JANT 한국항행학회논문지
Journal of Advanced Navigation Technology

항행 및 항법

J. Adv. Navig. Technol. 20(2): 127-133, Apr. 2016

DO-278A 표준을 적용한 A-SMGCS 개발 사례 연구

Case Study of Applying DO-278A for Advanced Surface Movement Guidance & Control System

조 상 훈[1*] · 이 홍 석[2] · 김 자 영[1] · 이 진 근[1]
[1]인천국제공항공사
[2]한국산업기술시험원 IT안전기술센터

Sanghoon Jo[1*] · Hongseok Lee[2] · Jha-young Kim[1] · Jin-geun Yi[1]
[1]Incheon International Airport Corporation, Incheon 22382, Korea
[2]IT Safety Engineering Center, Korea Testing Laboratory, Daejeon 34025, Korea

[요 약]

본 논문은 항공기 지상이동유도 및 통제 시스템(A-SMGCS)의 소프트웨어를 개발하기 위해 CNS/ATM 분야의 지상용 항공장비 소프트웨어 표준인 DO-278A를 적용한 사례를 제시하였다. DO-278A를 적용하기 위해 필요한 고려사항들을 기술하였으며, DO-278A에 명시된 프로세스 혹은 요건들을 테일러링한 방법에 대해 상세히 논의하였고, 테일러링된 결과에 대해 설명하였다. 그리고 DO-278A를 처음 적용하여 개발하고자 할 때 발생할 수 있는 사항들과 DO-278A를 적용하면서 발생했던 문제점들이나 이슈사항들을 정리하였다. 결론으로 DO-278A를 적용하면서 시행착오를 줄이기 위해 반드시 고려되어야 하는 교훈들을 설명하였으며 이러한 문제점들이나 이슈사항들을 해결하기 위한 방법을 제시하였다.

[Abstract]

This paper proposes a case study of applying to DO-278A, which is a software development standard of ground aviation equipment in communication, navigation, surveillance/air traffic management (CNS/ATM) domain, in order to develop software of advanced surface movement, guidance and control system (A-SMGCS). It explains considerations to application of DO-278A standard, discusses how to tailor processes or requirements specified in DO-278A in detail, and explain tailored results. And it also summary problems or issues occurred during application of DO-278A, which can be appeared when DO-278A is applied to for the first time by any development part. These are required considerations in order to reduce trial and error. To the conclusion, it explains lessons learned from application of DO-278A and suggest how to solve these problems or issues.

Key word : Advanced surface movement, guidance and control system (A-SMGCS), DO-278A, CNS/ATM, Software Development, Safety critical software, Process tailoring.

http://dx.doi.org/10.12673/jant.2016.20.2.127

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-CommercialLicense(http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 2 March 2016; **Revised** 22 March 2016
Accepted (Publication) 7 April 2016 (30 April 2016)

***Corresponding Author; Sanghoon Jo**

Tel: +82-32-741-4535
E-mail: sanghoonj@daum.net

Ⅰ. 서 론

항공분야는 산업의 특성상 시스템의 고장 또는 오작동 시 항공기에 탑승한 인명의 손실이나 심각한 상해 또는 장비의 손상이 발생하여 대형 참사로 이어질 수 있다. 이러한 이유로 항공분야의 연구개발에는 시스템이 본래 의도한 기능을 수행하는 동시에 수용할 만한 안전 수준을 제공하고 있는지 확인할 수 있어야 한다. 미국과 유럽에서는 FAA (Federal Aviation Adminstration)와 EASA (European Aviation Safety Agency) 등과 같은 기관을 통해 항공기 및 항공용 시스템 개발 시 안전부분을 고려하여 개발할 수 있는 표준을 만들어 권고하고 있다 [1],[2]. 국내에서도 이러한 국제 표준들을 참고하여 연구개발을 수행하고 있다.

항공기 지상이동유도 및 통제시스템 (A-SMGCS; advanced-surface movement guidance control system)은 공항 이동지역 내 항공기와 차량 등 이동 물체에 대한 감시 정보(surveillance)를 기반으로 이동물체의 목적지까지 경로를 지정(routing)하고, 항공등화시설의 점등을 통하여 안내(guidance) 정보를 제공함은 물론 위험상황 발생 등에 대비하여 이동 물체에 대한 통제(control) 기능을 제공하는 시스템이다 [3]. 유럽에서는 EMMA (European airport movement management by A-SMGCS) 프로젝트[4]를 통해 1990년도 중반부터 A-SMGCS 관련 연구를 수행하고 있고, 국내에서는 2013년 말부터 국가 연구개발 사업으로 A-SMGCS 개발을 추진하고 있다.

