

FACE 적합성을 위한 항공전자 소프트웨어
데이터 모델링 방안 및 검증조경연^{1,†} · 이두환¹ · 차상철¹ · 김정열¹¹LIG넥스원

Avionics Software Data Modeling Method and Test For FACE Conformance

Kyeong-Yeon Cho^{1,†}, Doo-Hwan Lee¹, Sang-Cheol Cha and Jeong-Yeol Kim¹¹LIG Nex1

Abstract

The avionics industry has recently adopted an open architecture to increase software portability and reduce the development schedule and cost associated with changing hardware equipment. This paper presents a data modeling method compliant with FACE, a widely used open avionics architecture. A FACE data model is designed and implemented to output data from VOR/ILS avionics equipment. A FACE Conformance Test Suite (CTS) program is utilised to verify that the data model meets FACE standards. The proposed data modeling method is expected to improve the development schedule and cost associated with modifying communication methods and ICDs (Interface Control Documents).

초 록

항공전자 시스템에서는 하드웨어 장비가 변경되면 통신 방식 및 ICD 변경이 빈번히 발생하며, 소프트웨어 또한 그에 맞게 수정이 필요하다. 이는 개발 일정 및 비용에 큰 영향을 주는 요소이다. 따라서 최근 항공전자 산업에서는 소프트웨어의 이식성을 높이기 위해 개방형 아키텍처를 도입하고 있으며, 관련 연구 또한 활발히 진행되고 있다. 본 논문에서는 개방형 항공전자 아키텍처 중 하나인 FACE를 준수하기 위한 데이터 모델링 방안 및 고려사항에 대하여 설명한다. 또한, 항전 장비 중 하나인 VOR/ILS에서 출력하는 데이터에 대해 FACE 데이터 모델을 설계 및 구현하고, FACE Conformance Test Suite (CTS) 프로그램을 이용하여 데이터 모델이 FACE를 준수하여 설계되었다는 것을 검증한다.

Key Words : FACE(미래 항공 운용능력 환경), Data Model(데이터 모델), Open System Architecture(개방형 시스템 아키텍처), Avionics Software(항공전자 소프트웨어), Conformance(적합성), Portability(이식성)

1. 서 론

최근 항공전자 시스템 설계 시 개방형 시스템 아키텍처(Open System Architecture)를 도입하는 추세이다. 항공전자 소프트웨어는 하드웨어 장비에 의존적이어서 장비 노후화나 공급업체 변경으로 인해 하드웨어

장비가 변경되면, 장비의 인터페이스에 맞게 소프트웨어를 수정해야한다. 이에 따라 시간과 비용이 소요되며, 기술 고도화에 따른 기능 요구사항 증가로 인해 항전 소프트웨어가 복잡해짐에 따라 개발 및 통합 비용이 더욱 증가하고 있다. 따라서 최근 항공전자 시스템에 개방형 아키텍처를 적용하려는 연구가 활발히 진행되고 있다[1-2]. 이는 비용 감소 목적뿐만 아니라, 타 플랫폼으로의 소프트웨어 이식성(Portability)을 높이고 재사용성(Reusability)을 향상시키기 위해서이다.

개방형 시스템 아키텍처 중 하나인 FACE(Future

Airborne Capability Environment)[3]는 미국 주요 항공전자 시스템 업체 주축으로 형성된 FACE 컨소시엄에서 관리하는 항공전자 시스템 아키텍처이다. FACE는 최초로 미 해군에서 소요 제기되어 현재는 미 공군에서도 채택하고 있다. FACE 아키텍처의 목적은 소프트웨어를 다른 플랫폼에 이식할 때, 영향을 최소화하기 위한 환경을 제공하는 것이다. 기존의 플랫폼 종속적인 항전 소프트웨어의 이식성과 재사용성을 향상시켜 여러 항공기 플랫폼에 적용할 수 있어 효율적이다.

국외에서는 FACE 아키텍처 관련 연구가 다양하게 진행되고 있다. Xiaoyan Du 등은 항공전자 시스템의 복잡도 증가에 따라 FACE 기반으로 항전 시스템 시뮬레이션 및 검증 아키텍처 프레임워크를 제안하고 설계하였다. 해당 연구에서는 제안한 프레임워크를 경험적으로 검증하였고, FACE 컨소시엄에서 주관하는 적합성 프로세스를 거쳐서 검증하지는 않았다[4]. Jin Xiao 등은 FACE 데이터 모델 기반으로 항전 소프트웨어 개발을 지원하기 위한 데이터 시스템을 제안하고, 제안한 시스템의 성능과 효율성을 검증하였다[5]. Gedare 등은 FACE OSS(Operating System Segment) Safety Base Profile을 준수하는 시스템을 설계하고 구현하였다. 해당 연구에서는 FACE 3.0 기술 문서에서 요구하는 POSIX와 ARINC-653을 동시에 지원하도록 구현한 Deos/RTEMS가 FACE 적합성 프로세스를 거쳐 FACE 적합성을 인증 받았다[6].

