Research Paper
Journal of Aerospace System Engineering
Vol.17, No.1, pp.33-41 (2023)
EISSN 2508-7150
http://dx.doi.org/10.20910/JASE.2023.17.1.33

DO-178C 기반 체크리스트를 활용한 무인동력비행장치 소프트웨어 인증 방안

권지훈[1,2]· 이동민[2]· 박경민[2]· 이은희[1]· 임석훈[1]· 최용훈[1]· 나종화[2,†]

[1]항공안전기술원 항공인증본부
[2]한국항공대학교

LUAV Software Certification Method using Checklists based on DO-178C

Ji-Hun Kwon[1], Dong-Min Lee[2], Kyung-Min Park[2], Eun-Hee Lee[1], Sauk-Hoon Im[1], Yong-Hun Choi[1] and Jong-Whoa Na[2,†]

[1]Aviation Certification Division of Korea Institute of Aviation Safety Technology
[2]Korea Aerospace University

Abstract

As seen in the case of the Boeing 737 Max accident, the proportion of aircraft software is rapidly increasing. However, it is vulnerable to safety issues. In case of domestic aircraft software, to operate a Light Unmanned Aerial Vehicle (LUAV) less than an empty weight of 150 kg, safety certification is required for an Ultra-Light Vehicle (ULV). However, software certification procedure is not included. Since the use of LUAVs has increased recently, software verification is required. This paper proposed a checklist of LUAV software that could be applied to LUAV referring DO-178C, an aviation software certification standard. A case study of applying the proposed checklist to the Model-based Development-based Helicopter Flight Control Computer (FCC) project currently used by domestic and foreign advanced companies and institutions was conducted.

초 록

보잉 737 맥스 사고사례에서 볼 수 있듯이 항공기 소프트웨어는 비중이 급속도로 증가하고 있으나 안전문제에서 취약한 것이 드러났다. 국내의 경우, 자체중량 150kg 이하 무인동력비행장치를 운용하려면 초경량비행장치 안전성인증이 요구되지만, 소프트웨어 인증 절차는 포함되지 않는다. 다만, 최근 무인동력비행장치의 활용이 증대됨에 따라 소프트웨어 검증이 요구되는 추세이다. 본 논문은 항공 소프트웨어 인증 규격인 DO-178C를 참조하여 무인동력비행장치에 적용할 수 있는 무인동력비행장치 소프트웨어 체크리스트를 제안하였다. 국내외 선진기업 및 기관에서 활용 중인 모델기반개발(Model-based Development) 기반의 헬리콥터 비행 제어 컴퓨터(FCC) 프로젝트에 제안된 체크리스트를 적용하는 사례연구를 수행하였다.

Key Words : : LUAV(무인동력비행장치), Software(소프트웨어), DO-178C, Checklists(점검표), Model-based Development(모델기반 개발), Certification(인증)

1. 서 론

무인기 산업의 발전으로 무인동력비행장치 신고 건수가 증가하고 있으며, 무인동력비행장치의 안전성 인증 불합격 및 사고율도 같이 증가하고 있다[1,2]. 특히, 사고 원인으로 조작 실수를 제외하면 GPS 수신불량, 시동불량 등 시스템 문제이며, 이 가운데 소프트웨어 문제가 큰 비중을 차지한다[3].

보잉 737 MAX 사고사례에서도 알 수 있듯이 항공용 소프트웨어는 비중이 급속도로 증가하고 있으나 결함 발생 시 치명적 사고로 이어질 수 있다[4]. 국내에서는 자체중량 150kg 이하 무인동력비행장치를 운용하려면 초경량비행장치 안전성인증이 요구되지만[5], 안전점검이 필요한 소프트웨어 인증 절차는 포함되지

Received: Oct. 25, 2022 Revised: Jan. 10, 2023 Accepted: Jan. 12, 2023
† Corresponding Author
Tel: +82-02-300-0410, E-mail: jwna@kau.ac.kr

않는다. 다만, 최근 무인동력비행장치의 활용이 증대됨에 따라 소프트웨어에 대한 검증이 요구되는 추세이다.

항공용 소프트웨어 인증 방안으로 DO-178C (Software Consideration in Airborne System and Equipment Certification) 지침이 사용되고 있으나, 인증에 소요되는 비용 및 기간이 과다하여 무인동력비행장치 개발업체와 같은 중소기업은 이에 대한 적용이 어려울 수 있다. 이런 문제를 해결하기 위해서 DO-178C 체크리스트를 사용하는 사례가 보고되고 있다[6-8]. 체크리스트를 활용한 점검이라도 개발업체가 감당해야 하는 많은 업무는 그대로 남아있다.

본 연구는 체크리스트 기법과 모델기반개발 방법을 통합하여 개발 및 점검 업무를 동시에 효율적으로 수행하는 방법을 제안하였다. 항공 소프트웨어 인증 규격인 DO-178C를 이용하여 개발이 완료된 무인동력비행장치의 소프트웨어의 검증 또는 인증 업무에 활용할 수 있는 체크리스트를 제시하였다. 또한 사례연구로서 헬리콥터 FCC(Flight Control Computer) 예제 프로젝트를 대상으로 체크리스트와 모델기반개발 (Model-Based Development) 적용하여 장단점을 분석하였다.

2. 동향분석

2.1 해외 무인기 인증 체계

미 연방항공청(FAA, Federal Aviation Administration)에서는 무인기의 최대이륙중량에 따라 인증 규정을 분류하여 적용하고 있다. 최대이륙중량 55lbs(25kg) 미만의 무인기를 sUAS(small UAS)로 정의하고 14 CFR Part 107 인증 규정을 적용한다. 그리고 최대이륙중량 55lbs (25kg)를 초과하는 무인기에 대해서는 규정 49 USC 44807에 따른 면제 또는 유인항공기 수준의 인증을 요구한다.

