

## 항공용 시뮬레이션 모델 프레임워크 성능 분석을 통한 적용성 평가

# The Applicability of Avionics Simulation Model Framework by Analyzing the Performance

서민기\* · 조연제 · 신주철 · 백경훈 · 김성우  
LIG넥스원 항공드론연구소

Min-gi Seo\* · Yeon-je Cho · Ju-chul Shin · Gyong-hoon Baek · Seong-woo Kim

Fixed Wing Drone System R&D Lab, LIG Nex1, Daejeon 34127, Korea

### [요 약]

항공전자시스템은 항공기의 뇌와 신경, 오감에 해당하는 것으로 비행 및 임무를 수행하기 위해 통신/식별/항법/무장/시현 계통의 항공기 탑재 전자장비로 구성된다. 비행체 체계에서의 비중이 50%에 육박하며, 4차 산업혁명 기반 기술이 발전됨에 따라 중요도 및 비중이 증가되는 추세이다. 항공기의 개발 소요시기에 따라 개발 기간은 점점 단축되고 있으며 항공전자시스템 통합 및 검증을 위한 안정적인 항공용 SIL의 적기 개발이 반드시 필요하다. 본 논문에서는 기존 항공용 SIL을 프레임워크 기반의 항공용 SIL로 대체하기 위한 방안을 제안한다. 그리고 대체 적용한 결과를 바탕으로 항공용 시뮬레이션 모델 프레임워크의 성능을 평가한다.

### [Abstract]

Avionics corresponds to the brain, nerves and five senses of an aircraft, and consists of aircraft mounted electronic equipment of communication, identification, navigation, weapon, and display systems to perform flight and missions. It occupies about 50% of the aircraft system, and its importance is increasing as the technology based on the 4th industrial revolution is developed. As the development period of the aircraft is getting shorter, it is definitely necessary to develop a stable avionics SIL in a timely manner for the integration and verification of the avionics system. In this paper, we propose a method to replace the legacy SIL with the avionics simulation model framework based one and evaluate the framework based on the result of alternative application.

**Key word** : Avionics simulation model framework, Discrete event system specification, System integration laboratory.

<https://doi.org/10.12673/jant.2021.25.5.336>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<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 24 August 2021; Revised 27 September 2021  
Accepted (Publication) 22 October 2021 (30 October 2021)

\*Corresponding Author; Min-gi Seo

Tel: +82-042-718-3565

E-mail: mingi.seo@lignex1.com

## I. 서론

비행/임무를 수행하는데 필요한 항공기 탑재 전자장비(이하 항전장비)들로 구성되는 항공전자시스템(이하 항전체계)은 항공기 제작의 기술/비용 비중 면에서 거의 50%에 육박하는 분야이다[1]. 이러한 항전체계는 기술의 발전에 따라 검증해야할 기능의 증가와 함께 고도의 신뢰성이 요구되기 때문에, 항전체계를 검증하는 시험 장비인 통합시험환경(SIL; system integration laboratory)의 개발이 중요해지고 있다[2]. SIL은 실제 탑재장비로 시험하기 어려운 항전체계의 고장 모의가 가능한 점과 비행체 및 탑재장비보다 앞서 개발하여 항공기 검증에 활용함으로써 항공기 개발 단계 간 10배가량 증가하는 고장탐구 비용을 효과적으로 절감할 수 있다는 점 등의 이점이 있기 때문에 항공기 개발에 필수적이다. 이러한 SIL 개발을 위해 서브시스템/전체 시스템의 개별 통합 검증 외에도 시간을 단축한 최적 접근 관점의 다중모드 통합 검증[3], SIL 모델의 검증 자동화를 통해 검증 시간을 단축하는 등의 다양한 시도들이 이어지고 있다[4].

항공용 SIL의 비행체 탑재장비 모델 프레임워크(ASMF; avionics simulation model framework)는 지금까지의 국내 항공용 SIL 개발 사례를 분석하여 효과적으로 SIL을 개발할 수 있는 방안을 제안한 기술이다. 국내 항공용 SIL은 In-house 방식으로 상용 도구 환경에 종속적으로 개발되어 타 개발 사업에 적용할 수 있는 이식성이 부족했다. ASMF는 모델 기반 개발(MDD; model driven development) 개념을 도입하여 이산사건 시스템 명세(DEVS; discrete event system specification) 형식론(formalism) 기반의 모델링 방법으로 이식성뿐만 아니라 개발 생산성, 유지보수성, 품질의 향상 효과를 제시하였다[5].

본 논문에서는 ASMF 기술을 무인기 항전체계에 적용하기 위해서 구체적인 적용방안과 그 검증 결과를 제시하겠다. 그리고 실제 운용 중인 무인기 항전체계 SIL(이하 legacy SIL)과 이를 대체 적용한 ASMF를 비교 평가하여 ASMF 기술의 효과를 입증하고자 한다.

