정형기법을 적용한 DO-178B 안전성 검증 및 인증 기준 개선

김창진° 최진영
고려대학교 정보통신대학 정형기법연구실
{cjkim°, choi}@formal.korea.ac.kr

KISS 33rd Fall Conference

Chang-Jin Kim°, Jin-Young Choi
Korea University Information & Communication College, Formal Methods Lab.

요 약

DO-178B는 항공분야 소프트웨어의 안전성 인증 기준으로서 실질적인 국제 표준으로 인정받고 있다. 그러나 목표달성 중심의 기준만을 제시함으로써 안전성 분석 및 검증에 대한 구체적인 가이드라인의 부족이 문제시 되어왔다. 본 논문을 통해 DO-178B의 취약점을 분석하고 정형기법을 적용한 개선방안을 제시한다. DO-178B 내용 중 정형기법을 적용하여 수행되어야 할 활동 요소들을 식별하고 정형언어를 통한 설계와 검증, 그리고 그 결과에 대한 증빙자료의 제출을 기준에 포함시킴으로써 개발 활동의 방향과 범위를 명확히 하고 인증 절차의 투명성을 향상시킬 수 있다.

1. 서 론

본 연구는 안전필수소프트웨어의 개발에 있어 소프트웨어의 안전성과 보안성을 보장하기 위한 통합 가이드라인의 필요성으로부터 시작되었다.

미국과 유럽 선진국들의 경우 RTCA/DO-178B나 Common Criteria (CC) 를 바탕으로 안전필수 소프트웨어의 개발과 인증에 대한 경험이 이미 상당 수준 축적되었을 뿐 아니라 그 동안 식별된 문제점을 인식하고 이제는 새로운 개선 방향에 대한 연구를 진행하고 있다. 그에 반해 국내에서는 기존의 표준 자체에 대한 기본적인 이해와 연구는 물론, 이러한 표준을 강제하고 인증의 기준으로 제시할 수 있는 수단 역시 없다는 사실은 문제가 아닐 수 없다.

영국 국방성(MoD)의 경우 안전등급 시스템의 개발 및 획득을 위해 정형성(Formalism)에 기반한 UK Def Stan 00-55를 적용해 왔으며 체계 인수 및 인증의 기준으로 사용해왔다. 그리고 이 규정의 폐기 이후에도 차기 국방 표준인Def Stan 00-56을 적용함과 동시에 소프트웨어에 대해서는 DO178B의 적용 및 그 적합성에 대한 연구를 병행하고 있다. DO178B는 사실상 민간표준이므로 별도로 군용에 대한 구분을 두지 않고 감항 (Airworthiness) 인증 기준을 제시하고 있는데 영국 국방분야의 경우뿐 아니라 이와 같은 민.군 통합표준의 추세는 계속될 것이다.

본 논문은 DO-178B를 중심으로 안전필수 소프트웨어의 개발과 관련된 기존의 주요 표준에 대한 분석을 통해 기존 표준들의 문제점을 식별하고 그 개선방법을 연구한다. 그리고 이를 통해 소프트웨어 개발의 오류를 최소화함과 동시에 명확한 인증 기준으로 활용할 수 있는 안전필수 소프트웨어의 개발 및 검증 가이드라인을 제시하고자 한다.

이미 DO-178C와 같은 새로운 표준의 등장이 예고되었지만 아직 정확한 내용을 확인할 수 있는 단계는 아니다. 따라서 본 논문의 주장과 명확히 실체가 드러나지 않은 새로운 표준과의 차이 또는 유사의 개연성은 문제의 범위에 포함시키지 않는다.

표1. 민간 표준과 관련 군사 표준의 매핑

민간표준	목 적	관련 (군사) 표준
FAR/CS 25.1309	Equipment, Systems & Installation	JSP 553 DS 00-56 Part 1
AMC 25.1309	System Design and Analysis	DS 00-56 Part 1 DS 00-56 Part 2
ARP 4754	Complex Avionics System	
ARP 4761	System Assessment Methods	ARP 4761
DO-254	Complex Hardware	DO-254
DO-178B	Software	DO-178B

2. 시스템 안전성에 대한 인식과 추세

시스템의 안전성과 관련된 산업계의 변화는 주로 안전성 관련 규제정책, 시스템의 안전성과 관련된 위험요소에 대한 사회적 인식, 증가된 제품의 기능성 요구, 그리고 시

스템 또는 소프트웨어 공학적인 기술의 발달과 같은 요인들이 주도해왔다.