국내에서 개발하는 A-SMGCS는 이동체 감시시스템, 운항정보시스템, 기상정보시스템, 항공등화시스템 등과 같은 여러 시스템들이 상호 협력적으로 운영되는 포괄적 시스템으로 개발품 대부분이 소프트웨어로 구성되어 있으며, 공항 이동지역 내 이동하는 이동체와 관련되어 있으므로 안전 관련 표준을 적용하여 개발할 필요가 있다. 항공분야 소프트웨어 개발관련 표준으로는 RTCA (Radio Technical Commission for Aeronautics)의 DO-178C [5]와 DO-278A [6]가 있다. DO -178C는 항공 탑재용 소프트웨어 개발에 적용되는 표준으로 FAA에서 항공용으로 사용하는 소프트웨어를 승인하기 위한 수단으로 오랫동안 사용되어 왔다 [7]. 반면에 DO-278A는 CNS/ATM 시스템과 장비에 대한 비항공용 소프트웨어 제작을 위한 지침을 제공하기 위한 것으로 CNS/ATM 시스템용 소프트웨어 개발을 위한 표준이며, 아직까지 FAA에서 DO-278A의 승인에 대한 고려사항은 존재하지 않는다.

본 논문에서는 DO-278A 표준을 A-SMGCS 개발에 적용하고자 할 때 고려해야 할 점과 적용상의 어려운 점들에 대해 논의하고자 하였다. 본 논문의 구성은 다음과 같다. 2장에서는 DO-278A를 적용하기 위한 관련된 시스템 분야의 표준에 대해 간단하게 소개하고, DO-278A 적용을 위한 사전 조건을 정의한다. 3장에서는 DO-278A기반 프로세스 테일러링에 대한 구체적인 내용을 소개하며, 4장에서는 DO-278A적용상 어려웠던 점들 및 교훈을 기술하였다.

Ⅱ. DO-278A 관련 표준

DO-278A 적용을 위해서는 소프트웨어 프로세스와 상호작용하는 관련 프로세스를 정의해야 할 필요가 있다. A-SMGCS 개발 프로세스는 시스템 엔지니어링 관련 표준을 적용하여 그림 1과 같이 정의하였다 [8]. DO-278A의 프로세스를 적용하기 위해서는 소프트웨어 보장 수준을 결정해야 할 필요가 있으며, 이는 시스템 수준의 안전 평가를 통해 도출되어야 한다.

