계진행간 기능분석결과의 요구조건 만족 여부를 지속적으로 확인하며 설계를 수행하였다.

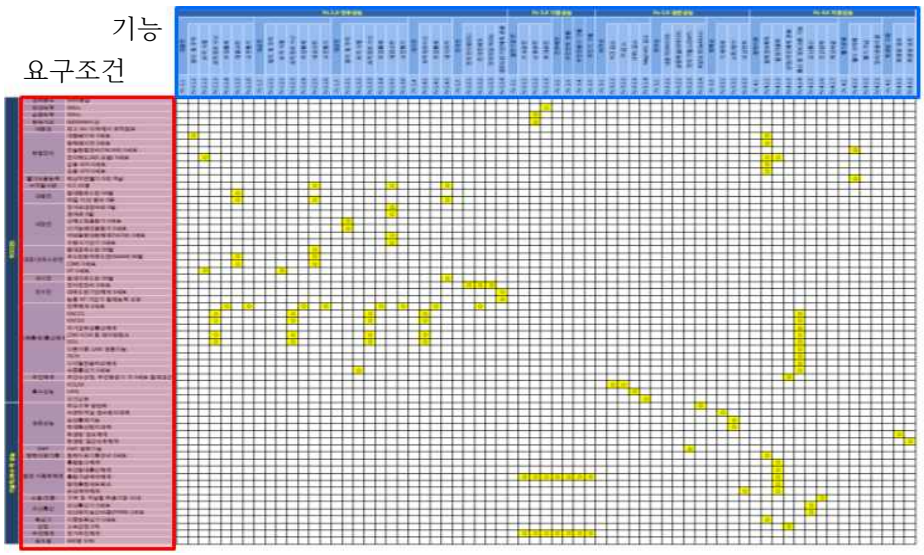

그림 38 요구조건-기능 Matrix

요구조건 분석결과를 토대로 <그림 39>와 같이 실제 구성할 물리구조를 분석하였다. 기능분석과 마찬가지로 현재 설계단계를 고려하여 3수준까지 물리구조를 분석하였으며, 이를 통해 100여개의 물리구조로 분해하였다.

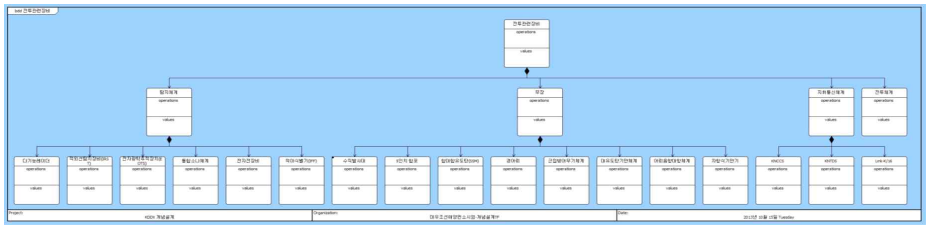

그림 39 물리구조 분석 결과 [1]

물리구조 분석결과를 검토하기 위하여 <그림 40>과 같이 기능-물리 Matrix를 작성하였으며, 이를 이용하여 설계진행간 물리구조가 가져야 하는 기능을 확인함으로써, 해당 구성품의 설계요구조건으로 반영할 수 있었다. 또한 설계진행간 물리구조의 기능 만족 여부를 지속적으로 확인함으로써 누락되거나 과다하게 적용되는 물리구조가 없도록 지속적으로 확인하며 설계를 수행하였다.

