

ADD 3.0 기반 전자전 소프트웨어의 계층적 아키텍처 연구

이정우^{*,1)} · 김수진¹⁾ · 허진혁¹⁾ · 장효정¹⁾ · 한진우¹⁾¹⁾LIG넥스원(주) 전자전연구소

Research on the Layered Architecture for Electronic Warfare System Software based on Attribute-Driven Design 3.0

Jeongwoo Lee^{*,1)} · Sujin Kim¹⁾ · Jinhyuk Heo¹⁾ · Hyejeong Jang¹⁾ · Jinwoo Han¹⁾¹⁾Electronic Warfare R&D Center, LIG Nex1 Co., Ltd., Korea

(Received 10 October 2023 / Revised 2 January 2024 / Accepted 19 January 2024)

Abstract

Development of electronic warfare(EW) software has become increasingly challenging due to stricter regulations, shorter development cycles, and increased reliability testing. Consequently, software development often proceeds without proper architecture design, which can lead to missing critical quality requirements and potential system redesigns. In this study, we propose using Attribute-Driven Design(ADD) 3.0 to design software architecture specifically tailored for EW systems, enabling a more systematic approach to address quality requirements. The paper presents an overview of EW software and ADD 3.0 methodology, followed by an analysis of the architecture design results using static and dynamic views. The paper concludes by discussing the effectiveness of the proposed architecture design.

Key Words : Electronic Warfare(전자전), Software Architecture(소프트웨어 아키텍처), Quality Attributes(품질 속성), Radar Warning Receiver(레이다 경고 수신기)

1. 서론

전자전 시스템은 적의 지휘, 통제, 통신 및 전자 무기체계의 기능을 마비 또는 무력화시키고, 적의 전자전 활동으로부터 아군의 지휘, 통제, 통신 전자 무기체계를 보호하는 제반 군사 활동을 말한다. 이 중 전

자전 소프트웨어는 전자전 시스템의 기능, 성능, 유연성 및 확장성을 결정짓는 주요소이며, 시스템의 크기와 복잡성이 증가함에 따라 소프트웨어의 신뢰성이 점차 중요해지고 있다. 반면, 국방 소프트웨어의 개발 기간 및 비용은 점차 축소되고, 신뢰성시험 종류의 증가 등 강화된 규정이 적용됨에 따라 개발 난이도는 기존보다 증가하는 추세이다.

이와 같은 상황에 유연하게 대처하기 위해서는 신뢰성, 재사용성 등 소프트웨어 품질을 보장할 수 있는

* Corresponding author, E-mail: jeongwoo.lee@lignex1.com
Copyright © The Korea Institute of Military Science and Technology

소프트웨어 아키텍처의 설계가 선행되어야 한다. 하지만 국방 소프트웨어 개발 프로세스에서는 소프트웨어 아키텍처 설계 단계를 엄격히 규제하고 있지 않아, 이를 배제한 채 개발이 이루어지는 경우가 많다. 이는 소프트웨어의 중요한 기능 요구사항 및 품질 속성의 누락을 발생시킬 수 있고, 잘못된 설계로 인한 시스템 재설계로 이어질 수 있다.

한편 다수 분야에서 소프트웨어 아키텍처 설계는 연구된 바 있으나 전자전 분야에 대한 아키텍처에 대한 선행연구는 없다. 따라서 본 연구에서는 Attribute-Driven Design(ADD) 3.0을 기반으로 전자전 시스템에 특화된 소프트웨어 아키텍처를 설계하여 이전보다 기능 요구사항 및 품질 속성에 체계적으로 접근하고, 관리하고자 한다. 이를 통해 재사용성, 유지보수성, 신뢰성 등 소프트웨어의 품질을 결정짓는 주요 속성들이 보장된 소프트웨어를 설계하고, 축소되는 개발기간 및 비용, 강화된 규정에도 군 무기체계가 요구하는 주요 기능 요구사항 및 품질 속성을 만족하는 소프트웨어를 개발할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 전자전 소프트웨어의 개념과 ADD 3.0 설계 방법론에 대해 설명하고, 3장에서는 ADD 설계 방법을 적용한 아키텍처 설계 결과에 대해서 분석한다. 해당 과정에서 아키텍처 요인을 식별하고 이를 반영한 설계 결과에 대해 정적 뷰 및 동적 뷰로 나타내며 근거를 제시한다. 4장에서는 본 아키텍처 설계에 대한 효과에 대해 언급하며 마무리한다.