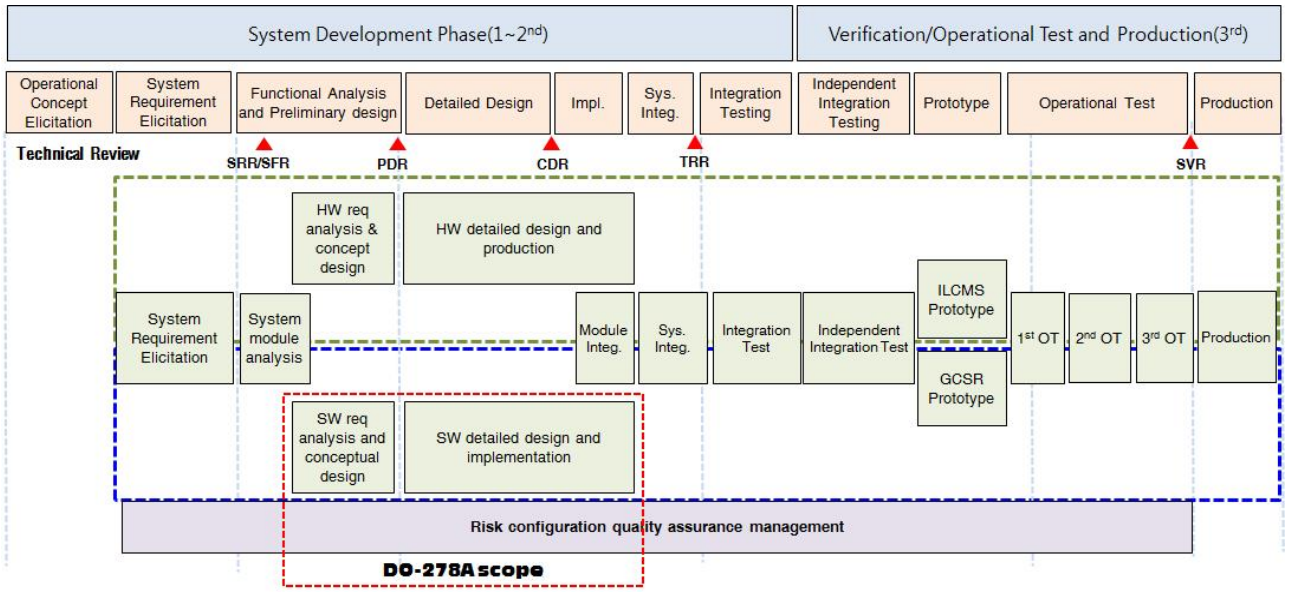

그림 1. A-SMGCS 연구개발 프로세스

Fig. 1. Research & development process of A-SMGCS.

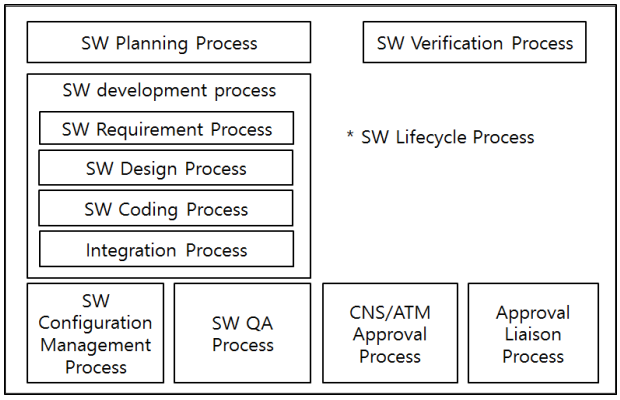

그림 2. SW 라이프사이클 모델
Fig. 2. Software life cycle model.

Ⅲ. DO-278A 기반 개발 프로세스 테일러링

A-SMGCS 개발품의 대부분이 소프트웨어로 구성되어 있어 지상용 항공장비에 탑재되는 소프트웨어 개발 기준인 DO-278A를 테일러링하여 적용하였다.

DO-278A를 적용하여 연구개발을 수행하기 위해 소프트웨어 개발범위 및 기준을 설정하는 것은 매우 중요한 일이다 [9]. DO-278A 프로세스의 소프트웨어 라이프사이클 모델은 그림 2와 같이 계획 프로세스, 개발 프로세스(요구사항, 설계, 코딩, 통합), 검증 프로세스, 형상관리 프로세스, 품질보증 프로세스, 승인 연락 프로세스, CNS/ATM 승인 프로세스가 있다.

본 연구에는 소프트웨어 라이프사이클 모델 프로세스에서 계획, 개발, 검증, 형상관리, 품질보증의 프로세스를 적용하였다. 승인 연락 프로세스 및 CNS/ATM승인 프로세스는 소프트웨어 승인 기관이 지정되어 있지 않고, 승인 기준을 제시할 기관이나 수행 주체가 명확하지 않으므로 미적용 하였다. A-SMGCS 개발 프로세스는 시스템 엔지니어링 프로세스를 같이 적용하고 있어 개발 프로세스 중 코딩과 통합 프로세스는 구현 및 통합 프로세스로 적용하였다.

3-1 개발 계획 프로세스

소프트웨어 계획 프로세스는 소프트웨어 개발 시작 단계에서 준비하는 활동으로 소프트웨어 개발 관련 계획 및 소프트웨어 개발과 관련된 표준을 수립한다.