FACE 관련 연구가 국외에서는 활발히 진행되고 있으나, 국내에서는 FACE 기반으로 무인 항공기 시스템 소프트웨어 개발 방법과 적합성 테스트에 대한 연구[7] 정도만 있을 뿐 상당히 미비한 실정이다.

기존의 연구에서는 FACE 아키텍처를 항공전자 시스템 전반에 적용한 방안을 제안하거나 특정 FACE 세그먼트의 인증을 위한 연구가 진행되었다. FACE 기반의 재사용 가능한 소프트웨어를 개발하기 위해서는 데이터 모델링이 선행되어야 한다. 개발한 소프트웨어의 FACE 준수 여부를 검증하기 위해 사용하는 FACE Conformance Test Suite(CTS) 프로그램에 필수적으로 데이터 모델을 입력해야 한다[8]. 본 논문에서는 FACE 데이터 모델링에 대하여 설명하고 항전 장비인 VOR/ILS에 대한 데이터 모델링 방안을 설명한다. 또

한 작성한 데이터 모델이 FACE CTS를 통과하는 것을 보인다.

본 논문의 구성은 다음과 같다. 2장에서 관련 기술을 소개하고, 3장에서는 FACE 데이터 모델에 대하여 설명한다. 3.1절에서는 VOR/ILS 장비 관리에 대한 FACE 데이터 모델링 방안을 제시하고, 3.2절에서는 FACE CTS을 이용하여 본 논문에서 기술한 데이터 모델이 FACE 요구사항을 준수하는 것을 보인다. 마지막으로 4장에서 결론을 맺는다.

2. 관련 기술

2.1 FACE 아키텍처

FACE는 미 해군 및 미국 주요 항공전자 시스템 개발업체 컨소시엄에서 연구 개발하고 있는 항공전자 시스템 아키텍처이다. FACE는 소프트웨어의 이식성과 재사용성을 향상시키는 것이 주요 목적으로, 항공 소프트웨어 공통 운용 환경(Common Operating Environment) 기술 표준을 제공하고 있다[5].

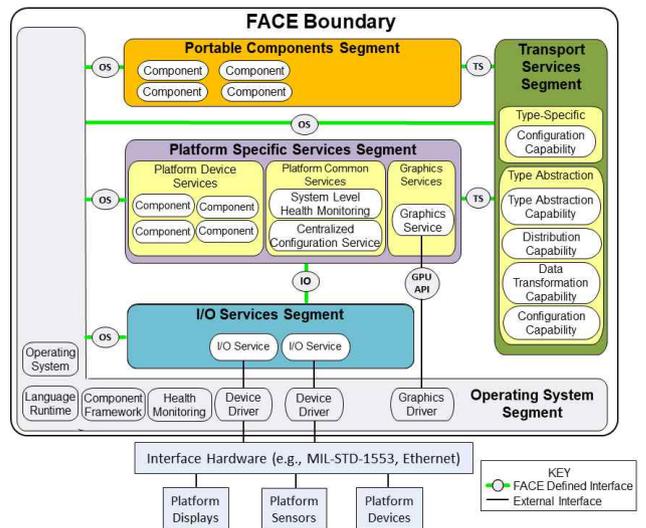


Fig. 1 FACE Architecture

FACE 아키텍처는 Fig. 1과 같이 OSS(Operating System Segment), IOSS(I/O Services Segment), PSSS(Platform-Specific Services Segment), TSS(Transport Services Segment), PCS(Portable Component Segment) 5개의 세그먼트로 구성된다

[9]. 각 세그먼트의 컴포넌트 간에는 FACE 정의 인터페이스를 통해 통신한다. OSS는 다른 FACE 세그먼트들의 기반을 제공하며, 운영체제 및 OS 레벨의 HM(Health Monitoring) 기능을 지원한다. IOSS는 연동 대상 하드웨어와의 통신을 수행하며, 표준 I/O 디바이스 드라이버 접근을 위한 공통 API를 제공한다. PSSS는 하드웨어 장비와 PCS 소프트웨어 컴포넌트 간 추상화 역할을 하며, 플랫폼에 의존적인 서비스를 제공한다. PSSS 컴포넌트는 이식이 가능하고, 대응되는 장비를 공유하는 플랫폼 간에 재사용할 수 있다. TSS는 PSSS와 PCS 간 데이터 전송 기능을 하며, 통신 메커니즘을 구현하기 위한 표준 인터페이스(Transport Service Interface)를 제공한다. PCS는 이식 가능한 컴포넌트와 공통 서비스의 집합으로 구성되며, 플랫폼 하드웨어에 독립적인 소프트웨어 컴포넌트를 포함한다.

FACE에서는 소프트웨어 기능을 수행하는데 필요한 서비스를 제공하는 “이식 가능한 단위”를 UoP(Unit of Portability)라는 개념으로 정의한다. 즉, 별도의 수정이나 재링크 없이 이식 가능한 단위 컴포넌트를 의미하며, 위에서 기술한 세그먼트 중 PSSS 컴포넌트, PCS 컴포넌트가 UoP에 해당한다[3].