2. 전자전 소프트웨어 아키텍처 개요

2.1 전자전 소프트웨어 개념

전자전 소프트웨어는 전자전 작전에 사용되는 소프트웨어 기술을 포함하고, 크게 전자전 지원, 전자 공격, 전자 보호의 3가지 하위 분야로 구성된다^[1]. 이러한 전자전 작전을 지원하기 위해 전자전 소프트웨어는 전자 정보 수집, 분석, 신호 감지, 주파수 스펙트럼 분석, 통신 및 레이더 시스템의 신호 방해 등의 다양한 소프트웨어 기능과 기술을 포함한다.

특히 전자전 소프트웨어의 주요 역할은 적의 전자 시스템을 조기에 탐지 및 식별하여 상황에 적절한 대응을 하는 것이다. 이를 위해 적의 통신망이나 레이더 시스템을 방해하는 등의 전자전 작전을 수행하고, 작

전 지휘관이 적절한 전략을 수립할 수 있도록 돕는다. 또한, 통합 기능 시험과 케이블 점검 등의 고장 탐구를 통해 장비의 정비를 지원하고, 특정 알고리즘을 이용하여 전자 방어 체계에 적용하는 등 전자적 지원 및 전자 보호의 역할도 수행한다.

이와 같이, 현대의 전자전 소프트웨어는 다양한 분야에서 다양한 플랫폼에 탑재되어 역할을 수행한다. 전자전 소프트웨어는 정해진 시간 내에 반드시 작업을 수행하도록 엄격하게 관리하는 실시간성과 안정적인 동작을 보장하는 신뢰성 및 안정성, 빠른 반응시간 등의 품질 속성을 주요 요소로 관리해야 한다. 더 나아가 다양한 플랫폼에서 운용되는 전자전의 특성 상, 각 분야의 다양한 이해 관계자들과의 원활한 소통을 위한 방안이 필요하다.

2.2 ADD 3.0 소프트웨어 설계 방법론 개요

ADD는 생성과 테스트 정확이 적용된 반복적 방법론이다^[2]. 즉, ADD를 통한 아키텍처의 설계 방법론은 설계를 위해 선택된 부분에 대해 각 요구사항을 만족시킬 수 있도록 최상위 수준부터 반복적으로 분할하여 요구사항을 충족시켜 아키텍처를 수립하는 방법이다.

ADD 프로세스는 Fig. 1과 같은 순서로 진행된다. 먼저 ASR(Architecture Significant Requirement)을 식별하고 컨텍스트 서술을 구성한다. 컨텍스트 서술은 설계될 시스템의 경계와 상호작용해야 하는 외부 시스템, 장치, 사용자 및 환경 조건 정보를 포함한다. ASR은 FR(Functional Requirement, 기능 요구사항), QA(Quality Attribute, 품질 속성), Constraint(제약조건)로 분류된다. 그다음, 설계할 시스템의 요소를 선택하여 ASR을 식별한다. 그다음, 각 ASR에 대해 후보 설계 접근 방법을 적용하여 솔루션을 개발한다. 이 단계는 ADD의 핵심이며, 이 과정을 통해 각 요소에 대한 설계를 생성하고 테스트한다. 프로세스 각 단계에서 타 개발자 또는 이해 관계자들과의 소통을 위해 다양한 방식의 뷰를 작성하여 이를 활용한다.



Fig. 1. Overview of software design methodology based on ADD 3.0

충족되지 않은 ASR이 있으면 ASR을 각 상황에 맞게 분산시켜 하위 제약사항을 도출한다. 충족되지 않은 ASR을 현재 설계로 만족시킬 수 없으면 철회 과정을 진행하는데, 이는 중요한 요구가 만족 되지 않았고 이 설계를 더 상세하게 한다 해도 만족시킬 수 없다는 것을 의미하여 설계 재검토로 이어진다.

마지막으로, 모든 ASR의 요구조건이 충족될 때까지 분할하여 해당 과정을 반복하고 모든 ASR이 충족되었다는 것이 명확해지면 ADD 프로세스를 마무리한다.

이와 같이 ADD 3.0을 활용하여 아키텍처를 설계할 경우, 시스템 요소 및 설계 단계 별로 품질 속성을 식별하여 요구사항을 체계적으로 관리하고, 이해당사자가 요구하는 기능 및 성능을 만족하는 아키텍처를 설계할 수 있다. 또한, 여러 방식의 뷰를 활용한 이해관계자와의 소통을 통해 요구사항의 변경에도 유연하게 대처할 수 있다.

