Research Paper

Journal of Aerospace System Engineering
Vol.14, No.특별호, pp.13-21 (2020)

EISSN 2508-7150
http://dx.doi.org/10.20910/JASE.2020.14.S.13

RTCA DO-178C와 새로운 RESSAC 소프트웨어 인증기술의 비교 분석

이동민[1] · 이동우[2] · 오승준[1] · 권오성[1] · 나종화[1,†]

[1]한국항공대학교
[2]항공전자연구소

Comparative Analysis of the Software Certification: RTCA DO-178C and RESSAC

Dongmin-Lee[1], Dongwoo-Lee[2], Seungjun-Oh[1], Oseong-Kwon[1] and Jongwhoa-Na[1†]

[1]Korea Aerospace University
[2]Aerospace and Aviation Electronics Research

Abstract

RTCA DO-178C is a development guideline to ensure aircraft system airworthiness. However, there is an opinion that the application of DO-178C to the development of UAV of more than MTOW 150 kg is over regulated because the severity of the risk from UAV is lower than that of normal aircraft. To address issue, EASA and FAA have been working on the Re-Engineering and Streamlining the Standards for Avionics Certification(RESSAC) project since 2016 with the goal of establishing a new certification scheme that simplifies existing aircraft certification procedures and standards. This paper analyzes the current DO-178C certification process and presents advantages by comparing and analyzing the new RESSAC certification process, which simplifies processes and outputs in comparing with the DO-178C certification process, while it ensures flight safety of the vehicle.

초 록

RTCA DO-178C는 항공기 시스템의 감항성을 보장하기 위한 소프트웨어 개발 지침서이다. 하지만, DO-178C는 최대이륙중량 150kg 이상의 무인기 인증에 적용하는 것은 무인기로 인한 위험의 심각도(severity)가 보통의 항공기보다 낮기 때문에 과도한 규제라는 의견이 있다. 이러한 문제를 해결하기 위해, EASA와 FAA는 기존 항공기 인증 체계를 간소화 한 새로운 인증 체계 수립을 목표로 항공전자 인증표준 재편 및 간소화(RESSAC: Re-Engineering and Streamlining the Standards for Avionics Certification) 프로젝트를 2016년부터 수행하고 있다. 본 논문에서는 현재 적용되는 DO-178C 인증 프로세스를 분석하였고, DO-178C 인증 프로세스보다 과정 및 산출물을 간소화하면서도 비행 안전성을 확보 할 수 있는 새로운 RESSAC 인증 프로세스를 비교 분석하고 장점을 도출하여 제시하였다.

Key Words : RTCA DO-178(항공용 소프트웨어 개발 지침서), RESSAC(항공전자 인증표준 재편 및 간소화), Certification(인증), System Safety Assessment(시스템 안전성 평가)

1. 서 론

무인기 시장이 확장됨에 따라 무인기에 탑재된 소프트웨어의 안전성이 중요해지고 있다. 민간 무인기 시장은 연평균 37% 이상의 급속도로 성장하고 있으며,

Received: Nov. 02, 2020 Revised: Mar. 03, 2020 Accepted: Mar. 03, 2020
† Corresponding Author
Tel: +82-02-300-0410, E-mail: jwna@kau.ac.kr

이에 따라 무인기 사고도 증가하고 있다[1]. 무인기 사고는 주변 장애물 등과의 접촉사고가 대부분이며, 이러한 문제를 해결하기 위해 주변 장애물 검출 알고리즘, 자율 회피 기능 등 소프트웨어 안전성이 중요하게 다루어지고 있다[2]. 국내 무인기 소프트웨어 개발은 소프트웨어 감항성을 보장받기 위해 항공기 전자장비 소프트웨어 개발 지침서인 DO-178C를 적용하고 있다[3, 4].

RTCA DO-178C 지침서를 최대이륙중량(MTOW) 150 kg 이상의 무인기 또는 4인승 이하의 소형 유인기 개발에 적용하는 것은 과도한 규제라는 의견이 있다. DO-178C 지침서의 개발 프로세스는 FAA 위임인증기술자(DER: Designated Engineering Representative, DER)이 SOI(Stage Of Involvement) 4단계 동안 설계보증수준(DAL: Design Assurance Level)에 따라 최대 22개의 산출물과 71개의 Objective들을 세밀하게 확인하는 과정이다[3]. DO-178C 지침서는 개발과정의 모든 산출물을 엄격하게 검토하기 때문에 안전도는 높지만 비용증가와 기간 연장 문제를 가진다. DO-178C에 따른 개발은 수백 명의 탑승객을 안전하게 이동시키는 여객기 전자 시스템에 적합하다. 그러나 항공안전 위해요인에 의해 유발될 수 있는 피해 또는 치명 정도를 나타내는 지표인 심각도(severity)가 대형 항공기보다 낮은 무인기 또는 소형 유인기 개발에 DO-178C를 적용하는 것은 과도한 규제라는 의견이 제시되고 있다[5].