## II. ASMF의 실체계 적용성 입증 방안 분석

### 2-1 실체계 적용성 확인을 위한 구현 범위/환경 구성

실체계 적용성 확인은 현재 개발하여 운용하고 있는 legacy SIL에서 수행한 항전체계 시험절차를 기준으로 수립했다. legacy SIL 모델을 ASMF의 항공용 시뮬레이션 모델(ASM; avionics simulation model)로 대체한 후 항전체계 시험절차를 적용하여 정상으로 확인되면 ASM이 legacy SIL 모델의 기능/성능을 대체했다고 판단할 수 있기 때문이다. 대체 모델들의 식별은 ASMF에서 제공하는 3가지 모델링 방식을 고려하여 결정하였다.

모델링 방법 중 첫 번째는 대상 항전장비의 특정 제조사/모델을

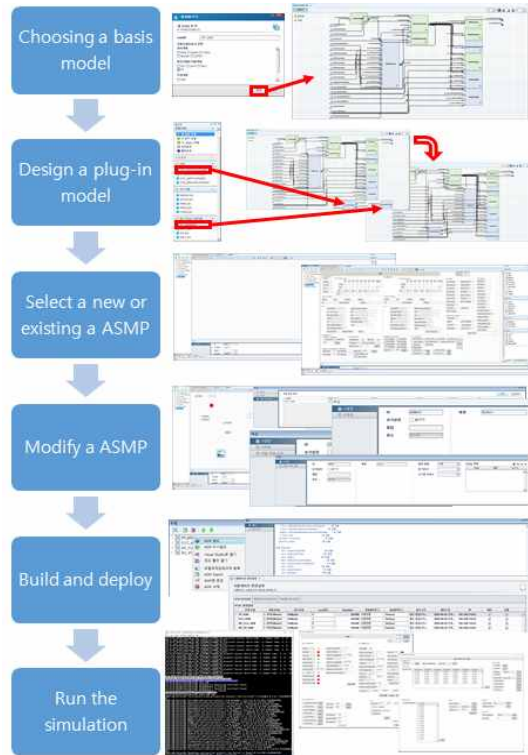


그림 1. ASMF 운용개념

Fig. 1. ASMF operational concepts.

완제품(完製品)으로 구현한 ASM을 선택하는 방법이다. 두 번째는 반제품(半製品)으로 구현한 기본 모델(basis model)을 활용하여 개발하는 방법이다. 기본 모델은 대상 항전장비의 특정 제조사/모델들이 공통적으로 보유하여 재활용성이 높은 기능들을 대상으로 한다. 그림 1은 기본 모델을 활용한 개발 절차 및 운용 개념이다. 마지막 방법은 재활용 구성요소를 사용하지 않고 신규 개발하는 것이다.

ASM의 데이터를 제어하고 시현하는 GUI 패널(ASMP; avionics simulation model panel)은 기본 모델을 활용하여 개발하거나 신규 개발하는 경우엔 통합개발환경(IDE; integrated development environment)의 팔레트에서 제공하는 컨트롤들을 화면에 배치하고, 컨트롤과 연결하고자 하는 데이터를 선택하는 등의 개발 작업이 필요하지만, 완제품 형태의 ASM을 선택한다면 해당 ASM과 연동되는 ASMP가 함께 제공되기 때문에 GUI 개발 작업이 필요하지 않다[5].

무인기 항전체계는 임무컴퓨터(MC; mission computer) 중심으로 항전 및 타계통장비들이 연동하는 구조이다. legacy SIL은 주요 검증 대상인 MC를 제외한 항전 및 타계통장비들을 모델로 제공한다. legacy SIL 모델 중 ASMF 기술로 대체할 대상은 항전체계 기능 비중이 크고, 3가지 모델링 방식을 각기 적용할 수 있는 관점으로 다음과 같이 선정하였다.

- 피아식별/비행조종/전파고도계/항법장치 : ASM 선택
- 음성통신중계장치 : 신규개발(완전 개발/basis model 활용)
- 데이터링크장치 : legacy model import

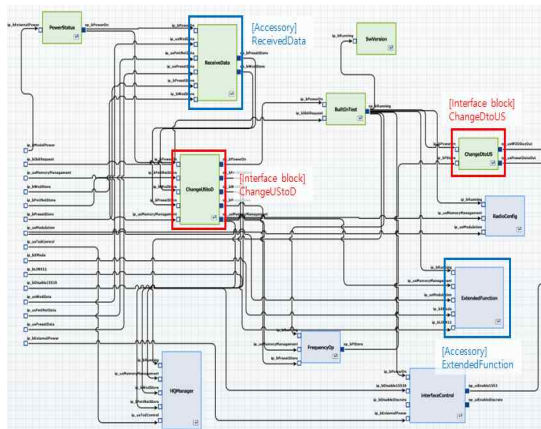


그림 2. 기본 모델을 기반으로 개발한 UVHF ASM 결과 예시  
 Fig. 2. An example of UVHF ASM results developed based on the basis model.

신규 개발은 DEVS 형식론 모델링과 개발자 코드를 삽입할 수 있으므로 이론적으로 모든 모델을 구현할 수 있다. 따라서 신규 개발만으로는 유의미한 적용성 평가 결과로 보기에 적합하지 않으므로, 데이터링크장치는 기존 모델의 재사용을 쉽게 하여 ASMF의 유용성을 극대화시키기 위한 기능인 legacy model import 기능을 이용하여 개발하였다. 여기서는 ASM 선택 개발 방안[5]을 제외한 나머지 개발 방안을 주로 살펴보겠다. 모델을 제외한 나머지 환경은 legacy SIL과 동일하도록 실장비 MC를 ASMF 환경에 연결할 수 있는 전원, 통신 신호 케이블 등을 설치하였다.