3. ADD 3.0을 활용한 전자전 소프트웨어 아키텍처 설계

아키텍처 설계는 시스템의 주요 품질 속성을 달성하기 위해 필수로 수행해야 한다. 아키텍처 설계를 수행함으로써 시스템의 안정성 및 신뢰성과 같은 군 무기체계에 적합한 주요 품질을 식별하고, 이에 집중할 수 있기 때문이다. 또한 아키텍처 설계 산출물들은 향후 시스템에 적용될 수 있는 재사용 가능한 자산이 되고 이는 소프트웨어 조직의 비즈니스 전략의 근거가 된다. 전자전 소프트웨어 아키텍처 설계 시 보다 체계적인 접근을 위해 ADD 3.0의 접근 방법을 사용하며, 아키텍처 요인, 아키텍처 설계 근거 기술 및 작성 양식을 적용한다.

3.1 아키텍처 요인

아키텍처 요인은 설계 목적, 최우선 기능, 품질 속성, 아키텍처 관심사, 제약사항으로 구성된다.

3.1.1 설계 목적

아키텍처의 설계 목적은 다음과 같다^[2].

- 전자전시스템 품질 예측 능력 향상
- 사업 초기 잠재적 문제 식별 가능
- 변경 근거 관리 가능
- 이해당사자 간 의사소통 향상

Table 1. Functional requirements

식별자	기능 요구사항
FR-01	동작 준비
FR-02	전원인가 자체진단시험 수행
FR-03	모드 변경 처리
FR-04	위협정보 관리
FR-05	위협우선순위 계산
FR-06	위협정보 전송
FR-07	위협에 대한 최적대응 선정
FR-08	위협신호 특성 분석
FR-09	위협 경보
FR-10	로그 저장 및 제어
FR-11	신호데이터 저장 및 제어
FR-12	파일 다운로드 및 삭제
FR-13	정보 제공
FR-14	위협신호 식별
FR-15	외부 시스템 연동(Blanking/Look Through 제어 등)

3.1.2 최우선 기능

최우선 기능은 전자전 소프트웨어에서 필요한 공통 기능을 기준으로 결정한다. 이를 위해 다양한 전자전 사업에서 고객과의 협의를 통해 결정하여 작성한 소프트웨어 요구사항 명세서를 분석하였고, 이 중 공통 항목을 식별하였다. Table 1은 식별된 최우선 기능 요구사항 중 일부를 나열하였다.

위 기능 요구사항을 모두 포함하도록 유즈케이스를 작성하였으며 Table 2는 주요 항목을 나열하였다. 각 유즈케이스에는 Fig. 2와 같이 입력, 출력, 해당 출력을 생성하기 위한 처리 순서를 기술한다^[3].

3.1.3 품질 속성

품질 속성은 개발 결과물에 대한 적합성을 판단하는 척도를 제공하는 요구사항으로 아키텍처에 주요한 영향을 미치는 요소이다. 항목 식별 기준은 전자전 소프트웨어의 요구사항 및 미래 전자전 사업을 위한 개선 사항으로 설정하였다. 이를 통해 기존 식별된 품질 속성 중 우선순위에 따라 선정된 5개의 품질 속성을 소프트웨어 아키텍처에 반영하고, Table 3에 기술하였다.

Table 2. Usecases

식별자	유스케이스 이름	관련 요구사항
UC-01	동작 준비	FR-01
UC-02	자체진단시험 수행	FR-02
UC-03	모드 변경	FR-03
UC-04	RWR 운용	FR-04
		FR-05
		FR-06
		FR-07
		FR-08
		FR-09
		FR-14
UC-05	RWR 정보 송신	FR-04
		FR-09
		FR-10
UC-06	로그 저장제어	FR-10
UC-07	로그 다운로드	FR-12
UC-08	신호 데이터 저장제어	FR-11
UC-09	신호데이터 다운로드	FR-12
UC-10	데이터 삭제 처리	FR-12
UC-11	모니터링 정보 전송	FR-13
UC-12	외부 시스템 연동	FR-15

유스케이스 이름: 동작 준비	식별자: UC-1
유스케이스 개요: 시스템에 전원이 인가되면 각 시스템 구성 요소의 초기화, PBIT를 수행하여 동작을 준비한다.	
액터: 운용자	
사전 조건: 운용자로부터 전원을 인가	
주요 이벤트 흐름	
<ol style="list-style-type: none"> 액터는 RWR 시스템의 각 구성품에 전원을 인가한다. 액터는 동작 준비 시간 확인을 위한 타이머를 시작한다. 시스템은 MDF, 시스템설정파일을 로드한다. 시스템은 시스템설정파일에 정의된 바와 같이 초기 정보를 설정한다. 시스템은 시스템 내의 각 LRU 모듈과의 통신을 연결하고, 초기화 결과를 수신한다. 시스템은 UC-2 전원인가 자체진단시험 수행을 시작한다. 시스템은 RWR 초기화, 자체진단시험 점검 결과 및 상태 정보, 준비 상태 이산신호를 전송한다. 액터는 RWR 초기화, 자체진단시험 점검 결과 및 상태 정보, 준비 상태 이산신호를 수신한다. 시스템은 UC-4 RWR 운용을 시작한다. 	
확장	
<ol style="list-style-type: none"> 통신 연결 실패 시 일정 횟수동안 통신연결 요청을 주기적으로 수행한다. 일정 횟수 이상 실패 시 연결 실패 처리, 오류 상황을 기록한다. 절차 6부터 수행한다. 	
사후조건: 시스템이 운용모드로 진입	