이러한 문제를 해결하기 위하여 EASA, FAA 등은 항공기 및 전자장비 개발 프로세스 간소화를 목표로 새로운 인증 체계를 수립하기 위한 항공전자 인증 표준 재편 및 간소화(RESSAC: Re-Engineering and Streamlining the Standards for Avionics Certification) 프로젝트를 2016년부터 수행하고 있다[5-7]. RESSAC 프로젝트에서는 감항성 검증을 위해, 여러 연구팀이 다양한 기법에 따른 개발 프로세스를 수립하고 검증하고 있는 중이다.

본 논문에서는 DO-178C와 RESSAC 프로젝트에서 제시하는 개발 프로세스를 비교 분석하였다. DO-178C와 RESSAC 프로세스 간 산출물을 비교표로 정리하였으며, RESSAC 프로젝트의 사례연구인 μXAV의 임무관리시스템(Mission Management System)을 분석하였다.

2. RTCA DO-178C 분석

2.1 DO-178C 개요

DO-178C의 목적은 항공 시스템 및 장비의 소프트웨어에 대한 개발 지침을 제공하는 것이다. DO-178C는 설계보증수준에 따라 달성해야 할 Objective를 정의하며, 이를 충족시키기 위한 활동을 제시하고 있다. 그리고 Objective가 충족됨을 보여주기 위한 데이터 형태의 증거로써 산출물들을 정의해야 한다. 추가로, 소프트웨어 환경에서 요구되는 추가 고려사항과 관련 용어 정의를 명확히 제시되어야 한다.

2.2 DO-178C 라이프사이클

DO-178C 지침서에서 제시하는 라이프사이클(Life Cycle)은 3가지(계획 프로세스, 개발 프로세스, 총괄 프로세스)로 구성되어 있다[8].

첫 번째, 계획 프로세스는 인증 양상(aspect), 개발, 검증, 형상 관리, 품질 보증 계획의 5가지 계획 문서와 요구사항, 설계 및 코드의 3가지 표준 문서를 정의하는 단계이다.

두 번째, 소프트웨어 개발 프로세스는 계획 프로세스에서 정의한 인증 데이터들을 기반으로 소프트웨어를 개발하는 단계이다. 다음은 소프트웨어 개발 프로세스의 4가지 하위 프로세스(요구사항, 설계, 소스 코드, 통합)에 대한 설명이다.

- 요구사항: 시스템 요구사항 및 시스템 안전 요구사항으로부터 소프트웨어 기능 구현을 위한 상위 수준의 요구사항을 구현하는 단계
- 설계: 상위 수준 요구사항으로부터 소스 코드를 구현하기 위한 하위 수준 요구사항 및 소프트웨어 아키텍처를 구현하는 단계
- 소스코드: 하위 수준 요구사항 및 소프트웨어 아키텍처를 기반으로 소스코드를 개발하는 단계
- 통합: 컴파일러 도구를 이용하여 소스코드로부터 목적코드(EOC: Executable Object Code)를 생성하며, 타겟 보드에 로드하여 확인하는 단계

세 번째, 총괄 프로세스는 검증 프로세스, 형상관리 프로세스, 품질 보증 프로세스, 인증 연락 프로세스의 4개의 하위 프로세스로 구성되어 있다.

- 검증 프로세스: 개발 프로세스에서 생성한 소프트웨어 개발 산출물에 대한 검증을 수행하는 프로세스
- 형상관리 프로세스: 결함보고 및 변경과 관련된 활동을 정의하고 통제하는 프로세스
- 품질 보증 프로세스: 전 라이프사이클과 산출물들이 요구사항에 만족하는지 등을 평가하는 프로세스
- 인증 연락 프로세스: 지원자와 인증기관 사이의 소통과 이해를 수립하여 인증 프로세스를 지원하는 프로세스

2.3 DO-178C 산출물

소프트웨어는 고장률 계산을 이용한 안전 평가가 불가능하기 때문에 Level A(catastrophic)부터 Level D(minor)까지 4단계의 설계보증수준에 따라서 프로세스의 활동 및 인증 목표를 엄격하게 확인 및 검증한다. 항공기 소프트웨어는 설계보증수준에 따라 최대 71개의 Objective를 수행하여야 하며, 이에 따른 증거로써 22종의 산출물을 생산해야 한다. Table 1과 Table 2는 설계 보증 수준에 따른 충족해야 할 Objective 수와 산출물이 제시하고 있다.

