항공소프트웨어 설계단계와 감항인증에서의 정형검증 적용연구

장준하O*,최진영**
*고려대학교 정보대학 컴퓨터학과
**고려대학교 정보보호대학원
{jhjang*,choi**}@formal.korea.ac.kr

Applied research in formal verification of certificates of airworthiness and aviation software design phase

JoonHa JangO*, Jinyoung Choi**
*Department of Computer Science and Engineering, Korea University
**Graduate School of Information Security, Korea University

기술의 발전에 따라 첨단무기체계인 군용항공기는 전투임무 수행시 내장형소프트웨어(Embedded Software)를 통해 기체내의 항공전자 장치, 항법장치, 조종장치 등의 물리적 기계적 움직임을 제어하고 있으며, 내장형 소프트웨어의 비율은 점점 증가하고 있다.[1,2] 기체의 물리적 기계적 움직임이 내장형 소프트웨어에 의해 제어 되기 때문에 군용항공기의 전투능력 보존과 국방 목적의 수행을 위해선 먼저 내장형 소프트웨어 고유의 특성을 만족하고, 나아가 소프트웨어 안전성, 신뢰성, 보안성을 확보하는 것이 필요하다. 본 논문에서는 설계 단계에서 스케줄성에 대해 정형검증 하여, 내장형 소프트웨어의 실시간성, 결정성, 생존성을 보증하고, 이러한 과정을 통해 전체적인 소프트웨어 안전성, 신뢰성, 보안성을 향상시키는 방안을 연구하며, 추가로 2011년 발표된 항공 소프트웨어 표준인 DO-178C에서 요구하는 정형검증[3,4]을 적용한 국내 감항인증 표준 제정의 확대방안을 연구한다.

Ⅰ. 서 론

현재 F-35, F-22 와 같은 최신의 군용항공기의 소프트웨어 의존도는 점점 증가하고 있으며, 이는 대략 80%, 90%의 의존도를 가지는 것으로 조사된 바 있다.[5] 또한 이러한 군용항공기의 소프트웨어는 내장형 소프트웨어 시스템의 특성을 지닌다. 군용항공기의 물리적 기계적 움직임은 소프트웨어를 통해 제어되기 때문에 소프트웨어의 안전성 신뢰성 보안성 보증은 군용항공기의 전투임무능력 보존과 국방목적 수행을 위해선 필수적이다. 또한 이러한 소프트웨어 속성을 확보하기 위해선 먼저 설계단계에서 명세한 내장형 소프트웨어의 특성을 만족해야하며, 명세한 속성의 만족을 검증하기 위해 설계단계에서 스케쥴성에 대한 정형검증을 통해서 내장형 소프트웨어의 실시간성, 결정성, 생존성을 보증할 수 있고, 이를 통해서 효율적이고 신속하게 소프트웨어의 안전성, 신뢰성, 보안성을 향상 시킬 수 있음을 연구하고 극복 추가로 군용항공기 소프트웨어의 안전성, 신뢰성, 보안성 확보를 위해서 정형기법의 적용을 요구하는 국제 항공소프트웨어 표준인 DO-178C[3]를 반영하여, 국내 감항인증 표준을 제정하기 위한 확대방안을 연구한다.

Ⅱ. 기본 속성 정의 및 특성 정의

2.1 소프트웨어 안전성, 신뢰성, 보안성

먼저 본 논문에서는 요구사항 명세과정에서의 의사소통의 정확성과 신뢰성 향상을 위해서 검증하고자 하는 소프트웨어 속성에 대한 정의를 선행하고자 한다. 소프트웨어 안전성이란 소프트웨어가 소프트웨어 실패시에 사람, 사회, 환경에 피해에 줄수 있는 피해를 방지할 수 있는가에 대한 판단 척도[6]를 말하며, 신뢰성이란 소프트웨어가 사용자가 명세한데로 기능을 수행하는 지에 대한 성질[6]이다. 마지막으로 보안성이란 명세된 입력을 제외한 의도적이건 비의도적인 침범에 대한 저항성[6]을 말한다. 현재 기술의 발전으로 인해 군용항공기는 하드웨어 중심에서 소프트웨어 중심으로 변하고 있으며, 이러한 양상을 통해서 볼 때, 현재 군용항공기는 살상능력을 보유한 공중 이동형 내장형 소프트웨어 시스템으로 분류 가능하며, 이러한 시스템의 목적인 영공방위와 평화유지를 위한 전투임무 수행을 위해선, 위 3가지 속성들의 보증은 필수적이다.