Fig. 2. Usecase scenario

Table 3. Quality attributes

식별자	품질 속성	품질 속성 정제	설명	중요도
QA-1	변경 용이성	인터페이스 변경 용이성	인터페이스는 사업 수행 중 변경이 빈번히 발생하기에 소프트웨어의 다른 부분의 수정 없이, 인터페이스를 변경할 수 있어야 한다.	상
QA-2	테스트 용이성	특수 인터페이스	테스트용 특수 인터페이스를 사용하여 시스템의 중요 데이터를 관찰할 수 있다.	중
QA-3	성능	반응 시간	위협신호 수신 후부터 Mission computer에 위협 정보 송신 시까지 x초 이내여야 한다.	상
QA-4	변경 용이성	신호 처리 알고리즘 교체 용이성	다양한 체계의 여러 위협신호에 대한 대응을 위해 신호처리 알고리즘을 교체할 필요가 있으며, 소프트웨어의 다른 부분의 수정 없이 교체가 가능해야 한다.	상
QA-5	변경 용이성	신호 처리 파라미터 변경 용이성	시스템 시험 중 신호처리 결과에 따른 파라미터의 변경을 위해 소프트웨어의 중단 없이 신호처리 파라미터를 변경할 수 있어야 한다.	중

3.1.4 제약사항

제약사항은 프로젝트 설계를 위해 사전에 이미 결정된 내용으로, 개발 시 준수하여야 하는 사항들이다. 특성에 따라 기술적 제약사항과 사업적 제약사항으로 구분하였고, 특성별 제약 사항들을 Table 4, Table 5에 기술하였다.

Table 4. Functional constraints

식별자	항목	설명
CON-1	MC 연동	시스템은 MC(Mission Computer)와 연동한다.
CON-2	CMDS 연동	시스템은 CMDS와 연동한다.
CON-3	Radar 연동	시스템은 Radar와 연동한다.
CON-4	File Transfer Program 연동	시스템은 File Transfer Program과 연동한다.
CON-5	Audio System 연동	시스템은 Audio System과 연동한다.
CON-6	Digital Receiver 연동	시스템은 Digital Receiver와 연동한다.

Table 5. Business constraints

식별자	항목	설명
CON-7	개발 인원 제한	SW 인력은 제한적이다.

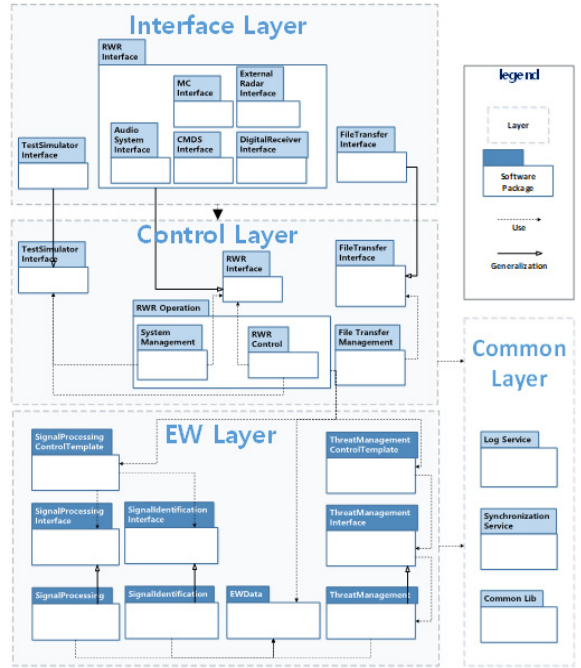


Fig. 3. Static view of electronic warfare software

3.2 아키텍처 설계

아키텍처 설계 결정은 여러 대안 후보를 식별하고 아키텍처 요인을 고려하여 비교 및 결정하는 과정이 필요하다. 이 과정에서 설계 인원 및 이해 관계자 간 소통 및 공유를 위해 뷰를 사용한다.

뷰는 시스템 요소 집합과 요소 사이의 관계를 나타낸다^[6]. 표현하고자 하는 구조에 따라 정적/동적 뷰를 이용하여 아키텍처 설계 결과를 보여주고, 설계 과정에서 고려한 품질 속성 기반의 설계 근거를 설명한다.