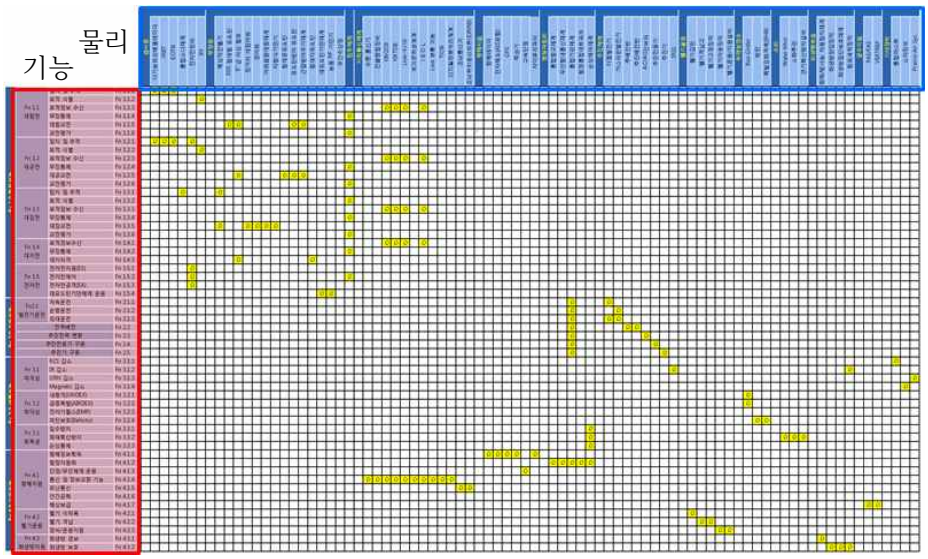

그림 40 그림 40 기능-물리 Matrix

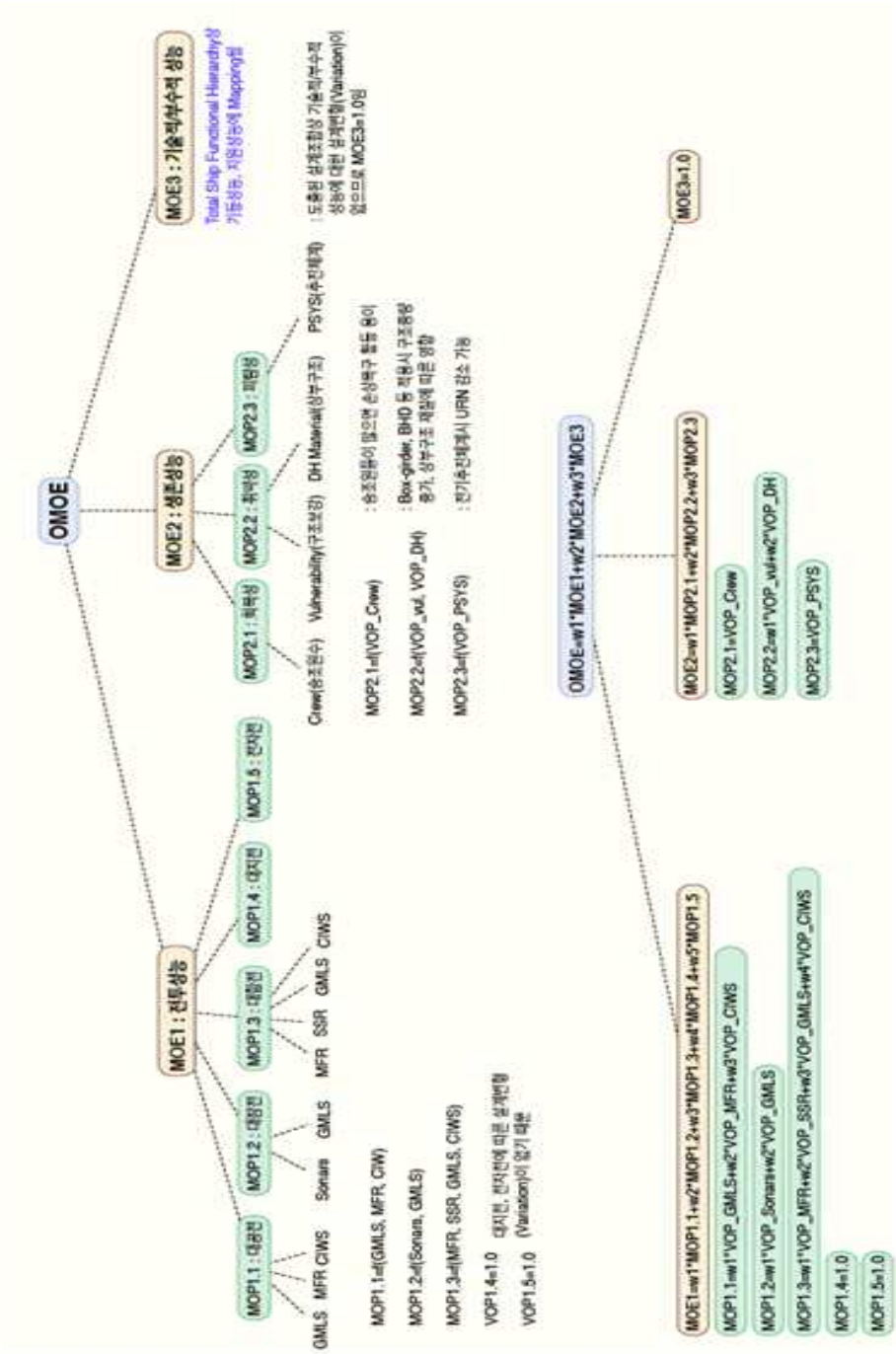

그림 41 함정 종합효과도체계(OMOE) [5]

설계조합 활동을 위하여 임무별 상대적 중요도 분석을 위하여 <그림 41>과 같이 종합효과도체계(OMOE, Overall Measure Of Effectiveness)를 정의하였으며, 설계참여자의 의견수렴을 통해 임무별 상대적 중요도 분석을 실시하였다.

함정의 임무 및 운용개념 그리고 개인이 갖고있는 지식과 경험에 따라 임무, 기능의 중요도가 다르기 때문에, 다수 관계자들의 의견과 그들의 비중을 함께 반영함으로써 보다 객관적인 평가를 할 수 있는 AHP (Analytic Hierarchy Process) 방법[10]을 이용하여 설계자부터 운용자까지의 의견을 반영하였다.

설계 관계자의 의견수렴 결과를 Expert Choice라는 프로그램을 이용하여 상대적 중요도를 분석, 가중치를 산정하였다.(<그림 42>)

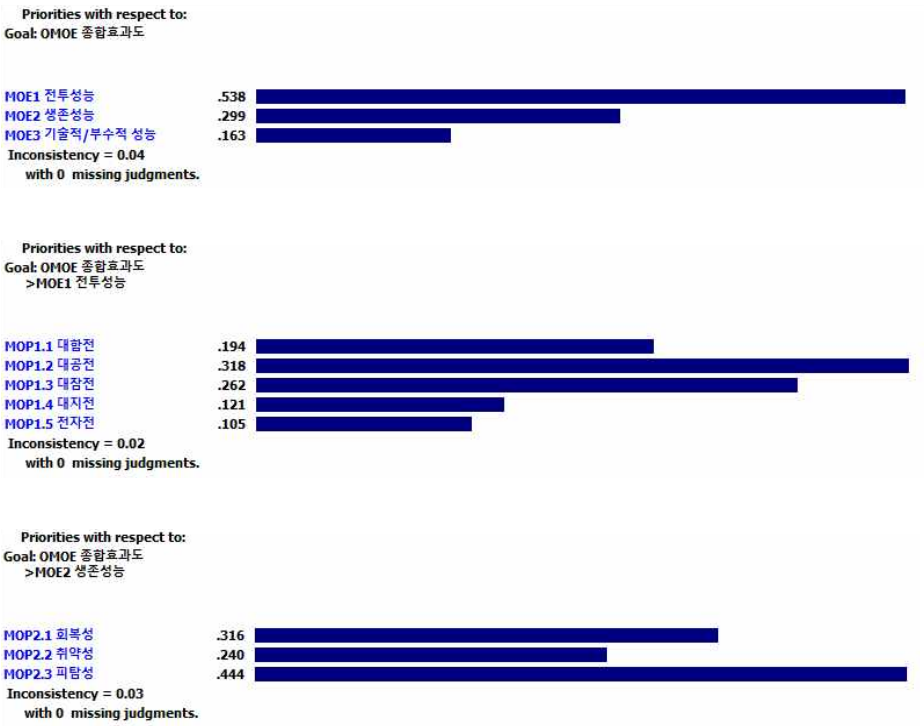

그림 42 가중치 산정 결과

다음으로 분석된 요구조건과 할당된 기능을 토대로 설계대안을 도출하고 설계조합을 실시하였다. 이를 위하여 설계에 영향을 미치는 인자를 식별하여 각 인자별 대안을 도출하였다. 설계인자를 도출하기 위한 고려사항으로 경하중량에 영향을 주는 정도와 주요 무기체계/장비 중 국내연구개발 또는 신규 적용되는 체계로 개념설계 중 구체화 되어야 항목 중에서 선정하였다.

이를 통해 승조원 수, 생존성 설계 수준, 추진체계 구성, 상부구조물 재질 등 2수준 1개, 3수준 5개의 총 6개 설계인자를 선정하였다. 각 설계인자의 설계대안이 함정 설계에 미치는 영향을 경하중량, 함 길이, OMOE 미치는 영향을 검토하여 6개 설계변수에 따른 486개 설계대안을 도출하였다. 모든 설계대안에 대한 OMOE와 경하중량에 따른 분포는 <그림 43>과 같다.