Table 1 DO-178C Certification Objectives based on Design Assurance Level

No.	Title	Design Assurance Level			
		A	B	C	D
A1	Software Planning	7	7	7	2
A2	Software Development	7	7	7	4
A3	Verification of Outputs of Software Requirements	7	7	6	3
A4	Verification of Outputs of Software Design	13	13	9	1
A5	Verification of Outputs of Coding&Integration	9	9	8	1
A6	Testing of Outputs of Integration	5	5	5	3
A7	Verification of Verification Process Results	9	7	6	1
A8	Software Configuration Management	6	6	6	6
A9	Software Quality Assurance	5	5	5	2
A10	Certification Liaison	3	3	3	3
Total		71	69	62	26

Table 2 Output Data according to DO-178C Process

Process	Output Data
Planning	Plan for Software Aspects of Certification
	Software Development Plan
	Software Verification Plan
	Software Configuration Management Plan
	Software Quality Assurance Plan
	Software Requirements Standard
	Software Design Standard
	Software Coding Standard
Development	Software Requirements Data
	Software Design Data
	Source Code
	Executable Object Code
	Parameter Data Item File
	Trace Data
Integral	Software Verification Cases and Procedures
	Software Verification Result
	Software Life Cycle Environment Configuration Index
	Software Configuration Index
	Software Configuration Management Records
	Problem Reports
	Software Quality Assurance Records
	Software Accomplishment Summary

3. FAA RESSAC 분석

3.1 RESSAC 개요

RESSAC(Re-Engineering and Streamlining the Standards for Avionics Certification)은 무인기 개발에 적용되는 기존 프로세스의 문제점인 높은 비용 및 개발 기간을 낮추기 위한 목적으로 EASA와 FAA가 2016년부터 추진한 새로운 인증체계이다.

RESSAC 프로젝트에서는 세 가지 핵심 특성(Overarching Property)을 정의하고 있으며, 이 특성을 만족하는 효율적인 개발 프로세스 정의 및 사례연구를 수행 중에 있다. 다음 절에는 세 가지 Overarching Property와 RESSAC 라이프사이클 및 사례연구를 제시하였다.

3.2 RESSAC 인증 프로세스

항공전자 인증표준 재편 및 간소화 프로젝트(RESSAC)에서 제시하는 Overarching Property는 기

존 개발 프로세스의 감항성을 유지하기 위한 최소한의 속성 집합을 의미한다. 이러한 Overarching Property는 다음과 같이 3가지로 정의할 수 있으며, 서로 다른 개발 영역에 의해 생성되는 장벽을 제거할 수 있다.

① Intents: 목표 시스템 구현에 필요한 요구사항(DIB: Defined Intended Behavior)들이 소프트웨어 개발문서들에 정확하고 완전하게 설명 여부 보장하는 특성

② Correctness: 구현된 시스템이 운용 조건 하에서 요구사항(DIB)대로 올바르게 구현 여부 보장하는 특성

③ Acceptability: 구현된 시스템이 요구사항에서 설명되지 않은 동작이 있을 경우 그 동작이 시스템의 안전요구사항을 위반하지 않음을 보장하는 특성

RESSAC 프로젝트는 4단계의 라이프사이클로 계획 프로세스, 시스템 정의 프로세스, 소프트웨어 개발 프로세스, 통합 프로세스를 제시하고 있다.

1단계: 계획 프로세스

계획 프로세스는 첫 번째 라이프사이클로 시스템 개발에 대한 전반적인 계획안과 가능한 표준 및 지침을 정의하고 있다. 계획안은 생명 주기, 활동, I/O, 구현 방법, 환경, 책임 등이 정의해야 한다. RESSAC 프로젝트는 시스템과 소프트웨어를 하나의 프로세스로 통합한 개발 프로세스를 제시해야 한다. ARP4754A의 개발 계획 프로세스에 DO-178C의 개발 계획 프로세스가 통합되어 수행하는 것으로 볼 수 있다.

2단계: 시스템 정의 프로세스

시스템 정의 프로세스는 시스템을 만족해야 하는 모든 기능, 제약 조건, 성능을 식별하고 이를 기반으로 검증하기 위한 시스템 시뮬레이션 절차 수행 및 분석을 수행해야 한다.

계획 프로세스에서 정의된 계획안을 기반으로 시스템 운영 사양을 생성한다. 시스템 운영 사양은 시스템 운영 상황, 시스템 임무 정의, 예상 성능 및 예측 가능한 조건과 고장 조건 등을 기록하며 시스템의 임무를 식별하여 각 임무에 따른 임무 시나리오를 정의하여야 한다.