3.2.1 정적 뷰

정적 뷰는 환경 안에서 리소스의 고정된 할당을 표현한다. 이를 통해 품질 속성을 만족시키기 위한 각 구성 요소를 식별하고, 책임을 할당할 수 있다. 또한, 구현하고자 하는 소프트웨어의 분할된 각 구성요소를 쉽게 파악할 수 있어, 다양한 설계안의 비교 분석을 용이하게 수행할 수 있는 강점이 있다.

Fig. 3은 계층적 아키텍처를 채택한 전자전 소프트웨어 시스템의 정적 뷰이다. 레이어는 Interface layer, Control Layer, EW Layer, Common Layer, OS Layer로 구성되며, 계층적 아키텍처 선택 및 각 레이어 구성의 설계 근거는 다음과 같다.

3.2.1.1 설계 근거(Layered architecture)

정적 구조 설계 대안은 Layered architecture와 Monolithic & Decomposition architecture를 적용한 대안을 비교 및 분석하는 방식으로 선정했으며, 속성 별 특징은 Table 6과 같다. Layered architecture에서 각 레이어는 독립적인 추상화로 구현되므로 계층적 구조는 논리적인 설계와 구현의 정확도 검증을 가능^[5]하게 하여 테스트가 용이하다. 또한 상위 레이어가 하위 레이어에 의존성을 가질 수 있으나, 하위 레이어는 상위 레이어에 의존성을 가질 수 없는 특징이 있다. 이에 따라 상위 레이어의 변경에 대한 하위 레이어의 영향을 감소^[4]시키기 때문에 연동 레이어의 변경에서 유리한 측면이 있다. 이에 따라 높은 유지보수성의 장점이 있다. 그러나 Monolithic 방식에 비해 상대적으로 수행 속도 및 복잡성 측면의 성능 저하가 발생한다는 단점이 있다^[2]. 외부 시스템 혹은 사용자의 요청으로부터 어떤 기능을 제공하기 위해서 Layered architecture 방식은 일반적으로 모든 하위 레이어를 순서대로 거쳐서 수행하기 때문이다. 이에 반해 Monolithic 방식은 레이어의 구분 없이 기능별로 Decomposition을 수행하고, 기능별 패키지가 서로 의존성을 가질 수 있다. 모든 기능에 직접 접근해서 사용하므로 성능 향상에 유

Table 6. Comparison of static architectural designs

품질 속성	Layered architecture	Monolithic & decomposition architecture
QA-1	상위 레이어 및 인터페이스 변경에 따른 하위 레이어의 영향 감소함 ^[4]	한 모듈의 변경이 다른 모듈의 변경에 영향을 미치는 범위를 제어하기 어려움
QA-3	레이어의 추가는 시스템 복잡성을 증가시키고 성능 저하를 야기함 ^[2]	관련 모듈을 바로 사용할 수 있으므로 반응 시간 확보에 유리
QA-4 QA-5	레이어의 독립적인 추상화로 인해 변경 용이성 향상 ^[5]	기능 간 강한 의존성으로 변화에 대한 대응 어려움 ^[8]

리하지만, 변경 범위의 제어가 어렵기 때문에 한 패키지의 변경이 여러 패키지에 영향을 끼칠 수 있다는 것을 알 수 있다.

따라서 전자전 소프트웨어의 아키텍처는 다양한 위협신호에 대응하기 위한 알고리즘을 수시로 교체하거나 관련 파라미터를 변경하기에 유리하며, 다양한 장비와의 연동을 위한 인터페이스를 독립적으로 교체하기에 강점을 갖는 Layered architecture를 적용하는 것이 적절하다. 시스템의 복잡성이나 성능 저하의 문제는 동적 구조 설계 시 보완하여 해결할 수 있다.

3.2.1.2 설계 근거(Interface layer)

Interface layer는 전자전 소프트웨어의 연동을 위한 계층이다. QA-1 인터페이스 변경 용이성을 위해서 연동 인터페이스를 도입했으며 이를 통해 연동 레이어의 구체 패키지들의 변경이 제어 레이어에 주는 영향을 최소화하여 실제 장비가 없는 모의 환경에서 주요 기능 및 성능 시험을 수행할 수 있도록 한다. 해당 레이어는 Test Simulator Interface, File Transfer Interface, RWR Interface로 구성된다. 각 연동의 인터페이스 기술에 따라서 관련 인원에 할당할 근거로 활용한다(CON-7).

QA-2 특수 인터페이스 품질 속성을 위하여 Test Simulator는 시험용 특수 인터페이스를 두어 모의기와 연동하여 모니터링 기능을 수행하도록 한다. 이를 통

해 소프트웨어 모듈의 운용 중단 없이 테스트용 인터페이스를 사용하여 시험 시간을 단축할 수 있다.