2.2 FACE 적합성 프로세스

FACE 적합성을 인증하기 위한 절차는 크게 세 가지로, 검증(Verification), 증명(Certification), 등록(Registration) 단계로 진행된다. 검증과 증명은 각각 FACE 컨소시엄에서 공인된 전문가가 각 단계의 산출물을 확인하여 진행된다. 먼저 검증은 소프트웨어가 FACE 기술 표준 문서의 요구사항을 충족하는지 결정하는 단계로, CVM(Conformance Verification Matrix)과 CTS(Conformance Test Suite)을 이용하여 수행한다. 증명은 모든 관련 문서가 완성되었는지 확인하여 비즈니스 측면에서 FACE에 적합하다는 증명서를 획득하는 과정이다. 마지막으로 등록은 FACE 인증 목록에 인증 받은 소프트웨어 항목을 추가하는 과정으로, 필수적인 사항은 아니다[10]. 본 논문에서는 FACE 검증 단계에서 이용하는 CTS를 통해 데이터 모델을 검증한다.

2.3 FACE Conformance Test Suite

FACE 컨소시엄에서 공식적으로 배포하고 있는 FACE Conformance Test Suite(CTS)는 개발한 소프트웨어가 FACE 적합성을 위한 검증 단계에서 사용하며, FACE에 정의된 제약사항을 준수하는지 검사한다. CTS를 이용해 검사할 수 있는 대상은 데이터 모델, PCS 컴포넌트, PSSS 컴포넌트, TSS/IOSS/OS 인터페이스이다. 이 때, 데이터 모델은 PCS, PSSS, TSS 인터페이스 검사를 하려면 반드시 필요하다. 해당 프로그램은 링크 검사(Link Test)의 원리로 동작하는 것으로, 테스트 대상이 되는 코드를 테스트 코드와 결합하여 준수 여부를 결정한다. 이 도구는 코드의 기능에 대한 검증은 수행하지 않으며, 코드의 함수 원형에 대한 FACE 적합성 여부만 검사한다[8].

2.4 FACE 데이터 아키텍처

개방형 아키텍처를 적용하는 항공전자 시스템에서는 소프트웨어 이식성과 재사용성을 위해 소프트웨어 컴포넌트 간 데이터 상호작용을 구현하는 것이 중요하다. 하나의 플랫폼에서 동작하는 소프트웨어를 여러 공급업체에서 개발하게 되면 소프트웨어 간 인터페이스가 달라서 데이터가 호환되지 않을 수도 있기 때문이다[5]. 이를 위해 FACE에서는 소프트웨어 간에 데이터를 교환하기 위한 상호운용 가능한 접근 방식을 제공하고 있다.

FACE 데이터 아키텍처는 TS 인터페이스를 통해 UoP 간 상호운용 가능한 데이터 교환 수단을 지원하기 위한 데이터 모델, 명세 및 관리 정책을 일컫는다. FACE 데이터 아키텍처는 다음을 포함한다[3].

- 데이터 모델 언어
- SDM(Shared Data Model)
- SDM Governance Plan 문서
- USM(UoP Supplied Model)
- 언어 바인딩 명세

2.5 FACE 데이터 모델

데이터 모델은 FACE 컴포넌트 간 데이터 상호작용을 구현하기 위한 기반으로, 개념 데이터 모델(Conceptual Data Model, CDM), 논리 데이터 모델(Logical Data Model, LDM), 플랫폼 데이터 모델

(Platform Data Model, PDM), UoP 모델로 구분된다. Fig. 2는 FACE 데이터 모델링의 흐름을 나타낸 것이다[3].

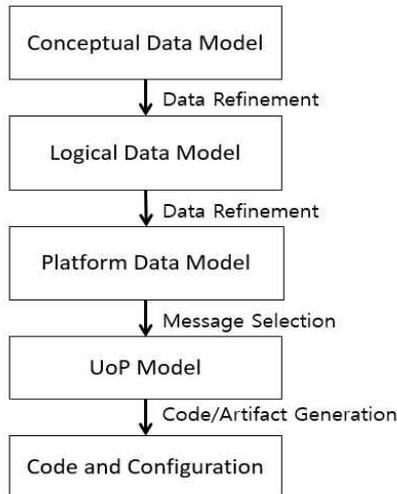


Fig. 2 FACE Data Modeling flow

FACE 데이터 모델링은 먼저 개념에 대한 아이디어를 정의한 다음, 세부적인 정보를 추가하면서 아이디어를 정교화 하는 과정으로 진행된다.

이는 MDA(Model Driven Architecture) 규칙과 DoDAF(DoD Architecture Framework) 메타모델과 일치하는 방식이다[11]. FACE 컨소시엄에서는 데이터 기본 요소 집합인 SDM(Shared Data Model)을 제공한다. SDM을 배포할 때, 이를 설명하는 문서도 함께 제공하고 있어 데이터 모델링 시 해당 문서를 참고할 수 있다. UoP를 개발하는 소프트웨어 개발자는 이를 활용하여 PCS, PSSS 컴포넌트에 해당되는 UoP 모델을 작성할 수 있고, 필요에 따라 SDM을 확장할 수도 있다. 또한, 작성된 UoP 모델을 프로그래밍 언어로 매핑 하여 코드를 생성할 수 있다.

