

A Study of Software Product Line Engineering application for Data Link Software

Jin-Woo Kim*, Woo-Sin Lee*, Hack-Joon Kim*, So-Yeon Jin*, Se-Hyeon Jo*

Abstract

In this paper, we have studied how to reuse common data link software by applying software product line engineering. Existing common data link software performed different stages of design, implementation, and testing without sharing the accumulated knowledge of different developers. In this situation, developers agreed that sharing the assets of each project and reusing the previously developed software would save human and time costs. Even with the initial difficulties, the common Data Link is a continually proposed project in the defense industry, so we decided to build a product line. The common data link software can be divided into two domains. Among them, the initial feature model for the GUI software was constructed, and the following procedure was studied. Through this, we propose a plan to build a product line for core assets and reuse them in newly developed projects.

▶ Keyword: Data Link, Common Data Link, Software Product Line Engineering, Software Engineering, Feature Model

I. Introduction

데이터링크는 그림 1과 같이 항공 탑재용과 지상 탑재용으로 구분된다. 감시체계, 타격체계 및 지휘통제체계와 연동하여 상황인식, 위협평가, 지휘통제 및 교전지원을 위한 필요한 전술/영상 정보를 실시간으로 전송하는 통신체계이다[1].

기능별로는 전술 데이터링크(TDL : Tactical Data Link)와 공용 데이터링크(CDL : Common Data Link)로 구분할 수 있다. 이 중 CDL은 ISR(군사정보, 감시 및 정찰/Intelligence, Surveillance and Reconnaissance, 이하 ISR이라 한다) 체계 간의 상호운용성과 대용량 데이터 교환을 위한 네트워크 통신체계이다. CDL은 양방향 동시 전송 방식, 항재밍(Anti-Jamming) 능력을 지니고 ISR 센서, 센서 플랫폼, 지상 터미널 간의 디지털 데이터 통신 능력을 제공한다. 주요 특징으로는 광대역 주파수를 사용하여 대용량의 정보를 전송할 수 있다는 것과 중계기를 이용할 경우 비가시거리까지 통신 가능하다는 것이다[2]. 감시 및 정찰 정보의 TDL의 전술 정보와는 다른 성격을 가지며, 우선 데이터의 양 측면에서 매우 큰 용량을 가지게 되며 수집된 정보를 모

든 전투원에게 전달하는 것이 목적이 아니라 정보 분석 부서에 전달하는 것을 기본 목적으로 한다.

최근의 CDL은 미래 네트워크 중심전 수행 시 요구되는 효과적인 정보공유를 위해 다중 플랫폼(MP : Multi-Platform) CDL로 발전하고 있으며 주로 감시 및 정찰을 위한 유.무인기에 적용될 예정이다. CDL의 데이터 처리를 위한 소프트웨어는 크게 데이터링크 장비의 임베디드 시스템으로 탑재되는 데이터링크 소프트웨어와 데이터링크 장비와 이더넷으로 연결하여 데이터링크 장비로부터 받은 정보들을 도시하고, 운용자의 데이터링크 장비 설정과 제어가 가능한 GUI 기반의 윈도우용 소프트웨어로 나눌 수 있다. 초기 장비 제어와 자원 할당부터 메시지 처리와 영상 재생, 로그 관리까지 다양한 기능들이 있다. 현재 공용 데이터링크 프로젝트는 기존의 개발제품 외에 현재에도 다수가 진행 중이며 프로젝트 별로 소프트웨어 개발 인력이 적게 투입된다. 유사 기능의 반복되는 개발로 인해 비용 및 업무의 효율성이 떨어지는 문제가 있다. 이러한 시간적, 인적 문

• First Author: Jin-Woo Kim, Corresponding Author: Jin-Woo Kim

*Jin-Woo Kim (stcjinu.kim@hanwha.com), Hanwha Systems Co.

*Woo-Sin Lee (woosin.lee@hanwha.com), Hanwha Systems Co.

*Hack-Joon Kim (hjn.kim@hanwha.com), Hanwha Systems Co.

*So-Yeon Jin (soyeon.jin@hanwha.com), Hanwha Systems Co.

*Se-Hyeon Jo (shn.jo@hanwha.com), Hanwha Systems Co.

• Received: 2018. 11. 15, Revised: 2018. 12. 10, Accepted: 2018. 12. 13.

제를 해결하기 위해 유사 기능의 개발물을 재사용할 수 있는 형태로 관리하고 사용한다면 개발에 투입되는 시간적인 요소를 줄일 수 있어 비용 절감이라는 효과를 기대할 수 있다. 이를 위해 공용 데이터링크 소프트웨어에 제품라인 공학(PLE : Product Line Engineering) 을 적용하여 공통된 기능을 식별하였다. 그리고 재사용 할 수 있고 미래 활용 가능성이 높은 기능 위주로 추출하여 핵심자산화하고, 추후 고객의 요구사항에 맞는 맞춤형 제품 개발의 효율성을 향상할 수 있는 프로세스에 대해 연구하였다.

본 논문의 구성은 다음과 같다. 2장에서는 현재 개발 중인 각 공용 데이터링크 프로젝트의 주요 기능들에 대해 살펴본다. 재사용 가능한 모듈들을 알아보고 3장에서는 데이터링크 소프트웨어에 대해 제품라인 공학의 절차 중 feature 모델링을 진행하고 그 후의 개발 절차에 대해 설명한다. 마지막으로, 4장에서는 제품라인 공학 적용의 결론 및 앞으로 계획을 논의한다.