대표적인 안전필수 시스템인 항공관제 시스템이나 철도 신호 시스템의 경우 그 안전성에 대하여 " rule-base" 규제로부터 " risk-based" 규제로 정책을 변화하는 추세이다[1]. 일반적으로 위험요소 기반의 규제정책들은 해당 시스템에 부여된 안전등급 목표에 맞추어 공급자들이 시스템 또는 서비스에 대하여 위험관리를 책임짐은 물론 그들이 위험관리를 적절히 수행했음을 입증하기 위한 데모를 요구하고 있다.

또한 시스템 및 소프트웨어의 요구사항은 다양한 측면에서 복잡해지고 있는데 그 대표적인 요인 중 하나는 기하급수적으로 증가하는 시스템의 복잡도에 있다. 현재 운용중인 군용 전투기에는 약 3~5MLoC의 소프트웨어 코드들이 탑재되어 있는데 차세대 전투기의 경우 1000만 라인 이상의 소프트웨어 코드가 탑재될 것이다.

그와 동시에 시스템 내부의 통합화 추세는 하부 시스템간의 인터페이스를 더욱 복잡하게 만들고 있다. 최근의 자동차 크루즈 시스템의 예를 들면 크루즈 기능이 정상적으로 작동하기 위해서는 엔진, 트랜스미션 그리고 브레이크 시스템과도 긴밀하게 인터페이스할 수 있어야 한다. 따라서 특정 기능을 분석하기 위해서는 연결된 모든 시스템에 대한 지식이 필요하며 다른 시스템의 변경에 의해 원래의 기능이 영향을 받기도 한다.

이와 같은 복잡하고 고도로 통합된 시스템일수록 그 정상적 동작 수행 과정에서 사용자의 역할이 차지하는 비중은 점차 줄어든다. 즉, 사용자의 운영상의 테크닉에 대한 의존도가 낮아질 뿐만 아니라 사용자가 시스템의 안전한 동작에 개입할 수 있는 여지도 줄어들었다. 반대로 시스템 자체에 대한 의존도와 시스템의 자율성이 급격히 증가하고 있다. 스텔스 기능으로 유명한F-117 전투기나 유로파이터 전투기의 경우 첨단 전투 성능 이면에 항공역학적으로는 매우 불안정한 기체를 보유하고 있기 때문에 더 이상 조종사의 감각이나 스킬만으로는 비행 자체가 불가능하다. 이들 전투기의 비행은 전적으로 컴퓨터시스템 통제하에 이루어진다고 보아야 할 것이다.

위와 같은 변화의 추세는 시스템을 보다 효과적으로 개발하는 방법, 그리고 완벽한 안전성을 보장함과 동시에 안전성에 대한 분석이 가능하도록 설계하는 방법을 종용하고 있다.

일각에서는 그 방편으로 UML과 같은 객체지향 설계기법을 제기하기도 했으며 실제로 객체지향 기법을 개발에 적용하려는 노력은 매우 보편화 되었다. 그러나 안전성 분석 및 검증 측면에서 객체지향 기법이 가지는 이점은 그리 크지 않다. 아직까지도 대부분의 프로젝트들은 검증 및 분석 수단으로서 테스팅이나 코드리뷰에 의존하고 있으며 객체지향 설계기법을 적용했다고 해서 이러한 전통적 기법으로도 완벽한 검증이나 분석이 가능한 것은 아니기 때문이다.

자동 코드 생성이나 체계적인 재사용(Systematic Reuse)을 통해 오류의 잠재 가능성을 줄이고자 하는 시도 역시 증가하고 있다. 그런데 문제는 기존의 테스팅이나 리뷰와 같은 안전성 분석 기법이 새로운 시스템에 적합하지 않다는 사실에는 대체로 공감하는 반면 자동 생성된 코드나 재사용된 컴포넌트의 안전성에 대한 증빙 방법 또는 그 증거자료의 재사용에 대해서는 명확한 대안이 없다.

3장에서는 대표적인 안전필수 시스템인 항공기의 안전성 인증 기준으로 정착된 RTCA/DO-178B를 살펴봄으로써 현재 어떤 기준들이 소프트웨어의 안전성 보장을 위해 적용되고 있는지 알아보고 그 문제점과 해결 방향에 대해 알아본다.