유럽항공안전청(EASA, European Aviation Safety Agency)은 유럽 내 개별 감항당국과 EUROCONTROL 전문 그룹으로 구성된 JARUS(Joint Authorities for Rulemaking Unmanned Aircraft Systems)를 통해 무인 인증체계 및 기준을 수립하고 있다. JARUS은 무인기에 대한 인증 기준으로 CS-LUAS(Certification Specification for Light Unmanned Aeroplane Systems)와 CS-LURS(Certification Specification for Light Unmanned Rotorcraft Systems)를 발표하였다.

북대서양조합기구(NATO, North Atlantic Treaty Organization)는 표준화 협정(STANAG, Standardization Agreement) 규격서를 발행하며, 무인기의 최대이륙중량 및 형식에 따라 분류하였다. 무인기의 형식은 고정익과 회전익으로 분류하고 150kg의 최대이륙중량을 기준으로 분류하였다. 다음의 Table 1은 앞서 설명한 무인기 인증 기준을 정리하였다[9].

Table 1 Classification of UAS Certification Standard

Issuing Agency/Organization	Standard	Type	MTOW
FAA (Federal Aviation Administration)	Part 107	Fixed, Rotary	<25kg
JARUS (Joint Authorities for Rulemaking Unmanned Aircraft Systems)	CS-LUAS	Fixed	<750kg
	CS-LURS	Rotary	
NATO (North Atlantic Treaty Organization)	STANAG 4671[1)]	Fixed	150 ~ 20,000kg
	STANAG 4703[2)]		<150kg
	STANAG 4702[3)]	Rotary	150 ~ 3,175kg
	STANAG 4746[4)]		<150kg

1) UAS Airworthiness Requirements
2) Light UAS Airworthiness Requirements
3) Rotary Wing UAS Airworthiness Requirements
4) Rotary Wing Light UAS Airworthiness Requirements

2.2 국내 항공기 인증 체계

국내 항공기 인증은 민수용과 군수용에 따라 다른 기준을 적용하고 있다. 민간 항공기는 항공기 기술기준(KAS, Korea Airworthiness Standard)을 적용하며, 군용 항공기는 표준감항인증기준(Standard ACC, Standard Airworthiness Certification Criteria)을 적용하고 있다. 민/군은 각각의 법률에 따라 인증을 수행하고 있다.

이러한 인증 체계에서 항공기 인증 시 소프트웨어는 DO-178C를 적용하여 적합성을 확인한다. 군용 항공 소프트웨어의 인증기준은 표준감항인증기준에서 제시되며, Table 2와 같이 나타낼 수 있다[10].

Table 2 Military UAV Software Certification Criteria

기반문서	적용대상	SW 기준	SW 개발보증
MIL-HDBK-516C	유무인 항공기	SW 안전계획	DO-178C 및 미 국방성 문서
STANAG 4671	고정익 무인기 (MTOW 150kg ~ 20,000kg)	제2권 수용 가능한 검증 방법 1309(b)	DO-178C
STANAG 4703	고정익 무인기 (MTOW 150kg 이하)	UL.31 소프트웨어 개발보증수준	DO-178C

STANAG 4703의 적용범위는 현재 무인동력비행장치의 범위에 속하며 소프트웨어 관련 기준과 적합성검증방법 (MOC, Means of Compliance)은 DO-178C 준수를 나타내고 있다. 결국 민수/군수용 항공 소프트웨어 인증 규격은 DO-178C를 수락 가능한 적합성검증방법으로 제시하고 있다.

2.3 DO-178C 개요

DO-178C/ED-12C는 항공 시스템 및 장비 인증에 대한 소프트웨어 고려사항으로 민간 항공기에 적용되는 시스템 및 소프트웨어 개발을 위한 하나의 수락 가능한 방법으로 인정된다. 이 지침은 항공통신 기술위원회(Radio Technical Commission for Aeronautics, RTCA)와 유럽 민간항공기 전자 기구(European Organization for Civil Aviation Equipment, EUROCAE)가 공동으로 1980년에 제정하였고, 2011년에 DO-178C로 개정되었다[11].

DO-178C 소프트웨어 레벨은 시스템 안전성 평가 절차에 따라 5개(Level A~E)로 분류되며, Table 3과 같이 정리될 수 있다. 각 레벨에 따른 고장조건과 목표(Objective)가 제시된다. 또한, 객관적인 검토를 위해서 3자 검증(독립성 필요 목표)이 필요한 항목이 제시되고 있다.

Table 3 DO-178C Software Level

SW Level	Failure Condition	Number of objectives	Required independence
A	Catastrophic	71	30
B	Hazardous	69	18
C	Major	62	5
D	Minor	26	2
E	No safety effect	0	0

2.4 체크리스트 방법론

소프트웨어 검사(Software Inspection)는 기술적 차원에서 Process, Roles, Products 그리고 Reading Technique 등 4가지로 분류된다. Fig. 1은 소프트웨어 검사에 대한 기술적 차원에 대한 내용이다[12].

이러한 요소 중에서 Reading Technique은 소프트웨어에 대한 검사 기법으로 대표적으로 결함 기반 읽기(DBR, Defect-based Reading), 관점 기반 읽기(PBR, Perspective-based Reading), 체크리스트 기법(Checklist technique) 등 여러가지 기법이 존재한다.

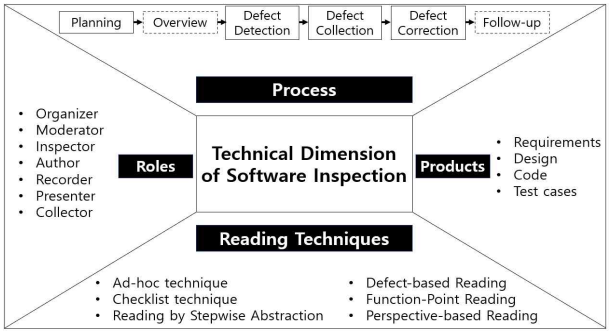


Fig. 1 Technical Dimension of Software Inspection

이러한 기법들 가운데 체크리스트 기법은 다양한 회사에서 표준 소프트웨어 검사 기법으로 많이 활용되고 있으며, 결함 발견에 있어서 가장 인기있는 기술로 간주되고 있다[12-13]. 또한, 소프트웨어 생명주기(Life cycle)의 여러 단계에 적용할 수 있어서 일반적으로 소프트웨어 각 프로세스의 초기 단계에서부터 결함을 발견 및 제거할 수 있다. 이로 인해, 개발 비용의 손실을 줄일 수 있어서 소프트웨어 분야에서도 많이 활용되고 있다[14].