뷰(View)는 여러 개체(Entity)에서 특정한 특성(Characteristic)들의 모음이다. 개념 뷰, 논리 뷰, 플랫폼 뷰가 있으며, 플랫폼 뷰는 UoP 모델에서 컴포넌트 간 주고 받는 메시지를 정의하는 포트로 사용되기 때문에 반드시 정의해야 한다[3, 11].

2.5.1 개념 데이터 모델링

개념 데이터 모델링은 모델링 하고자 하는 대상을

추상적인 아이디어로 표현하는 것이다. 먼저 기본적인 개념 아이디어인 개체(Entity)를 정의한 후, 개체를 구성하는 속성(Composition)을 정의한다. 그런 다음에 FACE SDM 문서를 참고하여 관찰 가능한 항목(Observable)을 개체의 속성에 매핑 한다. 개체와 속성을 기반으로 개념적인 메시지를 정의하는 과정이다 [11].

2.5.2 논리 데이터 모델링

논리 데이터 모델링은 개념 모델링에서 정의한 개념 개체를 구체화하는 과정으로, 개념 개체를 실현하는 논리 개체를 정의한다. 논리 개체는 관찰 가능한 항목의 속성이 수학적으로 어떻게 측정되는지 측정 항목(Measurement)을 정의한다. 측정 항목 또한 SDM 문서를 참고하여 선택한다. 개념 모델링에서 정의한 개념 요소들은 추가적인 정보를 어떻게 선택하느냐에 따라 논리적 수준에서 다양하게 표현될 수 있다[11].

2.5.3 플랫폼 데이터 모델링

플랫폼 데이터 모델링은 논리 모델링에서 정의한 논리 개체를 구체화하는 과정으로, 측정 항목이 IDL(Interface Definition Language) 데이터 타입에 매핑 되는 방법을 설명한다. 즉, 컴퓨터가 이해할 수 있는 방식으로 데이터를 표현한다. 특정 데이터 타입을 정의하고 결정해야 하며, FACE 컴포넌트 간 주고 받는 메시지에 데이터를 매핑하기 위해 뷰(View)를 추가해야 한다. 뷰는 UoP 모델에서의 포트를 정의하는데 사용된다. 플랫폼 데이터 모델링은 실제 물리적인 장치에 대한 모델링을 하기 때문에 물리적 모델링이라고도 한다[11].

2.5.4 UoP 모델링

UoP 모델링은 플랫폼 모델을 UoP에 해당하는 소프트웨어로 표현하기 위해 필요한 메시지 인터페이스 정보를 작성하는 과정이다. 개발하고자 하는 소프트웨어 컴포넌트의 특성에 따라 UoP 모델의 속성을 PCS 또는 PSSS로 지정하고, 입출력 흐름에 대하여 메시지 포트를 정의해야 한다. 그리고 플랫폼 모델링에서 정의한 플랫폼 뷰와 포트를 매핑 하여 포트를 통해 전달되는 데이터를 기술한다[3, 11].

3. 본 론

3.1 FACE 데이터 모델링 방안

FACE 데이터 모델은 CTS에 FACE XMI(XML Metadata Interchange) 형식의 파일(.face 확장자)로 제공되어야 한다. FACE XMI 형식으로 작성하는 방법은 직접 코드를 작성할 수도 있고, FACE 모델링을 지원하는 SCAD, MTF(Modeling Tool for FACE)와 같은 툴을 이용하여 모델링을 한 후에 파일로 추출할 수도 있다. 본 논문에서는 ANSYS사의 SCAD 툴을 이용하여 데이터 모델링을 수행하였다. SCAD의 Avionics Package는 FACE 2.1과 3.0 기반의 데이터 모델링을 지원하고 있다[12].

본 논문에서는 VOR/ILS 장비에 관련된 FACE 데이터 모델링을 진행한다. VOR/ILS는 민간 항공기의 계기비행 및 착륙에 사용되는 항전 장비이다. VOR(VHF Omni-directional Range)은 전방향으로 방사하는 신호를 수신하여 항공기의 위치를 파악하는데 사용되며, ILS(Instrument Landing System)는 로컬라이저와 글라이드 슬롭, 항공기와 활주로 간 거리를 알려주는 마커비컨으로 구성된다.

항공기의 임무 컴퓨터(Mission Computer, MC)는 I/O 서비스를 통해 VOR/ILS 장비와 통신을 한다. 임무 컴퓨터는 VOR/ILS로부터 수신한 항법 데이터를 이용해 계산한 후 MFD(Multi-Function Display)와 같은 시현 장치에 데이터를 전송하는 역할을 한다. 위에서 설명한 임무 컴퓨터의 기능은 다음과 같다.