3. DO178B

3.1 DO178B의 역사

1980년 5월 RTCA(Radio Technical Commission for Aeronautics)사는 " SC-145(Special Committee 145) - Digital Avionics SW" 를 설립하고 소프트웨어 기반의 항공기 시스템 및 장비 개발을 위한 가이드라인 구축을 시작하였다.

같은 해 10월 EUROCAE(European Organization for Civil Aviation Equipment)는 유사한 성격의 WG-12(Working Group 12)를 구성하고 역시 항공 시스템의 소프트웨어 개발을 위한 ED-35 " Recommendations on SW Practice and Documentation for Airborne Systems" 의 발간을 준비하고 있었다.

두 기구의 성격 및 목표가 비슷했으므로 EUROCAE는 ED-35를 발간하는 대신 RTCA와 협조하며 상호 공통성을 갖는 가이드라인을 구축하고자 했고, 결국1982년 SC-145와 WG-12는 각각 RTCA/DO-178와 EUROCAE ED-12 발간하기에 이른다. 이후 1985년 RTCA는 SC-152를 구성하고 가이드라인의 개정판인 DO-178A 발간하였고 이에 EUROCAE도 ED-12A를 발간하였다.

1989년 RTCA는 다시 SC-167를 구성하고 DO-178A 개정 작업에 착수하였고 EUROCAE의 WG-12는 SC-167과 협력을 추진하였다. 그 결과 RTCA/DO-178B와 ED-12B가 발간된 것이다.

발간 기구의 성격과 목표의 유사성에서 알 수 있듯이 RTCA/ DO-178와 EUROCAE/ED-12는 기술적으로 동일한 문서이다[2].

3.2 DO-178B의 목적 및 개요

DO-178B는 항공기 관련 부품 및 탑재시스템의 소프트웨어 생산 가이드라인으로서 감항 (Airworthiness) 요구사항을 충족하기 위한 기준들을 제시하고 있다. 가이드라인의 대상이 되는 소프트웨어는 그 목표 기능을 정상적으로 수행함과 동시에 안전에 대한 신뢰성을 보장해야 한다.

감항증명(Airworthiness Certification)은 항공기의 사고 방지를 위해 운항에 적합한 자체 안정성과 신뢰성을 갖고 있는지에 대한 증명으로서 이와 관련된 규정들을 살펴보면 우리나라 항공법 13조, 미 연방항공청(FAA) 및 연방감항규정(FAR) 등이 있으며 미국, 일본, 중국, 영국, 프랑스, 인도네시아 등의 국가간에는 감항성 상호인정협정(BAA)이 체결되어 있다. 그러나 대부분의 감항 규정은 소프트웨어에 대한 직접 언급보다는 장비 또는 기체 자체의 안전성이나 설치, 운영과 관련된 조항들을 다루고 있는데 비해

DO-178B는 오직 소프트웨어의 안전성에 대한 인증 기준만을 제시한다.

DO-178B의 주요 내용은 다음의 세 영역으로 구분된다.

- 소프트웨어 수명주기 프로세스의 목적
- 목표 달성을 위한 설계 고려사항 및 활동 내용 기술
- 목표 충족 여부를 결정할 수 있는 증빙 기준의 기술

DO-178B는 소프트웨어의 감항성 인증 측면 기술을 위해 시스템 수명주기와 소프트웨어 수명주기의 연관관계, 소프트웨어 인증 프로세스 이해를 위한 설명들을 포함하고 있다. 그러나 시스템 수명주기 프로세스 자체에 대한 설명은 포함시키지 않았으며 소프트웨어 범위를 벗어난 시스템의 안전성 평가나 검증 프로세스, 엔진 인허가 등도 다루지 않는다.

또한 생산된 소프트웨어의 운영적 측면은 배제하였으며 사용자가 변경할 수 있는 데이터에 대한 인증이나 인적 자원, 조직 구조, 피 인증조직과 부품제조업자의 관계, 책임 분할 등 인적자원의 품질 관련 사항은 포함하지 않는다.