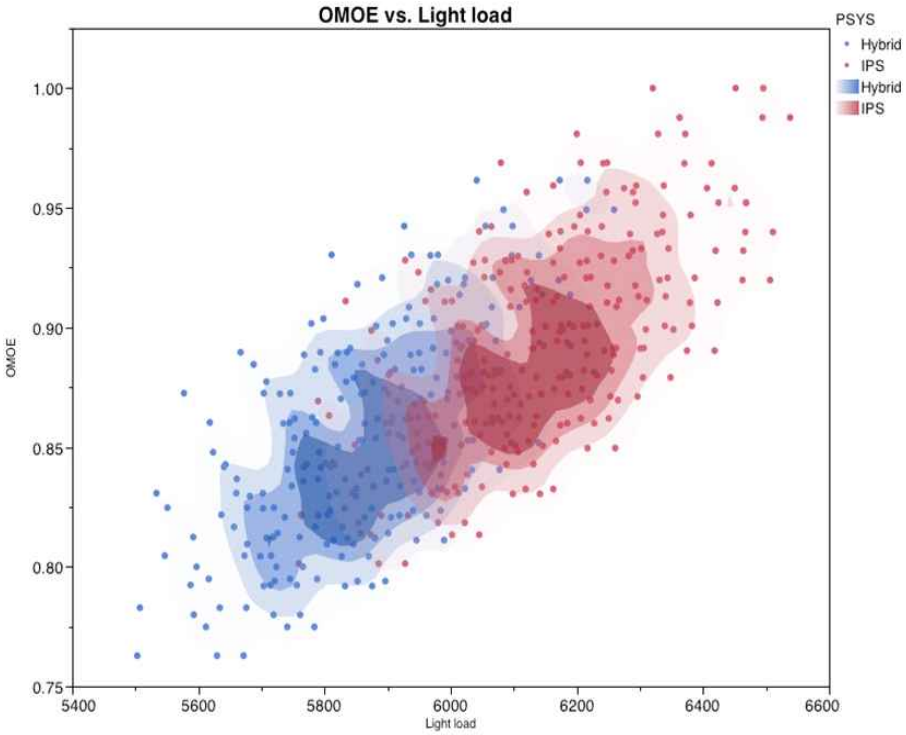

그림 43 모든 설계대안 도출(486개) [5]

모든 설계대안 중 일반적인 전투함의 방형비척계수(Cb) 0.55를 넘어서는 대안은 현실적으로 충족하기 어려운 대안으로 제외하였고 (<그림 44>) 유사실적함 현황, 해군관습, 상부지침 등을 고려하여 설계대안을 줄였으며 설계검토회의를 통해 최종 대안을 선정하였다. 이러한 방법을 통하여 설계대안을 제시함으로써 설계참여자들이 동의하는 대안을 식별할 수 있었다.

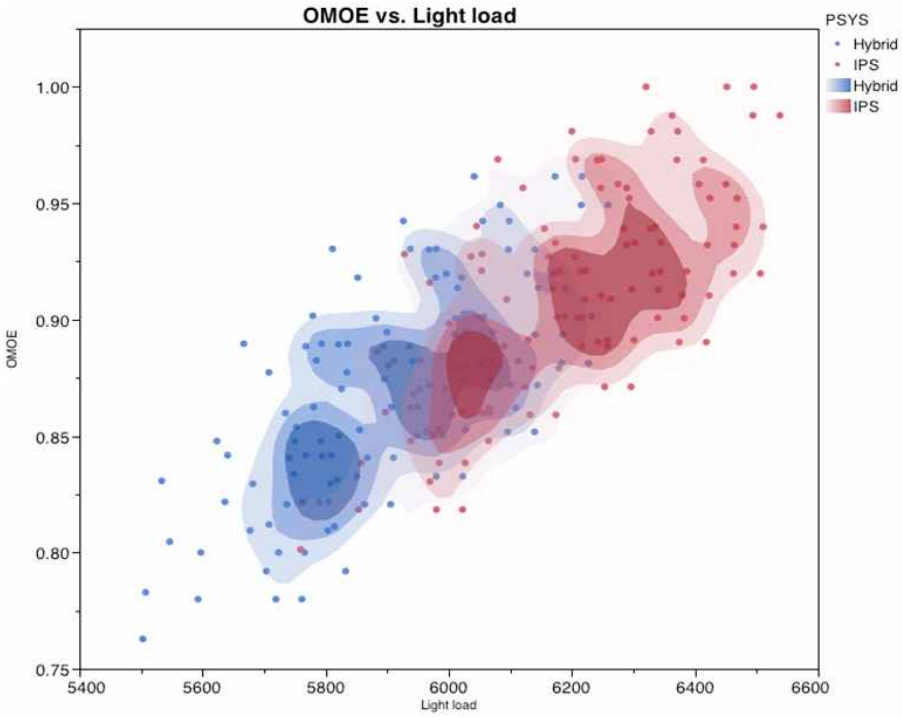

그림 44 조선공학적 제한사항 만족하는 설계대안(233개) [5]

4.2 적용효과

4.1장과 같이 초기설계 단계의 체계공학 설계방법을 적용하였으며 아래와 같이 향후 발생가능한 설계변경을 최소화하는 효과를 얻을 수 있었다.

먼저 설계 초기단계에서 요구조건 분석을 통해 운용자와 설계자 간의 공통된 요구사항을 식별 및 정의하였고, 이에따라 계약조건 불일치에 의한 설계변경 등 사용자에 의한 설계변경 가능성을 감소할 수 있었다.

다음으로 기능분석 및 할당을 통해 다이어그램 기반을 이용한 기능정의 및 분야별 의사소통시 활용하였으며, 설계 초기부터 각 장비의 요구사항 검토 및 요구조건 충족여부를 지속적으로 검증 및 확인함으로써, 연관된 장비 또는 설비의 사양 변경 가능성을 감소할 수 있었다.

마지막으로 설계조합을 통해 관련기관에서 수용 가능한 최적대안을 도출할 수 있었으며, 이를통해 운용자에 의해 요구되는 개선요구 및 연관된 장비 및 설비의 사양 변경 가능성을 감소할 수 있었다.

5. 기능분석을 이용한 인터페이스 식별/관리방법 적용

5.1 추진체계에 대한 인터페이스 식별/관리방법 적용

3장에서 제시한 기능분석을 이용한 인터페이스 식별/관리방법을 함정의 추진시스템의 인터페이스 식별에 적용함으로써 본 논문에서 제안하는 인터페이스 식별/관리방법 수행가능성을 검증하였다.

먼저 대상의 함정 추진시스템은 전기추진체계를 적용하는 최근 실적함 사례를 통하여 CODLAG(COmbined Diesel eLectric And Gas turbine)으로 가정하였다. 본 논문에서 적용한 2축 CODLAG 시스템의 구성은 <그림 45>와 같으며, 저속 및 순항 속력구간에서 추진전동기를 이용하여 추진하며 고속 기동시 가스터빈 및 추진전동기를 이용하여 추진한다. 또한 디젤 발전기를 이용하여 함내전원 및 추진전원 공급이 가능하여 보다 효율적인 전력 공급이 가능한 장점이 있다.