File Transfer Interface는 로그 다운로드, 파일 업로드를 수행하는 프로그램과의 연동을 위한 인터페이스이다.

RWR Interface의 경우 전자전에 특화된 신호처리 관련 기능 수행을 위한 연동 기능을 담당한다. 이를 위해 외부 및 내부 장치 별로 패키지를 식별했다. MC Interface는 전자전 시스템을 제어하는 항공기(또는 함정/헬기/지상 등)와 같은 플랫폼에 탑재된 Mission Computer와의 연동 기능(CON-1)을 수행한다. 또한 위협 대응 수행을 위한 CMDS Interface(CON-2)가 있다. External Radar Interface는 RF 송수신 외부 장비 간 상호운용을 위해 필요한 연동 기능(CON-3)이다. 주로 주파수 간섭 완화 설계와 연관된 전자전 기능 제어(Blanking, Look through, RF 대역 정보 송수신 등)를 수행한다. 이외에도 위협 식별 시 조종사에게 경보를 띄우기 위한 Audio System Interface(CON-5), 신호 분석에 필요한 기초 신호 데이터를 수신하는 Digital Receiver Interface(CON-6)가 있다.

3.2.1.3 설계 근거(Control layer)

Control layer는 기능에 따라 RWR Operation과 File Transfer Management로 분류된다. RWR Operation 내 RWR Control은 위협관리 및 대응, 수신기 제어, 신호 처리 제어를 담당하며, System Management는 RWR 시스템 관리 및 자체 점검을 수행한다. File Transfer Management는 로그 다운로드, 파일 업로드를 수행한다. RWR Operation 패키지는 외부 MC가 송신하는 제어 명령의 처리와 RWR 운용 기능을 분리하기 위해서 패키지를 분리했다.

3.2.1.4 설계 근거(EW layer)

EW layer의 경우 전자전 특화 기능에 대한 계층이다. 신호처리 제어 및 위협관리 기능은 정해진 순서로 진행되는 특징이 있어 템플릿 메소드 패턴을 채택하였다. 여기서 제어 템플릿 패키지는 정해진 제어 순서에 대한 코드 템플릿을 의미한다. 신호 분석 이후 신호 식별 단계로 진행되는데, 신호 분석 단계에서는 신호 데이터를 수집 및 처리하여 신호의 특징을 분석하고, 분석된 결과로 신호 식별 단계에서 라이브러리 내 신호와 매칭한다^[7]. 분석/식별 결과를 토대로 위협 관리 단계에서는 실시간으로 위협 리스트를 관리하고

대응 위협을 선정한다. 또한 QA-4 신호처리 알고리즘 교체 용이성을 위하여 신호분석, 신호식별, 위협관리 인터페이스를 도입하여 알고리즘 변경이나 신규 알고리즘 개발 시, 제공된 인터페이스를 사용하여 해당 패키지를 사용하는 RWR 운용 패키지의 변경을 최소화했다.

3.2.2 동적 뷰

전자전 소프트웨어의 동적 구조는 Fig. 4를 통해 파악할 수 있다. 연관된 외/내부 시스템과의 관계 및 송수신 정보를 나타낸다. 이를 통해 동작 주체를 식별하고, 구성 요소 별 책임을 할당할 수 있다. 또한 구성 요소 별 관계에 따라 책임을 할당하여, 기능 요구사항 및 품질 속성을 만족시키기 위한 전략을 채택하여 성능에 유리한 설계를 진행할 수 있는 장점이 있다.

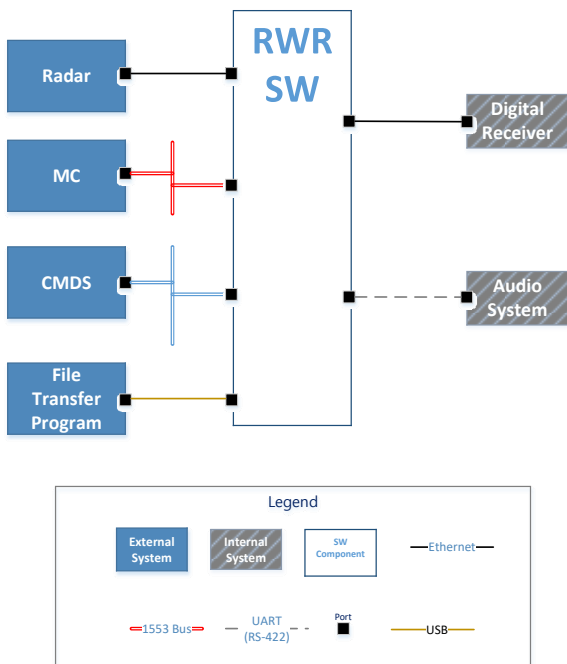


Fig. 4. Dynamic view of electronic warfare software

3.2.2.1 설계 근거(구성 요소 별 책임 할당)

