

技術 論文

DOI: <http://dx.doi.org/10.5139/JKSAS.2011.39.11.1060>

실시간 임베디드 시스템용 자바 가상머신을 이용한 다기능 시현 비행운용 프로그램 및 미들웨어 설계 및 구현

원현권*, 정재훈*, 최경식*, 김종필*, 김인규*

Design and Implementation of Multi-Function Display Operational Flight Program and Middleware Using Real-Time and Embedded System Java Virtual Machine

Hyeon-Kwon Won*, Chai-Hun Jeong*, Kyong-Sik Choi*, Jong-Pil Kim* and In-Gyu Kim*

ABSTRACT

In this paper, we present a development of an Operation Flight Program(OPF) on Real Time Operating System(RTOS) and Java Virtual Machine(JVM) of real-time and embedded system. The OPFs are consisted of Multi Function Display(MFD), Integrated Up Front Control(IUFC), Head Up Display(HUD) and Fire Control(FC) and loaded for localization Mission Computer(MC). This paper describes the structure and implementation of a MFD OPF and middleware based on Java.

초 록

본 논문에서는 실시간 운영체제(Real Time Operating System, RTOS)기반의 실시간 임베디드 시스템용 자바 가상 머신(Jamaica Virtual Machine, JVM)을 이용한 비행운용 프로그램(Operational Flight Program, OPF)의 객체 지향적 개발사례에 대해 기술하였다. 국산화 임무 컴퓨터(Mission Computer, MC)에는 다기능 시현(Multi Functional Display, MFD), 통합 전방 상황 제어(Integrated Up-Front Control, IUFC), 전방 시현(Head-Up Display, HUD) 및 화력 제어(Fire Control, FC)를 지원하는 OPF가 각각의 프로세스에 탑재되어 구동되는데, 본 논문에서는 자바(Java) 기반의 MFD OPF(다기능 시현기에 조종사 참고 정보를 제공하는 비행운용 프로그램)와 미들웨어의 설계 및 구현에 대하여 기술하였다.

Key Words : Real Time Operating System(실시간 운영체제), Mission Computer(임무 컴퓨터), Real Time and Embedded System JAVA VM(실시간 임베디드 시스템용 자바 가상 머신), Operational Flight Program(비행 운용 프로그램), Middleware(미들웨어)

1. 서 론

임무 컴퓨터는 항공기에 탑재되는 각종 구성품

(Line Replacement Unit, LRU)과의 1553B 등의 인터페이스를 통해 항법 시스템 및 무장 컴퓨터 제어 기능을 수행 하고, MFD, HUD 그리고 IUFC를 통한 각종 데이터 입력 및 비행정보를 시현하는 항전 시스템의 핵심 장비이다.

본 논문은 Fig. 1과 같이 T-50용 국산화 임무 컴퓨터(1)에 탑재될 항공전자 소프트웨어인 실시간 임베디드 시스템용 자바 가상 머신과 자바 네

† 2011년 7월 14일 접수 ~ 2011년 10월 26일 심사완료

* 정회원, 한국항공우주산업(주) 항공ES팀

교신저자, E-mail : maximin@koreaero.com

경남 사천시 사남면 유천리 802번지 우664-710



Fig. 1. Localization MC for T-50

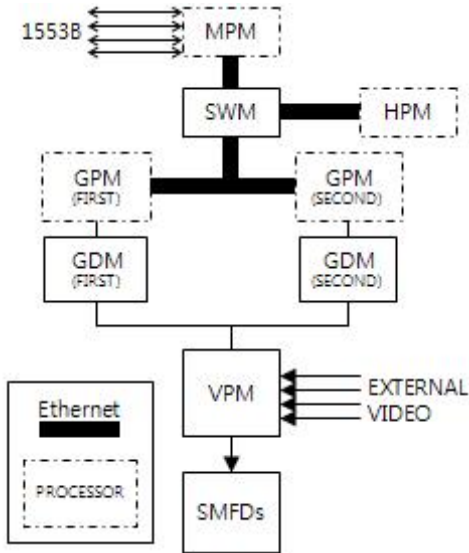


Fig. 2. Localization MC Structure

이티브 인터페이스(Java Native Interface, JNI)를 기반으로 한 MFD OFP와 다중 프로세스의 제어 및 진단 기능을 수행하는 미들웨어에 대한 설계 및 구현 방법을 제시하였다(2). 또한 플랫폼 독립적이고, 메모리 최적화(Garbage Collection)(3) 와 예외처리(Exception Handler)를 지원하는 신뢰성이 높은 자바(Java)를 이용한 객체 지향적 임베디드 소프트웨어 개발 과정을 기술하고자 한다.