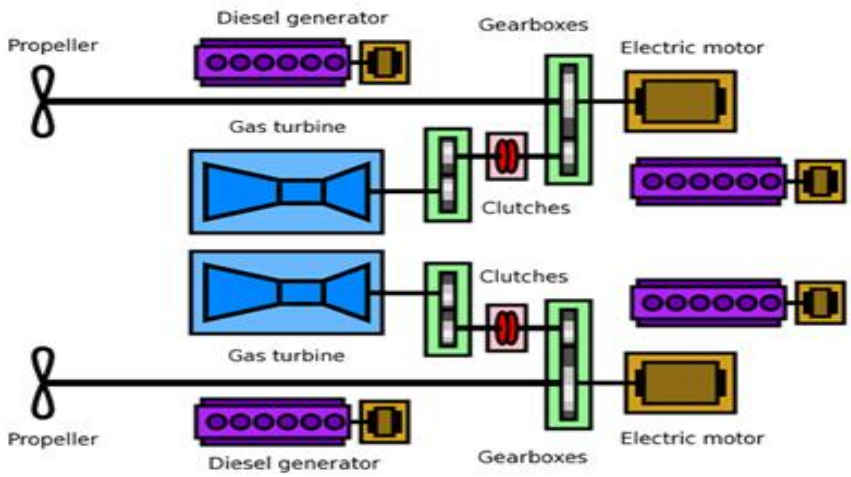

그림 45 일반적인 CODLAG 시스템의 구성 [14]

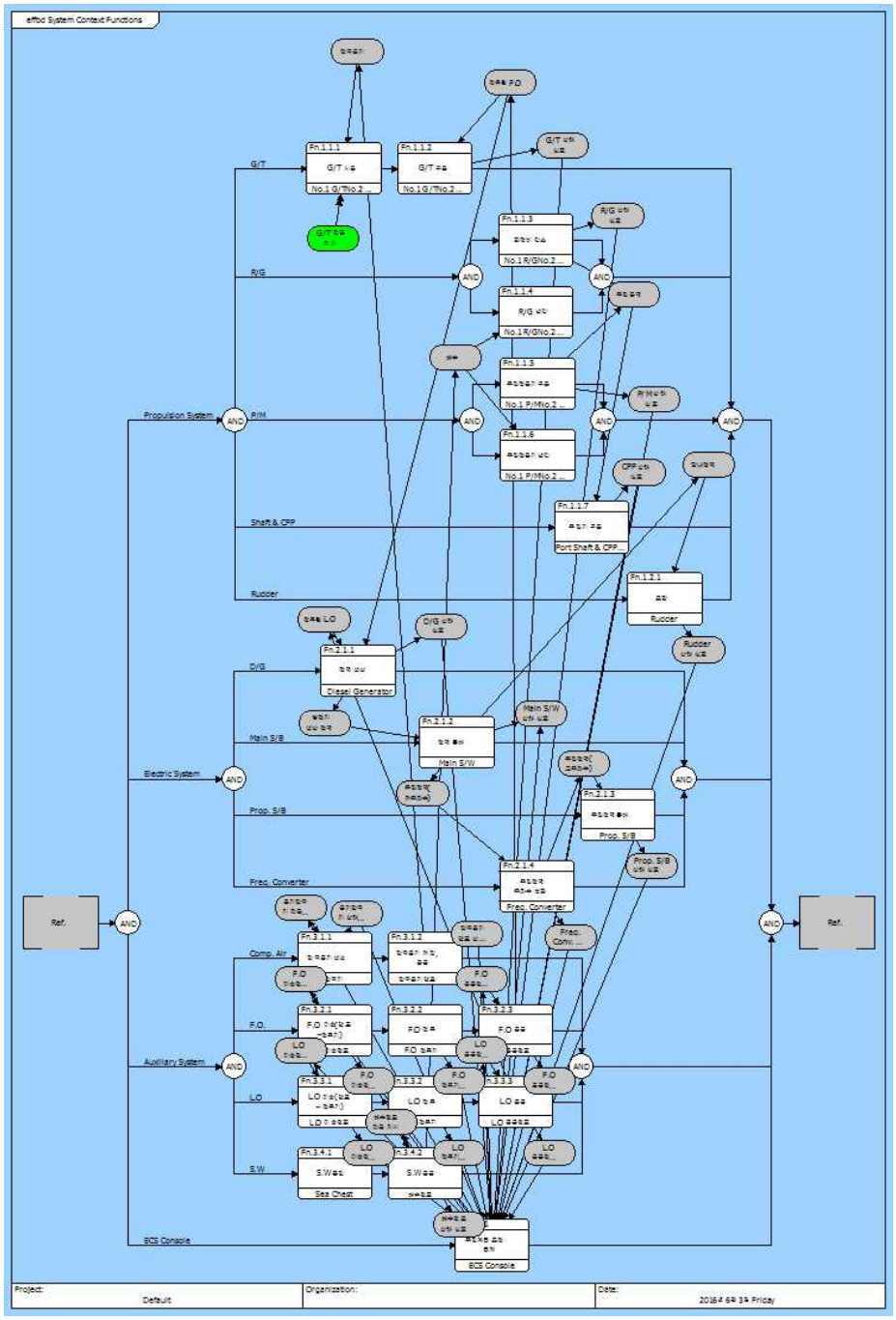

그림 46 System Context
(EFFBD, Enhanced Functional Flow Block Diagram)

분석된 기능들을 전체시스템 관점에서 <그림 46>과 같이 System Context를 작성하였으며, 시스템을 크게 추진, 전기, 보기계통 및 통제계통 4가지로 분해하였으며,(<그림 47>) 각 계통의 기능을 주요 장비별로 분해/분석하였다.

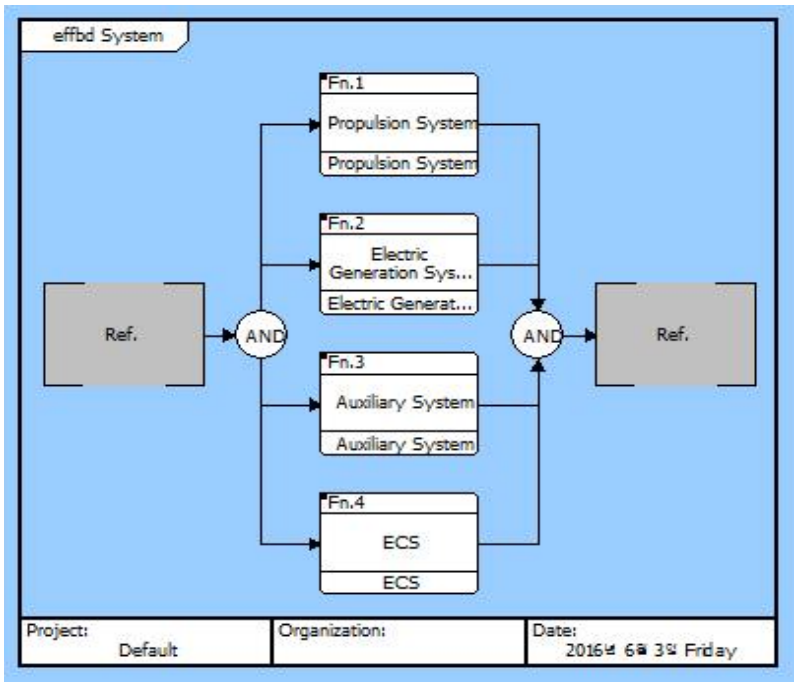

그림 47 추진체계 기능구성

추진계통은 구성되는 주요 장비인 가스터빈(G/T), 감속기어(R/G), 추진전동기(P/M), 축계(Shaft & CPP), 타(Rudder)로 구성하여, 각각의 장비가 수행해야하는 기능을 시간흐름 및 신호 등을 고려하여 구성하였다. 이를 통해 <그림 48>과 같이 8개의 기능으로 분해하였으며, 11개의 기능간 인터페이스를 식별하였다.