개발계획 단계에서 달성되어야 하는 목표는 소프트웨어 수명주기 프로세스의 활동이 정의되어야 하고 , 프로세스 간의 상호관계, 해당 프로세스들의 순서, 피드백 메커니즘 및 변환 기준을 포함한 소프트웨어 수명 주기를 결정하여야 한다. 또한, 소프트웨어 개발 표준을 정의하여 수명 주기 환경을 선택하고 정의하여야 한다. 표 1은 DO-278A를 적용한 A-SMGCS 소프트웨어 개발 계획 단계 프로세스의 활동 및 관련 산출물을 정의한 표이다.

표 1. 소프트웨어 개발 계획 프로세스
Table 1. Software planning process.

No	Activity	Applicable
1	Provide direction to the personnel performing the software life cycle processes	Yes
2	The software development standards to be used for the project	Yes
3	Methods and tools that aid error prevention and provide defect detection in the software development processes	Yes
4	Considerations for multiple-version dissimilar software, if planned	No
5	Change control and review process for software plans and software development standards	Yes
6	Considerations for deactivated code, if planned	No
7	Considerations for user-modifiable software, if planned	No
8	Planning of configuration data, or adaptation data items	Yes
9	Software life cycle environment planning	Yes
	- Software development environment	No
	- Language and compiler considerations	No
	- Software testing environment	Yes
10	Software development standard	Yes
11	Software planning verification	Yes
Work products		
Plan for software aspects of approval Software development plan Software verification plan Software configuration management plan Software quality assurance plan Software requirement standard Software design standard Software code standard Software verification results		

A-SMGCS 개발에는 개발계획 프로세스 중 다중 소프트웨어(diverse software)를 고려하지 않았으며, 비활성화 코드 및 사용자 변경 가능 소프트웨어를 고려하고 있지 않으므로 표1의 4, 6, 7번 활동은 미적용 하였다. 소프트웨어 개발 환경 및 언어와 컴파일러 고려사항이 있는 9번 활동은 표준에서 제시한 요건을 적용하기 위해서는 과제 기획단계에서 구체적으로 논의되어야 하고, 소프트웨어 승인기관에서 세부사항에 대해 조율되어야 하지만, 승인기관이 존재하지 않으며 개발 진행 단계에서 결정하기에는 무리가 있으므로 부분 적용하였다.

3-2 개발 프로세스

소프트웨어 개발 프로세스는 소프트웨어의 요구사항, 설계, 구현, 구현 및 통합 프로세스를 포함한다.

표 2. 소프트웨어 요구사항 프로세스
Table 2. Software requirements process.

No	Activity	Applicable
1	Software requirement specification - High level requirement specification - To provide derived high level requirement to system development process - Interface definition - Definition of high-level requirements that address system requirements allocated to software to preclude system hazards - The high-level requirements in quantitative terms with tolerances, if applicable	Yes
2	Software requirement review	Yes
3	Low level verification	Yes
4	traceability data (System requirement – Software requirement)	Yes
Work products		
Software requirement specification Software verification results		

1) 요구사항 프로세스

요구사항 프로세스는 시스템 수명 주기 프로세스의 출력을 사용하여 상위 요구 사항을 개발한다. 이 상위 요구 사항은 기능, 성능, 인터페이스 및 안전 관련 요구사항을 포함한다. 요구사항 단계에서 달성되어야 하는 목표는 상위 요구 사항을 개발하고, 파생 상위 요구 사항을 정의하며 시스템 안전 평가 프로세스를 포함한 시스템 프로세스에 제공하는 것이다. 표 2는 A-SMGCS 소프트웨어 개발 프로세스 중 요구사항 프로세스의 활동 및 관련 산출물을 정의한 표이다.

표 3. 소프트웨어 설계 프로세스
Table 3. Software design process.

No	Activity	Applicable
1	User modifiable software design	Yes
2	Design for deactivated code	Yes
3	Software design	Yes
	- Use case diagram	Yes
	- Class diagram	Yes
	- Sequence diagram	Yes
	- Activity diagram	Yes
	- UI design	Yes
	- Database structure	Yes
	- Software architecture	Yes
	- Low level requirement	Yes
	- Software partitioning	No
	- Reactions for fault conditions	Yes
	- Data flow monitoring for safety related requirement	Yes
	- Detection abnormal or wrong input data	Yes
4	Software design verification	Yes
5	Traceability data (Software requirement – Software design)	Yes
Work products		
Software design(conceptual, detailed) Software verification results Traceability data		