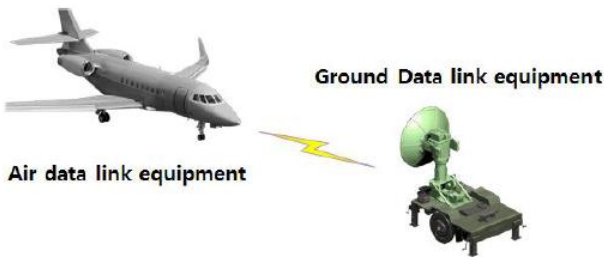


Fig. 1. Concept of Data Link

II. Preliminaries

1. Related works

본 논문은 공용 데이터링크 소프트웨어를 제품라인 모델링하여 차후 프로젝트에 적용하는 것을 연구하였다. 본문에서 제품라인 공학의 방법론을 적용하기 전에 제품라인 공학의 전반적인 내용에 관하여 연구하였다. 또 공용 데이터링크의 기능 구조에 대한 이해를 돕기 위해 기존의 공용 데이터링크의 기능들에 대해 살펴보고 기존의 프로젝트에서 구현한 공통적인 기능들과 차별화된 기능들이 어떤 내용이 있는지 연구하였다.

1.1 Overview of Software Product Line Engineering

일반적으로, 품질이 좋은 소프트웨어를 개발하기 위해서는 재사용성, 안정성, 성능 등을 고려하여 많은 시간과 노력이 필요하다. 소프트웨어 제품라인 공학은 시장이나 사용자의 요구나 목표를 만족하게 하기 위하여 특정한 도메인에 속하는 소프트웨어 제품들의 유사한 기능들을 공통된 핵심자산으로 만들고 공유하는 소프트웨어 시스템의 집합으로 기존에 제시되었던 코드의 재사용이 아닌 분석, 설계, 구현, 시험 등 모든 단계의 재사용을 강조하여 개발된 소프트웨어의 재사용성과 개발 효율성

을 증가시키고, 유지보수 비용을 감소시킬 수 있다. 도메인이란 지식의 분야 또는 일련의 개념 또는 현상 종사자들이 이해하고 있는 용어에 의해 특성화 되는 행위, 공통된 특성을 공유하는 현재와 미래의 어플리케이션 집합을 말한다. 소프트웨어 제품라인 공학의 목적은 이러한 일련의 소프트웨어 시스템의 공통성을 이해하고 특징을 구별시킴으로써 체계적으로 소프트웨어 제품 패밀리를 개발할 수 있도록 지원하는 방법이다[3].

이러한 내용으로 두 단계로 나뉘어서 활동하는데, 도메인 공학과 어플리케이션 공학으로 나눌 수 있다. 도메인 공학은 특정 제품 패밀리 내에서 재사용 가능한 핵심 자산을 추출하고 구성하여 새로운 제품 개발 시 재사용 가능한 모듈을 제공하는 활동으로 도메인 모델의 필수 요소와 가변 요소를 구분하여 재사용 가능한 아키텍처, 컴포넌트, 테스트케이스를 설계한다. 아키텍처 설계에서는 광범위한 의미의 설계 결정이 이루어지고 컴포넌트 설계에서는 좁은 의미의 설계 결정이 이루어진다. 컴포넌트 설계에서는 재사용성과 조합 적용을 고려하여 아키텍처 컴포넌트를 정제함으로써 제품라인을 위해 계획된 제품은 반드시 재사용 가능한 자산 컴포넌트로부터 조합할 수 있게 된다. 어플리케이션 공학은 도메인 공학의 핵심 자산을 바탕으로 새로운 제품에 대한 요구사항을 분석하여 기존의 핵심 자산 중 동일한 기능의 모듈을 재사용하고 새로운 기능은 새로 개발하여 어플리케이션을 개발한다.

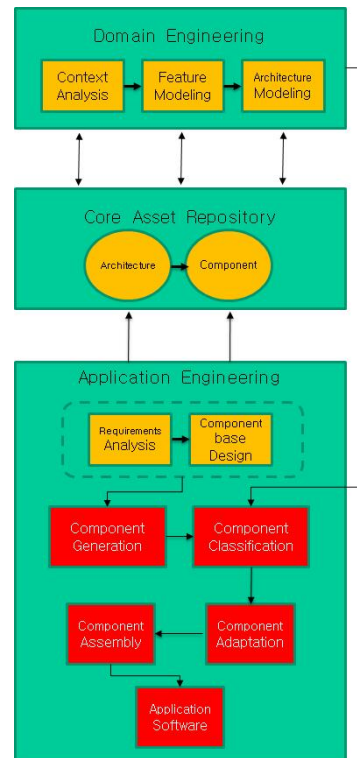


Fig. 2. Process of Software Product Line Engineering

이러한 소프트웨어 제품라인 공학의 절차는 그림 2와 같이 요약할 수 있다.