2-2 기본 모델 기반 개발

확장 모델은 장비의 ‘부가 기능(accessory)’ 및 ‘입출력 인터페이스 설정/제어 기능(interface block)’의 집합이다. 기본 모델은 확장 모델보다 재사용성이 상대적으로 높은 모델이다. 확장 모델의 ‘부가 기능’은 공용화 범위에 해당하진 않지만 임무 수행을 위해 추가될 수 있는 기능이고, ‘입출력 인터페이스 설정/제어 기능’은 장비 외부 입출력 통신 프로토콜에 맞추기 위한 ICD 모델이다. 통신 프로토콜 별 IDE에서 제공되는 ‘연동 블록’과 세부적인 변환 작업을 위해 추가된 결합/원자 모델이 ICD 모델에 해당한다.[5]

본 연구에서는 음성통신중계장치 기본 모델에 확장 모델을 추가하여 개발하였다. 음성통신중계장치는 주파수 수동 설정, 프리셋 설정, 프리셋 저장 등이 기본 모델로 지원되고, guard receiver 기능과 ICD 변환 기능 등은 확장 모델로 지원된다. 그림 2는 기본 모델과 확장 모델을 이용하여 개발 완료한 음성통신중계장치 ASM이다. 그림 2의 ‘Extendedfunction’ 결합 모델은 확장 모델의 ‘부가 기능’이며, xmode와 link11을 제어하는 무전기 설정(확장)과 guard receiver 기능인 주파수 설정(확장) 기능을 포함한다. 또 다른 ‘부가 기능’인 ‘ReceivedData’는 MC와 WOD, Preset, FMTNET table 데이터를 동기화하여 저장한다.

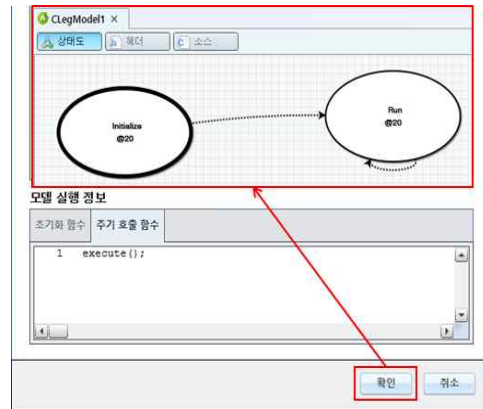


그림 3. legacy model import 기능의 화면 예시  
 Fig. 3. An example of GUI for legacy model import function.

‘ChangedUtoD’와 ‘ChangedDtoUS’는 확장 모델의 ‘입출력 인터페이스 설정/제어 기능’에 해당한다. 음성통신중계장치 ASM 내부에서 사용하는 실수형 데이터를 ICD의 필드 규칙에 맞춰 변환하는 역할이다.

2-3 Legacy Model Import 기능 설계 : DEVS 모델 변환

대체 대상으로 하는 legacy SIL를 포함하여 SIL 개발을 지원하는 상용 도구들은 별도의 모델링 형식은 제공하지 않고, 개발자가 코드를 작성할 수 있도록 구현부가 비어 있는 콜백 함수(callback function)를 제공한다. 개발자는 도구의 스케줄링 설정 기능으로 모델의 실행 빈도를 설정하면 시뮬레이션 실행 후부터 도구에서 설정 주기대로 콜백 함수를 호출한다. legacy model import는 이러한 특성을 반영하여 ASM으로 변환한다. 변환된 ASM은 legacy SIL과 마찬가지로 경성 실시간 시스템(hard real-time system) 성능을 보장하고, 나아가 legacy SIL의 주기적인 일괄처리 방식의 알고리즘을 이벤트 기반 시뮬레이션 알고리즘으로 해결하였다[6],[7],[8].

그림 3은 개발자가 legacy SIL 모델의 콜백 함수를 기입할 수 있도록 legacy model import 기능의 다이얼로그를 설계한 결과이다. 콜백 함수를 기입하고 확인 버튼을 클릭하면 콜백 함수 호출을 위한 상태도 템플릿을 자동적으로 생성한다. 그림 2의 출발 상태에서부터 도착 상태로 표기된 점선의 화살표는 내부 천이(internal transition)이다. 내부 천이는 현재 상태에서 머물고 있는 시간이 현재 상태에 기록된 시간 전진(TA; time advance)을 경과했을 때 현재 상태를 도착 상태로 변경하는 작업을 뜻한다. 즉, initialize 상태에서 20 ms 경과 후 run 상태로 천이하고, 그로부터 20 ms 경과마다 run 상태로 되돌아오는 천이를 실행한다. 내부 천이 화살표 내부에는 legacy SIL 모델과 연결되도록 다이얼로그에 기입한 콜백 함수의 호출이 삽입된다. initialize → run 내부 천이는 ‘초기화 함수’ 탭의 코드가 삽입되고, run → run 내부 천이는 ‘주기 호출 함수’ 탭의 코드가 삽입된다.