RWR SW는 MC와 연동하여 시스템 제어정보를 수신하고 상태/위협 정보를 송신한다(CON-1). CMDS와 연동하여 위협 대응을 위한 제어정보를 송신하고, 현재 대응 상태정보를 수신한다(CON-2). Radar와 연동하여 아군 레이더와 전자전 운용 정보를 송수신하며 상

호작용한다(CON-3). File Transfer Program과 연동하여 전자전 SW 실행파일, 위협 라이브러리 정보를 수신하고 로그 정보를 송신한다(CON-4). Audio System과 연동하여 정보를 위한 제어 정보를 송신하고 상태를 수신한다(CON-5). Digital Receiver와 연동하여 신호 수집을 위한 제어 명령을 송신하고 수집한 신호 데이터 및 상태 정보를 수신한다(CON-6). 외부 시스템 및 내부 장치와의 연동을 위한 개별 포트를 제공(CON-1~6)하여 변경 용이성(QA-1) 측면에서 우수하다.

3.2.2.2 설계 근거(RWR 운용 태스크)

태스크 설계는 시스템 성능과 사용자 경험에 직접적인 영향을 준다. 효과적인 태스크 설계는 작업을 최적화하고 병목 현상을 방지하여 응답 시간을 개선한다. 태스크 설계와 관련된 전략 중 하나로 병행성 도입(Introduce Concurrency)이 있다. 병행성은 여러 태스크를 동시에 실행하여 시스템의 전체 응답시간을 단축하는 방법을 의미하며, 이러한 병행성 전략은 사용자에게 빠르고 반응성 있는 시스템을 경험하도록 한다.

Control Layer의 RWR Operation 및 EW Layer와 연관된 개념인 전자전 특화 기능의 경우 3.2.1.4에서 언급된 것처럼 신호 수집, 신호 처리 그리고 위협 관리 및 대응의 순서로 진행되는데, 신호 환경에 따라 신호 처리 단계에서 시간이 길게 소요되어 병목 현상의 원인이 될 수 있다. 신호가 모호할 경우 레이더 스캔 주기 분석을 추가로 수행하는데, 레이더 스캔 시간의 배수만큼 소요되기 때문이다.

이에 따라 신호 처리 기능과 관련된 태스크에 대해 단일 태스크 처리를 적용한 대안과 다중 태스크 처리를 적용한 대안을 비교하였으며, Table 7에 요약하였다. 적절한 대안을 선택하기 위해 응답 시간 품질을 보완하는 방향으로 설계한다. 단일 태스크 방식의 경우 비교적 설계가 단순하여 이해가 용이하다. 또한 구현 및 디버깅 난이도가 비교적 낮고 메모리/CPU 사용량이 감소한다는 장점이 있으나, 탐색 대역이 많은 RWR 특성상 각 탐색 대역을 1개의 태스크가 모두 처리할 경우 반응시간이 길어 QA-3(반응시간) 조건 만족이 어렵다. 다중 태스크 방식의 경우 3~5개의 태스크를 이용하여 신호처리를 수행한다. Idle 태스크를 관리하는 제어 태스크(Producer-Consumer 구조)를 도입할 수 있고, 신호 수집이 완료되는 시점에 임무가 없는 Idle 태스크를 통해 신호처리 수행이 가능하다.

Table 7. Comparison of task design structures

Architecture driver	Single-task	Multi-task
QA-3	느린 반응시간	빠른 반응시간
CON-7	낮은 구현 난이도	상대적 높은 구현 난이도(다중 Task로 인한 디버깅 난이도 상승 및 Task 우선순위 고려)

이에 따라 다중 태스크 방식을 선택하였으며, Fig. 5와 같이 구조를 설계하였다. 높은 구현 난이도는 디버깅 포인트에 대한 로깅 위치, 프린트 위치, 데이터 정의 등의 상세설계에 반영되도록 하여 보완한다.



Fig. 5. Multi-task structure of RWR software

3.3 아키텍처 설계 평가

지금까지 제시된 기능 요구사항을 바탕으로 여러 아키텍처 요인을 도출하였고, 이를 만족시키기 위한 설계 대안 분석 및 각 설계에 대한 근거를 정적/동적 부를 이용하여 표현하였다. 설계 결과는 품질 속성 중 중요도가 높은 항목을 필수로 만족해야 한다.

