Journal of the Korea Academia-Industrial cooperation Society
Vol. 23, No. 8 pp. 466-471, 2022

https://doi.org/10.5762/KAIS.2022.23.8.466
ISSN 1975-4701 / eISSN 2288-4688

인공위성 탑재 소프트웨어의 결함 주입시험 도구 개발

배지훈[1], 최규옥[2], 김형신[3*]
[1]만도, [2]슈어소프트테크, [3]충남대학교 컴퓨터공학과

Development of Fault Injection Testing Tool for Spacecraft Flight Software

Ji-Hoon Bae[1], Gyu Ok Choi[2], Hyungshin Kim[3*]
[1]Mando
[2]Suresofttech
[3]Department of Computer Science and Engineering, Chungnam National University

요 약 인공위성의 탑재 소프트웨어는 고신뢰성을 갖는 임베디드 소프트웨어의 특징을 갖는다. 탑재 컴퓨터의 성능이 높아지면서, 탑재 소프트웨어는 보다 복잡한 임무를 수행하게 되었다. 탑재 소프트웨어는 소프트웨어 개발 표준에 따라 개발 주기 전체적으로 고신뢰성 소프트웨어 개발기법을 적용하고 있다. 이러한 고신뢰성 소프트웨어는 개발 과정에서 다양한 형태의 테스팅을 검증과정으로 수행해야 하는데, 본 연구에서는 인공위성 탑재 소프트웨어의 강인성을 검증할 수 있는 결함주입 테스팅 도구를 개발하였다. 개발한 결함주입 도구는 NASA에서 개발한 시뮬레이션 환경인 NOS3를 지원하며, 하드웨어용 인터페이스로 SPARC프로세서 기반 탑재 컴퓨터에서 많이 사용하는 GRMON 인터페이스를 지원하도록 하였다. 개발한 도구는 테스트 시나리오를 작성하고, 이에 따라 런타임에 변수 값을 변경하거나, 프로그램에 고장 유발 연결함수를 호출하는 기능을 가지고 있다. 개발한 도구를 STF-1 인공위성의 탑재 소프트웨어에 적용하여 실제 개발과정에 사용 가능성을 확인하였다. 이 연구에서 개발된 결함 주입시험 도구는 현재 인공위성에서 자주 사용되는 LEON프로세서기반 탑재 컴퓨터의 소프트웨어의 결함 주입시험 과정에 적용할 수 있다.

Abstract Spacecraft flight software is embedded software with high reliability. As the performance of flight computers is increasing, flight software is playing a more complex role in the mission success. Flight software is developed in accordance with development standards that impose reliability assurance throughout the development cycle. This kind of highly-reliable software should perform various verification processes, including tests. In this study, we developed a fault injection testing(FIT) tool that can verify the robustness of flight software when there are faults in the system. Our tool supports NASA's NOS3 simulator and the GRMON debug interface, which is a standard debug interface for SPARC processors. To inject a fault, a test scenario should be prepared, and faults are injected by either modifying variable values or redirecting execution flow. With the developed FIT tool, we performed a fault injection test on the NOS3 simulator target. Our fault injection tool can be applied to flight software testing when LEON processors are used as the flight computer.

Keywords : Fault Injection Test, Software Testing Tool, Satellite Software, Flight Computer, Software Testing

본 연구는 국방과학연구소의 "저궤도 고기동 정찰용 위성체 탑재 소프트웨어 신뢰성 검증방안 연구"(관리번호:UD190017FD)의 지원을 받아 수행되었음.
*Corresponding Author : Hyungshin Kim(Chungnam National Univ.)
email: hyungshin@cnu.ac.kr
Received May 18, 2022 Revised June 17, 2022
Accepted August 3, 2022 Published August 31, 2022

1. 서론

인공위성의 탑재 컴퓨터는 위성 운영의 두뇌와 같은 역할을 수행한다. 탑재 컴퓨터의 소프트웨어는 일종의 실시간 임베디드 소프트웨어로 개발 과정을 통해 고신뢰성을 보장해야 한다. 위성 소프트웨어와 같은 고신뢰성 소프트웨어의 개발은 개발과정 전 기간에 걸쳐 표준화된 규격에 의해 통제되며, 요구사항 분석 단계부터 통합 시험 단계까지 다중 검증과정을 통해 신뢰성을 확보하게 된다[1,2].