문서 구조를 간략히 살펴보면 소프트웨어 개발과 관련된 시스템적 측면 (Section 2), 항공기 및 엔진 인증의 개요 (Section 10) 그리고 소프트웨어 수명주기 프로세스 (Section 3 ~ 12)에 대한 내용으로 구성된다. DO-178B에서 다루는 수명주기는 계획, 개발, 검증, 형상관리, 품질보증, 인증 교섭 등 각 프로세스에 해당하는 개발 기준 및 활동들을 기술하고 있으며 수명주기 데이터 및 기타 고려사항들을 별도로 다루고 있다. 이 중 검증, 형상관리, 품질보증, 인증교섭 프로세스를 묶어 " 필수 프로세스 (Integral Process)" 로 분류하고 있는데 이 부분에 대한 설명은 3.4절에서 다시 하기로 한다.

3.3 소프트웨어 등급

DO-178B는 시스템 안전평가 프로세스를 통해 컴포넌트의 적정 등급 결정하고 있다. 문서 자체는 소프트웨어만을 가이드라인의 범위에 두고 있지만 그 소프트웨어의 등급을 결정하는 것은 시스템 수준의 안전평가 프로세스인 것이다.

안전평가 과정을 거친 안전필수 소프트웨어는 평가 결과로 선정된 원래 목표 등급보다 더 높은 등급으로 개발하는 것이 바람직하다. 수명주기 후반의 등급 상향조정은 더욱 어렵기 때문이다.

소프트웨어가 둘 이상의 시스템 실패조건에 영향을 줄 경우, 가장 높은 등급으로 선정되는데 진화적 시스템 설계 적용 시 소프트웨어 등급은 프로세스를 수행하면서 수정 가능하다.

시스템 기능이 둘 이상의 소프트웨어 컴포넌트로 병렬 개발된 경우 (둘 이상의 소프트웨어 컴포넌트가 비정상 작동해야 실패 조건이 발생) 최소한 하나의 컴포넌트는 시스템 내부 최상위 등급으로 지정되어야 하며 직렬 개발 (하나의 컴포넌트만 실패해도 실패조건이 발생)인 경우 모든 컴포넌트가 최상위 등급으로 지정되어야 한다. 여기서 주의할 점은 소프트웨어 등급이 실패율(failure rate)의 척도는 아니라는 것이다.

DO-178B는 시스템의 실패 조건을 다음의 5가지로 나누고 있는데 소프트웨어의 실패가 시스템의 어떤 실패 조건을 유발하는가에 따라 소프트웨어의 등급이 결정된다.

- Catastrophic: 지속적으로 안전한 비행 또는 착륙 불가
- Hazardous/Severe-Major : 시스템 안전성 또는 기능성의 격감, 승무원의 정상적 임무 수행 불가
- Major : 항공기의 성능 또는 승무원 임무 수행능력 저하로 인한 항공기 안전성 감소
- Minor : 심각한 안전성 저해 요건은 아니지만 안전성의 약간의 감소 또는 불편 초래
- No Effect : 항공기 성능, 승무원 업무 부하에 영향을 주지 않음

이에 따라 DO-178B가 정의하는 소프트웨어 등급 또한 5단계로 나누어진다.

아래 도표는 시스템의 실패 조건과 그와 연관된 소프트웨어의 등급간 상관 관계를 나타낸 것이다.

즉, 특정 소프트웨어의 실패가 " Catastrophic" 실패조건을 유발할 수 있다면 그 소프트웨어의 등급은 " Level A" 에 해당한다.

표 2. 시스템 실패조건과 소프트웨어 등급

Failure Condition		Software Level
Catastrophic	⇔	Level A
Hazardous/Severe-Major	⇔	Level B
Major	⇔	Level C
Minor	⇔	Level D
No Effect	⇔	Level E

3.4 DO-178B의 검증 프로세스

DO-178B의 정의에 따르면 검증 프로세스는 소프트웨어 개발 과정의 에러를 탐지하고 보고하는 과정이다. 단, 에러의 제거는 검증이 아니라 개발 프로세스에 해당하는 활동이다[2].

아래 도표는 각 수준별 검증대상과 그 검증활동이 목표로 하는 충족 범위를 보여준다. 이 때 각 활동에 사용되는 충족 수단은 기술적으로 오류가 없고 그 소프트웨어의 등급에 적합한 것이어야 한다.