우선 QA-1 인터페이스 변경 용이성 및 QA-4 신호 처리 알고리즘 교체용이성의 경우, Layered architecture 방식을 활용하여 계층 간 상호 의존이 발생하지 않도록 설계를 하였다. 그 결과, 신규 개발된 인터페이스

나 알고리즘의 개발 기간을 제외하고, 교체 및 검증에 의한 작업 중단 기간을 3일 이내로 단축시키는 효과를 확인했다.

다음 QA-3 반응시간의 만족 여부를 확인하기 위하여, Table 8과 같이 각 태스크 설계 구조 별, 탐색대역 개수 별 시험 수행 결과를 확인하였다. 최대 반응시간 기준으로, 단일 태스크 수행 대비 다중 태스크 구조에서의 반응시간을 최대 35 %로 단축시키는 효과를 확인하였고, 또한 QA-3 반응시간 품질 속성에서 제시한 시간 이내의 값을 갖는 것을 확인하였다.

이를 통해, 기능 요구사항 및 아키텍처 요인을 만족하고, 다양한 신호 환경에 대응하기 위한 신호처리 알고리즘 개선에 대한 확장성을 확보하며, 지속적인 인터페이스 변경에도 강건한 소프트웨어를 개발할 수 있는 구조를 설계했음을 확인했다.

Table 8. Evaluation of response time performance

설계 대안	탐색대역 N개		탐색대역 2N개	
	최소 반응시간	최대 반응시간	최소 반응시간	최대 반응시간
Single Task	50 ms	600 ms	70 ms	1,200 ms
Multi Task	50 ms	220 ms	70 ms	420 ms

4. 효과 및 결론

본 연구에서는 ADD 3.0을 활용하여 전자전 시스템에 특화된 소프트웨어 아키텍처를 제안했다. 이를 통해 전자전 소프트웨어 개발에 있어서 품질 속성에 체계적으로 접근하여 설계하는 과정을 나타냈다.

소프트웨어 정적 구조 설계 단계에서는 여러 대안을 비교하여 Layered architecture 방식을 선정했다. 이를 통해 상위 레이어 인터페이스 변경에 따른 하위 레이어의 변경 영향성을 감소시켜 체계에 유연하게 적용할 수 있음을 확인하였다. 또한 Layered architecture의 사용으로 인한 성능 저하 문제는 동적 구조 설계 과정에서 태스크의 Introduce concurrency 전술을 사용하여 보완했다.

향후 ADD 3.0 방법론을 기반으로 전자전 소프트웨어 아키텍처의 구조를 정립할 경우, 신뢰성, 재사용성, 변경 용이성 등 소프트웨어 품질에 직결되는 속성을

보장하기에 적합하며, 이해 당사자 간의 유연한 소통을 통한 생산성 향상에 유리하다. 앞으로 다양한 플랫폼에 탑재되어 역할을 수행할 전자전 소프트웨어의 여러 이해당사자와 유연한 소통을 통한 소프트웨어 개발 기간 단축 및 높은 안정성의 소프트웨어 개발을 기대할 수 있다.

후 기

논문 작성에 많은 도움을 주신 전자전연구소 소프트웨어 아키텍트 한진우 팀장님께 진심으로 감사드립니다.

References

- [1] Lee, I., "Radar Engineering and Applications in Electronic Warfare," Daeyoungsa, Inc., pp. 387-389, 2006.
- [2] Kazman, L. B. P. C. R., "Software Architecture in Practice, Third Edition," Addison-Wesley Professional, Inc., pp. 55-324, 2012.
- [3] Robert, C., "Clean Architecture: A Craftman's Guide to Software Structure and Design," Prentice Hall, pp. 200-205, 2019.
- [4] Bachmann, F., Bass, L., Nord, R., "Modifiability Tactics," Software Engineering Institute, 2007.
- [5] Dijkstra, E. W., "The Structure of the "THE"-Multiprogramming System," Communications of the ACM, pp. 341-346, 1968.
- [6] Clements, P., Garlan, D., Little, R., Nord, R., Stafford, J., "Documenting software Architectures: Views and Beyond, 2nd Edition," Addison-Wesley, pp. 22-23, 2011.
- [7] Kim, Y., Yu, K., Han, J., "Research for Radar Signal Classification Model Using Deep Learning Technique," Journal of the Korea Institute of Military Science and Technology, Vol. 22, No. 2, pp. 170-178, 2019.
- [8] ELCHERIANI, N. S., AHMED, N. A. S., "MICRO-SERVICES VS. MONOLITHIC ARCHITECTURES [THE DIFFERENTIAL STRUCTURE BETWEEN TWO ARCHITECTURES]," International Journal of Applied Sciences and Technology, Vol. 4, No. 3, pp. 484-498, 2022.