DO-178C 적용에도 체크리스트를 사용하는 사례가 보고되고 있다. 한 연구팀은 DO-178C 개발 목표 및 활동의 점검에 사용할 수 있는 체크리스트를 제작하였다 [6,7]. 이 연구에서 제시된 체크리스트는 항공 시스템 전문가를 대상으로 설문조사를 수행하여 타당성을 주장하였다. 한편 DO-178C 인증 컨설팅 업체도 DO-178C 기준에 따른 체크리스트를 개발하여 사용하고 있다[8]. 그러나 이러한 체크리스트가 점검은 쉽게 만들 수 있지만, 개발업체가 감당해야 하는 많은 업무는 그대로 남아있는 문제점이 있다.

본 연구는 체크리스트를 모델기반개발 도구와 통합하여 개발 업무와 점검 업무를 동시에 쉽게 만드는 방법을 연구하였다.

3. 체크리스트를 활용한 인증방안

3.1 국내 무인동력비행장치의 안전성인증 절차 및 산출물

국내에서 개발된 무인동력비행장치의 안전성인증은 Fig. 2와 같이 최초 형식에 대한 인증절차 및 기존에 인증받은 형식에 대한 인증절차로 분류된다.

무인동력비행장치 최초 형식에 대한 인증 프로세스의 첫 단계는 무인동력비행장치 연구개발을 위해 시험비행을 신청하게 된다. 이후 시험비행 결과 보고서를 제출함으로써 무인동력비행장치의 개발이 완료된 것으로 보며, 안전성인증 절차를 통해 개발된 기체를 감항당국에 의해 최초 검증을 하게 된다.

기존에 인증받은 형식의 인증 프로세스는 시험비행 절차를 생략하며 무인동력비행장치 신고 후 안전성인증을 받을 수 있다. 최초 인증 시 제출되었던 서류를 기반으로 동일하게 제작되었는지 확인하며, 형상변경이 있다면 기록 및 관리가 잘 이뤄지는지 확인한다.

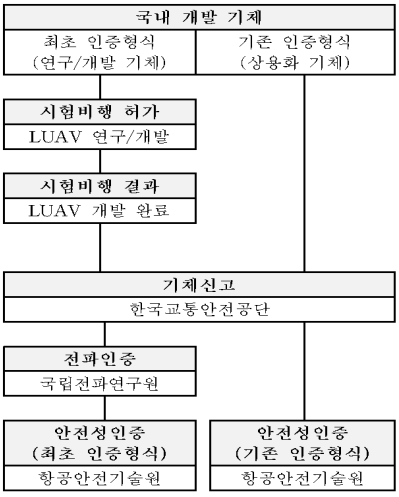


Fig. 2 LUAV Certification Process in Korea Regulation

본 연구에서는 국내 무인동력비행장치 최초형식 인증 절차에 대하여 DO-178C 인증 절차 적용에 따른 고려사항을 아래 Table 4와 같이 나타냈으며, 국내 무인동력비행장치 인증 프로세스에 따라 수행되어야 할 DO-178C 프로세스와 이에 대한 산출물을 정리하였다.

Table 4 Considerations of DO-178C for LUAV's Certification

국내 무인동력비행장치 초도인증 프로세스 (최초형식)		DO-178C 생명주기	
프로세스	산출물	프로세스	산출물
시험비행 허가	시험비행 계획서 등 9종	SW 계획	PSAC등 8종
시험비행 결과 승인	시험비행 결과 및 개발문서	SW 개발	개발문서 4종
안전성인증 (최초인증)	비행장치신고 증명서 등 10종	SW 검증, 형상관리, 품질보증	SVR 등 10종

3.2 개발보증수준 설정

유인항공기의 경우 탑재되는 항공전자장비 소프트웨어는 ARP 4761 내 안전성평가 절차를 통해 위험도가 할당되며, 이에따라 개발보증수준(DAL)을 설정한다. 통상적으로 유인항공기 시스템에서 비행제어장치 소프트웨어는 사고 치명도가 가장 높으므로 DAL A가 적용되고 있다.

하지만 사람이 탑승하지 않는 무인동력비행장치에 대한 안전성평가는 아직 이뤄지지 않고 있으며, 기체 사고 발생에 따른 피해규모가 유인기보다 적다. 군용 회전익 무인기 감항요건인 STANAG 4702에 따르면, 무인기는 사람이나 승객이 탑승하지 않으므로 유인항공기와 다르게 사건의 결과에 탑승자가 아닌 지상 재산이 고려되어야 한다고 명시되어 있다. 이에 따른 유인기 및 무인기의 고장조건 발생확률을 다음의 Table 5와 같이 나타낸다.

Table 5 Comparison of Probability of Failure Conditions for Aircraft and Rotory-wing UAS (STANAG 4702)

Probability / Severity of Failure condition	Classification	
	Aircraft	Rotory-wing UAS (150kg~3,175kg)
Extremely Improbable / Catastrophic	P<10E-9	P≤10E-6
Extremely Remote / Hazardous	10E-9≤P<10E-7	10E-6<P≤10E-5
Remote / Major	10E-7≤P<10E-5	10E-5<P≤10E-4
Probable / Minor	10E-5≤P	10E-4<P≤10E-3
Frequent / No effect	-	10E-3<P

회전익 무인기 인증 기준인 STANAG 4702에서 제일 치명적인 고장조건은 10E-6이며, 이를 유인기 소프트웨어 인증 기준인 DO-178C와 비교하였을 때 고장 확률이 Major인 10E-7≤P<10E-5의 범위 내에 있으므로 본 논문에서는 사례연구로 다룰 헬리콥터 비행제어장치(FCC)에 대한 개발보증수준을 C수준으로 설정하였다.