위성체 탑재 소프트웨어의 신뢰성 검증 사례 연구로는 Ariane 5 소프트웨어의 정적분석을 이용한 검증 프로세스 연구[3], NASA 화성탐사선의 플래시 파일시스템 신뢰성 분석을 수행한 연구[4], 우주왕복선 탑재 소프트웨어의 품질을 검증한 연구[5]가 있다. [4]에서는 랜덤 테스팅과 모델체킹, 단방향 테스팅 등의 기법을 이용하여 신뢰성을 분석하였다.

탑재 소프트웨어를 시험하기 위해서는 STB(Software Test Bed)를 구축하고, STB 상에서 시험 자동화 도구를 이용하여 테스팅을 진행한다. 본 연구에서는 국내에서 진행되는 인공위성 탑재 소프트웨어의 신뢰성 향상을 위해 STB에 적용 가능한 결함 주입시험 자동화 도구를 개발하였다. 결함 주입시험 도구는 슈어소프트테크(SuresoftTech)사에서 개발을 진행하였으며, 시뮬레이터 기반의 탑재 소프트웨어 STB와 GRMON[6] 인터페이스를 사용하는 탑재 컴퓨터 개발 보드에서 모두 적용 가능하도록 개발되었다.

시험도구의 프로토타입을 사용하여 NASA의 cFS 탑재 소프트웨어 개발환경인 NOS3 시뮬레이터에서 STF-1[6] 위성의 탑재 소프트웨어를 대상으로 결함주입 시험을 진행하였다. 시험 결과 랜덤 테스팅에서 검출하기 어려운 예외적인 오류 상황을 성공적으로 런타임에 주입할 수 있었으며, 결함이 주입된 후에 탑재 소프트웨어에서 오류를 검출하지 못하고 오동작하는 경우를 발견할 수 있었다.

이 논문의 기여도는 다음과 같다.

1) 가상머신(VM)에 결함을 주입할 수 있는 결함주입 시험 도구를 개발하였다. 이 도구는 테스트 시나리오를 설정하면, 이에 따라 런타임에 변수값을 변경하거나 제어흐름을 변경해준다.
2) 인공위성 탑재 컴퓨터로 사용되는 SPARC 프로세서 계열의 디버그 인터페이스와 연동되는 GRMON 기반 하드웨어를 위한 결함주입 도구를 개발하였다.
3) NASA의 표준 탑재 소프트웨어인 cFS기반 STF-1 탑재 소프트웨어에 대해 결함주입 시험을 수행하였으며, 그 결과로 시스템에서 주입된 결함을 검출하지 못하고 오동작을 하는 상황을 발견하여, 개발한 도구의 적용 가능성을 검증하였다.

이 논문의 구성은 다음과 같다. 2장에서는 결함 주입 시험 방법과 탑재 소프트웨어 개발 환경에 대해서 설명한다. 3장에서는 본 연구에서 개발한 결함 주입 시험도구에 대해 설명하고, 4장에서는 개발한 도구의 프로토타입을 이용하여 NASA의 NOS3 시뮬레이터에서 STF-1 탑재 소프트웨어의 결함 주입 시험을 수행하고 그 결과를 분석한다. 5장에서는 결론을 맺는다.

2. 배경지식

2.1 결함 주입시험 방법

결함 주입시험은 시스템이 정상적으로 동작하는 동안 인위적인 결함을 강제로 발생시켜 예외 상황에 대한 시스템의 반응을 테스트하는 시험방법이다. 주로 고신뢰성 시스템에서 발생 가능한 대표적인 결함에 대해 시스템이 강건한지를 검증하기 위해 사용한다. 시스템은 안전상태(Safe State)에서 결함이 주입되면 오류상태로 전환되고, 고장(Failure)이 발생한 후에 사고로 이어지게 되는 상태 흐름을 갖는다. 결함 주입시험은 안전 상태에 결함을 주입하고, 그 결과로 시스템이 사고로 이어지는지, 에러를 검출하고 시스템을 안전상태로 복원시키는지 검증한다.