1.2 Overview of Common Data Link

STANAG 은 북대서양조약기구(NATO : North Atlantic Treaty Organization)에서 제정한 군사적 표준 규격으로서 “Standardization Agreement”의 약자이며, STANAG 4586은 “Standard Interfaces of UAV Control System(UCS) for NATO UAV Interoperability” 로서 NATO 회원국 간의 연합 임무 시에 무인기 상호 운용성을 확보하기 위해서 표준화한 무인기 운용을 위한 표준 ICD 이다[4]. STANAG 4586의 목적은 여러 나라에 배치된 무인기 간에 밀접한 조직력을 갖추게 하고, 가용한 무인기를 통해 더 빠른 임무 수행을 하게하며, 상호 비행체/탐제 임무장비를 제어할 수 있도록 하여 시너지 효과를 도모하는 것이다.

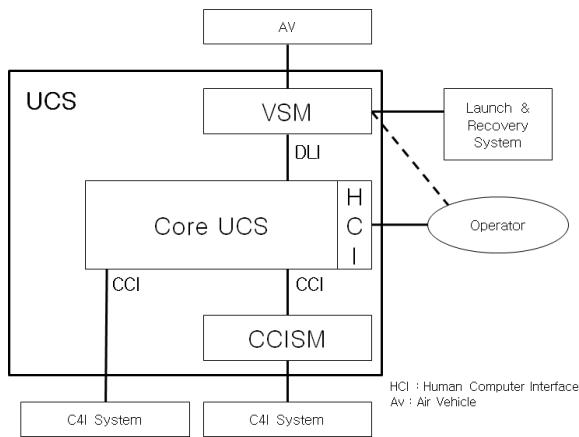


Fig. 3. Structure of UCS

현재의 공용 데이터링크는 주로 무인기 체계에서 운용하는데, 이러한 UCS(UAV Control System) 구조는 그림 3과 같이 CUCS(Core UCS), DLI(Data Link Interface), CCI(Command and Control Intereface), VSM(Vehicle Specific Module), CCISM(Command and Control Interface Specific Module)로 구성된다[5]. 그 중에서 CUCS 와 VSM 간의 DLI 인터페이스를 통해 AV 제어 및 상태 메시지, 새 AV 연동을 통해 중계 구조로 확장 가능한 네트워크를 지원하여 감시정찰 및 전술적 상황 인식을 가능하게 한다. CUCS 는 지상 조종사 및 임무 운용자에게 GUI 운용 환경을 제공하는 요소로서 이는 기존 GCS 구조에서 임무계획도시 컴퓨터, 비행통제 컴퓨터, 영상통제 컴퓨터 등의 임무 컴퓨터를 포함하는 요소이며, 각 기종에 공통적인 임무 기능을 수행하는 역할을 담당한다.

1.3 Domain of Common Data Link

데이터링크 소프트웨어의 도메인은 그림 4와 같이 데이터링크 장비에 탑재되는 임베디드(Embedded) 소프트웨어인 모뎀 소프트웨어와, 제어기에 탑재되어 장비와 연동하여 운용자가 장비를 제어하고 상태를 모니터링이 가능한 GUI 소프트웨어로 분리할 수 있다. 기존의 데이터링크 프로젝트들에서 모뎀 소프트웨어와 GUI 소프트웨어는 설계단계부터 각각의

ter Software Configuration Items) 로 분류하여 자산이 분리되어 있다. 각각의 도메인 내에 feature 들 사이의 관계를 파악하여 체계화하는 작업을 진행하였다.

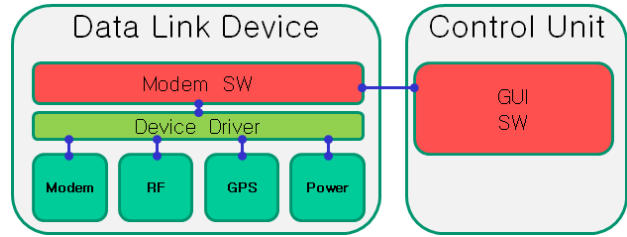


Fig. 4. Interface of Data Link Software Domain between GUI Software Domain

1.3.1 Modem Software Domain

공용 데이터링크의 소프트웨어 측면의 요구사항은 대동소이하여 그 기능이 크게 차이 나는 부분은 없었지만, 각 프로젝트의 개발자들이 모두 다르고, 개발자들 간의 지식 공유가 이루어지지 않았으며, 하드웨어 의존성이 요구되는 도메인의 특성 상 도메인 지식 통합하는 데에 시간과 노력이 필요하였다. 모뎀 소프트웨어는 데이터링크 모뎀의 CPU 상에서 동작하는 응용 프로그램이다. 표 1과 같이 각 프로젝트 별로 플랫폼 환경이 다르다. 그 중 C 프로젝트에서는 대용량으로 전송하는 영상 패킷을 처리하기 위해 WinPath3 CPU 를 탑재한 별도의 장비가 존재하여 모뎀 소프트웨어에서는 영상 데이터를 처리하지는 않았다.

Table 1. Modem Software Platform for each Project

Project	OS	CPU	Lang- uage	Uniqueness
A	linux 2.6.29	winpath3 (MIPS)	C	-
B	NEOS	Zynq7000	C	-
C	VxWorks	MPC8572E	C	extra WinPath3
D	VxWorks	MPC8569E	C	-

그림 5는 데이터링크 모뎀 소프트웨어의 구조도이다. 기본적으로 모뎀 소프트웨어는 다른 노드와 송수신하는 데이터를 처리, 데이터를 STANAG 4586 메시지와 같은 포맷으로 가공하여 메시지 처리하고 그 메시지를 GUI 소프트웨어와 연동, 메시지에 따른 모뎀, RF, 안테나 장비 제어, 장비 점검 및 관리를 하는 컴포넌트들로 구성되어 있다.