2) 설계 프로세스

설계 프로세스는 상위 요구사항을 소스코드로 구현하는 데 사용할 수 있도록 소프트웨어 아키텍처와 하위 요구사항을 개발하고 이를 위해 소프트웨어 설계 프로세스에서 한 번 이상의 반복을 통해 재 정의된다. 설계 단계에서 달성되어야 하는 목표는 소프트웨어 아키텍처와 하위 요구 사항은 상위 요구 사항으로부터 개발하고, 파생 하위 요구 사항을 정의하고 시스템 안전 평가 프로세스를 포함한 시스템 프로세스에 제공하는 것이다. 표 3은 A-SMGCS 소프트웨어 개발 프로세스 중 설계 프로세스의 활동 및 관련 산출물을 정의한 표이다.

설계 프로세스 중 사용자 변경 가능 소프트웨어에 대한 설계와 비활성화 코드에 대한 설계는 A-SMGCS 개발에 포함되지 않는 내용으로 미적용 하였으며, 소프트웨어 분할(partitioning) 부분도 운영체제(OS) 및 대상시스템에서 분할을 지원되지 않는 시스템으로 개발하기 때문에 설계시 분할을 고려하지 않으므로 적용하지 않았다. 유즈케이스, 클래스, 시퀀스, 액티비티, UI, 데이터베이스 구조는 DO-278A와 관련된 표준은 아니지만 일부를 UML 방법으로 적용하여 설계서에 반영하였다.

3) 구현 및 통합 프로세스

구현 및 통합 프로세스는 소프트웨어 코딩 프로세스에서 소스 코드는 소프트웨어 아키텍처와 하위 요구 사항으로부터 구현된다.

표 4. 소프트웨어 구현 및 통합 프로세스
Table 4. Software coding & integration process.

No	Activity	Applicable
1	Coding - Coding standard compliance	Yes
2	Source code implementation and verification - Code standard compliance - Constraints are conformed, if auto code generator used - Unit testing - Integration testing	Yes
5	Executable objective code	Yes
6	Configuration data	Yes
7	Traceability data(Design – code)	Yes
8	Review - Software unit testing report review	Yes
9	Software implementation verification	Yes
10	Object code generation	Yes
11	Executable object code generation	Yes
12	Compile	Yes
13	Link and load	Yes
14	Trace - Bi-traceability between high level requirement and low level requirement - Bi-traceability between low level requirement and source code	Yes
Work products		
Source code Software unit test procedure and results Executable objective code Configuration data Software verification results Traceability data		

구현 및 통합 단계에서 달성되어야 하는 목표는 소프트웨어 아키텍처와 하위 요구 사항은 하위 요구 사항으로부터 소스 코드를 개발하는 것이다. 표 4는 A-SMGCS 소프트웨어 개발 프로세스 중 구현 및 통합 프로세스의 활동 및 관련 산출물을 정의한 표이다.

3-3 검증 프로세스

소프트웨어 검증 프로세스는 소프트웨어 모듈의 시험 사례와 절차의 시험, 분석, 개발을 조합하고 이후 해당 시험 절차를 실행하여 달성된다. 검토와 분석은 소프트웨어 요구 사항, 소프트웨어 아키텍처 및 소스 코드의 정확성, 완전성 및 검증 가능성에 대한 평가를 제공한다. 시험 사례와 절차의 개발은 요구 사항의 내부 일관성과 완전성에 대한 추가적인 평가를 제공할 수 있다. 시험 절차를 이행하면 요구사항 준수 여부를 입증할 수 있다.

소프트웨어 검증 프로세스에 대한 입력은 시스템 요구사항, 소프트웨어 요구사항, 소프트웨어 아키텍처, 추적 데이터, 소스 코드, 실행 목적코드 및 소프트웨어 검증 계획을 포함한다. 소프트웨어 검증 프로세스의 출력은 소프트웨어 검증 사례와 절차, 소프트웨어 검증 결과 및 연결된 추적 데이터에 기록된다. 요구 사항을 소프트웨어에 구현한 후 검증 가능해야 할 필요성은 그 자체로 소프트웨어 개발 프로세스에 대한 부가적인 요구사항 또는 구속조건을 부과할 수 있다.

표 5. 소프트웨어 검증 프로세스
Table 5. Software verification process.