결함주입 시험을 위해 결함 대상을 결정하고 이에 따른 결함 유형과, 결함 조건(발생 시점 및 횟수)을 정의한다. 타겟 시스템에 결함을 주입하고 동작 상태를 모니터링 한다. 동작 상태 모니터링 결과 오류가 발견된 경우 하드웨어 또는 소프트웨어를 수정하여 오류를 제거한다.

2.2 탑재 소프트웨어 개발환경

본 연구에서는 탑재 소프트웨어의 신뢰성 검증을 위해 NASA의 NOS3(NASA Operational Simulator for Small Satellites)[7]를 STB로 사용하였다. NOS3는 NASA의 소형 위성인 STF-1(Simulation-to-Flight 1)의 소프트웨어 개발 과정에서 위성을 시뮬레이션하기 위해 개발된 오픈 소스 시뮬레이터이다. NOS3는 지상국 시스템(COSMOS), 우주 환경 시뮬레이터(42), NOS 엔

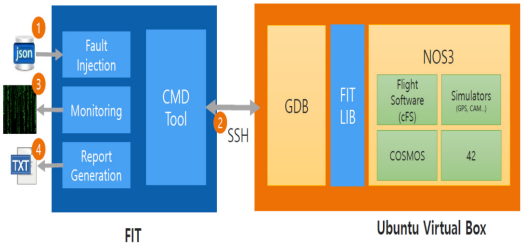
진 및 라이브러리, NOS 서버, 탑재 소프트웨어인 cFS 태스크[8], 운영체제인 cFE, OSAL로 구성된다. COSMOS는 지상국에서 명령을 보내는 시스템이고, 42는 위성의 이동을 계산하여 그것을 지도로 보여주는 시뮬레이터이다. 이 논문에서 결함주입 시험의 대상으로 하는 탑재 소프트웨어는 STF-1위성의 탑재 소프트웨어로, NASA의 표준 탑재 소프트웨어인 cFS를 운영체제로 사용한다. STF-1위성의 개발과정에서 테스트 베드로 NOS3를 사용하였으며, NOS3 시뮬레이터에 STF-1 탑재 소프트웨어의 일부가 포함되어 공개되고 있다. 본 연구에서는 STF-1위성의 cFS 태스크들을 NOS3 환경에서 실행하고, 개발한 결함 주입도구가 NOS3 환경에 결함을 주입할 수 있도록 하였다.

NOS3는 자력계, 전력 시스템, GPS, 카메라와 같은 하드웨어를 시뮬레이션할 수 있으며, 실제 하드웨어 위성 서브시스템을 시뮬레이터와 연동할 수도 있다. 이 시뮬레이터는 하드웨어가 개발되지 않은 시점에서 시뮬레이터를 탑재 소프트웨어의 테스트 베드로 이용하여 탑재 소프트웨어 개발 시에 단위 테스트 및 소프트웨어 통합 테스트에도 활용할 수 있도록 개발되었다.

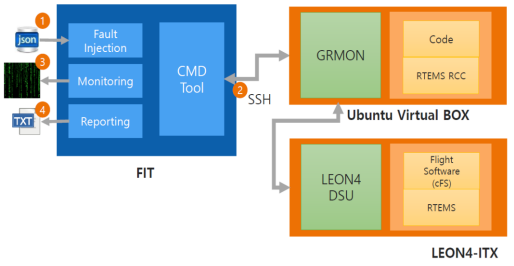
3. 결함 주입 시험도구 개발

인공위성 탑재 소프트웨어 시험에 사용할 목적으로 개발한 결함주입 도구는 FIT_CMD라고 부르며, 슈어소프트테크사가 보유하고 있는 결함주입 도구인 SECURESCROLL_FIT[9]과 달리 사용자 인터페이스를 생략한 커맨드 모드 형태로 개발되었으며, 상용 도구와 유사한 수준의 내부 기능 구현으로 결함 주입 시험을 지원한다.