3.3 체크리스트 개발

본 절에서는 3.2 절에서 항공기와 무인기 고장조건 분석을 통해 설정한 개발보증수준 C를 설정하여 체크리스트를 개발하였다. 개발보증수준 C는 62개의 Objectives를 충족시켜야 하며, 각 Objectives들은 DO-178C Annex A의 Table A-1 부터 10까지 10개의 표로 정리되어 있다. A-1은 계획 프로세스에 대한 Objectives를 나타낸다. A-2는 개발 프로세스에 대한 Objectives를 나타내고 A-3 부터 A-10 까지는 통합 프로세스에서 요구하는 Objectives를 나타내고 있다. Fig. 3은 Annex A의 Table 예를 나타낸다.

Table A-1 Software Planning Process

	Objective		Activity	Applicability by Software Level				Output		Control Category by Software Level			
	Description	Ref	Ref	A	B	C	D	Data Item	Ref	A	B	C	D
1	The activities of the software life cycle processes are defined.	4.1.a	4.2.a 4.2.c 4.2.d 4.2.e 4.2.g 4.2.i 4.2.l 4.3.c	O	O	O	O	PSAC	11.1	①	①	①	①
								SDP	11.2	①	①	②	②
								SVP	11.3	①	①	②	②
								SCM Plan	11.4	①	①	②	②
								SQA Plan	11.5	①	①	②	②
2	The software life cycle(s), including the inter-relationships between the processes, their sequencing, feedback mechanisms, and transition criteria, is defined.	4.1.b	4.2.i 4.3.b	O	O	O		PSAC	11.1	①	①	①	
								SDP	11.2	①	①	②	
								SVP	11.3	①	①	②	
								SCM Plan	11.4	①	①	②	
								SQA Plan	11.5	①	①	②	

Fig. 3 DO-178C ANNEX A Table

DO-178C의 Table A는 소프트웨어 수준에 대한 목표(Objective), 활동(Activity), 산출물(Output), 소프트웨어 수준에 따른 형상관리 수준(Control Category by Software Level) 등으로 구성되어 있다.

본 논문에서 개발한 무인동력비행장치용 체크리스트는 무인기 고장조건 분석을 통한 개발보증수준 C를 적용하였으며, 앞절에서 분석한 무인동력비행장치 인증 프로세스에서 고려될 DO-178C 기반 소프트웨어 인증 체크리스트를 개발하였다. 따라서 무인동력비행장치 소프트웨어 레벨과 프로세스에 맞춘 체크리스트 개발로 점검 효율성을 개선하였다. 또한, 프로세스 체크리스트는 기존 DO-178C Annex A Table 내 Objective의 포괄적인 항목을 참조 내용과 함께 질문 형식으로 구체화하여 점검항목을 상세하게 작성하였다. 산출물 체크리스트는 22종의 산출물에 대하여 DO-178C 11장 소프트웨어 생명주기 데이터에 명시된 내용을 점검항목으로 작성하였다. 그리고 체크리스트의 실효성 개선을 위해 현재 무인동력비행장치 안전성인증에 활용되는 안전성점검표의 형식에 맞춰 전자문서로 제작하여 개발중인 인증시스템에 탑재할 수 있도록 제작하였다. 마지막으로 체크리스트 내 참조를 별도의 탭으로 구성 및 하이퍼링크를 설정하여 점검 항목 간 추적이 쉽게 제작하여 사용성을 개선시켰다. 아래 Table 6은 인동력비행장치 초도인증 프로세스에 따른 무인동력비행장치 소프트웨어 체크리스트 제작에 활용한 템플릿이다.

Table 6 Checklist Template

번호	검토내용	참조	적합여부			근거자료
			적합	부적합	N/A	

이러한 체크리스트 템플릿을 기반으로 인증 프로세스별 점검 체크리스트와 산출물 점검 체크리스트를 각각 제작하였다.

다음은 개발한 체크리스트에 대한 예시로 Fig. 4는 프로세스에 대한 체크리스트 중 설계 프로세스 산출물에 대한 검증에 대한 체크리스트를 나타냈고, Fig. 5는 산출물 점검 체크리스트 중 소프트웨어 인증계획서(PSAC, Plan for Software Aspects of Certification)에 대한 체크리스트를 나타냈다.

소프트웨어 설계 프로세스의 산출물에 대한 검증

순번	점검 항목	참조	적합여부 Yes	No	N/A	근거자료(Evidence)
4-1	하위 수준의 요구 사항이 상위 수준의 요구 사항을 준수합니까? 파생 요구 사항과 그 존재에 대한 설계 기반이 올바르게 정의되어 있습니까?	6.3.2.a 6.3.2.a				
4-2	하위 수준 요구 사항이 정확하고 일관성이 있습니까?	각 하위 수준 요구 사항이 정확하고 모호하지 않습니까? 6.3.2.b 하위 수준 요구 사항이 서로 충돌하지 않습니까? 6.3.2.b				
4-3	하위 수준 요구 사항이 표준을 준수합니까?	소프트웨어 설계 과정에서 소프트웨어 설계 표준을 준수하였습니까? 6.3.2.e 표준에서 벗어난 것은 정당화었습니까? 6.3.2.e				
4-4	하위 수준의 요구 사항은 상위 수준의 요구 사항을 추적할 수 있습니까?	상위 수준 요구 사항 및 파생 요구 사항이 하위 수준 요구 사항으로 개발되었습니까? 6.3.2.f				
4-5	알고리즘이 정확합니까?	특히 불연속 영역에서 제안된 알고리즘의 정확성과 동작이 보장됩니까? 6.3.2.g				
4-6	소프트웨어 아키텍처가 상위 수준 요구 사항과 호환됩니까?	소프트웨어 아키텍처가 상위 수준 요구 사항, 특히 파티션 구성표와 같이 시스템 무결성을 보장하는 기능과 충돌하지 않습니까? 6.3.3.a				
4-7	소프트웨어 아키텍처는 일관성이 있습니까?	소프트웨어 아키텍처의 구성 요소 간에 올바른 관계가 존재합니까? (이 관계는 데이터 흐름과 제어 흐름을 통해 존재합니다.) 6.3.3.b 하위 소프트웨어 수준의 구성 요소에 대한 인터페이스가 있습니까? (상위 소프트웨어 수준 구성 요소가 하위 소프트웨어 수준 구성 요소의 잠재적인 잘못된 입력으로부터 자신을 보호할 수 있는 적절한 보호 메커니즘을 갖추고 있는지도 확인해야 합니다.) 6.3.3.b				
4-8	소프트웨어 아키텍처가 표준을 준수합니까?	소프트웨어 설계 과정에서 소프트웨어 설계 표준을 따랐습니까? 6.3.3.e 복잡성 제한 및 설계 구성 규칙에 대한 편차와 같이 표준에 대한 편차가 정당화됩니까? 6.3.3.e				
4-9	소프트웨어 파티션 무결성이 확인되었습니까?	파티셔닝 위반이 방지됩니까? 6.3.3.f				