표 3. 소프트웨어 검증의 수준별 검증 대상 및 충족 목표

Verification Target		Satisfaction Criteria
System Requirement	⇒	SW H-L Requirement
SW H-L Requirement	⇒	SW Archit. / L-L Req.
SW Archit. / L-L Req.	⇒	Source Code
SW Requirements	⇒	Executable Object Code

DO-178B에 명시된 검증 활동은 리뷰 및 분석, 테스트 케이스의 생성과 테스트 절차 수행 등으로 구성되는데 소프트웨어 요구의 구현물과 각 검증 활동 간에는 추적성이 보장되어야 한다. 이 때 요구사항과 테스트 케이스 간에는 " Requirement-Based Coverage 분석" 을, 그리고 코드와 테스트 케이스 간에는 " Structural Coverage 분석"

을 테스트의 요건으로 제시하고 있다. LynxOS-178을 개발한 LynuxWorks사의 자료에 따르면 Structural Coverage 충족을 위한 활동에 가장 많은 비용이 필요한 것으로 나타나는데 본 논문에서 지적하고자 하는 점도 바로 이 테스팅 과정의 비효율성과 불완전성이다.

표. 4 요구사항 기반 테스팅 방법론의 에러 탐지 범위

Methods	Objects	Error To Detect
HW / SW 통합 테스팅	High Level Requirements	인터럽트 핸들링, 실행시간 요건, HW 실패에 대한 SW 반응, Data Bus 및 자원 연결 문제, 자체 고장 탐지 실패, 피드백 루프 오류, 메모리관리 HW 비정상 제어, Stack overflow, 필드적재 SW 보증 메커니즘 오류 등
SW 통합 테스팅	SW Requirements / Architecture	변수/상수 비정상 초기화, 매개변수 전달 오류, 데이터 손상(전역데이터), 이벤트 및 동작 순서 오류
Low Level 테스팅	Low Level Requirements	알고리즘 오류, 비정상 루프, 로직 오류, 비정상적인 입력의 조합, 누락/손상 입력에 대한 비정상적 반응, 예외처리 실패, 계산 순서 오류, 알고리즘 정확도/성능 정밀도 오류

DO-178B에 명시된 대로 리뷰나 분석 활동을 수행하기 위해서는 피인증자(조직)가 소프트웨어의 요구사항과 아키텍처, 그리고 소스코드에 대한 정확성(accuracy), 완결성(completeness) 및 검증가능성(verifiability)을 제공해야 한다.

테스트 케이스의 생성에 있어서는 내부적인 일치성과 요구사항의 완결성에 대한 심층적인 분석이 필요하며 테스트 절차를 수행함에 있어 요구사항의 충족 여부를 시연할 수 있어야 한다.

3.5 DO-178B 관련 이슈 분석

DO-178B이 항공분야 소프트웨어의 안전성 인증에 있어 사실상 국제표준으로 자리 잡았지만 여전히 그 약점 중 하나는 소프트웨어 설계 또는 통합 등 각 단계별 수준에 맞는 안전성 분석을 위해 제공되는 구체적 가이드라인이 부족하다는 것이다[3]. 즉 인증 기준 충족을 위한 목표 중심적 표준의 성격이 강하기 때문에 어떻게 그 목표를 충족할 것인가는 전적으로 피인증자(조직)가 해결해야 한다.

또 한가지 산업계에서 가장 실질적인 이슈는 인증된 소프트웨어의 재사용과 그 인증 크레디트를 어떻게 재사용할 수 있는가 하는 것이다[4].

일반적인 테스팅 방법론으로는 대상 소프트웨어 내에 에러가 없다는 사실을 증명할 수 없고, 한 번 인증 받은 소프트웨어가 다른 시스템의 일부로 재사용되거나 이미 제품화된 소프트웨어가 약간이라도 변경이 진행된 채로 시스템 내부에 다시 적재되었을 때 시스템 전체를 다시 인증받아야 할지 아니면 바뀐 소프트웨어만 재검증하면 되는지 그 검증 범위에 대한 명확한 해답을 제시하기가 어렵다.

DO-178B 레벨 A 소프트웨어에 대한 코드 검증의 경우 모든 라인에 대해 Modified Condition/Decision Coverage(MCDC)로 검증되어야 하는데 이러한 테스트는 코드가 복잡도에 따라, 그 비용이 수백만 달러까지 늘어날 수도 있다.