시스템 운영 사양으로부터 시스템 상세 기능 및 유기적 절차를 정의하기 위한 시스템 아키텍처, 시스템 기능 구현 및 사양 정의를 위한 시스템 기능 사양, 구현한 요구사항 및 임무 시나리오 수행 능력 평가를 위한 행동 모델(Behavior Model)을 구현하여야 한다. 수행한 프로세스의 결과물에 대한 검증을 위해, 시스템 아키텍처, 시스템 기능 사양, 행동 모델을 기반으로 시스템 시뮬레이션 환경 구성 및 임무 시나리오를 기반으로 한 행동 모델 시뮬레이션 수행하여야 한다.

이러한 시스템 시뮬레이션은 구현하고자 할 시스템의 요구사항이 완전한지 판단(Intents)하며 구현 시스템이 요구사항대로 실행되는지(Correctness) 검사하고, 안전성에 영향을 주는지 확인(Acceptability)하게 된다. 시스템 시뮬레이션 결과가 부적합할 경우, 시스템 정의 프로세스를 다시 수행하게 되며, 결과가 만족할 경우 소프트웨어 개발 프로세스를 수행하여야 한다. 이러한 활동들은 ARP4761에서 수행하는 안전성 평가와 대응된다.

3단계: 소프트웨어 개발 프로세스

소프트웨어 개발 프로세스는 시스템 아키텍처 및 기능 사양을 기반으로 하는 소프트웨어를 개발하는 단계이다. 소프트웨어 개발 프로세스는 시스템 시뮬레이션으로 검증된 시스템 요구사항으로부터 소프트웨어 아키텍처를 개발하고 소프트웨어 모델을 구현 및 소프트웨어 모델 시뮬레이션을 진행하여 검증하여야 한다. 소프트웨어 아키텍처는 시스템 내의 소프트웨어 기능을 완전하고 일관되게 구현할 수 있도록 응용 소프트웨어 요소 및 해당 인터페이스(데이터 및 제어 커플링) 등을 정의하여야 한다.

소프트웨어 모델은 소프트웨어 아키텍처 및 시스템 기능 사양을 기반으로 모델 기반의 설계를 진행하여야 한다. 소프트웨어 모델은 DO-331 기반의 모델 검증 후, 소프트웨어 모델 시뮬레이션 진행하며, 소프트웨어 모델 시뮬레이션은 시스템 시뮬레이션의 소프트웨어 부분을 소프트웨어 모델로 변경하여 시험을 진행하여야 한다. 이러한 절차가 성공적으로 완수되면, TQL-1에서 정의된 DO-330/ED-215 인증 받은 코드 자동 생성 도구를 이용하여 코드를 자동 생성해야 한다.

4단계: 총괄 프로세스

총괄 프로세스는 소스코드 및 라이브러리 등 컴파일에 필요한 요소들을 통합하여 목적코드를 생성 및 검증하는 프로세스이다. 생성된 목적코드는 로드 절차에 따라 타겟 컴퓨터에 적재하여 테스트를 수행하며 시스

템 시뮬레이션 결과 및 소프트웨어 결과와 비교 검증을 수행해야 한다.

소프트웨어 중심의 DO-178C 라이프사이클과 다르게, RESSAC 라이프사이클은 시스템에서부터 소프트웨어까지 이루어져 있다. 임무 시나리오 기반의 시스템 시뮬레이션부터 소프트웨어 모델 시뮬레이션을 거쳐 통합 시뮬레이션까지의 확인 및 검증(validation & verification)을 수행하여 시스템 및 소프트웨어의 안전성을 보장한다. RESSAC 라이프사이클은 시스템과 소프트웨어를 강제로 분리하지 않으며 모델 기반 개발 기법을 이용하여 재사용성을 높인다.

3.3 RESSAC 산출물

RESSAC의 계획, 시스템 정의, 소프트웨어 개발 및 통합 프로세스에서 수행 활동의 산출물을 Table 3에 정의하였다. 먼저, 계획 프로세스 단계에서는 시스템 계획안, 시스템 표준 및 지침서를 정의하는 데이터를 개발하여야 한다. 두 번째 단계인 시스템 정의 프로세스는 시스템의 기능을 구현하고 시스템 운영 사양 및 아키텍처 모델, 기능 명세서 등을 생성해야 한다. 또한 정의된 모델 등을 검증하기 위한 활동으로 시스템 동작 모델, 시스템 시뮬레이션 시험 결과서 및 검토 보고서 등도 생성되어야 한다.

세 번째 단계인 소프트웨어 개발 프로세스는 검증된 시스템 정의 산출물로부터 소프트웨어 개발 활동을 통해 소프트웨어 아키텍처 모델 및 소프트웨어 모델과 소스코드를 개발되어야 한다. 또한 구현한 소프트웨어 모델을 검증하기 위해 모델링 지침 기반 정적 분석과 소프트웨어 모델 시뮬레이션 기반 동적 시험을 수행하여 소프트웨어 모델 보고서 및 소프트웨어 모델 시뮬레이션 시험 결과서 및 검토 보고서를 생성해야 한다.