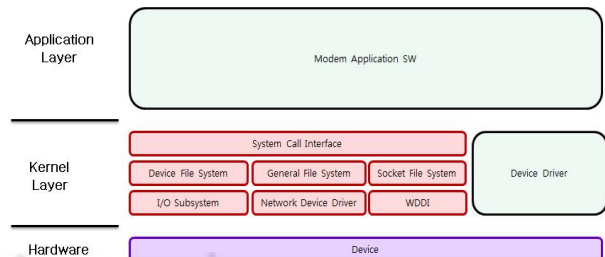


Fig. 5. DataLink Modem Software Architecture

이미 개발된 프로젝트들은 하위의 하드웨어 및 시스템을 제어, 관리하는 컴포넌트들은 재사용을 고려하지 않고 개발되어 왔기 때문에 하드웨어 의존성이 높다. 이 부분을 하드웨어 특성에 따라 변경하지 않고 재활용할 수 있게 device driver 와의 연동방법, 함수명 등을 정의한 도메인 사전을 구축하기로 하였다.

1.3.2 GUI Software Domain

GUI 소프트웨어는 사용자가 운용하는 응용 프로그램으로, 그림 6의 구조에서 사용자에게 데이터링크 장비 관리 및 장비의 상태와 성능을 보여주고, 데이터링크 네트워크 자원할당을 하는 기능 등을 제공한다.

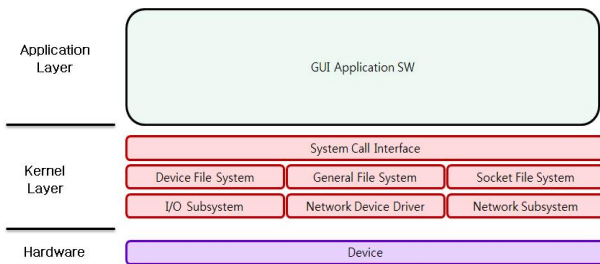


Fig. 6. DataLink GUI Software Architecture

타 노드로부터 수신한 메시지는 모뎀 소프트웨어가 GUI 소프트웨어에 전달하여 노드의 상태와 성능 정보를 도출할 수 있게 한다. 이렇듯, GUI 소프트웨어는 모뎀 소프트웨어로부터 수신한 메시지를 처리함으로써 기능 대부분이 시작된다. 또 운용자가 생성한 자원할당 정보를 모뎀 소프트웨어로 전달하면 데이터링크 모뎀은 해당 정보를 무선으로 네트워크의 노드들에 전달한다. GUI 소프트웨어의 특징은 기능적인 측면보다는 사용자에게 보이는 화면구성이 크다. 이는 고객별로 선호하는 화면 구성과 새로운 서비스 등의 요구사항 변화도 많은 도메인이다.

Table 2. GUI Software Platform for each Project

Project	OS	CPU	Language	Uniqueness
A	Windows 10	intel	MFC	-
B	Windows 10	intel	C#	-
C	Windows 7	intel	MFC	Some outsourced
D	Windows 7	intel	MFC	-

표 2는 기존 프로젝트들의 GUI 소프트웨어 운용 환경이다. B 프로젝트는 유일하게 C# 언어를 사용하여 개발하였는데, 이는 WPF(Windows Presentation Foundation)을 사용함으로써, 응용 프로그램의 화면구성 구현과 기능 구현을 분리할 수 있는 장점이 있다. WPF 는 응용 프로그램 개발에 필요한 UI 프레임을 지원한다. XAML(Extensible Application Markup Language)를 사용하여 화면구성 부분을 구현한다. 따라서 추후 새로운 기능 추가, 화면구성 변경, 새로운 화면 추가 등의 소스코드 수정이나 구현 후의 소프트웨어 신뢰성 시험에서 인적, 시간적 비용을 절감할 수 있었다. C 프로젝트는 일부 기능을 외주 제작을 하여 개발과정이 외주

개발자와의 협업이 필요하였다. 외주 제작사와의 유지보수 기간이 끝난 이후부터는 X 사의 개발자가 유지보수를 하였기 때문에 컴포넌트의 구현내용 파악에 문제가 없었다.

GUI 소프트웨어 역시 모뎀 소프트웨어와 마찬가지로 개발자들 간의 지식 공유가 없고, 독립적으로 개발했다는 문제가 있었다. 본 사례 연구는 이러한 과정을 최소화시키고자 제품라인 모델링을 하는 것이고, 앞으로 컴포넌트들의 조합으로 새로운 프로젝트 개발함으로써 개발 비용을 절감하고자 한다.