II. 비행운용프로그램

2.1 국산화 임무 컴퓨터 (Mission Computer, MC) 구성

Fig. 2처럼 국산화 MC는 MPM(Main Processor Module), HPM(Head-Up Processor Module), GPM(Graphic Processor Module)으로 이루어진 다중 프로세스를 중심으로 내, 외부 인터페이스가 연결된 구조로 디자인 되었으며, 세부적인 하드웨어 구성요소는 Table 1과 같다. 각 프로세스들은 개략적으로 다음과 같은 업무를 수행한다.

Table 1. Hardware Configuration of Localization MC

구성품	기능
MPM	FC OFF 탑재, 1553B 및 Analog/Digital I/O 제어를 통한 데이터를 관리
GPM	MFD/IUFC OFF 탑재, GDM 및 VPM 제어
VPM	내부 심볼 및 외부 비디오 신호 제어
GDM	그래픽 심볼(OpenGL) 생성
HPM	HUD OFF 탑재, HDM제어
HDM	그래픽 심볼 생성
SWM	프로세스간 통신 제어

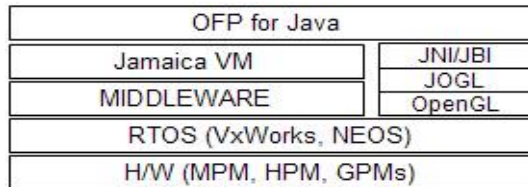


Fig. 3. Embedded Software Structure on Processor

2.2 OFP에 적용된 임베디드 소프트웨어 개발환경 및 계층구조

MC에 탑재되는 응용프로그램인 OFP는 Fig. 3과 같은 계층 구조로 이루어져 있다. MC 내부의 프로세스(MPM, HPM, GPM) 상에 VxWorks와 NEOS RTOS(Real Time Operating System) 및 OpenGL (Open Graphics Library)의 인터페이스 역할을 하는 실시간 JVM(Java Virtual Machine)인 Jamaica Virtual Machine(Jamaica VM)을 바탕으로 자바 기반의 객체 지향적 OFP가 설계 및 구현 되었으며, 또한 JVM은 자바 네이티브 인터페이스(Java Native Interface, JNI) 와 자바 바이너리 인터페이스(Java Binary Interface, JBI)를 통한 C언어 기반의 라이브러리 형태로 설계 및 구현된 미들웨어로의 접근을 지원한다.

상위 구조를 바탕으로 Fig. 4와 같이 1) 통합 모델링 언어(Unified Modeling Language UML)를 지원하는 통합 개발 도구 Rhapsody(IBM)를 이용하여 디자인 및 코딩을 하고, 2) 유닛테스트 도구인 JUNIT(Java Unit Test Tool) 4.0을 이용한 개별 함수에 대한 유닛 테스트를 수행 하였으며, 3) 정적 분석 툴인 Resort for JAVA(Soft 4 soft)를 이용한 규칙 기반(Rule Base) 소스 분석, 4) Jamaica Builder들 통해 OFP의 최종 실행 이미지를 생성하는 순으로 객체 지향적 OFP를 설계 및 구현하였다.