No	Activity	Applicable
1	Software review and analysis - Review and analysis for high level requirement - Review and analysis for low level requirement - Review and analysis for software architecture - Review and analysis for source code - Review and analysis for integration process	Yes
2	Software testing - Subsystem testing - Subsystem test results review - Test environment - Requirement based test • Normal boundary test case • Robustness test case - Test coverage • Requirement based coverage • Structural coverage(Statement, Branch, MC/DC) • Structural coverage review - Review of test case, procedure, results	Yes
3	Software verification process traceability	Yes
4	Configuration data verification	Yes
Work products		
Sub-system test plan Sub-system test procedure Sub-system test results Sub-system test review report Software verification report Traceability data		

표 6. 소프트웨어 형상관리 프로세스
Table 6. Software configuration management process.

No	Activity	Applicable
1	Software configuration management activity	Yes
2	Data control(CC1/CC2)	Yes
3	Software load control	Yes
4	Software life cycle environment control	Yes
Work products		
Configuration management records Configuration index Software build setting information Executable program		

검증 단계에서 달성되어야 하는 목표는 개발 단계에서 수행된 작업들에 대한 검토 및 시험이 수행되는 것이다. 표5는 A-SMGCS 소프트웨어 검증 프로세스의 활동 및 관련 산출물을 정의한 표이다.

3-4 형상관리 프로세스

형상관리 프로세스는 소프트웨어 코딩 프로세스에서 소스 코드가 소프트웨어 아키텍처와 하위 요구 사항으로부터 구현되었는지 확인하고 개발 단계 동한 형상관리 대상을 식별하고 유지 및 관리한다. 형상관리 단계에서 달성되어야 하는 목표는 소프트웨어 개발 단계 동안 형상관리 대상을 식별하고 유지 관리하는 것이다. 표 6은 A-SMGCS 형상관리 프로세스의 활동 및 관련 산출물을 정의한 표이다. A-SMGCS 개발에는 형상관리 수준을 CC2 (control category 2) 수준의 프로세스 활동을 수행하도록 정의하였다. CC1수준으로 형상관리를 수행하기 위해서는 산출물의 버전관리 체계가 구축되어야 하며, 강력한 변경 통제 프로세스가 구축되어 있어야 하지만 개발사들 각각을 변경 통제하기 위한 인력과 시스템이 구축되어 있지 않으므로, CC2수준에서 형상관리를 수행하였다.

3-5 품질보증 프로세스

품질보증 프로세스는 목적이 달성되고, 결함을 탐지하며, 제품을 평가하고, 요구사항을 추적한다. 또한 소프트웨어 제품과 소프트웨어 수명 주기 데이터가 승인 요구 사항을 준수한다는 보장을 획득하기 위해 소프트웨어 수명 주기 프로세스와 그 출력사항을 점검한다. 표 7은 A-SMGCS 품질보증 프로세스의 활동 및 관련 산출물을 정의한 표이다.

표 7. 소프트웨어 품질보증 프로세스
Table 7. Software quality assurance process.

No	Activity	Applicable
1	Software quality assurance(SQA) records	Yes
2	Audits for software development standard	Yes
Work products		
Software quality assurance report		

품질보증 단계에서 달성되어야 하는 목표는 소프트웨어 품질보증 계획 문서에 정의된 기준의 준수와 일관성을 위한 소프트웨어 계획과 표준을 개발하고 검사하는 것과 공급자의 소프트웨어 수명 주기 프로세스를 포함한 프로세스가 승인된 소프트웨어 계획과 표준을 준수하는지, 소프트웨어 수명 주기 프로세스에 대한 변환 기준을 만족하는지 등을 검사하여 소프트웨어 제품의 적합성 검사를 수행하는 것이다.

3-6 승인 연락 및 CNS/ATM 프로세스

승인 연락 프로세스 및 CNS/ATM프로세스는 시스템과 장비의 소프트웨어 측면에 관한 승인 프로세스이다. 승인 기관은 소프트웨어를 CNS/ATM 시스템의 일부로 검토한다. 즉, 승인 기관은 소프트웨어를 고유한 독립적으로 실행되는 형태의 제품으로 승인하지 않는다. A-SMGCS 개발에는 소프트웨어 승인 기관이 지정되어 있지 않고, 승인 기준을 제시할 기관이나 수행 주체가 명확하지 않으므로 미적용 하였다.