Fig. 4 Verification of Outputs of Software Design Checklists(Process)

소프트웨어 인증 계획서(Plan for Software Aspects of Certification) 체크리스트

순번	점검 항목		참조	적합여부 Yes	No	N/A	근거자료(Evidence)
1	시스템 개요	시스템 기능과 하드웨어 및 소프트웨어의 할당, 아키텍처, 사용된 프로세서, 하드웨어/소프트웨어 인터페이스 그리고 안전성 특성을 포함하는 시스템의 개요를 제공하는가?	11.1.a				
2	소프트웨어 개요	제안된 안전성과 파티션 컨셉성에서 강조되는 소프트웨어 기능들을 설명하고 있는가? (예시: 리소스 공유, 중복, 고장 허용 범위, 단일 이벤트 혼선의 완화 그리고 타이밍 및 스케줄링 전략을 포함)	11.1.b				
3	인증 고려사항	소프트웨어의 특성과 관련하여 적합성준수 방법을 포함한 인증 기준의 요약을 제공하는가? 제안된 소프트웨어 레벨이 있으며, 고장 조건에 대한 잠재적인 소프트웨어 역할을 포함하는 시스템 안전성 평가 프로세스에 의해서 제공되는 정당성이 설명되어 있는가?	11.1.c				
4	소프트웨어 생명주기	소프트웨어 생명주기를 정의하고 있는가?	11.1.d				
		해당 소프트웨어 계획에 대한 자세한 정보가 정의된 각 소프트웨어 생명 주기 프로세스에 대하여 요약되어 있는가?	11.1.e				
		요약에는 각 소프트웨어 생명 주기 프로세스의 목표가 어떻게 충족되는지 설명되어 있는가? 요약에는 관련되는 조직, 조직의 의무, 그리고 시스템 생명주기 프로세스와 인증 고섭 프로세스의 책임이 구체화되어 있는가?					
5	소프트웨어 생명주기 데이터	소프트웨어 생명주기 프로세스에 의해 생성되고 컨트롤될 소프트웨어 생명주기 데이터가 구체화되어 있는가? 데이터 상호간 또는 시스템을 정의하는 다른 데이터 간 관계와 인증당국에 제출되어야 할 소프트웨어 생명주기, 데이터의 정태, 데이터가 인증당국에 사용할 수 있도록 만들어지는 방법이 설명되어 있는가?	11.1.f				
6	일정	소프트웨어 생명주기 프로세스 활동에 대해 인증당국이 볼 수 있도록 제공하여 리뷰가 계획될 수 있도록 하기 위한 방법이 설명되어 있는가?	11.1.g				
7	추가 고려사항	인증 프로세스에 영향을 줄 수 있는 특정한 고려사항이 설명되어 있는가? (예시: 적합성준수 대체 방법, 툴인증, 이전에 개발된 소프트웨어, 옵션 선택가능 소프트웨어, 사용자 수정가능 소프트웨어, 비활성화 코드, COTS 소프트웨어, 현장 적재가능 소프트웨어, 파라미터 데이터 아이템, 다중 버전 비유사성 소프트웨어, 그리고 제품 서비스 히스토리 등을 포함)	11.1.h				
8	납품업체 감독	납품 프로세스와 결과물이 승인된 소프트웨어 계획과 표준 이행에 대한 보장 방법이 기술되어 있는가?	11.1.i				

Fig. 5 PSAC Checklists(Output)

다음의 Fig. 6은 무인동력비행장치 인증 절차에 따른 소프트웨어 체크리스트 개발 결과이다.

국내 무인비행장치 인증 프로세스	DO-178C 소프트웨어 생명주기	SW 프로세스 체크리스트	산출물 체크리스트
시험비행허가 절차	SW 계획	7	59
시험비행 결과 승인	SW 개발	7	28
	요구사항 프로세스	2	
	설계 프로세스	3	
	코딩 프로세스	1	
	통합 프로세스	1	
안전성인증(최초 초도인증)	SW 총괄	48	43
	SW 검증	34	
	요구사항 프로세스 검증	6	
	설계 프로세스 검증	9	
	코딩 프로세스 검증	8	
	통합 프로세스 검증	5	
	검증 프로세스 결과에 대한 검증	6	
	SW 형상관리	6	
	SW 품질보증	5	
	SW 인증교섭	3	
Total		62	130

Fig. 6 Checklists Development Results

4. 사례연구

4.1. 모델기반개발 방법

최신 항공우주시스템의 시스템이 복잡도가 높아짐에 따라 모델링 및 시뮬레이션을 활용한 모델기반 개발 방법으로 개발되는 제품이 많아지고 있다. 이러한 개발방법을 모델기반 개발(Model-based Development, MBD)이라고 하며, RTCA는 DO-331(Model-Based Development and Verification Supplement to DO-178C and DO-278A)을 이용하여 인증하는 방법을 제시하고 있다. 본 논문에서는 DO-331 도구 인증은 Mathworks 사에서 수행하는 것으로 갈음하고 개발 내용만 인증받는 것으로 가정하였다. Fig. 7은 Mathworks사에서 제공하고 있는 DO-178C 개발 인증에 준하는 모델기반개발 프로세스 및 검증된 도구에 대한 내용이다[15].