네 번째 단계인 통합 프로세스는 검증된 소스코드로부터 목적코드를 생성하고 이를 검증하기 위해 타겟 보드에 목적코드를 로드하여 수행하는 통합 시뮬레이션을 진행하고 최종 검토 보고서를 제작하여야 한다.

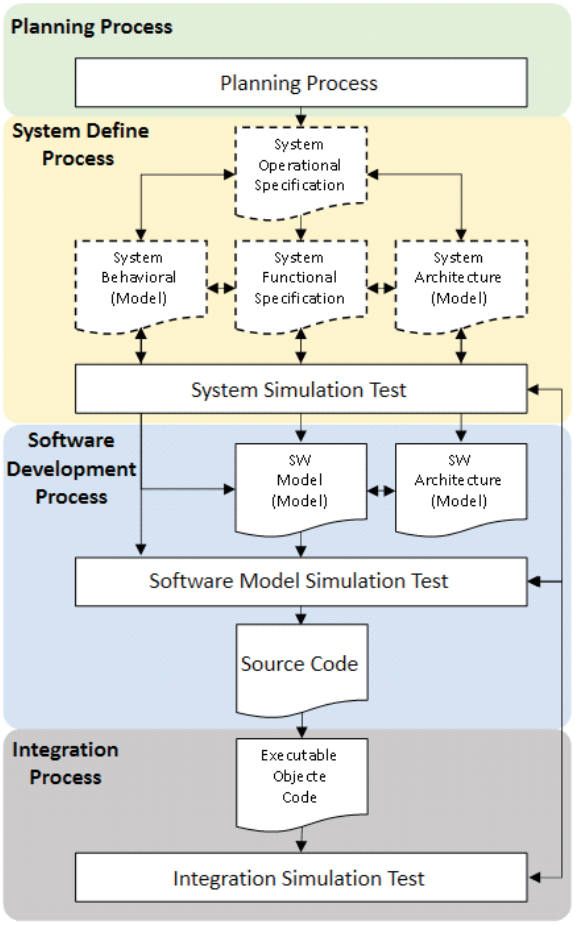

Fig. 1 RESSAC Software Model Process

Table 3 Output Data According to RESSAC Process

Process	Output Data
Planning	Planning Report
System Define	System Operational Specification
	System Operational Review
	System Architecture Model
	Behavior Model
	System Simulation Result
	System Simulation Report
Software Development	Software Architecture Model
	Software Architecture Model Review
	Software Model
	Source Code
	Software Simulation Result
	Software Simulation Report
	Software Development Report
Integration	Executable Object Code
	Integration Simulation Result
	Integration Simulation Report

4. DO-178C와 RESSAC 비교분석

2장에서 분석한 DO-178C와 3장에서 분석한 RESSAC를 비교하여 유사점과 차이점을 분석하였다. 구체적으로는 DO-178C의 A1~A10 Objective를 달성하기 위한 개별 활동들을 RESSAC의 라이프사이클 활동들과 비교하였다. 다음은 정의 및 검증이 진행 중인 형상관리, 품질관리, 인증연락을 제외한 DO-178C의 A1~A7까지의 Objective를 달성하기 위한 프로세스 및 활동을 RESSAC 프로젝트의 활동과 비교한 결과를 Table 4에 제시하였다.

소프트웨어 개발 과정을 중심으로 설명하기 위해 DO-178C의 A2 Objective를 대상으로 한정한다. DO-178C의 A2 Objective는 개발 프로세스에서 다음 네 개의 활동을 수행하여 달성할 수 있다.

1) 상위 수준 요구사항 정의
2) 소프트웨어 아키텍처, 하위 수준 요구사항 정의
3) 소스코드 개발
4) 목적코드 생성 및 타겟 보드 로드

이 DO-178C A2 Objective의 네 가지 활동에 대응하는 RESSAC 활동은 Software Model Process 활동이며 다음과 같이 구성된다.

① 시스템 기능 사양, 시스템 아키텍처, 시스템 행동 모델 정의
② 소프트웨어 아키텍처 모델, 소프트웨어 모델 정의
③ 코드 자동 생성 도구를 이용한 소스코드 개발
④ 목적코드를 생성 및 타겟 보드 로드

RESSAC 활동은 MC/DC 시험 그리고 소스코드와 목적코드 간 추적성 자료를 요구하지 않기 때문에 개발 보증수준 Level A에서 요구하는 Objective는 충족하지 못하였다. 또한, 형상 관리 및 품질 보증과 인증 연락 프로세스는 생략된다. DO-178C A1~A10 프로세스 목표, 활동 및 산출물과 RESSAC 인증의 목표, 활동, 데이터를 비교한 표를 Table 6로 제시하였다.