III. The Proposed Scheme

본론에서는 서론에서 제기한 소프트웨어 제품라인 공학을 적용하여 재사용 모델링을 하는 방법에 관하여 서술한다. 현재까지 소프트웨어 재사용을 위한 접근방법 중 소프트웨어 제품라인 공학의 보편적인 방법론으로 FORM(Feature Oriented Reuse Method)을 들 수 있다. FORM 은 포항공대 소프트웨어 공학연구실에서 개발한 소프트웨어 제품라인 공학 방법론으로 세계에서 가장 많이 사용되고 참조되는 방법론 중 하나이다. FORM 에서 도메인 분석에 가변성 분석을 위해서 사용하는 feature 모델은 암묵적인 표준(de facto standard) 으로 인정받는다. feature 모델로 특정 도메인의 가변성을 식별하고, 그것을 반영한 재사용 가능한 컴포넌트를 자산화하여, 이 컴포넌트를 조합하여 새로운 솔루션을 만들어낼 수 있다. 이미 다양한 분야에 제품라인을 구축하는 데에 이 방법론이 사용되어 참고 사례가 많고, 데이터링크 체계의 특성상 기본적으로 서비스하는 기능들이 어느 정도 대동소이하므로 재사용이 가능한 형태의 컴포넌트 자산을 추출하여 제품에 적용하는 FORM 방법론을 연구하여 데이터링크 소프트웨어에 적용하기로 하였다.

feature 모델, 요구사항 모델, 아키텍처 등과 같은 특정한 모델링 언어를 확장하여 가변성을 통합하여 모델링한다고 할 수 있다. FORM 방법론은 feature 중심으로 도메인을 분석하여 시스템의 공통점과 차이점을 파악하고 그것을 조합하여 사용하는 것으로 제품라인을 효과적으로 모델링할 수 있다. FORM 방법론은 도메인을 분석하여 참조모델을 만드는 과정인 도메인 공학과 도메인 공학에서의 산출물을 통해 새로운 시스템을 만드는 과정인 어플리케이션 공학으로 나누어진다. 시스템 사용자, 분석자, 설계자, 개발자의 관점에 따라 모델링을 하고 상호간에 효율적으로 의사소통이 가능한 모델을 제시한다[6-8].

데이터링크 소프트웨어의 제품라인 구축을 위해 X 사의 이미 개발된 4개의 프로젝트에서 feature 모델을 정립하고 분석하여 추출식 모델을 구성하는 것을 목표로 하였다.

1. Feature Modeling

본 논문에서는 GUI 소프트웨어 도메인에 대하여 feature 모델링을 하였다. 업무가 제각각인 기존의 개발자들이 모두 모여 참여하기

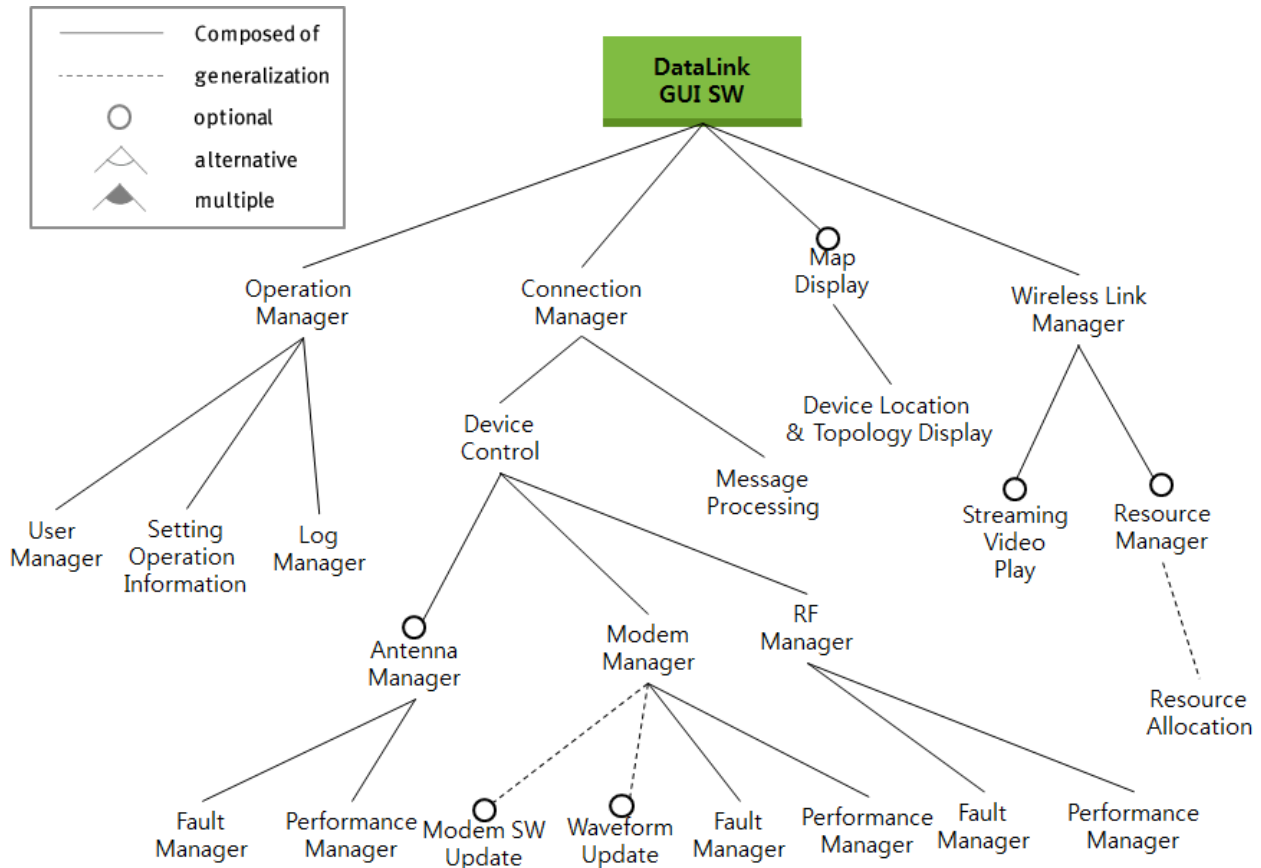


Fig. 7. Data Link GUI Software Feature Model

에는 무리가 있어서 GUI 소프트웨어 도메인 활동을 우선적으로 시작하였고, 모뎀 소프트웨어 도메인은 차후 진행하기로 하였다. GUI 도메인은 하드웨어 의존도가 낮고 모두 윈도우용 응용 프로그램이라 컴포넌트 코드를 선정하는 과정도 모뎀 소프트웨어 도메인에 비해 진행이 빠를 것으로 예상했기 때문이다.