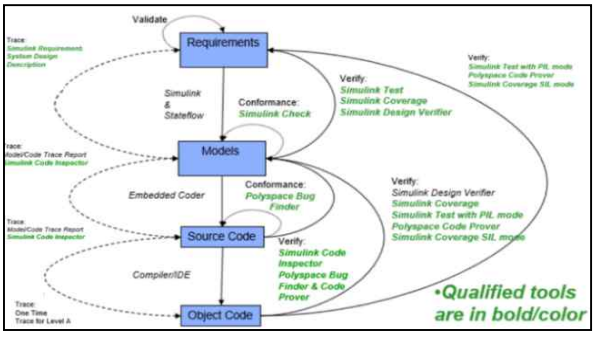


Fig. 7 MBD Workflow with Qualified Tools

모델기반 개발 방법은 개발 단계에서 소프트웨어 아키텍처를 모델로 구현하므로 요구사항 추적성이 용이하다. 궁극적으로는 기존 소프트웨어 개발 방법 대비 개발 기간이 단축되며 높은 신뢰성을 확보할 수 있다.

4.2. 사례연구용 헬리콥터 FCC

본 절에서는 DO-178C 사례연구용으로 Mathworks 사에서 제공하는 헬리콥터 시스템의 FCC (Flight Control Computer) 모델에서 인증 데이터를 출력하였다[16]. 인증 데이터 출력은 항공 소프트웨어 인증에 사용되는 모델기반개발 도구이자 검증된 도구로 DO-178C 및 DO-331 인증 워크플로에 대한 TQL-4(Tool Qualification Level 4)를 준수하는 Simulink Tool를 활용하였다[17]. 그리고 본 연구에서 개발한 체크리스트의 일부 검증을 수행하였다.

Simulink로 구현된 헬리콥터 FCC는 비행장치의 자세와 방향을 제어해주는 역할이며, Fig. 8와 같이 6개의 모델로 구성되어 있다. 각각의 모델은 왼쪽부터 Attitude Heading Reference System(AHRS) Voter, Helicopter Outer Loop Control(HOLC), Helicopter Inner Loop Control(HILC) 그리고 3개의 Actuator Control loop로 구성된다.

AHRS Voter는 헬리콥터 자세 및 방향에 대한 정보를 제공하는 3개의 AHRS 센서로부터 정보를 받아 Voting algorithm에 의해 유효 값을 선정하여 비행제어시스템에 제공하는 역할을 하며, 제어시스템은 2가지의 피드백 루프로 구성되어 있다. 3개의 Actuator 루프는 각각 Position Feedback과 Command를 입력받아 Actuator로 데이터를 제공해주는 모델이다.

4.3. 체크리스트를 활용한 인증 시뮬레이션

본 절에서는 앞 절에서 소개한 헬리콥터 FCC Simulink 모델을 활용하여 Fig 1 및 Table 4의 국내 무인동력비행장치 초도인증(최초형식) 프로세스에 고려해야 될 DO-178C 인증 사례연구를 수행하였으며, 개발한 체크리스트를 인증 프로세스에 일부 적용하여 실용성을 확인하였다.

4.3.1. 계획 프로세스(Planning Process)

무인동력비행장치 최초형식 개발에 대한 계획을 하는 단계로 시험비행 허가 절차에 해당된다. 소프트웨어 인증 프로세스에서는 계획 프로세스에 해당되며, 무인동력비행장치 개발에 앞서 전반적인 계획을 수립하고 이를 이행하는 방법 등을 정의한다.

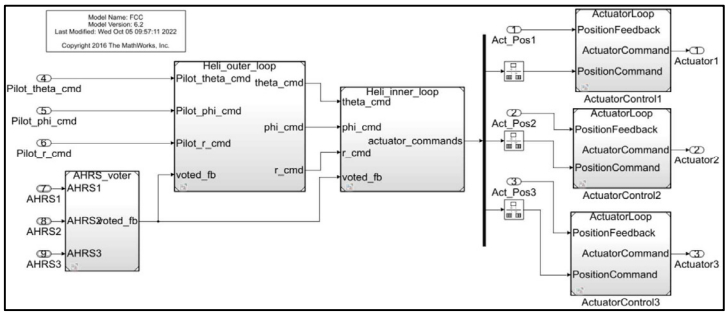


Fig. 8 FCC(Flight Control Computer) Model

계획 프로세스에서 무인동력비행장치 제작사가 인증 당국에 제출해야 하는 문서로 PSAC(Plan for Software Aspects of Certification)이 있다. PSAC은 소프트웨어 인증 계획서로 소프트웨어 라이프사이클 데이터와 프로세스에 대한 전반적인 내용을 다루고 있는 계획 문서이다.

Simulink Tool의 DO-Qualification Kit는 DO-178C 지침 내에서 PSAC에 포함되어야 할 요건인 11.1의 시스템 개요, 소프트웨어 개요, 인증 고려사항, 라이프사이클 등을 포함한 PSAC template를 제공한다[18]. 그리고 DO-178C 지침 내 계획 프로세스에서 PSAC 외 8가지 산출물에 대한 template도 제공되어 계획 프로세스에 해당되는 8가지 Objectives를 충족할 수 있도록 지원한다.

아래의 Fig. 9는 DO-Qualification Kit에서 제공하는 PSAC template의 목차와 Fig. 4에서 제시한 PSAC 체크리스트 점검 항목 중 일부를 같이 나타냈으며, 점검 항목에서 요구하는 내용에 대해 목차에 반영되어 있음을 확인하였다.