전기계통은 마찬가지 방법으로 <그림 49>와 같이 4개 기능으로 분해하였으며, 10개 기능간 인터페이스를 식별하였고, 보기계통은 <그림 50>과 같이 10개 기능으로 분해하였으며, 19개 기능간 인터페이스를 식별하였다.

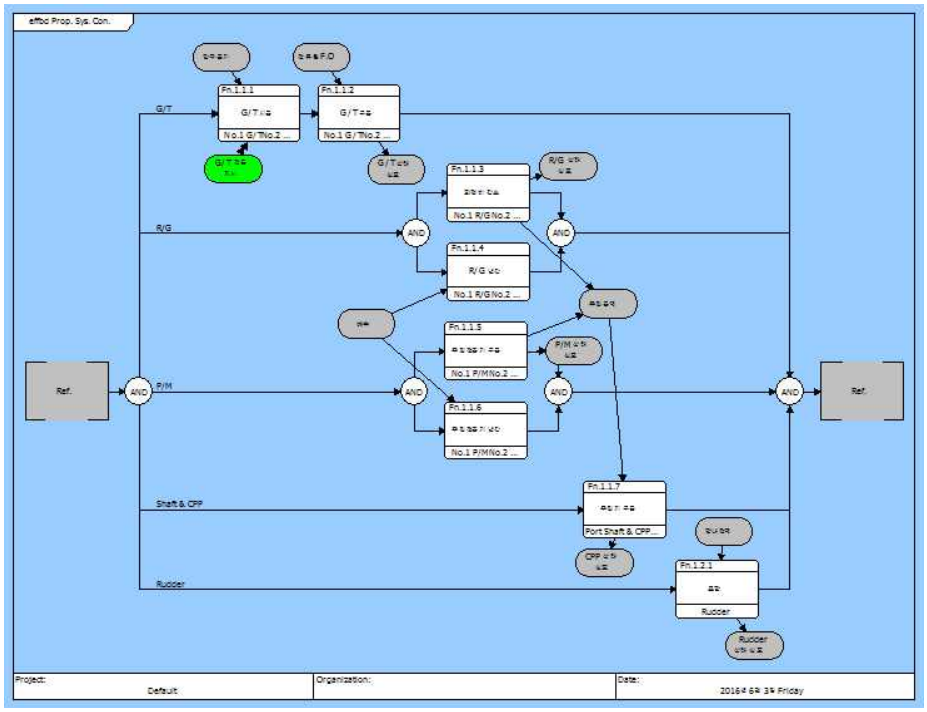

그림 48 추진계통 기능 구성

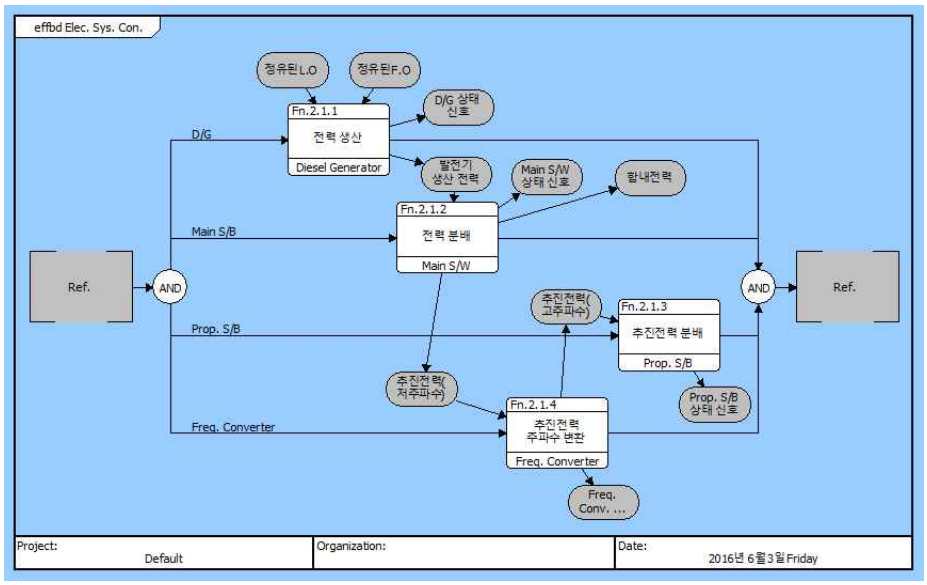

그림 49 전기계통 기능 구성

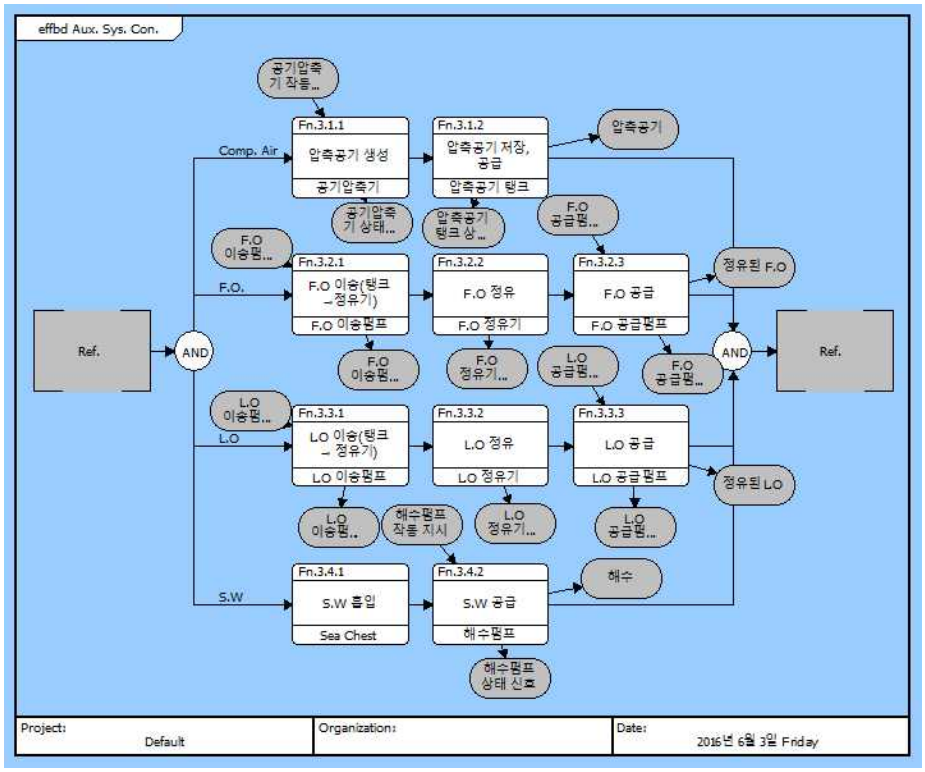

그림 50 보기계통 기능 구성

앞서 구성된 기능을 수행하기 위한 물리구조 또한 <그림 51>과 같이 기능과 동일하게 추진, 전기, 보기계통 및 통제계통으로 구성하였으며, 각 계통은 추진계통 9개(<그림 52>), 전기계통 4개(<그림 53>), 보기계통 10개(<그림 54>), 통제계통 1개로 구성하였다.