기 개발된 4가지 프로젝트들의 CSCI 구조를 파악하고 CSC 단위의 기능을 파악하였다. 개발자들 간의 토의를 통해 이 CSC 단위의 기능을 하나의 feature 로 하여 feature 모델을 정립하기로 하였다. 프로젝트 별로 같은 기능을 담당하지만 다른 이름을 가진, 대응되는 feature 에 대해서는 이름을 통일시켰다.

Feature 모델과의 관계는 모든 제품에 들어가는 것은 공통 (mandatory) feature 로, 고객의 요구사항에 따라 특정 제품에만 들어가야 할 feature 는 선택적(optional) feature 로, 하나의 feature 가 포함되면 다른 feature 는 배제되어야 하는 택일 (alternative) feature, 하나의 feature 와 함께 선택 가능한 다중선택(multiple) feature 로 분류하였다[6].

그림 7은 GUI 소프트웨어 도메인의 초기 feature 모델이다. Feature 간 구성관계를 정할 때, 재사용 가능성과 요구사항 빈도 등을 고려하여 개발자들 간의 협의를 통해 결정하였다. GUI 소프트웨어 도메인의 주요 서비스 feature 는 운용 관리, 연결 관리, 장비 제어, 무선헤크 관리 등이며, 연결 관리 feature 는 장비 제어, 메시지 처리 feature 로 구성된다. 장비 제어 중 안테나 관리는 안테나 특성과 프로젝트 별 운용개념에 따라 선택적으로 포함되는

feature 이다. 4 개의 프로젝트 중 1개의 프로젝트가 안테나를 제어하는 모듈을 개발하였고, 제어가 필요한 경우, 안테나 개발업체가 제공하는 별도의 제어 프로그램을 사용하여 안테나를 제어하고 점검하였다. 모뎀 관리와 RF 관리는 각각 장애 관리 feature 와 성능 관리 feature 를 포함한다. 모뎀 관리 feature 에는 모뎀 소프트웨어 업데이트와 웹브라우저 업데이트 feature 가 선택적 feature 로 포함된다. 운용 관리 feature 는 사용자 관리, 운용정보 설정, 로그 관리 feature 를 포함한다. 지도 도시 feature 는 고객 요구사항에 따라 선택적이고, 장비위치와 토폴로지 도시 feature 를 포함한다. 무선헤크 관리 feature 는 스트리밍 영상 재생과 자원 관리 feature 가 선택적으로 포함된다. 그림 7의 모델에는 나타나지 않았지만 차후 개발자들과 더 연구하여 모델을 세분화하여 정립하면 나올 수 있는 관계라서 택일 feature 와 다중선택 feature 를 범례에 포함하였다.

이렇게 한 도메인을 분석하여 설계한 feature 모델의 의미는 다음과 같다[9].

- 제품라인에 속해 있는 제품 사이의 공통점과 차이점을 분석하게 해준다.
- 공통점으로 분석된 feature 는 재사용의 가능성이 높은 부분을 나타낸다.
- 차이점으로 분석된 feature 는 아키텍처 및 컴포넌트에서 변화를 수용해야 하는 부분을 나타낸다.
- feature 들 사이의 관계가 가시화된다.

2. Architecture and Component Configuration

Feature 모델은 feature 내부의 기능을 더 세분화하면 하나의 CSCI 구조도와 유사한 형태가 된다. 이렇게 feature 모델링을 하여 개발자들은 한 도메인의 이해도를 높일 수 있었고, 차후 제품라인화 되었을 때 feature 모델링에 참여하지 않은 개발자도 전체적 기능 구조에 대해 파악하기 쉬울 것이다.

feature 모델을 정립하여 초기 제품라인 모델을 구축하였다. 이 모델은 추출식 제품라인으로, 각 feature의 컴포넌트 코드들을 공용화하는 작업이 필요하다. 기존의 시스템으로부터 컴포넌트 자산을 확보하는 과정이 필요한데, 하나의 컴포넌트 안에 한 가지 기능이 아닌 다양한 기능이나, 공통적 요소가 없는 기능이 포함되어 이를 명확하게 분리하여야 한다. 이를 위해서 컴포넌트 역할에 따른 컴포넌트 타입 분류와 정의, 컴포넌트 타입 탐색, 컴포넌트 타입 기반 재공학의 프로세스를 밝아야 한다[10].