이러한 비용, 일정의 부담, 그리고 그에 상응하는 완전성을 확보해야 하는 심각한 요구에 직면한 소프트웨어 개발 업체들은 다방면에서 독자적인 해결책을 제시하고 있다. 그러나 FAA/DER과 같은 인증기구 입장에서 시스템 개발 비용이나 일정 등은 안전성 인증에 있어 그다지 중요한 요소가 아니다[4]. 안전성의 보장만이 인증의 가부를 결정하는 기준인 것이다.

DO-178B 인증을 목표로 개발된 제품들 중 LynxOS-178의 예를 들면 어플리케이션과 디바이스 드라이버가 안전성이 요구되는 시스템 상에서 하나의 덩어리로 실행될 수 있도록 링크할 필요 없이 모듈 구성요소화 되어 실행되도록 해준다. 디바이스 드라이버를 수정 할 필요가 있을 때 전체 패키지 소프트웨어를 재인증 받을 필요가 없도록 어플리케이션과 드라이버를 독립시키고자 한 것이다. 이러한 특징을 지니지 않는 소프트웨어는 모든 어플리케이션, 커널, 라이브러리를 포함한 전체 시스템에 대하여 단순 시스템 업그레이드, 강화에 대해서도 막대한 비용을 들여 재검증, 재인증을 받아야 한다.

4. DO-178B 수명주기 데이터의 정형화를 통한 검증 및 인증 기준의 구체화

안전성의 확보를 위한 분석 및 검증 방법 중 현실적으로 가장 신뢰할 수 있는 방법은 정형기법이다. 소프트웨어의 규모나 복잡도에 따라 실제 적용에 따르는 제약사항들이 있기는 하지만 CC의 인증 기준으로 명시될 만큼 그 당위성이 인정받고 있다.

그러나 DO-178은 본문 12장에서 대체방법론(Alternative Methods)으로서의 정형기법을 개론적으로 언급하고 있을 뿐이다. 또한 인증을 위해 정형기법을 적용하고자 하는 시도는 계속되고 있으나 표준화되지는 못했다.

본 논문에서는 우선 정형기법을 적용할 수 있는 DO-178B의 검증 활동을 식별하고자 한다.

안전성과 보안성 인증의 통합을 위한 활동 중 DO-178B와 CC의 인증 기준을 통합하고자 하는 연구들이 진행되기도 했는데 DO-178B의 프로세스와 CC EAL5의 보증 클래스간 매핑을 통해 정형 또는 준정형(Semi-Formal) 기법이 필요한 활동을 식별한 연구결과도 있다[5].

현실적으로 DO-178 문서의 모든 문장을 CC 문서의 문장들과 대응시키기는 불가능하고, 그 목적이나 범위도 다른 부분이 많으므로 CC에서 정형기법을 요구하는 클래스만을 대상으로 그에 해당하는 DO-178B의 활동을 구분하기로 한다.

본 논문에서는 CC의 EAL(Evaluation Assurance Level)의 최고 수준인 EAL7을 기준으로 DO-178B의 프로세스

와 CC 클래스를 매핑함으로써 정형기법이 적용 가능한 DO-178B의 개발 활동을 식별하였다. 그리고 그 결과로서 DO-178B 11장 " 수명주기데이터" 의 보완을 제시하는데 DO-178B에서 각 수명주기 데이터와 그에 해당하는 활동들을 연결하기는 어렵지 않기 때문이다.

표. 5 CC EAL7 개발 클래스의 정형기법 요구수준

CC Assurance Class	Assurance Family	Formal Methods in EAL 7
Development	ADV_FSP	Semi-Formal
	ADV_HLD	Formal
	ADV_IMP	Formal
	ADV_INT	Formal
	ADV_LLD	Formal
	ADV_RCR	Formal
	ADV_SPM	Formal

DO-178B의 11장에서는 수명주기 프로세스의 각 활동에 대한 계획 또는 산출물이라고 볼 수 있는 수명주기 데이터의 개발 속성을 다음과 같이 규정하고 있다.

- Unambiguous : single interpretation
- Complete : all defined
- Verifiable : for correctness
- Consistent : no conflict
- Modifiable : changes in structure
- Traceable : to origin
- Form : for efficient retrieval
- Control : CC2 minimum

이러한 속성을 모두 충족할 수 있는 방법으로서 정형기법은 가장 적합한 수단이다.