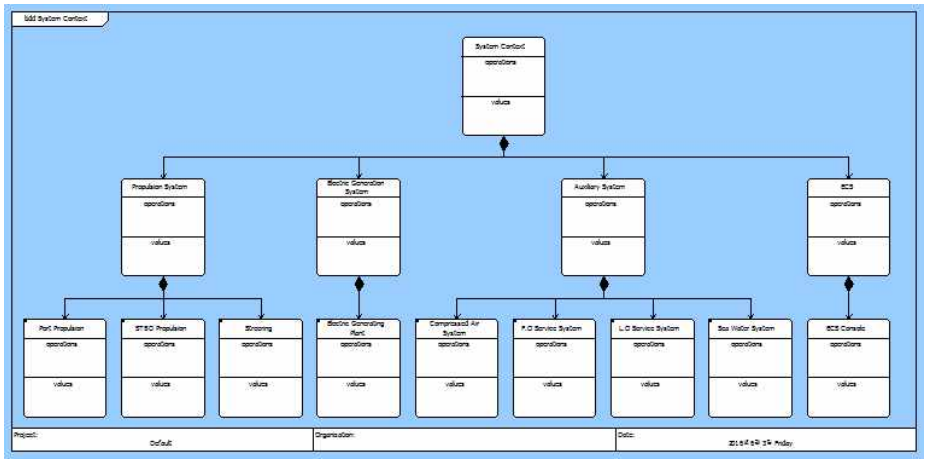

그림 51 추진체계 구조

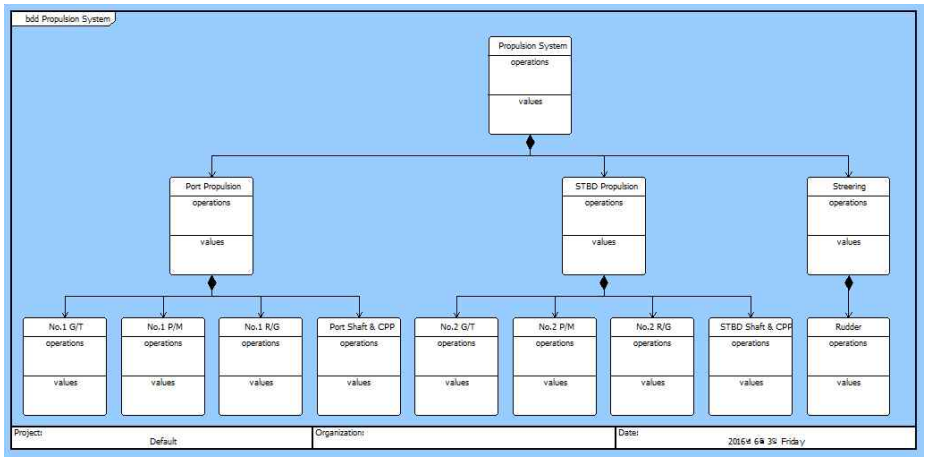

그림 52 추진계통 구조

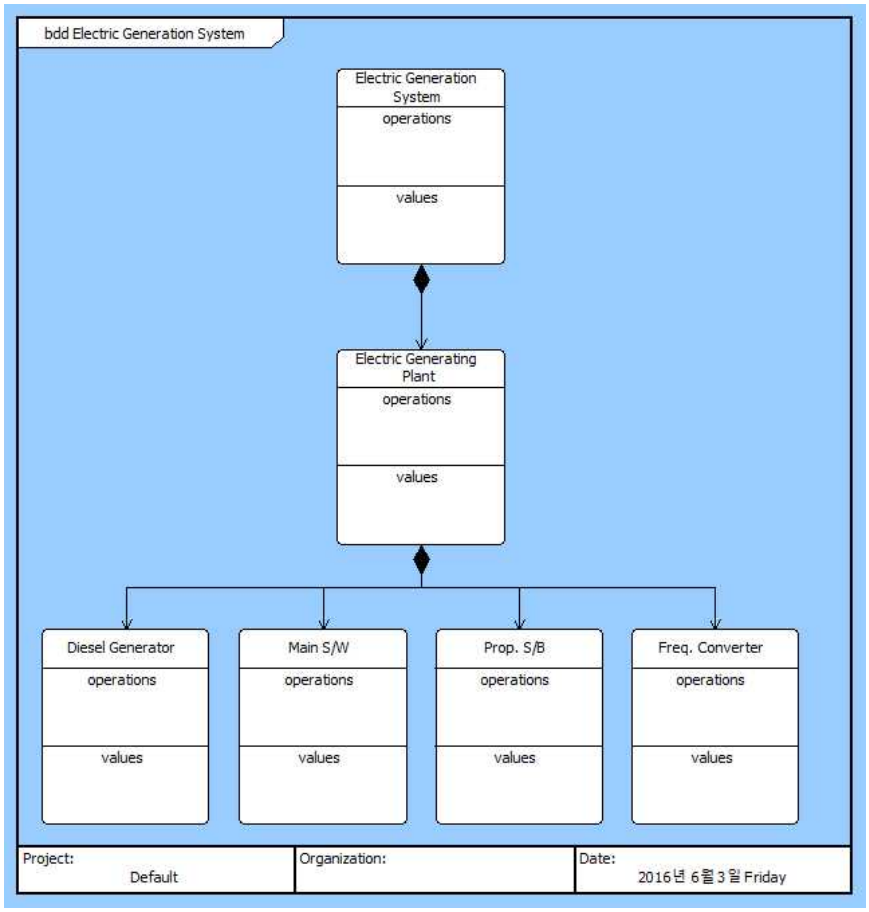

그림 53 전기계통 구조

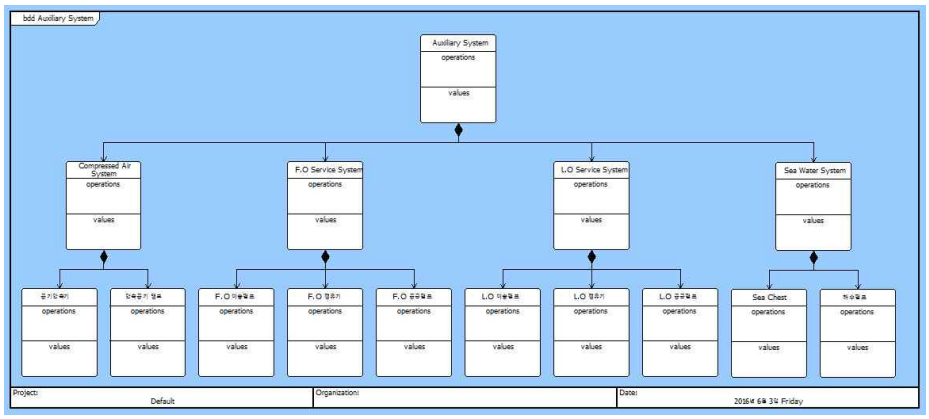

그림 54 보기계통 구조

<그림 55>와 같이 앞서 분석한 기능을 물리구조에 할당함으로써, 기능간에 식별된 인터페이스를 물리 구성품으로 할당하였고, 이를 통하여 <그림 56>과 같이 구성품의 인터페이스를 식별하여 Matrix를 작성하였다.