개발 도구	P C S I M U L A T O R	S/W 산출물
개발지원 툴 -Eclipse -JUnit -Jamaica Builder -Resort -for Java (정적 분석)		코드 -OFFP -미들웨어
설계 -Rhapsody		JamaiciaVM -JNI/JBI -Jamaica Builder
문서작성 -FrameMaker		실행 이미지
요구도 추적 -DOORS		시험 환경
형상관리 -PVCS		SIL -가상비행 시뮬레이션 -MC 확장 모듈 -EDTU -DMM

Fig. 4. Development Configuration and Structure

2.3 T-50에 적용된 C 언어 기반의 OFF와 Java 기반의 OFF 비교

본 논문에서 설계 및 개발된 OFF는 객체 지향적 방법을 사용함으로써 재사용성을 높이고, 자바 가상머신을 사용함으로써 플랫폼 독립적인 개발이 가능한 Java 언어를 사용 하였다. 기존 T-50용 임무컴퓨터에 탑재된 OFF에 사용된 주 개발언어는 C 언어지만, 본 논문에서는 Java언어를 기반으로 한다. 그리고 각종 하드웨어에 종속적인 미들웨어(1553B, ADID, UART, System Management 등 관련 API)에 대해서는 기존 OFF는 Ada 기반의 API를 참조하지만, 본 논문에서의 OFF는 C 언어 기반의 API를 JNI/JBI를 통해 참조한다. 또한 본 논문에서는 시스템간의 이식성이 좋고, 고급 및 저급 언어 사이의 인터페이스가 좋은 C언어와 객체 지향적 언어의 장점 및 플랫폼 독립적이고 메모리 최적화(Garbage Collection)를 지원하는 자바 언어가 가진 언어적 장점을 최대한 이용함으로써 보다 효율적인 프로그램 개발 방식을 임무 컴퓨터에 탑재되는 소프트웨어 개발에 적용 하고자 하였다.

2.4 실시간 임베디드 시스템용 자바 가상머신 기반의 소스 빌드 구조

임무 컴퓨터에서 실행 파일 생성을 위한 OFF 빌드는 다음과 같은 과정을 거친다. Fig. 5와 같

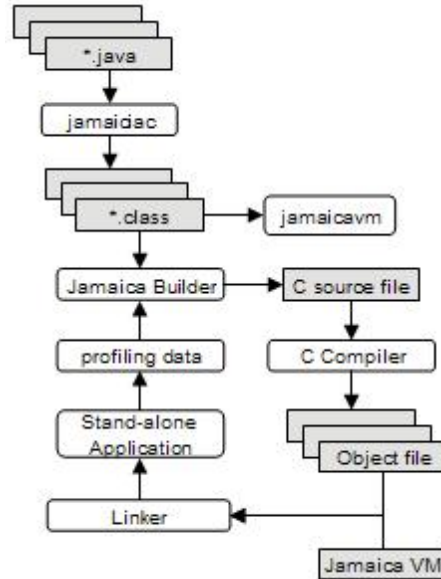


Fig. 5. Cross Compile Environment (Jamaica Tool Chain)

이 JVM인 Jamaica VM을 기반으로 자바로 구현된 MFD OFF 소스 파일들은 Jamaica Builder에 의해 C 소스로 변경된다. 이후 링커에 의해 Jamaica VM과 C 컴파일러를 통해 생성된 오브젝트 파일의 링크이후 실행 가능한 이미지가 생성된다(4). 이러한 빌드 구조는 상대적으로 수행속도가 느린 인터프리터(Interpreter)의 방식을 개선한 것으로서, Fig. 6과 같이 PC상에서 OFF를 개발할 경우 자바 가상 머신 기반의 자바 바이트 코드를 이용한 시뮬레이터(PC BENCH)를 개발하여 시뮬레이션을 수행한다. 하지만 바이트코드에 대하여 인터프리터하면 시스템에 대한 이식성은 뛰어나지만 성능 저하의 문제가 발생하므로, 실제 임베디드 시스템에 로드 시에는 바이트 코드 형태로 되어있는 자바 어플리케이션을 수행되기 전에 미리 컴파일(Cross Compile)해서 최적화된 바이너리 코드를 생성하는 AOTC(Ahead Of Time Compile) 형태의 기술을 사용함으로써 수행속도에서 최대 성능을 낼 수 있도록 하였다. AOTC의 핵심인 자바 바이트 코드를 기계어 코드로 바꾸는 방식에는 직접 기계어 코드를 생성하는 방식과 C코드를 생성하는 방식이 있는데, 본 논문에서는 바이트 코드를 C코드로 변환한 후 C컴파일러로 컴파일 하는 방식이 사용 되었다(5).