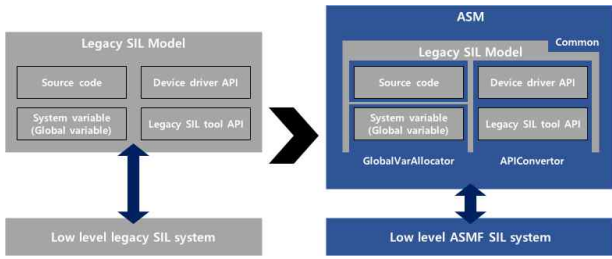


그림 4. ASM converting layer를 이용한 legacy SIL 모델과 ASMF ASM의 연결 관계

Fig. 4. The relationship between the legacy SIL model and ASMF ASM as a converting layer of ASM

2-4 Legacy Model Import 기능 설계 : ASM Converting Layer

ASM Converting Layer(이하 CL)는 legacy SIL 모델이 ASMF에서 정상적으로 구동될 수 있도록 ASM으로 변환/연결하는 서브루틴의 모음이다. CL의 정의에 의하면 legacy SIL 모델의 콜백 함수 호출을 위한 상태도 템플릿도 CL에 해당하지만, 여기서는 소스코드만으로 한정하였다.

ASMF는 하드웨어와 OS를 포함한 구동 환경(RTSE; real-time simulation environment)과 개발된 도메인 영역은 서로 간의 변경으로부터 독립적이어야 한다는 개발 철학을 바탕으로 ASM과 RTSE를 서로 다른 계층으로 분리하였다[5]. 따라서 이러한 개발 철학이 반영되지 않은 legacy SIL 모델에 대해 CL은 legacy SIL 모델과 하드웨어 I/O 및 OS 간의 종속적인 환경 요소들을 조정해야 한다.

그림 4는 legacy SIL 모델과 ASMF 시스템의 CL 연결 구조를 나타낸 그림이다. CL은 legacy SIL 모델의 소스코드를 온전히 활용하고, 시스템 수준에서 제공되는 전역 변수는 GlobalVarAllocator를 통해 ASMF 시스템과의 연결을 제공한다. APIConvertor는 위에 언급한 환경 차이를 조정하는 것으로 하드웨어 I/O API와 legacy SIL 도구 전용으로 제공되는 API 및 OS API가 해당한다. CL의 모든 구성요소는 개발자가 수정할 수 있는 소스코드로 제공되기 때문에 기능 확장을 위한 추가/수정이 가능하다.

표 1은 CL의 구성 요소를 설명하였다. 데이터링크장치를 모의하는 legacy SIL 모델은 도구 전용 및 OS API를 제외한 하드웨어 I/O가 RTSE 종속 요소로 포함되어 있다.

APIConvertor는 Legacy와 ASMF 하드웨어 I/O의 연결을 위해 연동 블록(ICD 모델; interface control document)의 control 속성으로 분류되는 메시지와 연결한다. control 속성 메시지는 T, R, TR 타입으로 분류된다. T 메시지는 ASM에서 RTSE로 전달하는 명령으로 RTSE는 명령에 대한 응답을 발생시키지 않는다. 통신 인터페이스의 활성화/비활성화 제어, 통신 상태 설정 등이 T 메시지에 해당한다. R 메시지는 RTSE에서 ASM으로 전달되는 응답을 뜻한다. 원하는 데이터가 수신되었는지 주기적으로 확인하는 용도로, RTSE에서 ASM으로 주기적인 보고가

표 1. ASM converting layer 구성 요소

Table 1. Components of the converting layer of ASM.

Layer	Description
APIConvertor	Adjustment to apply the legacy SIL with strong coupling for RTSE(HW, SIL Tools, OS) to RTSE-independent ASMF
GlobalVarAllocator	Connect interface attribute of ASMF to system global variable of legacy SIL
Common	1) Main header of the wrapper 2) Declare callback function for legacy SIL model 3) Implements the interface between APIConvertor and GlobalVarAllocator

필요한 경우에 해당한다. TR 메시지는 ASM에서 RTSE로 전달하는 명령인 점이 T 메시지와 동일하지만, RTSE가 명령에 대한 응답을 ASM으로 보내는 점이 다르다. 불필요한 것으로 식별된 하드웨어 I/O는 더미(dummy) 처리하였다.

GlobalVarAllocator는 legacy SIL 모델에서 정의한 시스템 전역 변수들과 ASMF의 연동 속성(interface attribute)을 연결하는 역할을 한다. IDE는 상태도 템플릿 생성 시 legacy SIL 모델의 변수 정의를 읽어 동일한 구조로 연동 속성을 생성한다. 주요 기능은 생성된 연동 속성을 참고하여 포인터 변수로 정의하고, 포인터 변수와 legacy SIL 모델의 시스템 전역 변수 간 전환을 수행하는 것이다. legacy SIL 모델의 콜백 함수는 이 영역에서 호출되는데, 매 프레임 실행 전/후마다 시스템 전역 변수와 연동 속성 간의 동기화가 필요하기 때문이다.