2.2 내장형 소프트웨어의 특성

본 논문에서는 군용항공기 내장형 소프트웨어에 대한 설계단계에서 모델을 명세하고 스케쥴성을 정형검증함으로써, 내장형 소프트웨어의 특성 중 실시간성, 결정성, 생존성을 보증할 수 잇으며, 이를 통해서 궁극적으론 안전성, 신뢰성, 보안성을 향상시킬 수 있음을 연구하였다. 이러한 연구 수행에 선행 과정으로 모델의 명세와 정형검증 과정의 정확성, 신뢰성 확보를 위해서 검증하고자 하는 내장형 소프트웨어의 특성을 다음과 같이 정의한다. 실시간성이란 올바른 입력에 대한 결과가 미리 결정된 시간 전에 정확하게 나와야 하는 성질[1]을 말하며, 결정성이란 수행 시간 및 자원의 사용량이 예측가능성[1]을 말하며, 이는 수행시간의 최대값, 최대 스택 사용량, 최대 메모리 사용량, 폴링 주기 등의 자원 사용량은 설계 시점에 명시되어야 함을 의미한다. 반응성은 외부의 환경 변화에 반응하여 동작하는 성질1)을 말하고, 생존성은 발생하지 않을 이벤트를 기다리면서 무한히 멈춰있거나, 동작의 중단을 방지하는 성질[1] 마지막으로 이질성은 다양한 하드웨어 및 소프트웨어와 상호작용하며 동작하는 성질[1]을 의미한다.

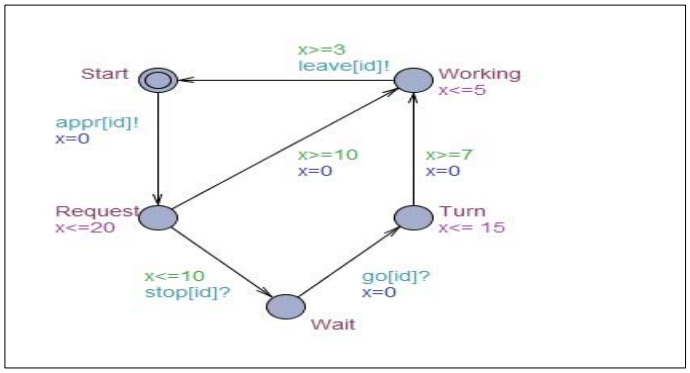
Ⅲ. 설계 단계에서의 스케줄성 정형검증

3.1 UPPAAL.

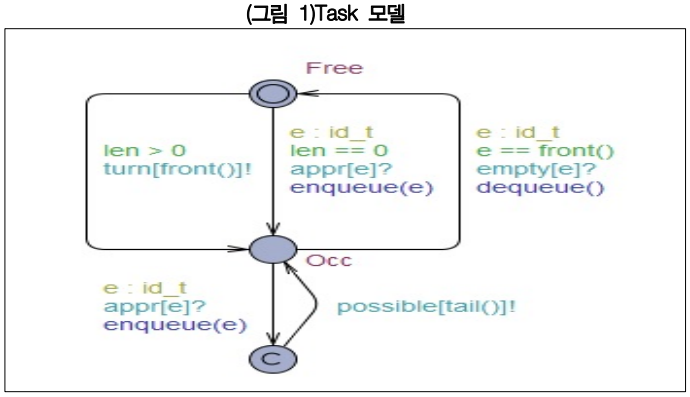
본 논문에서 검증을 위해 사용한 UPPAAL은 실시간(Real-time)시스템 모델검증 도구로써, 이는 타임드 오토마타(Timed automata)와 TCTL(timed computation tree logic)을 사용하며, 이때, 타임드 오토마타는 시스템 행위를 모델링하기 위한 명세 수단이며, TCTL은 모델의 시스템 검증 속성을 명세하는 데 사용한다. UPPAAl 도구는 크게 정형검증기(Verifier)와 시뮬레이터(Stimulator)로 구성되어 있다. 속성 검증 이외에도 반례를 생성해 주며, 이를 통해 속성 검증시의 논리적 오류를 추적가능하다. 시뮬레이션을 통해 각각의 모델들의 관계와 시간에 따른 상태의 전이, 속성의 변화등을 확인 가능하다. 설계단계 정형검증 도구로써 UPPAAL을 선정한 이유는, 먼저 도구가 가지는 실시간성 때문이다. Clock 변수를 통해 모델링 하기에 결정성을 검증가능하며, 둘째로는 Location에서 Invariant 설정을 통해 , 실시간성을 검증가능하기 때문 이다. 셋째로 모델의 생존성 도구에서 지원하는 DeadLock 점검을 통해 검증 가능하기 때문이다. 또한 Simulator를 통해 시각적으로 state의 전이를 쉽게 확인 할 수 있다는 장점이 있기 때문이다. 추가로 정형검증 중 모델검증의 장점으로는 원하는 속성을 불만족 하였을 때 반례를 통해 논리적 오류를 추적가능하다는 점이다.