표. 6 CC EAL7 개발 클래스와 DO-178B 매핑 결과

Action Element	CC Paragraph	DO-178B Paragraph	Remarks
Functional Specification	ADV_FSP. 2.1C	11.6	SW Requirement Standard
	ADV_FSP. 2.1D	11.9	SW Requirement Data
	ADV_FSP. 2.2C	11.9/11.14	SW Verification Results
	ADV_FSP. 2.3C	11.9	SW Requirement Data
	ADV_FSP. 2.4C	11.14	SW Verification Results
	ADV_FSP. 2.5.C	11.9	SW Requirement Data
High-Level Design	ADV_HLD. 2.1D	11.10	Design Description
	ADV_HLD. 2.2C	11.14	SW Verification Results
	ADV_HLD. 2.3C	11.7	SW Design Standard
	ADV_HLD. 2.4C	11.10	Design Description
	ADV_HLD. 2.5C	11.10	Design Description
	ADV_HLD. 2.6C	11.10	Design Description
	ADV_HLD.2.7 C	11.10	Design Description
	ADV_HLD. 2.8C	11.10	Design Description
	ADV_HLD. 2.9C	11.7	SW Design Standard
Low-Level Design	ADV_LLD. 1.1D	11.10	Design Description
	ADV_LLD. 1.2C	11.14	SW Verification Results
	ADV_LLD. 1.3C	11.7	SW Design Standard
	ADV_LLD. 1.4C	11.10	Design Description
	ADV_LLD. 1.5C	11.10	Design Description
	ADV_LLD. 1..6C	11.10a	Design Description
	ADV_LLD. 1.7C	11.10c	Design Description
	ADV_LLD. 1.8C	11.10c	Design Description
	ADV_LLD. 1.9C	11.10c	Design Description
	ADV_LLD. 1.10C	11.7	SW Design Standard
Representation Correspondence	ADV_RCR. 2.1C	11.14	SW Verification Results
	ADV_RCR. 2.1D	11.14	SW Verification Results
	ADV_RCR. 2.2C	11.14	SW Verification Results
Security/Safety Policy Modeling	ADV_SPM. 3.1C	11.7	SW Design Standard
	ADV_SPM. 3.1D	11.9	
	ADV_SPM. 3.2C	11.14	SW Verification Results
	ADV_SPM. 3.2D	11.14	SW Verification Results
	ADV_SPM. 3.2C	11.9	SW Requirement Data

위의 도표를 통해 현재 CC EAL7에서 요구하는 정형기법의 적용을 DO-178B에서는 어떻게 적용할 수 있는지를 판단할 수 있다. 정형기법의 적용은 두 가지 측면의 문제를 동시에 해결해준다.

먼저 개발자 혹은 피인증자 측면에서 문제시 되는 안전성 분석 및 검증 가이드라인의 부족 문제이다. 위와 같이 정형기법이 적용되어야 할 해당 패러그래프의 데이터를 명시하고 정형기법 적용에 대한 증빙 자료를 요구하는 경우 피인증자가 수행해야 할 활동방향과 그 산출물이 명확해진다. Statechart, Z를 비롯하여 현재 정형언어로 인정되는 언어들이 해당 데이터의 기술 수단이 될 것이며 정형언어를 사용한 설계와 검증의 결과는 모호함과 불완전성이 배제됨을 전제하고 있으므로 논란의 소지를 제거할 수 있다. 단, 정형기법 자체의 난해함과 기술적으로 적용 가능한 범위는 그 사용자가 감수해야 할 과제이다.

다음의 그림은 고도의 완전성을 요구하는 Ada Ravenscar Profile[6]을 적용하여 작성된 Ada 코드에 대하여 역공학적 방법을 동원하여 동일한 행위를 수행하는 Statechart로 변환한 예이다. 이와 같이 비교적 적용이 용이한 Statechart로 작성된 모델만으로도 그 행위의 명확성뿐 아니라 모델체킹을 통한 검증이 가능함으로써 DO-178B의 설계와 검증 기준 모두를 충족시킬 수 있다.