Table 4 RESSAC Activities for Satisfy DO-178C Process Objectives

No.	DO-178C	RESSAC
A1	Software Planning	Planning Process
A2	Software Development	System Define Software Development
A3	Verification of Outputs of Software Requirements	Scenario Review
A4	Verification of Outputs of Software Design	SW Model Review SW Architecture Review
A5	Verification of Outputs of Coding&Integration	Source Code Review Model Verification
A6	Testing of Outputs of Integration	Simulation Result
A7	Verification of Verification Process Results	Simulation Review

Table 5 RESSAC Activities for Satisfy DO-178C A2 Certification Objectives

Objective	DO-178C	RESSAC
A2.1	High-Level Requirements are developed.	Operation & Function Specification are developed.
A2.2	Derived High-Level Requirements are developed	
A2.3	Software architecture is developed	Software architecture is developed
A2.4	Low-Level Requirements are developed.	Software Model is developed.
A2.5	Derived Low-Level Requirements are developed	
A2.6	Source Code is developed.	
A2.7	Executable Object Code and Parameter Data Item Files are produced and loaded in the target computer.	Executable Object Code is produced and loaded in the target computer.

Table 6 Comparison of the Certification Data at DO-178C and RESSAC

DO-178C		RESSAC	
Process	Output	Output	Process
Planning	PSAC, SDP, SVP, SCMP, SQAP, SRS, SDS, SCS	PP*	Planning
Develop ment	SRD	SOP*, SFS*, SA*, SST*	System Define
	SDD	SM* , SAM*, SMST*	Software Develop ment
	SC, TD	SC*	
	EOC, PDIF	EOC*	Integral
Integral	SVCP, SVR, SECI, SCI, SCMR, PR, SQAR, SAS	IST*	

*PP: Planning Process Plan
*SOP: System Operational Specification
*SFS: System Functional Specification
*SA: System Architecture
*SST: System Simulation Test
*SM: SW Model
*SAM: SW Architecture Model
*SMST: SW Model Simulation Test
*SC: Source Code
*IST: Integration Simulation Test

5. RESSAC 사례 분석

5.1 사례 분석 대상 시스템 구성

RESSAC 프로젝트 중 하나로 유럽과 미국의 실무자들이 Overarching Property를 준수하는 마이크로 UAV(μXAV)를 개발하는 연구를 수행하였다. μXAV는 Fig. 2에서 보는 바와 같이 μXAV는 4개의 주요 서브시스템으로 구성되어 있다.

RESSAC에서는 Overarching Property를 적용한 개발 프로세스를 정의 및 검증하기 위하여 4개의 서브시스템들을 개발할 때 서로 다른 도구 및 기법들을 사용하여 개발하였다. 여기서는 μXAV의 임무관리시스템(MMS: Mission Management System)을 SCADE도구를 이용한 모델 개발 프로세스에 대해 설명하였다.

5.2 μXAV의 임무관리시스템 검증

μXAV에 장착되는 임무관리시스템(MMS: Mission Management System)의 인증 개발 프로세스를 검증 절차로 한정하여 설명한다. RESSAC 인증 개발에서 시스템 정의 프로세스는 시스템 아키텍처 명세서, 기능 명세서, 운영 명세서, 고장모드 요소 명세서 등을 개발하고, 이를 이용하여 행동 모델 및 시스템 시뮬레이션 모델을 개발한다.

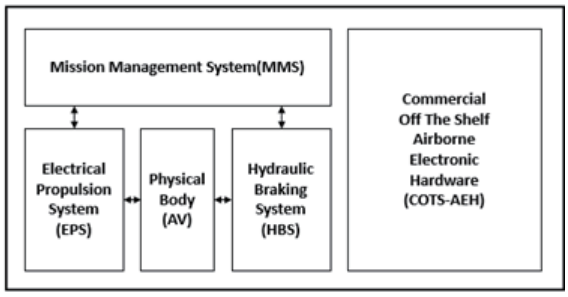

Fig. 2 Subsystem of the μXAV

개발된 모델은 시뮬레이션을 이용하여 세 가지 Overarching Property인 Intent, Correctness, Acceptability 특성을 검증해야 한다. 이 검증 작업은 시스템의 다른 구성요소 들과의 상호작용을 고려해야 하기 때문에 4개의 서브시스템 MMS, EPS, HBS, AV Model 들과 연동되는 통합 시뮬레이션 환경의 구축이 필요하다. Fig. 3과 같이 통합 시뮬레이션 환경은 MMS, EPS, HBS, 그리고 AV에 대해 시스템 수준의 Simulink 모델을 개발하고 이들의 입출력을 연결하여 개발한다.