3.2 UPPAAL을 이용한 스케줄성 정형검증 모델.

본 연구에서는 설계 단계에서의 스케줄성을 위해서 3가지 속성의 검증을 목표로 하였다. 첫째, 임베디드 소프트웨어 정해진 시간내에 입력을 보내는지에 대한 실시간성 검증, 수행 시간의 예측가능성인 결정성, 마지막으로 소프트웨어가 스케줄링시 기아현상이 발생하거나, 동작이 멈추지 않음을 검증하는 생존성을 검증하였다. 이를 위해서 아래와 같은 모델을 명세하였으며, 이 모델은 설계단계에서의 정형검증 적용의 가능성과 이러한 방법 적용의 필요성 그리고 장점을 보이기 위한 모델이다. 모델의 명세시에 구체적인 스케줄링 주기와, 스케줄링 정책은 배제하였다. 이 모델은 여러 작업들의 동시에 수행 될 때의 스케줄링을 명세 하였으며, 작업 시작(Start), 자원할당 요청(Request), 대기(Wait), 작업가능(Turn), 작업중(Working)을 Location으로 모델링하고, 시간변수(Clock)는 x로 주었으며, 해당 Task의 수행의 실시간성 검증은 Invariaint를 통해 명세하였다.

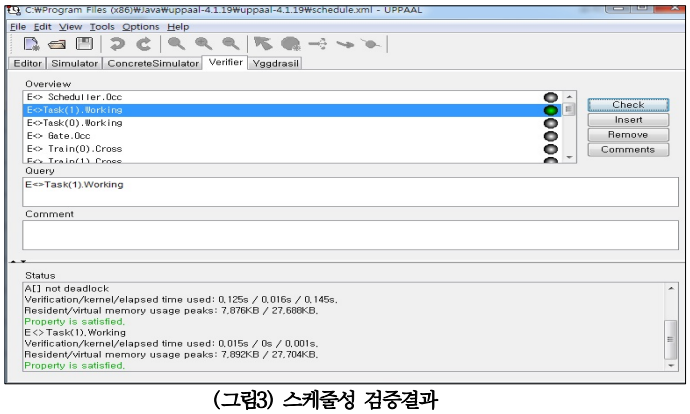


(그림 1)Task 모델



(그림2) Scheduller 모델

실제 검증시, 총 5개의 Task의 스케줄성을 검증하였으며, 속성을 검증하기 위해서 사용한 TCTL로는 실시간성은 E<>Task().Working을 통해 검증하였고, 결정성은 E<>Task().Working and E<>Task().Wait를 통해 검증하였다. 마지막으로 생존성은 A[] not deadlock을 통해 검증하였다. 아래의 그림은 검증의 결과를 보여주며, Property is satisfied는 속성의 만족을 보여준다. 만약 모델이 속성을 만족하지 않을 경우는 반례를 생성하며, 이러한 결과를 바탕으로 재모델링이 가능하다. 이를 통해 내장형 소프트웨어에 대한 UPPAAL 설계단계에서의 스케줄성 검증 가능성을 보였다.



(그림3) 스케줄성 검증결과

3.3 설계 단계에서의 스케줄성 정형검증의 장점과 단점

설계 단계에서 내장형 소프트웨어의 스케줄성을 정형검증함으로써 얻을 수 있는 장점은 다음과 같다. 첫째로 스케줄성의 정형검증 전 요구사항 명세 과정에서 속성에 대한 정의를 선행하기 때문에 소프트웨어는 보다 정확하게 요구사항을 반영하게 된다. 추가로 설계단계의 정형검증 시에 하드웨어 속성과 시스템 요구사항을 명확히 반영한다면, 기반 하드웨어와의 최적화에 유리하며 전체 시스템의 안전성, 신뢰성, 보안성이 향상된다.[3,8] 둘째로는 경제성이다. 스케줄성 검증을 설계단계에서 적용 함으로써 소프트웨어의 결함을 사전에 파악할 수 있으며 이를 통해 결함으로 발생하게 되는 비용이 감소한다. 셋째 효율성이다. 최근 항공소프트웨어는 개발시 모델기반 개발[7] 기조로 설계 단계 이후 개발 과정에서 소프트웨어모델을 기반으로 개발하며, 감항인증시 이모델에 대한 정형검증이 요구하는데, 설계 단계시 정형검증을 수행 함으로써 이후 단계의 소프트웨어 모델에 있어서 보다 신속하게 작성 이 가능하며, 요구사항 반영에 있어서 일관성있는 추상화가 가능하다. 설계 단계에서의 스케줄성 정형검증의 적용이 가지는 장점은 위의 3가지이며, 단점은 다음과 같다. 첫째로는 설계단계에서의 모델 검증을 완료 한 후 구현과정에서 차이가 발생한다는 것이다. 둘째로는 모델검증시 명세를 하기 위해선 기반 하드웨어에 대한 이해가 선행되어야 한다는 점이다. 이러한 단점들을 해결하기 위해선 소프트웨어개발 절차 및 연구 협업과 하드웨어 전문가의 모니터링을 통해 충분히 개선 가능한 하며, 이러한 방법 적용의 성공사례로는 미국 국방부에서 진행중인 HACMS프로젝트[9]가 있다.