또한 Jamaica Builder에 제공하는 Profiling Data기법을 통해 OFF의 수행 속도를 최적화 하

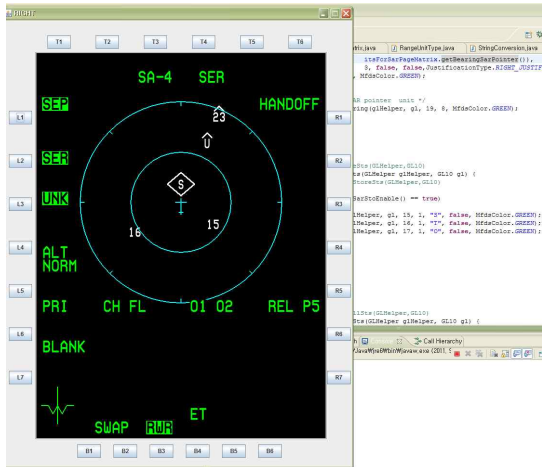


Fig. 6. PC Bench for MFD OFF

였다. 이는 작은 메모리와 CPU성능을 가진 임베디드 시스템 일수록 중요한 기능으로서, 빠른 컴파일된 코드와 작은 자바바이트 코드 사이의 수행시간 및 객체의 참조비율 등의 상관관계를 이용하여 최적의 실행이미지를 생성하는 기술이다.

2.5 OFF 개요 및 개발 절차

비행운용 프로그램은 프로세서 모듈 MPM, HPM에 각각 FC, HUD OFF가, 그리고 GPM에 MFD와 IUFC OFF가 탑재된다. FC OFF는 항법, 적아식별, 통신, 공대공, 공대지 임무 기능, 고장 시현 및 기록 그리고 각 프로세스간의 통신 역할을 수행하며. HUD OFF는 전방 시현기에 항공기 속도, 고도, 각종 비행 정보 심볼 및 임무 관련 정보를 시현 시켜 주는 역할을 수행한다. 또한 IUFC OFF는 임무 계획 및 수행을 위한 항공 전자 모드 및 기능 선택, 표적 정보 등의 데이터 입력을 수행하며. MFD OFF는 다기능 시현기(MFD)에 레이더 및 엔진 정보, 항공기 장착 무장 그리고 각종 조종사 참조정보를 시현시켜 주는 역할을 수행한다.

개발된 각각의 OFF는 개발 보드 없이 PC(Personal Computer)상에서의 단위 테스트를 거쳐, 실제 국산화 MC에 탑재되어 SIL(System Integration Laboratory)에서 소프트웨어 통합시험 및 시스템 통합시험을 수행 한다. 이후, 항공기에서 지상 시험 및 비행 시험이 이루어진다. 특히 항공기에 MC 탑재 전 SIL에서 수행되는 시스템 통합시험에서는 실제 항공기 탑재 장비와 기타 비행 시뮬레이션으로 부터 입력된 데이터를 이용하여 각 OFF의 정상동작 유무 확인 및 실제

항공기 탑재 장비와의 통신 적합성 점검은 물론, 실제 비행 중 MC에서 발생 할 수 있는 문제점을 미연에 방지 할 수 있다.

2.6 소프트웨어 공학 기반의 OFF 개발

항공전자 임베디드 소프트웨어는 체계 규격서(System Specification), 개발 규격서(Development Specification) 그리고 제품 규격서(Product Specification)에 명시된 산출물에 근거하여, 1) 요구도 분석, 2) OFF 설계, 3) 코딩, 4) 통합시험의 절차로 개발 된다(6).

각각의 OFF는 CMMI(Capability Maturity Model Integrated) 레벨 3 환경을 기반으로 개발이 이뤄 졌으며, 기타 DO-178B Level B, OOTiA(Object Oriented Technology in Aviation)(7)를 만족할 만한 수준으로 개발 되었다. 특히 OOTiA는 객체지향 프로그램 개발 시 준수해야 할 가이드라인을 제시하는데, 그중 C++와 다르게 다중 상속을 금지하고 참조만 허용하는 자바의 특성을 그대로 반영하고 있다.