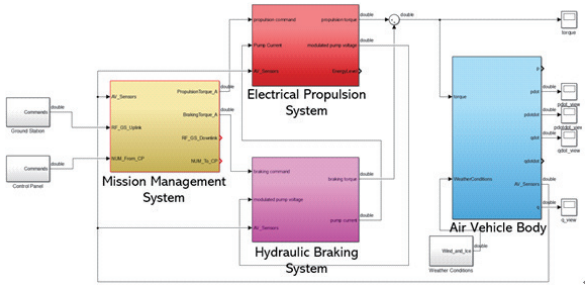

Fig. 3 Simulink Model for the μXAV

RESSAC의 모델 개발은 알고리즘 수준의 모델 제작 단계와 타겟 보드에서 실행 가능한 소스코드의 생성을 위한 모델제작의 2단계로 구성된다. 먼저, 알고리즘 수준의 모델제작은 Simulink 시뮬레이션 환경을 사용할 수 있다. 이 단계에서 모델이 개발되고 시스템 시뮬레이션 테스트를 수행하여 검증이 완료되면, 소프트웨어 아키텍처를 구현하고 타겟용 소스코드의 자동 생성을 위한 MMS의 SCADE 모델을 개발한다. 이와 같이 개

발된 MMS 모델은 모델 개발 프로세스에서 함께 개발된 임무 시나리오에 기반한 테스트 케이스들을 이용하여 소프트웨어 모델 시뮬레이션 테스트를 수행한다.

Simulink 및 SCADE 모델의 개발 및 검증이 완료되면, 최종적으로 타겟 보드에서 소프트웨어를 검증하는 통합 테스트를 수행한다. SCADE의 경우에는 인증 코드 자동생성도구인 SCADE Suite KCG 코드 생성기를 이용하여 소스코드를 제작한다. 이 소스코드를 컴파일하여 획득한 목적코드를 타겟 보드에 로드하고 테스트 케이스들을 이용하여 최종 통합 테스트를 수행한다.

5.3 임무관리시스템 산출물

앞서 설명한 임무 관리 시스템(MMS)의 RESSAC 라이프사이클에서 생성되는 산출물에 대하여 제시한다. RESSAC 라이프사이클은 계획, 시스템 정의, 소프트웨어 개발, 및 통합의 4단계 프로세스에서 제작되는 산출물이 필요하다. 본 논문에서는 지면관계상 시스템 정의 및 소프트웨어 개발 프로세스의 인증데이터를 위주로 기술되어 있다.

Table 7은 RESSAC이 제시한 MMS 개발 프로세스에서 제작된 인증 데이터를 정리한 표이며, 각 문서에 대한 간략한 설명은 다음과 같다.

muXAV Operational Specification은 uXAV의 임무 시나리오 정의를 목표로 작성되었다. 임무 시나리오를 정의하기 위해, 시스템 운용 절차, 구성 요소, 예측 상황 등을 정의한다.

muXAV Architecture Specification은 uXAV의 기능 사양과 이를 구성하는 서브 시스템들(MMS, EPS, HBS)의 기능 사양과 아키텍처 사양을 정의한다. 전반적인 시스템 아키텍처와 시스템을 구성하기 위한 물리 구성 부품, 전기 배선, 통신 네트워크 등을 정리한다.

muXAV Functional Specification은 uXAV의 서브 시스템들(MMS, EPS, HBS)들의 입력 사항을 정의한다. 기능 사양에 맞는 각 서브 시스템들의 입력, 출력 등을 정의하고 각 서브 시스템들간 데이터 흐름 및 동작과 이러한 동작을 하기 위한 가정 등이 기술되어 있다.

MuXAV SysML Architecture Draft는 SysML을 이용하여 MMS SW 아키텍처를 구현한 문서이다. MMS를 구성하기 위한 서브 기능들을 정의하며, SysML 도구를 이용하여 MMS 내부의 기능들 간의 데이터 흐름을 구현한다.

MuXAV SCADE SW model는 SysML에서 구현한 SW 아키텍처를 기반으로 SCADE 도구를 이용하여 MMS SW 모델을 구현 및 설명한 문서로, SysML에서 구현한 MMS의 서브 기능들을 SCADE 도구를 이용하여 논리 수준의 구체적인 소프트웨어 모델을 구현한다.

Simulink 모델은 시스템 수준의 시뮬레이션을 수행하기 위해 uXAV의 서브시스템들을 simulink 모델로 구현한 시뮬레이션이다. 해당 시뮬레이션의 서브시스템들은 기능 수준으로 머물고 있으며, 시스템 아키텍처 및 기능 사양이 올바르게 구현되었는지, 그리고 시스템 안전 요구사항에 위반하는지 등을 검증할 수 있다.