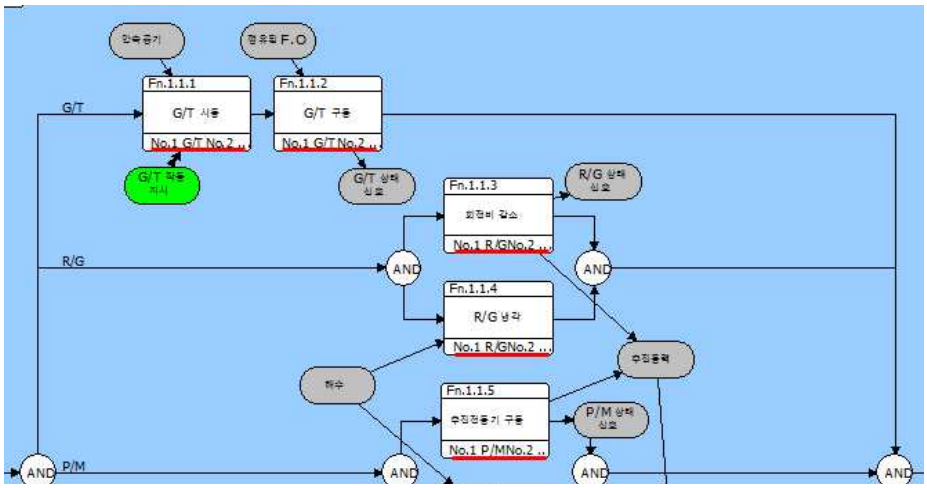

그림 55 기능할당

Table 23 C.1.1.1 No.1 G/T External I/O

Functions	Interface Items	Interfacing Elements
Fn.1.1.1 G/T 시동	← 압축공기	Fn.3.1.2 압축공기 저장, 공급 C.3.1.2 압축공기 탱크
Fn.1.1.2 G/T 구동	→ G/T 상태 신호	Fn.4.1.1 추진계통 조정 통제 C.4.1 ECS Console
	← 정유된 F.O	Fn.3.2.3 F.O 공급 C.3.2.3 F.O 공급펌프

Table 24 C.1.1.2 No.1 P/M External I/O

Functions	Interface Items	Interfacing Elements
Fn.1.1.5 추진전동기 구동	→ P/M 상태 신호	Fn.4.1.1 추진계통 조정 통제 C.4.1 ECS Console
	→ 추진동력	Fn.1.1.7 추진기 구동 C.1.1.4 Port Shaft & CPP Fn.1.1.7 추진기 구동 C.1.2.4 STBD Shaft & CPP
Fn.1.1.6 추진전동기 냉각	← 해수	Fn.3.4.2 S.W 공급 C.3.4.2 해수펌프

그림 56 Interface Matrix

앞서 식별된 인터페이스 Matrix를 이용하여 <그림 57>과 같이 인터페이스 종류를 표시한 DSM(Design Structure Matrix)을 작성하였다. 인터페이스 종류는 앞서 3.2.1장에서 설명한 NASA에서 분류한 인터페이스 분류를 적용하였다.(1:Electrical/Functional, 2:Mechanical/Physical, 3:Software, 4.Supplied Services)

		C.1.1.1	C.1.1.2	C.1.1.3	C.1.1.4	C.1.2.1	C.1.2.2	C.1.2.3	C.1.2.4	C.1.3.1	C.2.1.1	C.2.1.2	C.2.1.3	C.2.1.4	C.3.1.1	C.3.1.2	C.3.2.1	C.3.2.2	C.3.2.3	C.3.3.1	C.3.3.2	C.3.3.3	C.3.4.1	C.3.4.2	C.4.1.1
		No.1 G/T	No.1 P/M	No.1 R/G	Port Shaft & CPP	No.2 G/T	No.2 P/M	No.2 R/G	STBD Shaft & CPP	Rudder	Diesel Generator	Main S/W	Prop. S/B	Freq. Converter	공기압축기	압축공기 탱크	F.O 이송펌프	F.O 정유기	F.O 공급펌프	LO 이송펌프	LO 정유기	LO 공급펌프	Sea Chest	해수펌프	ECS Console
C.1.1.1	No.1 G/T															4			4						1
C.1.1.2	No.1 P/M												1											4	1
C.1.1.3	No.1 R/G																							4	1
C.1.1.4	Port Shaft & CPP		2	2																					1
C.1.2.1	No.2 G/T															4			4						1
C.1.2.2	No.2 P/M												1											4	1
C.1.2.3	No.2 R/G																							4	1
C.1.2.4	STBD Shaft & CPP						2	2																	1
C.1.3.1	Rudder											1													1
C.2.1.1	Diesel Generator																		4			4			1
C.2.1.2	Main S/W										1														1
C.2.1.3	Prop. S/B													1											1
C.2.1.4	Freq. Converter											1													1
C.3.1.1	공기압축기																								1
C.3.1.2	압축공기 탱크														4										1
C.3.2.1	F.O 이송펌프																								1
C.3.2.2	F.O 정유기																4								1
C.3.2.3	F.O 공급펌프																	4							1
C.3.3.1	LO 이송펌프																								1
C.3.3.2	LO 정유기																			4					1
C.3.3.3	LO 공급펌프																				4				1
C.3.4.1	Sea Chest																								
C.3.4.2	해수펌프																						4		1
C.4.1.1	ECS Console	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	

그림 57 DSM으로 표기된 인터페이스

5.2 적용효과

5.1장과 같이 기능분석을 이용하여 구성장비간 인터페이스를 식별, 문서화 방안을 제시함으로써, 추가작업을 최소화하는 인터페이스 식별 방법을 제시하였다. 식별된 인터페이스를 통해 설계변경사항 전파경로를 식별하였으며, 이를 사용자가 직관적으로 이해할 수 있는 인터페이스 종류를 포함한 DSM으로 표기하여 제공함으로써, 설계변경에 따른 후속 설계변경 미실시 감소를 기대할 수 있었다.