2.7 MFD OFF 개발 절차 및 구성

비행운용 프로그램 중 다기능 시현장치에 각종 시각 정보를 시현하여 조종사에게 각종 비행 정보를 제공하는 MFD OFF의 세부 개발 환경 및 절차는 다음과 같다.

자바 리얼타임 규격(Real time Specification for Java, RTS)을 만족하는 실시간 임베디드 시스템용 자바 가상머신을 기반으로 개발된 MFD OFF는 UML2.1기반의 모델 중심 개발(Model Driven Development, MDD)을 지원하는 자바 통합 개발 툴인 Rhapsody에서 디자인 되고, 일부 소스 코드를 자동 생성 하였다. 또한 이클립스(Eclipse)라는 통합 개발 툴과 Rhapsody(8)와의 연동을 통하여 코딩한 후, 최종적으로 Rhapsody에서 모델을 생성하였다. 이렇게 생성된 모델에 대하여 버전 관리를 수행 하였다.

개발된 MFD OFF는 JUNIT 4.x로 유닛 테스트를 수행하고, 룰 기반(Rule Base)의 정적 코드 분석 툴(Static Code Analysis Tool)인 Veriflux(AICAS)와 Resort for Java(Soft 4 Soft)을 이용하였다. 코딩 개발 표준에 대한 만족도를 분석 툴에 의한 객관적인 결과를 도출함으로써, 결과적으로 소스 코드의 완성도를 높였다.

T-50용 국산화 MC에 탑재되는 MFD OFF는 FCS(Flight Control System), FCR(Fire Control Radar), WPN(Weapon), SMC(Store Management Control), RWR(Radar Warning Receiver) 관련

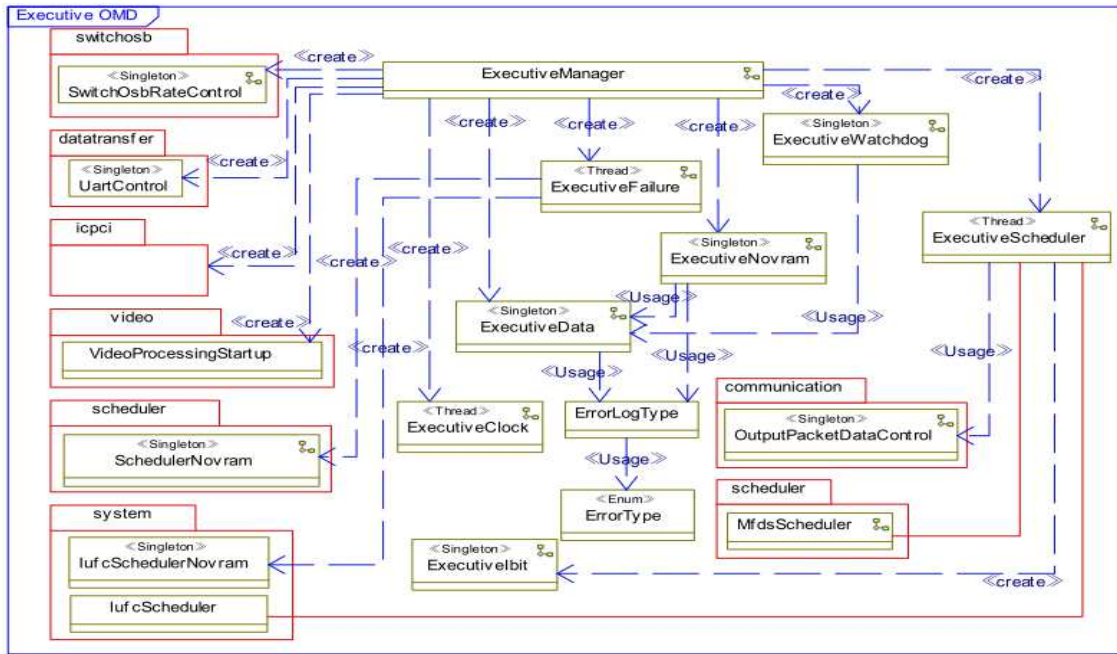


Fig. 7. Design of Executive Control

Table 2. Detailed Module of MFD OFF

모듈	내용
실행 관리	MFD OFF 초기화 수행
시스템 제어	프로세스 스케줄링 관리
데이터 전송	프로세스간 데이터 관리
입력 해석	OSB 입력 처리
비디오 제어	비디오 채널 관리
화면 관리	심볼 시현 및 시현 페이지 제어