Common은 APIConvertor, GlobalVarAllocator 모두에 해당하는 역할들을 구현한 컴포넌트이다. ASM에서는 특정 헤더만을 참조하여 CL을 이용할 수 있도록 ASM과 CL을 연결할 메인 헤더를 추가하였고, GlobalVarAllocator가 legacy SIL 모델의 콜백 함수 호출을 위해 직접 포함하여 발생하는 종속성을 방지하기 위해서 별도의 인터페이스를 부여하였다. 시스템 전역 변수와 연동 속성 간 전환은 주기적인 처리로 문제가 없지만 인터럽트 등으로 인한 변수의 비주기적인 변동은 발생 시점으로부터 주기 처리까지의 지연을 감수해야 한다.

이벤트 메시지(이하 EventMessage)는 지연 감수에 대처하기 위해서 추가된 것으로 이벤트의 특징에 따라 L2A, A2L, A2A로 분류한 그림은 표 2와 같다. L2A는 legacy → ASM 이벤트, A2L은 ASM → legacy 이벤트, A2A는 ASM 내부 통신용 이벤트를 의미한다. 데이터링크장치 CL에서 정의된 이벤트는 표 2와 같다. IO\_ACTIVE 이벤트는 legacy SIL 모델 내에서 I/O 활성화/비활성화를 제어하는 API의 호출과 연관된 이벤트이다. L2A로 분류되는 이벤트는 ASM의 포트 출력을 실행하므로 IO\_ACTIVE 이벤트는 IOActive 포트 출력을 실행하여 연동 블록에서 T 속성의 control 메시지로 정의된 인터페이스 활성화/비활성화 제어 메시지를 발생시켜 RTSE를 제어한다. COMM\_STS\_SET 이벤트는 IO의 설정을 처리하는 이벤트로

**표 2.** ASM converting layer에서 지원하는 이벤트 목록  
**Table 2.** The list of events supported by converting layer of ASM.

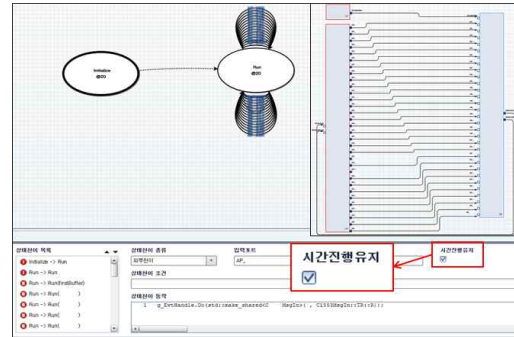
Category	Event name	Description
L2A	IO_ACTIVE	A command to enable/disable the I/O interface is issued.
	COMM_STS_SET	A command is issued to set the state of the I/O interface.
A2L	COMM_MSG_IN	Receive aperiodic messages into I/O interfaces.
A2A	NO_EVENT	In case the event condition is not met in the event handling class.
	EMPTY_Q	An event that is raised if the event queue is empty when checking the event queue.
	ERROR	An event dispatched to ASM when an error occurs during simulation.

L2A와 동일한 구조로 처리된다. A2L로 분류되는 COMM\_MSG\_IN 이벤트는 메시지를 수신 여부를 import된 legacy SIL 모델로 알리는 이벤트이다. APIConvertor에서 참조하는 정보에 접근하여 수정함으로써 legacy SIL 모델의 소스코드에서 호출하는 API의 실행 결과를 변경한다. NO\_EVENT는 EventMessage와 ASM 간의 인터페이스로 ‘이벤트 처리’ 상황은 발생했지만 ‘이벤트 조건’이 부합하지 않는 경우에 해당한다. 조건에 해당하여 IO\_ACTIVE 이벤트를 반환하는 앞선 예에서 조건에 해당하지 않는 경우에는 NO\_EVENT 이벤트가 반환된다. EMPTY\_Q 메시지 큐가 비어 있음을 알리기 위한 용도로 ASMF 시스템 내부에서만 사용된다. ERROR 이벤트는 실행 중 발생하는 오류 상황을 포착하여 예외처리를 할 수 있도록 정의한 이벤트이다. 본 연구에서는 예외처리 목적에 맞는 상황이 식별될 때 추가할 수 있는 예약(Reserved) 영역으로 구분하였다.

그림 5는 legacy model import 기능을 이용하여 개발한 데이터링크장치 ASM 결과이다.