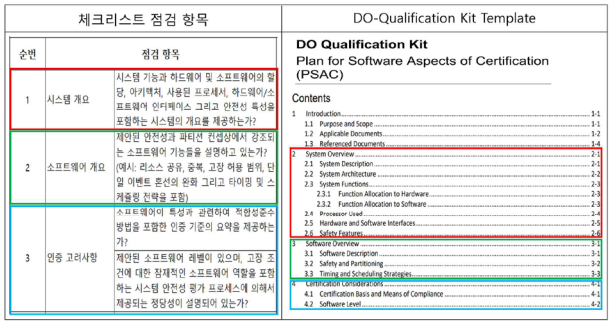
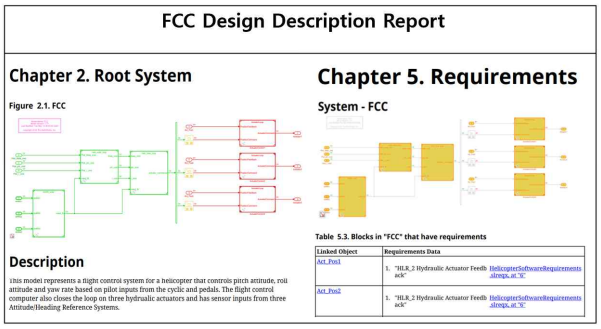


Fig.9 PASC Template Table of Contents for Checklists Item

4.3.2. 개발 프로세스(Development Process)

소프트웨어 개발 프로세스는 무인동력비행장치 최초 형식 인증 프로세스에서 시험비행절차에 해당된다. 본 논문의 4.3.1의 계획 프로세스에서 정의한 내용 및 프로세스를 소프트웨어 개발을 수행하게 된다.

Simulink 도구에서는 소스코드를 구현하기 위한 Low Level Requirement 및 아키텍처가 잘 구현되었는지 확인할 수 있는 DD(Design Description) 레포트를 생성하는 기능을 지원한다. 생성된 DD는 모델의 Root System, Subsystem에 대한 설명과 Parameter등의 내용을 포함하고 있다. DD외 Software Requirement, Embedded Coder Tool을 활용하여 DO-178C 지침 내 개발 프로세스의 7가지 Objective를 충족할 시킬 수 있다. Fig. 10은 FCC 모델의 Design Description Report에 대한 체크리스트 점검 결과를 일부 나타냈다.



체크리스트 점검 항목

순번	점검 항목	참조	적합여부 Yes	No	N/A	근거자료(Evidence)
1	알고리즘, 데이터 구조, 소프트웨어 요구사항이 어떻게 프로세서와 태스크로 할당되는지를 포함해서 소프트웨어가 어떻게 상세화된 상위수준 요구사항을 충족하는지에 대한 설명이 있는가?	11.10.a	V			Design Description Chapter 5 Requirements
2	요구사항을 구현하기 위해 소프트웨어 구조를 정의하는 소프트웨어에 대한 아키텍처에 대한 설명이 있는가?	11.10.b	V			Design Description Chapter 2 Root System Chapter 3 Sub System
3	입/출력에 대한 설명, 예를 들면, 소프트웨어 아키텍처 전반에 걸쳐 내부적 및 외부적인 데이터 사전이 있는가?	11.10.c	V			Design Description Chapter 4 System Design Variables

Fig. 10 Checklist Result for FCC Design Description

4.3.3. 총괄 프로세스(Integral Process)

DO-178C의 총괄 프로세스는 검증, 형상관리, 품질보증 프로세스로 구성되며, 최초형식 인증 프로세스에서 안전성인증 절차에 해당된다. 본 절에서는 앞 절에서 개발한 체크리스트를 모델기반 도구를 이용하는 검증 프로세스에 대하여 수행하였으며, 형상관리 및 품질보증 프로세스는 모델기반 도구의 추적성 관리도구와 모델 및 코드 검증도구로 점검하는 방법을 향후에 연구할 계획이다.

4.3.3.1. 검증 프로세스 인증 데이터 출력

Table 7과 같이 사례연구를 통해 FCC 모델에서 출력한 모델기반개발 방법의 프로세스별 인증 데이터를 출력 결과를 나타냈다. 출력된 DO-178C의 DAL C 수준에서 요구하는 검증 단계에서의 목표(Objectives) 34가지 모두를 충족시킬 수 있음을 확인하였다.

4.3.3.2. 검증 프로세스 체크리스트 확인

개발보증수준 C에 해당하는 검증 프로세스의 목표(Objectives) 34가지에 대한 Simulink 인증 데이터는 14종이 있다. 본 절에서는 검증 프로세스 결과에 대한 검증 프로세스를 체크리스트와 Simulink Tool을 통하여 점검하였다.

DO-178C 지침에서는 상위, 하위 수준 요구사항과 소프트웨어 구조에 대한 Test Coverage의 달성 여부를 확인해야 한다. Simulink에서는 상세설계 후 생성된 모델에 대한 검증으로 요구사항 기반의 테스트 케이스를 작성해야 하며, 작성된 테스트 케이스가 요구사항에 잘 반영되었는지를 Test Coverage Report를 통해 확인하며, Test Report를 통하여 시험 결과를 확인할

Table 7 Verification Process Case Study Results(FCC Simulink Certification Data)

Simulink Certification Data	DO-178C Table A-3 ~ 7 (Verification/DAL C Objectives)
SW Requirements	A-3.1 HLR comply with SR
Simulink Model	A-4.1 LLR comply with HLR
SW Design Description	A-4.8 SW architecture is compatible with HLR A-4.13 SW partitioning integrity is confirmed
Traceability Report	A-3.6 HLR are traceable to SR A-4.6 LLR are traceable to HLR.
Model Coverage Report	A-3.4 HLR are verifiable A-3.5 HLR conform to standards. A-7.3 Test coverage of HLR is achieved
REQ_based Test Report	
Model Advisor Conformance Report	A-4.5 LLR conform to standards. A-4.12 SW architecture conforms to standards
Design Verifier Report(Error Detect)	A-3.2 HLR are accurate and consistent. A-3.4 HLR are verifiable. A-3.7 Algorithms are accurate A-4.2 LLR are accurate and consistent A-4.7 Algorithms are accurate A-4.9 SW architecture is consistent
Test Report (LLR_based test generate)	A-7.1 Test procedures are correct A-7.2 Test results are correct and discrepancies explained A-7.4 Test coverage of LLR is achieved
Test Coverage Report(LLR_SIL)	
Source Code (SC)	A-7.7 Test coverage of SW structure (statement coverage) is achieved.
Code Inspection Report	A-5.1 SC complies with LLR A-5.2 SC compiles with SW architecture A-5.4 SC conforms to standards A-5.5 SC is traceable to LLR A-5.6 SC is accurate and consistent
Source Code Standard(SCS) Conformance Report	
EOC Certification Data (PIL MODE)	A-6.1 EOC complies with HLR A-6.2 EOC is robust with HLR A-6.3 EOC complies with LLR A-6.4 EOC is robust with LLR A-6.5 EOC is compatible with target computer. A-7.1 Test procedures are correct A-7.2 Test results are correct and discrepancies explained A-7.4 Test coverage of LLR is achieved

수 있다. 이와 관련하여 체크리스트 항목과 Simulink Test Coverage Report를 Fig. 11과 같이 나타냈다.