정보를 시현 및 제어해야 함은 물론 계기 착륙 및 전술 항법 정보가 포함된 EFI(Electronic Flight Instrument)의 시현이 요구되는데, Table 2는 이러한 요구사항을 구현하기 위해 MFD OFF를 구성하는 전체 모듈을 도식화 해 놓은 것으로서, 본 장에서 세부 내용에 대하여 기술하고자 한다.

2.7.1 실행 관리 (Executive Management)

본 모듈은 OFF 수행 및 스레드(Thread)를 제어 및 관리 하며, 시스템 관리 모듈을 통한 시스템 제어 및 예외처리 시나리오를 수행 하는 모듈로서, Fig. 7과 같이 설계 되었다. 메인 클래스인 Executive Manager를 통하여 프로세스 정보, VIDEO 및 화면 관리 모듈에서 제어하는 데이터에 관한 초기화 루틴을 호출하고, 각종 스레드(Clock, Failure, Schedule, Communication 등)를

생성하며, 각각의 스레드는 세마포어(Semaphore)를 통해서 구동하도록 설계하였다(9). 또한 JNI/JBI 인터페이스를 본 모듈 내에 정의하고, 접근 인터페이스를 제공함으로써 외부 모듈이 해당 인터페이스를 통하여 미들웨어를 제어 할 수 있도록 하였다. 본 모듈의 대표적인 클래스인 Failure Manager 는 Bit Log Manager에 의해 수집된 시스템 오류 및 OFF 구동 오류에 대한 정보를 진단하여 시스템 재시작에 대한 제어를 담당한다.

2.7.2 스케줄 제어 (System Control)

본 모듈은 OFF가 각 Hz별(25Hz, 12Hz, 6Hz, 3Hz, 1Hz, 1Hz Offset)로 수행 하는 명령을 스케줄링 하는 모듈로서, Fig. 8과 같이 설계 되었다. 또한, 본 모듈은 실시간 임베디드 시스템의 가장 중요한 요건중의 하나인 시간 결정적(Time Deterministic) 스케줄링을 보장하기 위 RTSJ에 의해 정의된 리얼타임-쓰레드(Realtime-Thread)를 사용하여 스케줄 제어를 수행한다. 최상위 우선순위를 가진 리얼타임-쓰레드는 25Hz 주기로 하위 리얼타임-쓰레드의 스케줄을 제어하게 된다.

2.7.3 데이터 전송 (Data Transfer)

본 모듈은 FC, HUD, 그리고 IUFC와의 패킷 통신 및 FC를 통한 LRU와의 1553B MUX 통신에 대한 데이터를 처리하는 부분으로 각 Hz별