6 결 론

본 논문은 함정 설계시 발생하는 설계변경에 의한 주요 설계오류 발생을 최소화하기 위하여 설계변경 감소를 위한 체계공학 적용방안 및 후속설계변경 미실시에 의한 설계오류 감소를 위한 인터페이스 식별/관리방법을 제안하였다.

설계변경 감소를 위한 초기설계 단계의 체계공학 적용 방안을 제시하기 위해 설계변경 발생의 주요 이유를 분석하였고, 일반적인 체계공학 및 MBSE 방법론에 대해 연구하여 이를 함정 설계에 적용하기 위한 방법을 제시하였다. 이를 위해 요구조건 분석, 기능분석 및 할당, 설계조합 각 단계에서 실시하여야 하는 활동을 정의하였다.

설계변경에 의한 설계오류를 줄이기 위하여 후속설계변경 미실시로 인한 설계오류 발생 이유를 검토하였고, 인터페이스 및 DSM에 대해 연구하여 이를 이용한 설계 진행간 인터페이스 식별 및 관리 방법을 제시하였다. 이를 위해 기능분석을 통한 인터페이스 식별 및 DSM 표기방법을 제시하였다.

초기설계 단계의 체계공학 설계 방법을 함정 개념설계에 적용하였고, 앞서 제시한 단계별 활동을 실시하였으며, 이를 통해 설계진행 중 발생 가능한 설계변경을 감소 할 수 있음을 확인하였다. 또한 설계오류를 줄이기 위한 인터페이스 식별, 관리방법을 함정 추진체계 시스템에 적용하였고, 이를 통해 직관적인 설계변경 전파경로 식별을 통해 후속 설계변경 미실시 감소를 기대할 수 있음을 확인하였다.

향후에는 일반화된 체계공학 설계방법 정립 및 효율적인 인터페이스 관리방법에 대한 추가적인 연구 및 적용으로 설계오류 감소 효과 검증 및 설계수행을 위한 검증된 절차 정립이 필요하다.

참고문헌

[1] 곽해민, 강희진 등 (2013). 체계통합설계 Tool을 이용한 초기설계단계 기능분석 및 할당기법 연구, 2013 함정기술·무기체계 세미나, BEXCO

[2] 곽해민, 양영순, 송현석 (2015). 인터페이스 관리를 이용한 효율적 함정설계방법, 2015 함정기술·무기체계 세미나, BEXCO

[3] 권용수 편역, 미 국방획득대학교 (2010). 신 시스템엔지니어링 입문. 서울 : 아이워크북

[4] 박진원, 정치훈, 곽해민 등 (2013). 소요군 중심의 개념설계 수행 교훈과 발전방향, 2013 함정기술·무기체계 세미나, BEXCO

[5] 박진원 (2015). Early Naval Ship Systems Engineering, 함정설계 특별강좌, 방위사업청

[6] 송현석 (2015). DSM(Design Structure Matrix)을 활용한 해양플랜트 설계변경 관리 방안, 석사학위논문, 서울대학교

[7] 송현석, 양영순, 곽해민 (2015). 설계구조행렬(DSM)을 이용한 FPSO Topside Module 구성. 한국해양과학기술협의회 공동학술회의, 218-221

[8] 송준태 등, 대한조선학회 (2015). 함정, 파주 : 텍스트북스

[9] 신정호 등(공편), 방위사업청 (2013). 체계공학(SE)을 기반으로 하는 함정 연구개발 업무편람, 서울 : 방위사업청

[10] 양영순, 장범선 (1999). AHP 기법을 이용한 최적 구조 설계. 대한조선학회 논문집, 36(1), 82-83

[11] 한창수 (2015). 외국 잠수함 건조 사례 교훈 분석. 한국해양안보포럼 E-저널, ISSN 2465-809X

[12] Arena, Mark V (2005). Monitoring the progress of

Shipbuilding Programmes. Retrieved from http://www.rand.org/

[13] Browning, Tyson R. (2001). Applying the Design Structure Matrix to System Decomposition and Integration Problems : A Review and New Directions, IEEE Transactions on Engineering Management, 48(3), 292-306

[14] CODLAG. (미상). http://ko.wikipedia.org/wiki/CODLAG 에서 검색

[15] Eppinger, Steven D. (2012). Design Structure Matrix Methods and Applications. London : The MIT Press

[16] Estefan, Jeff A. (2007). Survey of Model-Based Systems Engineering (MBSE) Methodologies, INCOSE MBSE Focus Group

[17] Kapurch, Stephen J(Ed.) (2007). NASA Systems Engineering Handbook. Washington,D.C : NASA Headquarters

[18] Kelly, Brian (2006). Interface management: Effective communication to improve process safety, Journal of Hazardous Matrials, 130(3), 321-325

[19] Kerns, Corey (2011). Naval Ship Design and Synthesis Model Architecture Using a Model-Based Systems Engineering Approach. Masters Thesis, Virginia Tech

[20] Lalli, Vincent R. (1997). Training Manual for Elements of Interface Definition and Control, Retrieved from http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19970018043.pdf

[21] NAVSEA 2005 Cost Estimating Handbook. (n.d.). Retrieved from http://www.navsea.navy.mil/Organization/SEA05C/handbook.a

spx
[22] NAVSEA (2007), Summer Naval Surface Ship Design Course
[23] U.S. Department of Transportation (2007). Systems Engineering for Intellignet Transportation Systems. Washington D.C : Department of Transportation
[24] Vitech (2012). CORE를 이용한 모델기반 시스템 엔지니어링, Core 사용자교육, 함정기술처
[25] Yun, J.Y, Kim, J.H, Kang, S.H. & Seo J.W. (2013). A study on an Optimal Plant Design Collaboration System Using a Design Structure Matrix. Journal of The Korean Society of Civil Engineers, 33(1), 337-346

Abstract

Early-phase Design of Naval ship using Systems Engineering and Interface Management

Hae-Min Kwak
Dept. of Naval Architecture
and OceanEngineering
The Graduate School
Seoul National University

Naval ship has multidiscipline characteristic unlike commercial ship because of complex of naval ship system and needs complex interactions. So it changes concurrent design that each results influence other disciplines from spiral design. And naval ship needs over 10 years from the start to the end of the project. So sometimes it needs some changes of components during the design.

Because of these characteristics, naval ship needs some design changes 20 times more than commercial ship. And there is a high possibility of design error when the design change happens.

To resolve this problem, this paper researches how to apply systems engineering in early-phase design to reduce the design changes and finds some methods to reduce the design errors by the interface managements.

To reduce the design changes, this paper researches some operations that could applied at the early-phase design by modifying the process of systems engineering like requirement analysis, functional analysis and allocation, and design synthesis. And it distinguishes some interfaces by functional analysis, manages the interfaces by DSM. So that it can be used to distinguish the way how the design change affects, and applied to the ship design.

Keywords : Systems engineering, Functional analysis, Design change propagation, Interface management

Student Number : 2014-20652

※ 보안성 검토 완료 : 방사청 운영지원과-19726호('16. 7.19) 보안성 검토 결과 통보