**2-5 Legacy Model Import 기능 설계 : 개발환경 고려사항**

legacy model import 기능 개발 및 적용 과정 중 2가지 문제가 발생하였다. 첫 번째는 컴파일러 버전 문제이다. vxworks 5 환경에서 구동되는 legacy SIL 모델은 함수 선언/정의 간 시그니처가 다른 경우와 타입 변환 등에 대해 빌드 오류가 없었지만, ASMF의 대상 OS(windows, linux, vxworks 7)에서는 모두 빌드 오류가 발생했다. 따라서 legacy SIL 모델에서 문제가 발생한 코드를 수정하여 legacy SIL과 ASMF에서 정상 빌드를 확인하였다. 두 번째는 메시지에 대한 데이터 매핑(mapping) 구조 차이로 인한 문제였다. ASMF는 프로세서의 endian 특성으로부터 독립적으로 모든 통신 인터페이스에 대한 데이터 매핑 단위를 bit 단위로 지원한다. 이를 위해 구조체 변수와의 매핑될 경우, 구조체 멤버 변수 간 비어 있는 메모리 공간이 없어야 일관적인



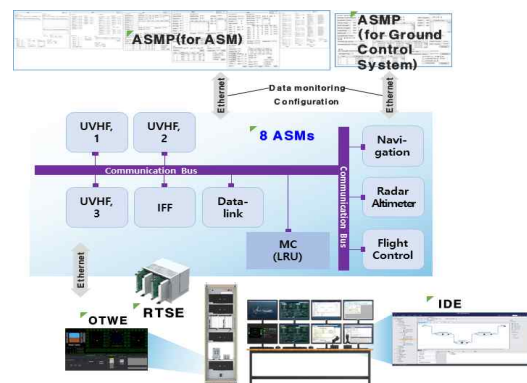
**그림 5.** legacy model import 기능의 실행 결과(원자/결합모델)  
**Fig. 5.** Execution result of legacy model import function(atomic/coupled model).

bit 단위 매핑이 가능하므로 구조체 정렬 크기를 1byte로 통일하였다. 반면 legacy SIL은 매핑 단위를 byte 단위로 지원하는 통신 인터페이스가 있으며, big endian 환경으로 고정되어 있어, 정렬되지 않은 구조체 사용을 개발자가 코드 수준으로 디코딩/인코딩하여 사용하고 있었다. ASMF와 legacy SIL 모델의 구조체 정렬 차이를 통일하는 쉬운 방법으로 legacy SIL 구조체 정렬 크기 정의를 추가할 수 있지만, 정렬 크기를 수정한 legacy SIL 모델을 다시 legacy SIL 환경에서 실행하면 정상적으로 동작할 수 없으므로 CL 내 GlobalVarAllocator의 포인터 변수와 legacy SIL 모델의 시스템 전역 변수 간 전환 작업 때 이 차이를 고려하도록 작업을 추가하였다.

**III. ASMF의 실제 적용성 입증 수행**

**3-1 실제 적용성 입증 환경**

그림 6은 시험을 위한 환경 구성도이고, 그림 7은 실제 하드웨어와 소프트웨어를 탑재하여 구성한 결과이다. legacy SIL 비행



**그림 6.** 실제 적용성 입증을 위한 시험 환경 구성도  
**Fig. 6.** Test environment configuration diagram to prove applicability.



그림 7. 실체계 적용성 입증을 위한 시험 환경 구성  
 Fig. 7. Test environment configuration to prove applicability.

조종 모델의 타계통 기능 시험을 위해서는 전과고도계와 항법 장치 연동이 수반되어야 한다. 데이터링크장치는 항전체계와 지상체 간을 중개하며, 비행체 관리 3개 분야 전 시험 항목 수행의 기반이 된다. ASM 별 연관된 ASMP는 대상 ASM의 데이터 모니터링과 내부 설정 용도로, 지상체 ASMP와 연결되는 ASM은 데이터링크장치 ASM이다. 외부환경(OTWE; out the window & environment)은 항전체계가 탑재된 비행체의 비행환경을 모의한다. 주요 기능은 비행 중 상황에서의 비행 데이터를 모의하여 비행조종 및 전과고도계, 항법장치로 전달하는 것이다.

### 3-2 실체계 적용성 입증 결과 및 효과 평가

ASMF의 실체계 적용성 입증 시험의 결과는 기능/성능 모두 만족하여 legacy SIL 대상의 대체 적용성을 입증하였다. 수동 코드 작성 관점으로 평가한 개발 비용 효과는 ASM은 약 83%의 절감 효과가 있었고, ASMP는 약 100%의 효과가 있었다.

먼저 실체계 적용성 입증의 시작은 시험절차 수립이었다. 시험절차는 비행체 관리 3개 분야(피아식별, 음성통신 중계, 타계통 기능) 20개로 구성하였고, 20개의 시험절차 별 하위 시험절차가 포함된 구조로, 모두 legacy SIL 환경에서 검증한 항전체계 시험절차를 활용하였다. ASMF의 성능은 실장비 MC의 기능으로 검증하였다. 실장비 MC는 연동하는 모델의 통신 성능 저하 등의 문제로 적절한 시기와 적합한 데이터를 모의하지 못하면 실패로 처리하는 기능이 포함되어 있다. 시험절차는 앞서 서술한 실체계 적용성 입증 환경에 실장비 MC를 장착하여 수행하였다. 모든 시험절차의 결과는 정상으로 확인하였다.