- VOR/ILS 데이터 수신 기능
- 수신 데이터 이용한 항법 계산 기능
- 시현 장치에 데이터 전송 기능

Table 1 Data Message from VOR/ILS

Data	Type	Units
VOR Bearing	short	Degrees
ILS Localizer Deviation	float	Degrees
ILS Glide Slope Deviation	float	Degrees
ILS Marker Beacon Signal	enum	-

이와 같은 기능을 수행하는 컴포넌트를 FACE 아키텍처에 적용한 설계는 Fig. 3과 같다. 본 논문에서는 Fig. 3에서 PSSS 컴포넌트에 해당하는 MC의 VOR/ILS 장비 관리 기능만 제한적으로 모델링한다. MC는 VOR/ILS 데이터뿐만 아니라 장비의 상태를 모니터링하기 위해 BIT(Built-In Test) 결과와 같은 상태 정보도 수신하는데, 본 논문에서는 VOR/ILS의 데이터 메시지에 대한 모델링만 수행하였다. 임무 컴퓨터가 VOR/ILS로부터 수신하는 데이터 메시지를 간략하게 표현하면 Table. 1과 같다.

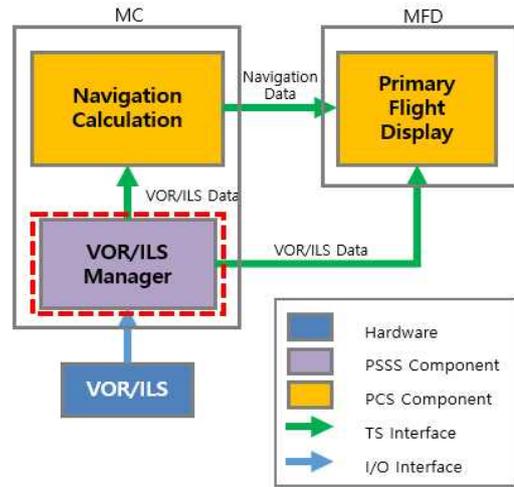


Fig. 3 Design of VOR/ILS Manager based on FACE Architecture

3.1.1 개념 데이터 모델링

본 논문에서 모델링한 개념 데이터 모델을 도식화하여 표현하면 Fig. 4와 같다. VOR/ILS 장비는 개념 개체로 정의하였고, 개념 개체의 구성요소는 장비에서 출력되는 데이터를 기반으로 정의하였다. 개념 개체를 구성하는 각 구성요소에 대하여 관찰 가능 항목(Observable)을 정의해주어야 하며, SDM 문서를 참고하여 Table 2와 같이 VOR 방위(Bearing)는 “Angle”로, ILS 데이터인 로컬라이저 편차와 글라이드 슬롭 편차 또한 “Angle”로 정의하였다. ILS 마커 비컨 신호 데이터는 항공기와 마커 비컨이 설치된 활주로 간의 거리에 따라 얻어지는 데이터이므로 “Distance”로 정의하였다. 개체의 고유 식별자는 FACE 기술 표준 문서에서 요구하는 필수적인 사항이다.

Table 2 VOR/ILS Conceptual Modeling

Entity	Composition	Observable
VORILS_Conceptual	UniqueID	UniqueIdentifier
	Bearing	Angle
	Deviation	Angle
	Distance	Distance

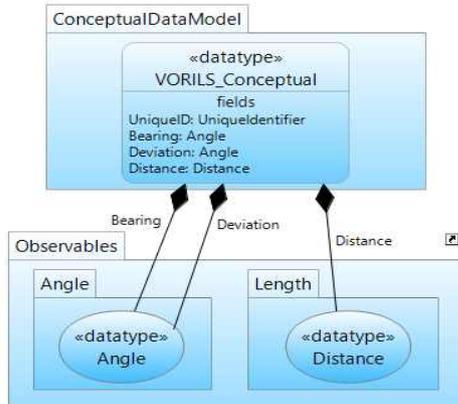


Fig. 4 VOR/ILS Conceptual Data Model

3.1.2 논리 데이터 모델링

Figure 5은 논리 데이터 모델을 도식화 한 것으로, 개념 모델링에서 정의한 개념 개체의 속성을 측정 항목(Measurement)으로 매핑 해야 한다. Table 3과 같이 자북 기준으로 측정되는 VOR 방위는 “CourseMagneticNorthMeasurement” 측정 항목으로 설정하였고, ILS 편차는 로컬라이저 편차와 글라이드 슬롭 편차 각각 측정 항목으로 설정하였다. 마커 비컨 데이터는 “DistanceMeasurement” 항목으로 설정하였다.

Table 3 VOR/ILS Logical Modeling

Entity	Composition	Measurement
VORILS_Logical	UniqueID	UniqueID_UUID_Text
	Bearing	CourseMagneticNorthMeasurement
	LateralDeviation	LateralRotationMeasurement
	VerticalDeviation	VerticalRotationMeasurement
	Distance	DistanceMeasurement

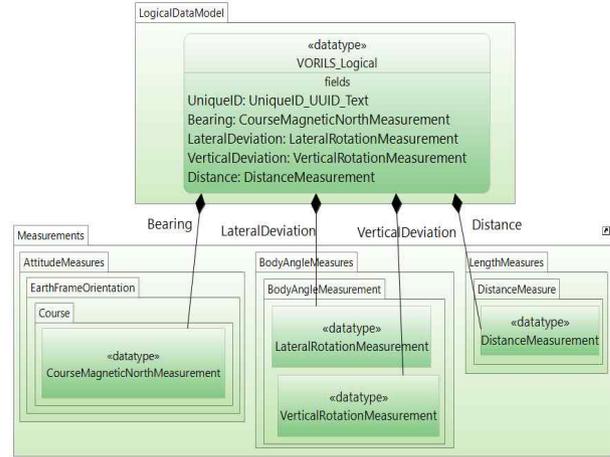


Fig. 5 VOR/ILS Logical Data Model

개념 개체에서와 같이 논리 개체에서도 고유 식별자는 필수적인 사항으로, 텍스트 문자로 가정하여 측정 항목을 정의하였다.