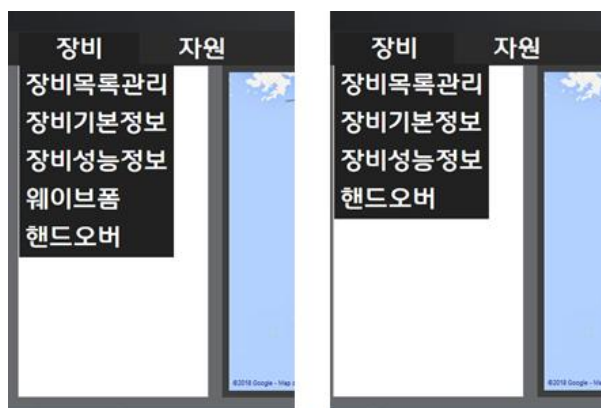
아키텍처 설계 및 정련 그리고 컴포넌트 설계 과정에서 반드시 고려해야 할 것은 feature 모델, 제품라인 요구사항에서 식별된 품질 요구사항 및 제약사항을 만족하도록 하는 것이다 [10]. 컴포넌트 코드를 생성할 때 어떤 기준으로 코드를 조합하여야 할지를 결정하기 위해 그림 7의 feature 모델에서 정의한 feature 에 대응하는 4가지 프로젝트의 컴포넌트들에 각각 응집도, 복잡도, 이해도, 하드웨어 의존도라는 평가 지표를 설정하여 조사하였다. 평가 지표에 대한 설명은 다음과 같다.

- 응집도 : 모듈의 독립성을 나타내는 개념으로 하나의 모듈 내부 처리 요소 간에 기능적 연관도를 평가
- 복잡도 : 모듈 내의 메소드 간의 연관 관계로 평가
- 이해도 : 전체 코드에 대한 주석 코드의 비율로 평가
- 하드웨어 의존도 : 특정 하드웨어 모듈에 의존하여 기능의 수행 정도를 평가

위 지표를 이용하여 품질이 우수한 프로젝트의 컴포넌트 코드를 기본으로 하여 나머지 프로젝트의 코드와의 차이점을 코드에 포함하도록 하여야 한다. 이러한 컴포넌트 타입 재공학에는 높은 응집도를 위하여 탐색된 컴포넌트가 2가지 이상의 타입을 가지고 있는 경우, 하나의 타입만을 가지고 있도록 컴포넌트를 분리하여 응집도가 높은 컴포넌트를 생성해 재사용성을 높일 수 있는 컴포넌트 분리 재공학과 컴포넌트 간의 의존성 강도를 계산하여 의존성이 높은 부분을 하나의 컴포넌트로 묶는 방법 등이 다른 사례에서 제시되었다[10]. 재공학 후의 결과물로 나온 컴포넌트들에 대해서 단위 테스트와 같은 신뢰성 시험 수행과, 위의 4가지 평가지표를 포함하여 코드의 품질을 측정할 수 있는 유의미한 평가지표를 도입하여 자산 컴포넌트의 품질을 관리하여야 한다.

3. Application Engineering Phase

새로운 프로젝트 개발을 시작하면, 요구사항 분석 후 제품라인에서 요구사항에 매치되는 feature 들을 선택하고, 해당 feature 들의 자산 컴포넌트 코드를 조합한다.



(a) include optional feature (b) exclude optional feature
Fig. 8. Product comparison due to feature selection difference

컴포넌트 코드를 조합하여 컴파일하면 요구사항에 맞는 소프트웨어 제품이 생성된다. 이러한 과정을 통해 필요한 기능을 포함하여 생성할 수 있었다. GUI 소프트웨어 도메인을 대상으로 소프트웨어 제품라인 공학 프로세스를 테스트하여 제품을 생성해 보았다. 그림 7을 보면 웹브폼 업데이트 feature 는 선택적이다. 이 feature 를 포함하여 제품을 생성한 그림 8(a)과 포함하지 않고 생성한 그림 8(b)에 제품 실행화면을 캡처하여 나타내었다.

요구사항 중 feature 모델에 대응되는 기능이 없으면 새로이 개발하여 제품에 포함하고, 새로 개발한 컴포넌트는 앞서 모델링한 feature 모델에 포함하고 공용화 가능한 형태의 코드로 제품라인을 확장한다. 이처럼 기술의 발전과 함께 생겨날 수밖에 없는 새로운 요구사항에 유연하게 대응할 수 있도록 반응식 (Reactive) 방식을 통해 제품라인 모델을 확장할 예정이다. 이러한 과정을 통해 효율적이고 체계적으로 제품라인 모델을 관리하고 제품 개발을 가능하게 할 것이다.

IV. Conclusions

본 논문에서 우리는 공용 데이터링크 소프트웨어의 기능과 구조, 그리고 소프트웨어 제품라인 공학의 절차와 방법에 대해 서론과 관련 연구에서 알아보았고, 데이터링크 소프트웨어 제품들에 소프트웨어 제품라인 공학을 적용하여 feature 모델을 구성하는 도메인 활동을 하였다. 이미 개발된 도메인 소프트웨어 프로젝트 패밀리의 feature를 개발자들의 평가지표와 인터뷰를 통해 선정하고 공통 feature, 특일 feature, 선택 feature로 분류하여 feature 모델을 정제하였다. 이후의 활동으로 다음과 같은 과정을 수행하여야 한다. feature 내의 컴포넌트 타입 분류하고 컴포넌트를 검색, 추출해 낼 수 있는 방법론 모색, 그리고 추출한 컴포넌트들을 기반으로 하여 자산화하여 사용할 컴포넌트 제품을 만들어 내는 재공학 방법을 연구하고 본 논문