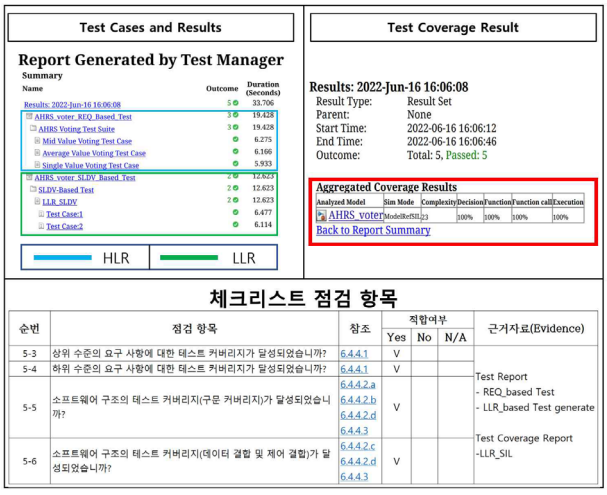


Fig. 11 Verification of Verification Process Checklist Results

5. 결 론

본 논문은 헬리콥터 FCC 모델을 모델기반 개발 도구와 무인동력비행장치 소프트웨어 인증에 활용할 수 있는 체크리스트를 활용하여 사례연구를 수행하였다.

본 논문에서 제안한 무인동력비행장치 소프트웨어 체크리스트의 점검항목은 유무인기의 고장조건 발생확률 및 심각도 분석을 통하여 개발보증수준(DAL) C 수준에 맞게 도출하였다. 그리고 국내 안전성인증 절차를 고려하여 프로세스 및 산출물을 효율적으로 점검할 수 있는 체크리스트와 모델기반 개발 도구를 활용한 개발 및 인증 방안을 제시하였다.

향후, 국내 무인동력비행장치 FCC 소프트웨어를 본 논문에서 제안하는 모델기반 개발 도구와 체크리스트를 활용한다면 소프트웨어의 계획 단계부터 개발과 양산까지의 개발 및 인증 업무를 합리적으로 수행할 수 있다. 그리고 제작사도 개발초기단계에서 위험요소를 효과적으로 줄이는 등 개발 비용과 개발 기간을 줄일 수 있을 것으로 기대할 수 있다.

References

[1] Drone “near-miss” incidents with aircraft up more than 25% –UK Airprox Board, https://www.unmannedairspace.info/uncategorized/drone-near-miss-incidents-aircraft-25-uk-airprox-board/, 2019

[2] Sauk-Hoon Im “Introduction of UAV(drone) safety certification”, Agricultural drone company site visit and industry meeting (2022)

[3] Veronica L. Foreman, Francesca M. Favaró, Joseph H. Saleh, Christopher W. Johnson, Software in military aviation and drone mishaps: Analysis and recommendations for the investigation process, *Reliability Engineering & System Safety*, Volume 137, Pages 101-111, 2015

[4] Christopher A. Hart(Team chiar), “Boeing 737 MAX Flight Control System, JATR(Joint Authorities Technical Review) Report”, pp.Ⅲ~Ⅳ, 2019.10.11.

[5] Korean Aviation Safety Act, Chapter 10 Ultra-Light vehicle.

[6] A. Jiménez, J. A. M. Merodio, and L. F. Sanz, ‘‘Checklists for compliance to DO-178C and DO-278A standards,’’ *Computer Standards Interfaces*, vol. 52, pp. 41–50, May 2017.

[7] J. Andres-Jimenez, J. -A. Medina-Merodio, L. Fernandez-Sanz, J. -J. Martinez-Herraiz and J. Gonzalez-De-Lope, "A Framework for Evaluating the Standards for the Production of Airborne and Ground Traffic Management Software," in *IEEE Access*, vol. 8, pp. 149142-149161, 2020

[8] Pharus-Tech / https://pharus-tech.com/en/

[9] Jun-Mo Yang, Hyo-Won Yeom, Min-Sung Kim.(2021).A Study on Global Trends and Domestic Studies for Establishing the Unmanned Aircraft Certification System. *Journal of the Korea Academia-Industrial cooperation Society*, 22(7), 259-265.

[10] Jin Gu Heo, Min Sung Kim, Man Tae Kim, Yong Ho Moon.(2019).The Study on Airworthiness Certification Process on Military Airborne Safety Critical Software based on DO-178. *Journal of Aerospace System Engineering*, 13(1), 62-68.

[11] Software Considerations in Airborne Systems and Equipment Certification, document RTCA, DO-178C, 2011

[12] O. Laitenberger, J.M. DeBaud, An encompassing life cycle centric survey of software inspection, J. Syst. Softw. 50 (1) (2000) 5–31.

[13] L.M. Muriana, C. Maciel, F.F. Mendes, QualiCES: a method for verifying the consistency among documents of the engineering phase, in: Proceedings of the 30th ACM international conference on Design of communication, ACM, 2012, pp. 105– 114.

[14] T. Capers Jones, Estimating Software Costs, 2nd ed., McGraw-Hill Professional, New York, USA, 2007, pp. 1–644.

[15] https://kr.mathworks.com/videos/using-qualified-tools-in-a-do-178c-development-process-part-11-tool-qualification-1509539322829.html

[16] https://github.com/wfpotter/DO178 Case Study

[17] https://kr.mathworks.com/company/newsroom/simulink-code-inspector-and-polyspace-qualified-under-do-330.html

[18] DO-Qualification Kit User Manual / https://manualzz.com/doc/1113036/do-qualification-kit-user-s-guide