Ⅳ. DO-178C를 반영한 국내감항인증 표준 연구

현재 국내 감항인증은 방사청의 “군용항공기 표준감항인증기준에 관한 고시”의 내용을 통해 이루어 지고 있으며 이의 주된 기반은 DO-178B이다. 소프트웨어 기술의 발달에 따라 소프트웨어 개발 방식변화와 함께 보안성 고려에 대한 필요성이 증가하고 있으며, 이에 따라 DO-178C,MISRA-C 개정 등의 국제 표준의 변화가 있다. 국내 감항인증에서도 이러한 변화를 반영할 필요성이 대두되고 있다. DO-178C는 DO-178B와의 차이로는 무엇보다 첫 번째로는 소프트웨어 단계 전반에서 요구사항 의사소통과정에서 명확한 언어와 용어의 사용을 통한 일관성을 유지할 것을 강조하였고 다음으로는 소프트웨어 인증에 사용하는 도구에 대해 등급을 분류하고 자격을 부여한 점, 모델기반의개발과 객체지향의 추세를 반영한점, 소스코드 검증 시 추적성의 강조, 정형기법의 사용 등이 있다. 현재 국내 감항인증표준에서는 위의 소프트웨어 기술변화 추세를 반영하고 현행 용어집10)의 소프트웨어 속성 추가 정의를 통해 의사소통의 명확성을 개선하고 소프트웨어 안전성, 신뢰성, 보안성을 향상해야 한다.

Ⅴ 결론

본 논문에서는 군용항공기 내장형 소프트웨어의 설계 단계에서의 스케줄성의 정형검증의 적용 필요성과 이를 통한 장점과 극단점, 단점의 해결 방안 그리고 항공소프트웨어에 대한 국제표준인 DO-178C와 이에 내재된 소프트웨어 기술 발전이 반영한 국내 감항인증 표준 제정의 필요성에 대해 제안하였으며, 향 후 이러한 일련의 과정과 방법론의 적용한 군용항공기 소프트웨어의 안전성, 신뢰성, 보안성 향상과 확보를 위한 추가연구가 필요하다.

참고문헌

[1]안보경영 연구원 “무기체계 SW 국산화 실태분석 및 확대방안연구” 2011.12

[2]Jack Ferguson, Crouching Dragon, Hidden Software : Software in OdO Weapon System,2001

[3]RTCA DO-178B,“Software Considerations in Airborne Systems and Equipment Certifications”, December 1, 1992

[4]RTCA DO-178C,“Software Considerations in Airborne Systems and Equipment Certifications”, December 13, 2011

[5]RTCA DO-333,“Formal Methods supplement to DO-178C and DO-278A”

[6]The Method Framework for Engineering System Architecture 2009

[7]Sommerville Software Engineering, 10th Edition 2016

[8]박무혁, “ 고신뢰도 소프트웨어 인증규격 및 시험 기술 연구”, 한국 항공우주연구원 2006

[9]Gabriella Gigante and Domenico Pascarella, Formal Methods in Avionic Software Certification : The DO-178C Perspective “ 5th International Symposium, ISoLA 2012 Heraklion, Crete, Greece, October2012, Proceedings, Part II

[10]DRPA,BroadAgencyAnnouncement“HighAssuranceCyberMilitarySystem” 2012

[11]방사청,무기체계 소프트웨어 개발 및 관리 매뉴얼 2014

[12]방사청,군용항공기 표준감항인증기준에 관한고시 2012

[13]김창진, 최진영 “정형기법을 적용한 DO-178B 안전성 검증 및 인증 기준 개선”, 한국정보과학회 가을 학술발표 논문집, 2006

[14]Alessandro Biondi, Giogio C.Buttazzo, “Schedulability Analysis of Hierachicla Real-Time Systems uder Shared Resources”, IEEE

[15]김지연,장준하,최진영 “보안성이 강화된 MISRA-C 기반 무기체계 소프트웨어 코딩 규칙 연구, 한국정보보호학회 하계 학술대회 논문집