개발 비용 효과 평가의 기준은 음성통신중계장치 ASM과 legacy SIL 모델이 모두 동일 MC와 연동하여 기능/성능을 입증한 점에 착안하여 음성통신중계장치를 비교 대상으로 선정하였다. 비교 평가 지표는 ASM과 legacy SIL 모델 개발의 담당자가 동일한 점에서 '개발 기간'이 적절해 보일 수 있지만, ASM과 legacy SIL 모델의 개발 착수 시점이 서로 다르기 때문에

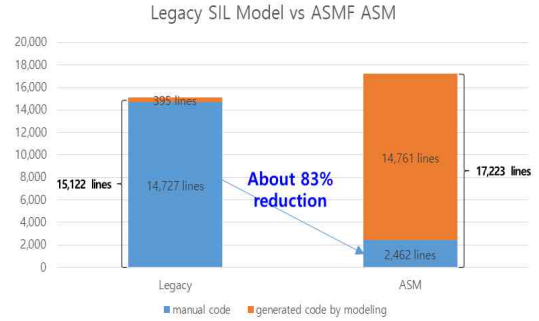


그림 8. legacy SIL 모델을 대체하는 ASMF ASM의 효과 평가  
 Fig. 8. Assessing the effectiveness of ASMF replacing legacy SIL model

모델 요구사항 이해도에 차이가 있어 적절하지 않다고 판단했다. 반면에 '소스코드 라인 수'를 적합하다고 판단한 근거는 '소스코드 라인 수'가 개발자의 작업량에 비례하기 때문이다. 모델링 관점에 의한 차이에 더 주안점을 두고 비교 평가할 수 있었던 배경엔 알고리즘과 같이 DEVS 형식론만으로 표현하기 어려운 legacy SIL 모델의 소스코드를 ASM에 재사용한 점이 있었다.

비교한 결과는 그림 8과 같다. legacy SIL 모델부터 살펴보면 전체 소스코드는 15,122 라인이다. 개발환경 내 기능을 통해 설정한 값에 따라 자동으로 생성되는 395 라인을 제외하면 개발자가 직접 작성한 수동 코드는 14,727 라인이다. ASM의 전체 소스코드는 17,223 라인이며, IDE로 설정하거나 DEVS 형식론 기반의 상태도/블록도 작성으로 자동 생성되는 14,761 라인을 제외하면 개발자가 직접 작성한 수동 코드는 2,462 라인이다. 결론적으로 legacy SIL 음성통신중계장치 모델을 음성통신중계장치 ASM으로 개발하면 개발자가 직접 작성해야 하는 수동 소스코드 분량이 83% 가량 감소된 점을 알 수 있다. 정확히 하면 음성통신중계장치 ASM을 개발하기 위한 약 83% 가량의 수동 코드 작업 시간이 일관된 표현으로 가시화되어 생산효과를 높이는 DEVS 모델링으로 전이되었다고 볼 수 있다.

앞선 설명에서 컨트롤들을 드래그하여 화면에 배치하고 변수에 연결하는 작업들로 구성된 ASMP는 동작하기 위해 개발자가 추가로 작성해야 하는 수동 코드가 없다. legacy SIL 모델과 연결된 legacy SIL GUI 모델 패널은 자동으로 생성되는 프레임워크 코드를 제외한 수동 코드가 7,632 라인이다. 실제로는 패널들의 데이터 연동을 통합 관리하는 조정자 모듈도 함께 수정되어야 하는 분량을 제외했기 때문에 더 많은 코드가 발생될 수 있다.

추가적으로 legacy SIL이 갖는 한계였던 CPU/OS 종속성을 해결한 ASMF의 CPU/OS 독립성을 확인하였다. ASMF의 RTSE는 2종으로, RTSE 간의 차이는 CPU와 OS이다. intel RTSE는 intel SBC에 linux OS를 설치하였고, powerpc RTSE는 powerpc SBC에 vxworks 7을 설치하였다. 2종의 RTSE를 대상으로 ASM을 교차 적재하여 동일한 항전체계 시험절차를 각각 수행함으로써 정상 결과를 확인하는 것으로 CPU/OS 독립성을

확인하였다.

#### IV. 결 론

본 논문에서는 MDD 개념을 반영한 ASMF 기술을 체계 운용 중인 무인기 항전체계의 legacy SIL 모델에 적용함으로써 ASMF 기술의 적용성을 확인했다. 진행 과정 중 검증 완료된 legacy SIL 모델을 재사용할 수 있는 legacy model import 기능이 식별되어 설계하여 적용하였고, ASMF 기술 기반으로 MC와 legacy SIL 모델을 대체한 ASM을 운용하여 항전체계 시험 절차를 수행해봄으로써 정상 결과를 확인하였다. 그리고 음성통신중계장치를 기준으로 legacy SIL 모델과 ASM의 개발 비용 관점의 효과를 평가하였다.

당초의 ASMF 연구 취지는 개발 완료한 SIL의 타 개발 사업으로의 이식성 향상을 목표로 개발생산성, 유지보수성, 품질 향상, 이해관계자와의 의사소통 효과를 확보하는 것이었다. 이에 따라 따라서 획득되는 효과와 상충 관계에 놓인 신규 개발 비용은 ASMF가 legacy SIL보다 높을 것으로 짐작했기 때문에, ASM의 전체 코드 수는 legacy보다 현격히 많고, 수동 코드의 감소는 50% 미만으로 예측했었다. 실제 비용 효과 결과가 예상과 다르게 나온 것은 legacy SIL 모델에서 지원하지 않는 bit 단위의 인터페이스 데이터 접근 때문이었다. legacy SIL 모델은 byte 단위로 매핑한 데이터에 대해 인코딩 및 디코딩 변환 작업이 있기 때문이다. 따라서 legacy SIL 대비 전체 코드 증가량이 약 14%에 그쳤고, 수동소스 코드 분량이 50%보다 큰 83%의 감소 효과로 결과가 도출된 것은 ASM 구동 엔진에 해당하는 항공용 시뮬레이션 프레임워크(ASF; avionics simulation framework)에서 bit 단위의 디코딩/인코딩을 자동으로 처리하는 점이 가장 큰 요인으로 보인다.