에서 적용한 방법론인 FORM 을 지원하는 도구를 사용하여 공학 프로세스를 다시 체계적으로 진행할 예정이다. 또한, 이후 새로운 프로젝트를 개발 시에 해야 할 어플리케이션 활동의 방법에 대해서도 계획을 세웠다. 데이터링크 소프트웨어의 특성상 기존의 제품들이 제공하는 기능들 중 다수 기능이 중복된다. 이에 대하여 제품라인 구축은 유사 기능에 대한 개발시간을 감소하게 할 수 있어 개발원가의 절감과 인적 자원의 효율적 사용이라는 효과를 기대할 수 있다. 그리고 검증된 컴포넌트를 재사용함으로써 품질에 대한 신뢰도 역시 확보할 수 있다. 또한, 제품라인화 과정에 참여하지 않은 개발자나 신규 개발자도 데이터링크 소프트웨어의 전체적 기능 구조에 대한 빠른 이해를 도울 수 있다. 다만, 이러한 제품라인 공학을 적용한 경험이 없는 개발자들이 이를 적용하려고 하면 제품라인 모델링에 참여했던 전문가가 반드시 해당 프로젝트에 참여해야 개발 중 실수와 개선에 들이는 비용을 줄일 수 있을 것이다. 제품라인 개발에 초기 비용이 많이 들지만 성공적으로 모델이 완성된다면 생산성을 향상할 수 있다. 본 연구와 같은 접근 방식을 통해 지속해서 다른 다양한 국방 분야의 소프트웨어 제품 개발에 소프트웨어 제품라인 공학을 적용하여 개발 비용을 절감할 방법을 연구할 것이다.

REFERENCES

- [1] Joo-Yeon Kim, "RF Transceiver Design and Implementation for Common Data Link," Journal of IKEEE, Vol. 19, No. 3, pp. 371~377, Sep, 2015.
- [2] Jongsun Um, Byung-Oh An, "An Efficient Dynamic Workload Balancing Strategy," Korea Defense Issue & Analysis, Vol. 1404, pp. 1-11, Apr. 2012.
- [3] Mi-Young Ahn and Hyun-Gi Min, "A Case Study: Applying the Product Line Engineering to NEXCORE Code Inspector," Proceedings of the Korean Information Science Society, pp. 529~531, Jun, 2013.
- [4] Se-Hyeon Jo, Hack-Joon Kim and Cha-Youn Lim, "Study for Common Control Datalink Terminal Structure Design between UAV Systems," Proceedings of the IEEK Conference, pp. 334~337, Nov, 2014.
- [5] Seung-Mi Song, "Study on DLI(Data Link Interface) Protocol based on STANAG 4586," Proceedings of Symposium of the Korean Institute of communications and Information Sciences, pp. 834~835, Nov, 2013
- [6] Hyunsik Choi, Hyesun Lee, YoonHo Cho and Kyo Chul Kang, "Legacy System-Based Software Product Line Engineering: A Case Study on Cable Set-Top Box Software," Journal of KISS : Software and Applications, Vol. 36, No. 7, pp. 539~547, Jul, 2009.

- [7] Sehoon Kim and Jeong-Ah Kim, "Consistency Checking Rules of Variability between Feature Model and Elements in Software Product Lines," Proceedings of KIIT Summer Conference, , pp. 519~524, Nov, 2012.
- [8] Kyungmo Yang, YoonHo Jo and Kyo Chul Kang, "Modeling FORM Architectures Based on UML 2.0 Profiling," Journal of KISS : Software and Applications, Vol. 36, No. 6, pp. 431~442, Jun, 2009.
- [9] Jaejoon Lee and Kyo-Chul Kang, "Product Line Software Development Process," Communications of the Korean Institute of Information Scientists and Engineers, Vol. 20, No. 3, pp. 23~30, Mar, 2002.
- [10] Sungbae Cho, Hyesun Lee and Kyo Chul Kang, "Asset Component Identification and Re-engineering Method for an Extractive Software Product Line Engineering," Journal of KISS : Software and Applications, Vol. 39, No. 10, pp. 775~786, Oct, 2012.

Authors



Jin-woo Kim received the B.S. degrees in Computer Engineering from Kwangwoon University, Korea, in 2010. Jin woo Kim is currently a engineer in Hanwha systems. He is interested in tactical data links, Link-16, Link-K.



Woo-Sin Lee received the B.S., M.S. and Ph.D. degrees in Computer Engineering from Kwangwoon University, Korea, in 2001, 2003 and 2007, respectively Dr. Lee is currently a chief engineer in Hanwha Systems. He is interested in data links,

tactical networks.



Hack-Joon Kim received the B.S. degree in Computer Engineering from Hongik University, Korea, in 2004. He is currently a senior engineer in Hanwha systems and also a M.S. graduate student in Defense Fusion Engineering, Yonsei University,

Korea. He is interested in Common/Tactical data links, UAS and machine learning.



So-Yeon Jin received the B.S. degree in Computer Engineering from Chonbuk National University, Korea, in 2003. So Yeon Jin is currently a senior engineer in Hanwha systems. She is interested in data links, machine learning, military

communications, unmanned systems.



Se-Hyeon Jo received the B.S. degree in Computer Science and Engineering from Hanyang University, Korea, in 2010. In 2010, he joined Hanwha Systems Co., Republic of Korea, and he is currently an engineer. He is interested in datalink and

deep learning.