3.1.3 플랫폼 데이터 모델링

Figure 6은 플랫폼 데이터 모델을 도식화 한 것으로, 상단은 플랫폼 개체를 정의한 것이고 하단은 플랫폼 뷰이다. 논리 모델링에서 정의한 논리 개체를 플랫폼 개체로 실현하였다. 이 때, 플랫폼 개체를 구성하는 속성들의 타입을 설정해주어야 한다. Table 4와 같이 VOR 방위는 정수 타입, 로컬라이저 편차와 글라이드 슬롭 편차는 실수 타입으로 설정했다. 마커비컨 신호의 경우, 열거형(enum) 타입을 정의하여 설정하였다. 플랫폼 뷰는 UoP 모델에서 포트로 사용하며, 뷰의 속성 또한 플랫폼 개체의 타입과 동일하게 설정하였다.

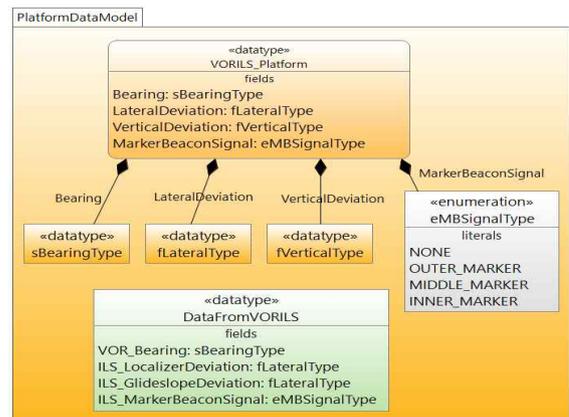


Fig. 6 VOR/ILS Platform Data Model

Table 4 VOR/ILS Platform Modeling

Entity	Composition	Type
VORILS_ Platform	Bearing	sBearingType (short)
	LateralDeviation	fLateralType (float)
	VerticalDeviation	fVerticalType (float)
	MarkerBeaconSignal	eMBSignalType (enum)

3.1.4 UoP 모델링

UoP 모델링 단계에서는 개발하고자 하는 소프트웨어 컴포넌트의 특성에 따라 PCS 또는 PSSS로 지정해야 한다. 본 논문에서는 UoP가 VOR/ILS 장비와 관련된 기능을 수행하기 때문에 Fig. 7과 같이 PSSS 컴포넌트로 정의하였다. 해당 UoP 모델의 메시지 포트는 플랫폼 데이터 모델링에서 정의한 플랫폼 뷰이다.

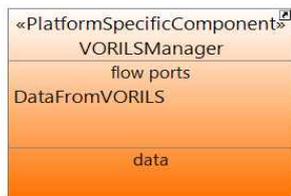


Fig. 7 VOR/ILS UoP Model

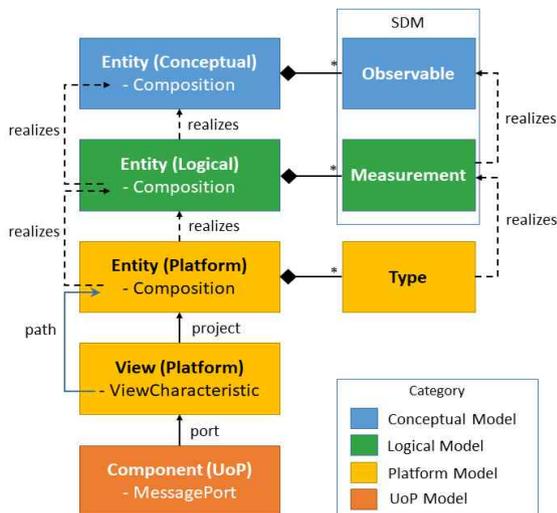


Fig. 8 Relationship between FACE Data Models

개념 모델, 논리 모델, 플랫폼 모델, UoP 모델 간의 관계를 나타내면 Fig. 8과 같다. 논리 모델의 개체는 개념 모델을 실현하고, 플랫폼 모델은 논리 모델을 실제화한다. 개념 개체, 논리 개체, 플랫폼 개체는 여러 개의 구성 요소로 이루어지며, 개념 모델, 논리 모델, 플랫폼 모델의 구성 요소는 각각 관찰 가능 항목, 측정 항목, 타입으로 지정된다. 플랫폼 뷰는 여러 특성으로 구성되며, 각 특성은 플랫폼 개체의 구성 요소들을 path로 지정해야 한다. UoP 모델은 플랫폼 뷰를 포트로 지정한다.