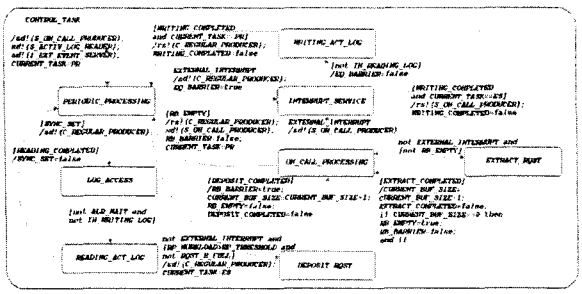

그림 2. Statechart를 이용한 시스템 행위 모델의 검증

인증자 측면에서도 더욱 명확한 기준으로 제시할 수 있는 이유 또한 정형검증 결과를 증빙자료로 요구할 수 있기 때문이다. 즉 주관적 판단이나 부분적 결과를 놓고 전체를 평가해야 하는 위험 부담을 줄일 수 있으며 피인증자 측의 제출 자료에 대해 그 충족 여부를 명확히 응답할 수 있다.

또 하나의 이점은 정형성을 기반으로 인증 받은 제품 혹은 소프트웨어 모듈의 재사용 가능성 확대이다.

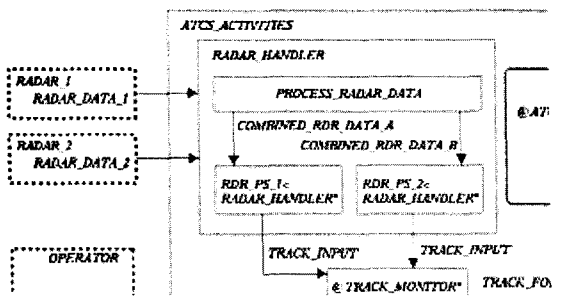

그림 3. Generic Chart를 사용한 재사용 모듈의 설계

위의 그림은 Statemate Magnum[8]으로 작성된 항공관제시스템의 예로서 특정 기능을 수행하는 모듈을 재사용이 가능하도록 작성한 것이다. 그림의 RADAR HANDLER라는 모듈은 다수의 외부 인터페이스에 대하여 변경 없이 사용할 수 있도록 " Generic Chart" 를 이용하여 작성되었는데 이 모듈 역시 내부에 Statechart를 포함하고 있어 그 행위의 정확성과 안전성을 정형검증 할 수 있다[9]. 특히 Generic Chart와 같이 독립성이 강한 재사용 모듈의 인증은 곧 이 모듈의 재사용에 대한 재인증의 필요성을 줄여준다.

5. 결 론

DO-178B는 항공분야 안전성 인증 기준으로서 사실상의 국제표준으로 정착했다. 그러나 안전성 분석과 검증에 대한 구체적인 가이드라인이 부족하고 그나마 국내에서는 이러한 기준의 적용 자체가 소극적인 현실이다.

본 논문에서는 DO-178B의 간략한 소개와 함께 DO-178B와 CC 인증 기준 비교를 통한 정형기법 적용 가능성을 제시하였다.

정형기법의 적용은 DO-178B의 목적이나 문서 구조에 심각한 영향을 주지 않는 상태에서 개발자와 인증기관 모두에게 보다 명확한 기준으로 작용할 수 있도록 도와주며 인증 절차의 투명성 향상에도 기여할 것이다.

6. 참고문헌

[1] John A McDermid, " Trends in System Safety: A European View?" , 7th Australian Workshop on Safety Critical Systems and Software, Adelaide, 2002

[2] RTCA DO-178B, RTCA, 1992

[3] Carolyn Salmon, " The Certification of Systems containing Software Developed using RTCA DO-178B, ERA, 2006

[4] Hoyt Lougee, " DO-178B Certified Software: A Formal Reuse Analysis Approach" , The Journal of Defense Software Engineering, 2005

[5] Jim Alves-Foss, et al.," Towards Common Criteria Certification for DO-178B Compliant Airborne Software Systems" Center for Secure and Dependable Systems, University of Idaho, 2002

[6] Alan Burns, et al., " Guide for the use of the Ada Ravenscar Profile in high integrity systems" , University of York Technical Report YCS-2003-348, 2003

[7] Chang-Jin Kim, Jin-Young Choi, " Transformation of the Ravenscar Profile based Ada real-time application to the verification-ready statecharts : Reverse engineering and Statemate approach" , SERP2006, 2006

[8] David Harel, Michal Politi, " Modeling Reactive Systems with Statecharts: The Statemate Approach" , I-Logix, 1999

[9] 김창진, 최진영, " 소프트웨어 설계 모듈의 재사용을 위한 Statemate 일반화 차트의 확장" , KCC2006