이번에 개발된 도구는 기존의 GDB기반의 결함 주입 기능 외에, 탑재 소프트웨어에서의 다양한 결함 주입 시나리오를 모사할 수 있도록 대상 함수의 연결함수(hook function)를 이용하여 결함 주입 코드가 실행될 수 있는 기능을 추가하였다. FIT_CMD 도구는 GDB와 GRMON과 같은 디버그 인터페이스를 원격으로 제어하여 결함을 주입하고 모니터링하는 기능을 구현하였다. 결함의 주입과 모니터링 방법은 시나리오 파일에 정의할 수 있으며, JSON파일 포맷을 사용한다. 결함 주입 후 타겟 시스템의 실행 결과는 호스트에 연결된 모니터 창으로 확인할 수 있으며, 테스트 결과는 보고서로 생성되어 저장된다.



(a) FIT_CMD tool for NOS3 simulator



(b) FIT_CMD tool for GRMON interface

Fig. 1. FIT_CMD tool interactions

결함 주입시험은 호스트 컴퓨터의 커맨드 창에서 FIT_CMD를 실행하면, 빌드 시스템과 대상 함수의 연결함수(hook function) 결함 주입 코드가 저장되어 있는 가상머신으로 SSH연결을 한다. SSH를 통해 교차 개발환경의 빌드 시스템을 제어해서 타겟을 위한 바이너리 이미지를 자동으로 생성해준다. 빌드한 바이너리 이미지를 타겟에 로딩하고 실행하는 작업은 타겟에 따라 달라진다. 시뮬레이터 타겟인 경우에는 GDB를 연결하여 제어하고, 타겟 시스템에는 GDB와 고장주입을 위한 라이브러리 코드가 함께 저장되어 있다.

FIT_CMD는 가상머신에서 타겟의 실행 코드를 빌드 후 NOS3 시뮬레이터에 바이너리를 적재하여 실행하는 작업을 제어한다. Fig. 1(a)는 FIT_CMD 도구와 NOS3 시뮬레이터와의 결함주입 인터페이스를 보여준다. 타겟 시스템이 실제 타겟 하드웨어인 경우에는 GRMON의 원격 GDB를 이용하여 제어하고 우분투 가상머신에는 GRMON과 GDB와 결함 주입을 위한 연결 함수(hooking function) 코드가 함께 저장되어 있다. FIT_CMD는 가상머신에서 타겟의 실행코드를 빌드 후 하드웨어 타겟 시스템에 바이너리를 LOAD하고 실행하는 작업을 제어한다. Fig. 1(b)는 FIT_CMD 도구와 실제 타겟 하드웨어인 LEON4 탑재 컴퓨터에서 GRMON 인터페이스를 이용하여 결함을 주입하는 과정을 보여준다.

```
"hookFunction": [
  {
    "hookFunctionName": "HK_SureTest",
    "hookEnableType": {
      "type": "counter",
      "condition": {
        "comparisonVariableName": "FIT_hookEnable1",
        "comparisonOperators": "==",
        "comparisonValue": "0",
        "comparisonVariableDataType": "Integer"
      },
      "counter":{
          "countLimit": "2"
      }
    },
    "faultArgs":[
      {
          "faultValueDataType": "Integer",
          "faultValue": "1"
      }
    ]
  }
```

Fig. 2. Hook function scenario in JSON format

결함주입 시험을 정의한 시나리오 파일은 결함 주입 시나리오 기본설정, 결함 주입 방법 설정, 결함 모니터링 설정의 세 가지 설정으로 구성된다. Fig. 2는 연결 함수(hook function) 유형으로 결함을 주입하기 위한 시나리오 파일을 보여준다. HK_SureTest 라는 대상 함수가 2회 실행될 때 까지는 정상적인 함수 코드가 실행된다. 이후 3회째 실행될 때는 연결 함수(hook function)의 결함 주입 코드가 실행된다. FIT_CMD 도구에서는 결함 주입 조건 즉 counter 조건에 만족하게 되면 연결 함수(hook function)에서 결함 주입 코드가 실행될 수 있게 FIT_hookEnable1 변수를 true로 세팅해 준다. counter 유형을 이용하면, 프로그램의 흐름에서 특정 조건를 일정 횟수만큼 실행한 후에 결함을 주입하게 된다.