본 논문에서 작성한 FACE 데이터 모델 파일(.face 확장자)의 개략적인 구조를 나타내면 Fig. 9와 같다.

```

<?xml version="1.0" encoding="UTF-8"?>
<face:DataModel xmlns:face="http://www.omg.org/XMI" ...>
  <cdm xmlns:face="face:ConceptualDataModel" ...>
    <element xmlns:face="face:ConceptualEntity" ...>
      <composition xmlns:face="face:ConceptualComposition" ... rolename="Bearing"/>
      <composition xmlns:face="face:ConceptualComposition" ... rolename="Deviation"/>
      <composition .../>
    </element>
  </cdm>
  <!-- Conceptual Data Model (SDM) -->
</cdm>
  <ldm xmlns:face="face:LogicalDataModel" ...>
    <!-- Logical Data Model (SDM) -->
  </ldm>
  <ldm xmlns:face="face:LogicalDataModel" ...>
    <element xmlns:face="face:LogicalEntity" ...>
      <composition xmlns:face="face:LogicalComposition" ... rolename="Bearing"/>
      <composition xmlns:face="face:LogicalComposition" ... rolename="LateralDeviation"/>
      <composition .../>
    </element>
  </ldm>
  <pdm xmlns:face="face:PlatformDataModel" ...>
    <element xmlns:face="face:PlatformEntity" ...>
      <composition xmlns:face="face:PlatformComposition" ... rolename="Bearing"/>
      <composition .../>
    </element>
    <element xmlns:face="face:PlatformView" ... name="DataFromVORILS">
      <characteristic xmlns:face="face:PlatformCharacteristicProjection" ...
        path=".Bearing"/>
      <characteristic .../>
    </element>
  </pdm>
  <uopModel xmlns:face="face:UoPModel" ...>
    <element xmlns:face="face:UoPPlatformSpecificComponent" ...>
      <port xmlns:face="face:UoPMessagePort" ... name="DataFromVORILS" />
    </element>
  </uopModel>
</face:DataModel>

```

Fig. 9 VOR/ILS Data Model File Structure

3.2 FACE 데이터 모델링 고려사항

FACE 데이터 모델링에서 고려해야 할 사항은 FACE 기술 표준 문서의 요구사항을 충족했는지 확인하는 것이다. FACE CTS에서는 데이터 모델의 적합성 검사 시, FACE 메타 모델 검증, OCL 제약사항 검사, SDM 적합성 검사를 수행한다. 데이터 모델은 요구사항에 맞게 설계되어야 하며, 이를 위해 SDM 문서와 SDM Governance Plan 문서[13]를 참고하는 것이 좋다. 또한 개념 모델, 논리 모델, 플랫폼 모델, UoP 모델 간의 관계를 이해하고 각 모델에서 필수적인 요소를 정

의하는 것이 중요하다.

3.3 FACE 준수 검증

3.3.1 검증 환경

본 논문에서 제시한 데이터 모델을 검증하기 위한 환경은 다음과 같다.

- FACE Technical Standard: 2.1
- FACE Shared Data Model: v2.1.35
- FACE Conformance Test Suite: v2.1.0r9
- OS: Windows 10
- FACE Data Modeling Tool: SCADE 2021 R1

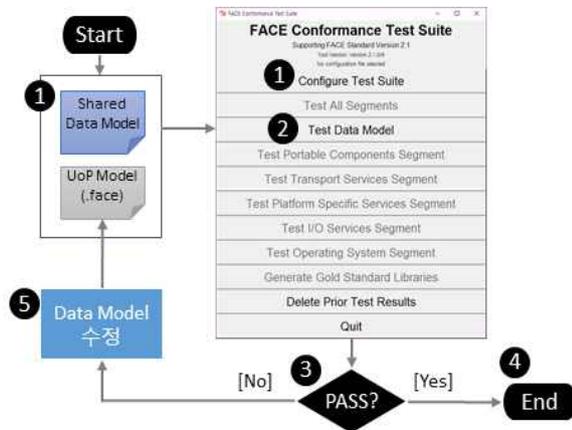


Fig. 10 Data Model Test Procedure using FACE Conformance Test Suite

3.2.2 검증 방법

FACE Conformance Test Suite를 이용하여 데이터 모델의 FACE 적합성 여부를 테스트하는 과정을 나타내면 Fig. 10과 같다. FACE 데이터 모델 검증을 위한 과정은 다음과 같이 수행하였다.

- 1) FACE CTS의 환경설정에서 Shared Data Model 과 작성한 Data Model 파일을 설정한다.
- 2) FACE CTS에서 데이터 모델 테스트를 수행한다.
- 3) 테스트 결과를 확인한다.
- 4) 테스트 결과가 모두 통과하면 완료한다.
- 5) 테스트 중 하나라도 통과되지 않으면 데이터 모델을 수정하여 1번부터 반복하여 모두 통과할 때까지 수행한다.

3.3.3 검증 결과

본 논문에서 기술한 방안으로 작성한 FACE 데이터 모델을 FACE CTS에서 테스트한 결과는 Fig. 11과 같다. FACE 메타 모델 검증, OCL 제약사항 검사, SDM 적합성 검사 3가지 검사에 대하여 모두 통과를 하여 결과적으로 데이터 모델에 대한 테스트가 통과하였다. 이는 작성한 모델이 FACE의 요구사항에 적합하다고 볼 수 있다.