Table 7 Certification Data of Mission Management System

Process	RESSAC	Mission Management System
Planning	Planning Process Plan	-
System Define	System Operational Specification	muXAV Operational Specification
	System Functional Specification	muXAV Functional Specification
	System Architecture	muXAV Architectural Specification
	System Simulation Test	Simulink model
Software Development	SW Architecture Model	muXAV SysML Architecture Draft
	SW Model	MuXAV SCADE SW model
	Source Code	-
	Software Model Simulation Test	-
Integration	EOC	-
	Integration Simulation Test	-

위의 6종의 문서는 RESSAC 프로세스에 따른 개발 과정 중에 제작된 산출물로서 인증 수행 과정의 이해를 돕는 역할을 할 수 있다. 추가로 모델 기반 외의 개발을 하는 개발자들을 위해서, RESSAC은 SPARK,

FPGA, SoC 등 다양한 도구를 이용한 개발 프로세스 및 산출물을 제공한다.

6. 결 론

본 논문은 최대이륙중량(MTOW) 150 kg 이상의 무인기 항전장비의 개발 프로세스를 효율적으로 수행하기 위하여 FAA와 EASA가 추진하는 RESSAC 프로젝트를 분석하였고 DO-178C 지침서와 비교하였다. DO-178C 프로세스의 Objective와 RESSAC 프로젝트의 개발 활동을 DO-178C의 A2 과정에 한정하여 비교하였다. 개발 프로세스 및 산출물의 구체화하기 위해, RESSAC 프로젝트에서 수행한 μXAV의 임무 관리 시스템의 개발 사례를 분석하였다.

현재의 항공전자장비의 개발 프로세스는 RTCA 및 SAE 지침서에 따라서 시스템, 하드웨어, 소프트웨어, 시스템 안전성 평가를 각 영역별로 정의하고 복잡하게 분산되어 있으며 모든 전자 장비에 대하여 모든 높은 수준의 안전성을 요구하는 등의 무인기 개발에 적용하기에는 어려운 문제들이 있다. RESSAC 프로젝트에 따른 개발 프로세스에서는 하나의 개발 과정으로 통합하여 간소화되고, 품목별 특성을 고려한 안전성 평가를 수행할 수 있게 된다. 특히, RESSAC 프로젝트는 최근 항공우주전자시스템 개발의 트렌드로 자리 잡은 모델기반 설계(MBD: Model Based Design) 기법을 개발 프로세스에서 광범위하게 활용함으로써 개발자가 생성 및 관리해야 하는 산출물이 DO-178C의 산출물보다 줄어들게 되었다.

아직 국내에서는 RESSAC 프로젝트를 이용한 연구의 시도가 없으나 항공선진국에서는 다양한 사례 연구들이 진행되고 있다. 소프트웨어 인증이 DO-178C 개발 프로세스에서 RESSAC 개발 프로세스로 전환하려는 소프트웨어 개발 프로세스의 변화는 전자 시스템에 따른 개발도구기술의 발전과 안전성 프로세스의 중용이라는 흐름을 반영한 것으로 다른 항공우주 장비들의 인증기술에도 파급효과가 높을 것으로 예측된다.

References

[1] M. J. O'Donnell, "Investigation of UAS Accidents and Incidents," Federal Aviation Administration, Washington D.C., USA, 30 pages, Sep. 2017.

[2] H. B. Lee, "Guidelines for Unmanned Aircraft System Accident/Incident Investigation and Case Studies", *Proc. of SASE Spring Conference 2018*, Jeju Island, Korea, pp. 179-180, Apr. 2018.

[3] Radio Technical Commission for Aeronautics, "RTCA DO-178C: Software Considerations in Airborne Systems and Equipment Certification," Washington D.C., USA, Dec. 2011.

[4] H. J. Ahn, J. H. Park and S. W. Yoo, "A Study of the Status of UAS Certification System and Airworthiness Standard," *The Korean Society for Aeronautical and Space Sciences*, vol. 42, no. 10, pp. 893-901, Oct. 2014.

[5] D. Brown, (2016, Nov. 28) "An Alternative Approach to DO-178B," Message posted to https://www.slideshare.net/AdaCore/an-alternative-approach-to-do178b, Nov. 2016.

[6] J. Chelini, et al. "Avionics Certification: Back to Fundamentals with Overarching Properties," *Proc. of ERTS 2018*, Toulouse, France, pp. 1-6, Jan. 2018.

[7] M. Graydon, "Retrospectively Documenting SAFE-GUARD's Possession of the Overarching Properties," *Proc. of 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, Portland, USA, pp. 27-28, Jun. 2019.

[8] Y. M. Jun, J. H. Lee and B. H. Kim, "Lessons and learned from Aero-engine Application Software Development according to DO-178/331," *Proc. of SASE Fall Conference 2017*, Busan, Korea, pp. 276-278, Nov. 2017.