본 연구의 적용 사례를 시작으로 더 다양한 항전체계 검증에 ASMF 기술을 접목하여 기초 데이터를 확보해나간다면 ASMF 연구 취지에 맞는 이식성과 유지보수성 등까지 평가 가능할 것으로 보인다.

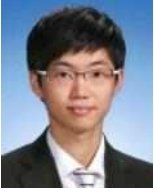
#### Acknowledgments

본 연구는 방위사업청과 방위산업기술지원센터의 지원(사업명 : 항공용 SIL의 비행체 탑재장비 모델 프레임워크 기술 개

발, 계약번호 : UC170001D)하에 수행되었습니다.

#### References

- [1] S. H. Han, "4th industry breakthrough avionics," in *Proceedings of the Institute of Electronics Engineers of Korea Conference*, Pusan, pp. 1456-1458, 2017.
- [2] Y. K. Kim, M. C. Kim, W. W. Choi, and W. S. Oh, "Development of the MEP integration test environment for surion," *Journal of The Korean Society for Aeronautical and Space Sciences*, Vol. 39, No. 7, pp.666-673, Jul. 2011.
- [3] W. H. Chang, J. S. Park, Y. W. Jo, and J. K. Byun, "Verification of hierarchically structured avionics system utilizing multi-mode system integration laboratory," *Journal of The Korean Society for Aeronautical and Space Sciences*, Vol. 45, No. 11, pp.998-1005, Nov. 2017.
- [4] S. G. Yang, Y. J. Cho, K. Y. Jo, and C. M. Ryu, "Design of automatic model verification for system integration laboratory," *Journal of Advanced Navigation Technology*, Vol. 23, No. 5, pp. 361-366, Oct. 2019.
- [5] M. G. Seo, J. C. Shin, G. H. Baek, and S. W. Kim, "Development of real time simulation environment based on DEVS formalism applicable to avionics system integration laboratory," *Journal of Advanced Navigation Technology*, Vol. 23, No. 5, pp. 345-351, Oct. 2019.
- [6] J. C. Shin, M. G. Seo, Y. J. Cho, G. H. Baek, and S. W. Kim, "Event-driven modeling and simulation method applicable to avionics system integration laboratory," *Journal of Advanced Navigation Technology*, Vol. 24, No. 3, pp. 184-191, Oct. 2020.
- [7] Y. J. Cho, S. W. Kim, and G. H. Baek, "Method of avionics system integration laboratory modeling based on DEVS formalism," in *Proceedings of Korean Society for Aeronautical and Space Sciences 2020 Fall Conference*, Jeju, pp. 713-714, 2020.
- [8] K. T. Kim, J. C. Shin, and M. G. Seo, "Event-driven simulation method applicable to avionics system integration laboratory," in *Proceedings of Korean Society for Aeronautical and Space Sciences 2020 Fall Conference*, Jeju, pp. 1316-1317, 2020.



**서민기 (Min-gi Seo)**

2012년 02월 : 한국항공대학교 컴퓨터공학과 (공학사)  
2011년 10월 ~ 현재 : LIG넥스원 항공연구소 선임연구원  
※ 관심분야 : 항공전자, 내장형 소프트웨어



**조연제 (Yeon-je Cho)**

2016년 02월 : 경희대학교 전자전파공학부(공학사)  
2016년 01월 ~ 2018년 03월 : GS ITM 플랜트 ICT 팀 사원  
2018년 07월 ~ 현재 : LIG넥스원 항공전자연구소 연구원  
※ 관심분야 : 항공전자시스템, 내장형 소프트웨어



**신주철 (Ju-chul Shin)**

2008년 2월 : 포항공과대학교 컴퓨터공학과 (공학사)  
2009년 1월 ~ 현재 : LIG넥스원 항공연구소 선임연구원  
※ 관심분야 : 항공전자, 내장형 소프트웨어



**백경훈 (Gyong-hoon Baek)**

2002년 02월 : 한국과학기술원 기계공학과 (공학사)  
2002년 01월 ~ 2016년 05월 : ㈜도담시스템스 수석연구원  
2016년 09월 ~ 현재 : LIG넥스원 항공연구소 수석연구원  
※ 관심분야 : 항공전자, 실시간 시뮬레이션



**김성우 (Seong-woo Kim)**

2002년 08월 : 부산대학교 정보통신공학과 (공학석사)  
2002년 10월 ~ 현재 : LIG넥스원 항공연구소 수석연구원  
※ 관심분야 : 실시간 시뮬레이션 기법 및 시험환경 응용 개발