Ⅳ. 연구의 한계

A-SMGCS 개발에 DO-278A를 적용하면서 다음과 같은 한계점들이 드러났다.

1) DO-278A와 관련된 시스템 프로세스의 선정 문제; 시스템 프로세스 표준은 ARP4754[2], ISO15288[10], 시스템 엔지니어링 매뉴얼[11]등 여러 가지 참조표준이 존재하며, 각 표준별로 준수해야 하는 기준이 상이하기 때문에 어떤 범위로 어디까지 수행해야 하는지에 대한 결정이 어려웠다.

2) 프로젝트 성격상 DO-278A의 일부 항목에 대한 준수가 어려운 점들에 대한 고민; DO-278A의 요건들 중에 A-SMGCS 프로젝트에서 만족시키기 어려웠던 대표적인 것들로는 다음과 같은 것들이 있다. a) 언어와 컴파일러에 대한 고려 b) 개발 및 검증 도구에 대한 도구 인정 c) 운영체계, 상용 라이브러리와 같은 상용 제품에 대한 인정 d) 프로젝트 조직 구조상 형상관리 활동의 데이터 관리의 CC1 수준의 준수 불가능.

위의 이슈들은 개발을 착수하는 단계에서 검토되고 해결되어야 할 사항이었으나, 프로젝트 착수 시점에 이미 결정된 사항들도 있었기 때문에 이미 결정된 사항들을 되돌려서 하는 것은 현실적으로 불가능했다. 첫 번째로 언어와 컴파일러의 경우 항공용 소프트웨어 개발에 필요한 도구인정[12]에 부합되도록 도구 선정을 수행해야 한다. 하지만, 그와 같은 활동을 수행하기 위한 일정과 비용이 계획되지 않았기 때문에 진행시키기에 매우 큰 어려움이 있었다.

두 번째로 운영체계, 상용 라이브러리의 경우 DO-278A의 12.4에 따라서 수행되어야 한다. 하지만 이것 역시 기 선정된 솔루션이 DO-278A의 요건에 부합하는지의 여부를 판단하고, 그렇지 않다면 새로운 솔루션을 찾기 위한 활동 및 새로운 솔루션에 대한 DO-278A의 12.4항목에 대한 적합성 여부를 판단하는 것은 개발 일정이나 비용에 고려되지 않았으며, 그렇기 때문에 이를 적용하는 것은 현실적이지 않았다.

마지막으로 형상관리 활동에서 CC1수준의 관리를 수행하기 위해서는 각각의 소프트웨어 개발 업체들에서 형상관리 시스템과 통일된 프로세스를 구축 및 운영하여야 하지만, 비용을 고려하여 PMO (project management office)조직을 통해 통합 프로젝트에 대한 형상관리 활동을 수행하였으나 이 부분에 대한 DO-278A요건이 만족되는 것은 아니다.

3) 소프트웨어 승인 기관의 부재; DO-278A에서는 승인 연락 프로세스를 통해서 소프트웨어의 승인 요건에 대한 기준 및 여러 가지 기술적 이슈에 대한 해결을 하도록 되어 있다. 하지만 A-SMGCS 프로젝트에서는 소프트웨어에 대한 승인 기준을 적절하게 수립할 만한 역할을 가진 기관이 없었으며, 그로인해 DO-278A를 적용하면서 발생하는 여러 가지 이슈들을 적절하게 해소하지 못했다.

Ⅴ. 결 론

DO-278A는 지상기반의 CNS/ATM 소프트웨어 개발 및 검증을 지원하기 위한 가이드라인이다. DO-278A가 소프트웨어 개발 및 인증 절차를 제공할 목적으로 만들어진 것은 아니지만 소프트웨어 개발 절차를 설명하는 것으로 지상용 항공장비의 소프트웨어를 만드는데 적절한 역할을 수행할 수 있다. 본 논문에서는 국내에서 국가 R&D로 개발하고 있는 A-SMGCS의 소프트웨어 개발 및 검증에 DO-278A 표준을 적용한 사례를 제시하였다.