4. 결함 주입 도구를 사용한 테스팅

본 연구에서 개발한 FIT_CMD 도구의 적용성을 확인하기 위해 STF-1 위성의 탑재 소프트웨어를 대상으로 결함 주입 시험을 수행하였다. STF-1 위성 탑재 소프트웨어는 NASA의 표준 탑재 소프트웨어인 cFS를 활용하여 개발되었으며, cFE 커널 위에 HK(Housekeeping), HS(Health and Safety), TO(Telemetry Output), CI(Command Injection)의 cFS 앱들과 다수의 cFE 서비스 앱들을 탑재하였으며, 이들을 NOS3 시뮬레이터 환경에서 실행하면서 결함을 주입하였다. 테스팅에서 주입한 결함들은 탑재 소프트웨어에서 자주 발생하는 결함들과 STF-1 미션에서 정의하고 있는 주요 결함들로 구성하였으며, 주요 테스팅 항목들과 이들이 주입되었을 때 에러 검출방법은 Table 1과 같다.

STF-1 탑재 소프트웨어에는 HS, EVS, HK, SB등 주요 태스크들에 오류와 결함을 검출하고, 시스템을 안전상태(Fail-Safe)로 복원하는 루틴이 포함되어 있다. HS 태스크는 cFS 태스크들이 정상적으로 실행이 되는지를 확인하는 App 실행 카운터를 관리하고 있어서 태스크들이 오동작 하는 경우에 이를 감지해서 HS 태스크에서 에러 메시지를 전송하게 된다. SB서비스는 cFS 탑재 소프트웨어에서 태스크들을 소프트웨어 버스로 연결하고, 지상국, 온보드 태스크들 간의 메시지 교환 서비스를 제공하며, 특정 태스크가 메시지를 읽어가지 않는 것을 에러로 판단하고 이벤트 서버를 통해 지상국으로 에러 발생 상황을 보고한다. 이렇게 탑재 소프트웨어에 구현되어 있는 에러 검출 기능을 활용해서 결함 주입의 효과로 발생한 에러를 탑재 소프트웨어가 검출하였는지를 확인하게 된다.

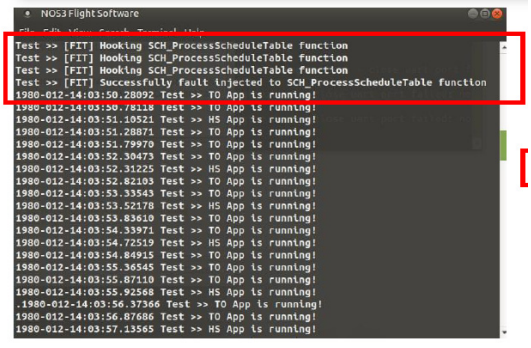
Table 1. Flight software faults list

ID	Fault	Detection Method
1	Application not executed	HS App monitor
2	Exception handler fails	Exception handler
3	Application hogging	HS App monitor
4	Memory error	System error
5	Application restart	Application restart
6	Undefined instruction	Exception handler
7	Abrupt processor reset	Software reboot
8	Incorrect SB message	HK, SB detection
9	Abnormal operation data	CI, SB detection
10	Incorrect telecommand	CI detection