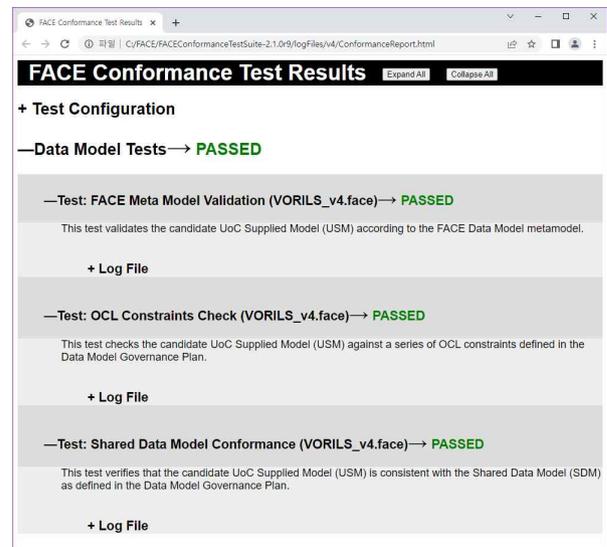


Fig. 11 Data Model Test Results using FACE Conformance Test Suite

4. 결 론

본 논문에서는 개방형 항공전자 소프트웨어 아키텍처인 FACE에 대하여 소개하고, FACE를 준수하는 소프트웨어를 개발하기 위해 필수적인 FACE 데이터 모델링에 대한 방안을 기술하였다. 이를 기반으로 VOR/ILS의 데이터 모델링을 수행하고, 해당 모델이 FACE Conformance Test Suite의 검사를 통과하는 것을 보였다. 다만 데이터 모델만으로는 FACE를 준수한다고 할 수 없고, 데이터 모델을 기반으로 구현한 PCS, PSSS 컴포넌트가 FACE 준수의 단위(Unit of Conformance)이다. 하지만 PCS, PSSS 컴포넌트의 준수를 위해서는 반드시 데이터 모델링이 선행되어야 한다. 데이터 모델링 자체는 수행하는 사람에 따라 다르게 할 수 있다. 그러나 FACE 요구사항에 맞게 모델

링을 수행하지 않으면 FACE 적합성 검사를 통과할 수 없다.

추후 FACE 데이터 모델에서 정의한 포트와 인터페이스를 기반으로 PSSS, PCS 컴포넌트의 소프트웨어를 구현하고 이를 검증하는 방안에 대한 연구가 필요하다.

후 기

본 논문은 산업통상자원부 재원으로 한국산업기술평가관리원(KEIT)의 지원을 받아 수행된 연구 결과입니다. [사업명: 항공우주부품기술개발사업-수출유망부품 및 핵심기술개발 / 과제명: 중소형 항공기급 개방형 항공전자 시스템 아키텍처 및 소프트웨어 개발 / 과제 고유번호: 20005378]

References

- [1] T. Joyce L. "A comparison of avionics open system architectures.", *ACM SIGAda Ada Letters*, vol. 36, no. 2, pp. 22-26, December 2017.
- [2] H. Mark, et. al., "A comparison of open architecture standards for the development of complex military systems: GRA, FACE, SCA NeXT (4.0)." *MILCOM 2012-2012 IEEE Military Communications Conference. IEEE*, pp.1-9, 2012.
- [3] The Open Group, "Technical Standard Future Airborne Capability Environment (FACE™), Edition 2.1.1", June 2017.
- [4] D. Xiaoyan, et al., "A FACE-based Simulation and Verification Approach for Avionics Systems." *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 1509-1513, October, 2019.
- [5] X. Jin, et. al., "Design of a Data System for the Avionics System based on the Open System Architecture", *2019 14th IEEE Conference on Industrial Electronics and Applications(ICIEA)*, pp. 1958-1962, June 2019.
- [6] G. Bloom, et al., "Harmonizing ARINC 653 and Realtime POSIX for Conformance to the FACE Technical Standard." *2020 IEEE 23rd International Symposium on Real-Time Distributed Computing (ISORC)*, pp.98-105, May 2020.
- [7] H.C, Cho and K.Y. Park, "Development and Verification Methodology for Small Civil Unmanned Aerial Vehicle System based on Open System Architecture", *Journal of Platform Technology*, Vol.8, No.2, pp.32-43, June 2020.
- [8] Vanderbilt University, "Conformance Test Suite Manual", 2019.
- [9] Future Airborne Capability Environment (FACE™) Consortium, "https://www.opengroup.us/face/documents.php?action=show&dcat=&gdid=20893"
- [10] The Open Group, "FACE™ Conformance Certification Guide", August 2016.
- [11] The Open Group, "Reference Implementation Guide for FACE™ Technical Standard, Edition 2.1", April 2016.
- [12] G. Goretkln, et al. "Developing Portable and Reusable Applications with SCADE, the FACE™ Technical Standard, and ARINC 661", ANSYS, September 2019.
- [13] The Open Group, "Shared Data Model Governance Plan, Edition 2.1", October 2015.