제시한 사례를 통해 계획, 요구사항, 설계, 구현 및 통합, 검증, 형상관리, 품질보증, 승인 연락 및 CNS/ATM프로세스 각 영역에 대한 AL4기준으로의 활동을 식별하고, DO-278A에서 요구하는 활동의 수행 여부를 판단하였고, 수행하지 않는 경우에 대한 사유를 정리하였다.

하지만, 프로젝트 성격 또는 사업특성 및 제약사항들로 인해 DO-278A를 적용하지 않은 사항들로 인한 한계들이 노출되었으며, 이런 점들에 대한 개선하기 위해서는 소프트웨어 개발에 관한 승인 조직이 필요하며, 개발 초기에 소프트웨어의 개발 기준 수립 및 개발 범위의 명확화가 필요하다. 이러한 개선사항들에 대한 구체적인 연구는 향후 연구에서 수행할 예정이다.

감사의 글

본 연구는 국토교통부/국토교통과학기술진흥원의 지원으로 수행되었습니다(과제번호:15ATRP-C069188-03).

참고 문헌

[1] SAE International, Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment, SAE International, Warrendale, USA, Aerospace recommended practice(ARP) 4761, 1996.

[2] SAE International, Guidelines for development of civil aircraft and systems, SAE International, Warrendale, USA, Aerospace recommended practice(ARP) 4754A, 2010.

[3] ICAO, Advanced surface movement guidance and control system (A-SMGCS) manual, ICAO, Montreal, Canada, ICAO DOC 9830, 2004.

[4] EMMA project [Internet]. Available: http://www.dlr.de/emma/.

[5] RTCA, Software considerations in airborne systems and equipment certification, RTCA, Washington, USA, RTCA DO-178C, 2011.

[6] RTCA, Software integrity assurance considerations for communication, navigation, surveillance and air traffic management(CNS/ ATM) systems, RTCA, Washington, USA, RTCA DO-278A, 2011.

[7] S. A. Jacklin, Certification of safety-critical software under DO-178C and DO-278A, NASA Ames Research Center, Moffett Field, CA, 2012.

[8] S. H. Jo, J. Y. Kim and J. G. Kim, “Development of advance surface movement guidance & control system utilize system engineering process,” in *Proceeding of the The Journal of Korea Navigation Institute*, Seoul: Korea, pp.337-340, 2015.

[9] H. S. Lee, G. H. Kwon and B. G. Ko, “A study for evaluation method of safety critical software in avionics industry,” *The Journal of Korea Navigation Institute*, Vol. 2, No. 19, pp. 91-97, Apr. 2015

[10] ISO/IEC/IEEE, Systems and software engineering - System life cycle processes, ISO/IEC/IEEE, Switzerland, ISO/IEC/IEEE 15288:2015, 2015.

[11] FAA, Systems Engineering Manual, FAA, Washington, USA, 2014.

[12] RTCA, Software tool qualification considerations, RTCA, Washington, USA, RTCA DO-330, 2011.

조 상 훈 (Sanghoon Jo)
2013년 2월 : 한국항공대학교 항공운항관리학과 (이학석사)
2016년 2월 : 한국항공대학교 항공운항관리학과 박사수료
2014년 12월 ~ 현재 : 인천국제공항공사 전문연구원
※ 관심분야 : CNS/ATM, 운항관리, 시스템 안전평가, 시스템 V&V

이 홍 석 (Hongseok Lee)
2011년 2월 : 아주대학교 전자공학과 (공학박사)
2010년 9월 ~ 현재 : 한국산업기술시험원 선임연구원
※ 관심분야 : 시스템 안전평가, 기능 안전 소프트웨어, 소프트웨어 V&V

김 자 영 (Jha-young Kim)
1999년 2월 : 전북대학교 전기공학과 (공학사)
1999년 10월 ~ 현재 : 인천국제공항공사 차장
※ 관심분야 : 공항운영 및 개발, 항공안전, CNS/ATM

이 진 근 (Jin-geun Yi)
2006년 2월 : 서울과학기술대 전기공학과 (공학석사)
2011년 7월 : 한국항공대학교 항공경영학과 박사수료
1990년 7월 ~ 1999년 6월 : 한국공항공사
1999년 6월 ~ 현재 인천국제공항공사 항공등화팀장
※ 관심분야 : 항공등화, 전자전기, CNS/ATM, A-SMGCS