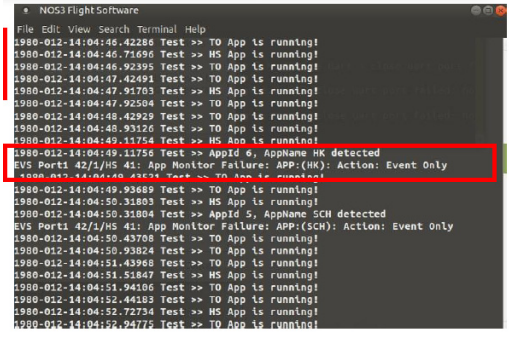
Table 1의 테스트 ID.1번의 애플리케이션 태스크의 실행을 중단시키는 결함을 주입하는 시험에서는 cFS에서 SCH 서비스가 HK애플리케이션을 주기적으로 깨워주는 wakeup 메시지를 보내지 않도록 설정하는 방식으로 결함을 주입하였다. 정상적인 경우에 HK 태스크는 SCH가 1Hz로 보내 주는 메시지를 기다리고 있다가 메시지가 도착하면 깨어나서 작업을 수행하는 구조를 가지고 있다. 만일 SCH에서 메시지가 오지 않으면, HK는 SB서비스의 큐에서 wakeup 메시를 무한히 대기하게 되어 태스크가 실행되지 않게 되고, 이를 감시하는 HS 태스크에서 오류 발생을 검출하게 된다. 이 테스트의 결함

주입은 FIT_CMD의 연결함수를 설정하여 시스템이 부팅한 이후 일정 시간 이후부터 SCH 태스크가 메시지를 전송하지 않도록 하였다. Fig. 3은 결함 주입시의 다운링크 메시지와 HS 태스크에서 HK 태스크가 실행되지 않았다는 오류를 검출한 화면을 보여준다.

대부분의 결함 주입에 대해서 STF-1의 탑재 소프트웨어는 설계된 에러 검출 기능이 정확히 동작하였다. 그러나 일부 결함에 대해서는 에러를 검출하지 못하였다. 탑재 컴퓨터가 부팅될 때 탑재 소프트웨어에서 읽어들이는 데이터 파일인 startup 스크립트 파일 내부에 태스크들의 스택크기를 지정하게 되어 있는데, 이 크기를 0으로 설정하여 테스트 ID.9의 결함주입하였다. 테스트 결과 탑재 소프트웨어가 아무 오류도 검출하지 못했다. 그 이유는 탑재 소프트웨어가 부팅될 때 스크립트의 설정 값대로 스택을 할당하지 않고, 고정된 값을 할당하고 있기 때문으로 분석되었다. 이 경우는 설계와 다르게 태스크가 구현된 사례로 일종의 구현 오류로 볼 수 있다.



(a)Fault injected into SCH task and a hook function called



(b) EVS message showing HS App monitor detected HK task error

Fig. 3. FIT Test #1 : Application not executed

ID.8번 테스팅에서는 태스크 간의 메시지 통신 수단인 SB 서비스에서 지상국에서 수신한 원격명령 데이터가 위성 내에서 변경되는 결함을 주입했을 때, 원격 명령을 수신하는 태스크에서 이를 검출할 수 있는지를 관찰하였다. 위성내 태스크간 메시지는 CCSDS표준 포맷을 이용하여 교환하게 되는데, 메시지 포맷은 정확하게 유지하고 명령 내용을 수신한 것과 다른 명령을 변조하였더니 오류를 검출하지 못하고 다른 명령을 실행하였다. 이것은 SB 태스크에서 메시지의 명령어 포맷의 위변조를 체크하는 기능이 부재하기 때문에 발생하였다.

결함 주입시험 결과 탑재 소프트웨어 설계에 명시한 결함 조건들에 대해서는 설계한 대로 태스크들이 결함으로 인한 오류를 검출하였다. 그러나, 설계 명세에 명확하게 정의되어 있지 않은 결함들이 주입되었을 때는 탑재 소프트웨어는 오류를 검출하지 못하였고, 잘못된 동작을 실행하였다. 개발한 FIT_CMD도구를 이용하여 STF-1 위성의 탑재 소프트웨어를 대상으로 결함 주입시험을 수행할 수 있었으며, 개발한 결함 주입 도구를 사용하여 다양한 유형의 결함을 주입할 수 있었으며, 탑재 소프트웨어의 구현 오류를 발견할 수 있었다.

5. 결론

이 연구에서는 인공위성 탑재 소프트웨어의 신뢰성을 검증하기 위해 결함 주입 도구를 개발하였다. 개발한 도구는 기존의 일반적인 응용에서의 결함 주입 도구와는 달리 인공위성 탑재 소프트웨어에서 정의한 다양한 형태의 결함을 주입할 수 있도록 연결함수 기능을 포함하도록 설계하였다. 또한, 인공위성 시뮬레이터인 NOS3 환경을 가상머신을 통해 접근하여 고장을 주입할 수 있으며, 인공위성 탑재 컴퓨터에서 많이 사용하는 SPARC 프로세서의 디버깅 인터페이스인 GRMON을 지원하고 있다.

개발한 결함 주입 도구의 유용성을 확인하기 위해 큐브위성인 STF-1위성의 탑재 소프트웨어를 대상으로 10가지 항목의 결함 주입시험을 수행하였다. 시험 결과 일부 결함에 대해 탑재 소프트웨어에서 결함으로 인해 발생한 오류를 검출하지 못하는 사례를 발견할 수 있었으며, 일정 수준의 유용성을 확인할 수 있었다. 본 연구로 개발한 결함 주입도구는 향후 업체와 협력하여 상용화를 추진하고, 국내 인공위성 탑재소프트웨어 검증과정에 적용할 계획이다.

References

[1] NASA (2019). NPR 7150.2C: Software Engineering Requirements.

[2] ESA (2017). ECSS-Q-ST-80C Rev.1: Software product assurance.

[3] LACAN, Ph, et al. “Ariane 5-the software reliability verification process”, In: DASIA 98-Data Systems in Aerospace. 1998. p. 201.

[4] GROCE, Alex, et al. “Establishing flight software reliability: Testing, model checking, constraint-solving, monitoring and learning”, Annals of Mathematics and Artificial Intelligence, 2014, 70.4: 315-349. DOI: https://doi.org/10.1007/s10472-014-9408-8

[5] SCHNEIDEWIND, Norman F. "Validating metrics for ensuring Space Shuttle Flight software quality", Computer, 1994, 27.8: 50-57. DOI: https://doi.org/10.1109/2.303613

[6] Aeroflex Gaisler AB, GRMON User’s Manual, Ver.1.1.58, April 2013.

[7] GELETKO, Dustin M., et al. "NASA operational simulator for small satellites (NOS3): the STF-1 cubesat case study", arXiv preprint arXiv:1901.07583, 2019.

[8] MCCOMAS, David. "NASA/GSFC's Flight Software Core Flight System", In: Flight Software WorkshopFlight Workshop. 2012.

[9] SECURE_SCROLL FIT, Suresofttech, https://suresofttech.com/ko/html/tool/secure_fit.php

배 지 훈(Ji-Hoon Bae) [정회원]

- 2020년 8월 : 충남대학교 컴퓨터공학과 (학사)
- 2022년 2월 : 충남대학교 컴퓨터공학과 (석사)
- 2021년 12월 ~ 현재 : (주)만도 연구원

〈관심분야〉
SW신뢰성, 기능안전, 차량 보안

최 규 옥(Gyu Ok Choi) [정회원]

- 2012년 2월 : 성결대학교 정보통신공학부 (학사)
- 2018년 2월 ~ 현재 : 슈어소프트테크 임베디드연구소 책임연구원

〈관심분야〉
정보통신, 임베디드 제어기

김 형 신(Hyungshin Kim) [정회원]

- 1990년 12월 : Univ. of Surrey, 위성통신공학과 (공학 석사)
- 2003년 2월 : 한국과학기술원 전산학과 (전산학 박사)
- 2004년 2월 ~ 현재 : 충남대학교 컴퓨터공학과 교수

〈관심분야〉
임베디드 시스템, 모바일 시스템, 에지 컴퓨팅