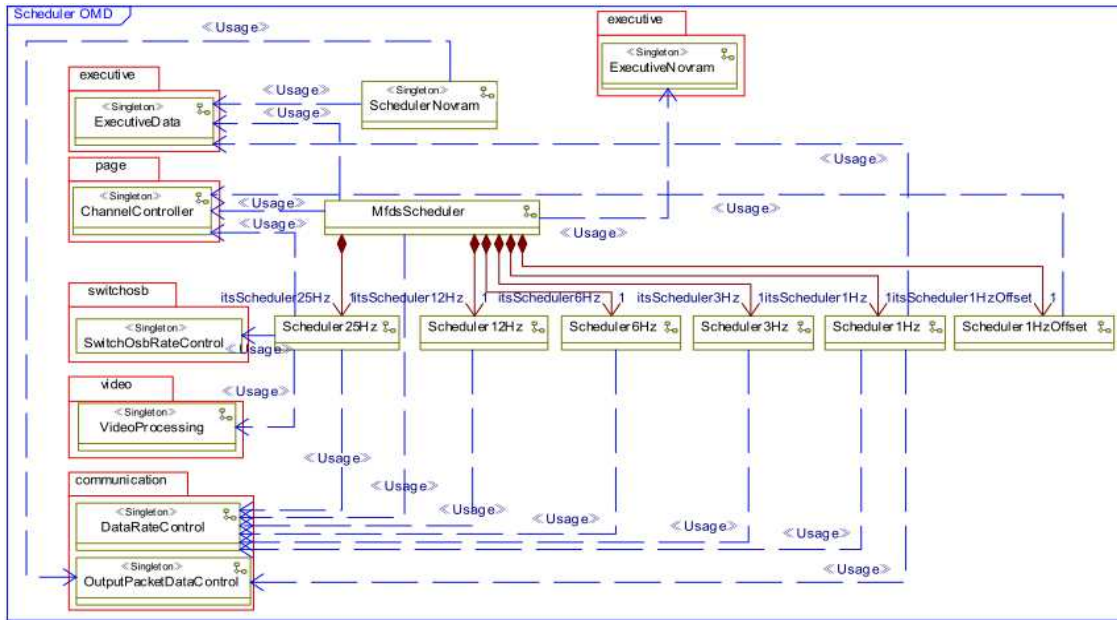


Fig. 8. Design of Schedule Control

데이터 관리와 수집된 데이터에 대한 무결성 점검을 수행한다.

2.7.4 입력 해석 (Switch Interpreter)

본 모듈은 화면 제어를 위한 입력 장치인 OSB(Option Switch Button) 신호를 제어하기 위한 부분으로, MFD의 형태에 따라 두 가지 모듈로 작동하게 되는데, 먼저 일반적인 MFD 타입에서는 UART 인터페이스를 통한 직접 신호 처리를 수행한다. SMFD (Smart Multi Functional Display) 타입에서는 1553B MUX 통신을 수행하는 관계로 FC와의 통신 제어 모듈을 통한 신호 처리를 수행한다.

2.7.5 비디오 제어 (Video Control)

본 모듈은 미들웨어의 일부분인 VPM API를 이용하여 외부 비디오를 제어한다. 입력된 비디오를 MFD OFF가 관리하고 있는 4개의 채널에 적절한 심볼 및 심볼과 혼합된 비디오가 시현 될 수 있도록 제어한다.

2.7.6 화면 관리 (Page Management)

본 모듈은 통신모듈을 통하여 수집된 데이터를 바탕으로 각종 심볼을 조합하여, 특정 페이지로 구성하고, 이를 화면에 시현 시키는 모듈로써, MFD를 통하여 시현되는 최종 화면을 제어한다.

또한, Fig. 9와 같이 본 모듈은 페이지(Page), 심볼(Symbol), 그리고 매트릭스(Matrix) 단위로

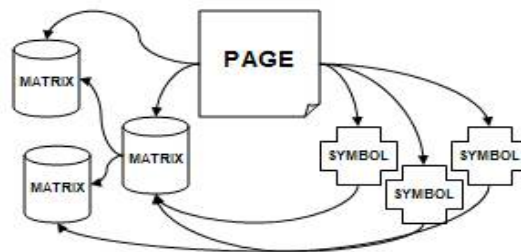


Fig. 9. Page Control Structure

구성된다. 먼저 페이지 부분은 입력 제어로부터 입력되는 신호를 처리하고, 심볼 부분은 OpenGL API(10)를 이용한 MFD로 시현할 화면을 구성한다. 매트릭스 부분은 통신 제어 모듈로부터 입력 받은 최신 데이터를 업데이트 하여 심볼 및 페이지에 제공하는 역할을 수행한다.

2.8 확장 모듈 제어 (Extended Module Control)

임무 컴퓨터(Mission Computer)는 다양한 LRU를 통하여 확장된 기능을 제공한다. 그중에서도 1) 내장 데이터 훈련 유닛(Embedded Data Training Unit, EDTU)과 2) 전자지도(Digital Moving Map, DMM)는 OFF가 제어 및 관리 하는 주요 유닛에 대해 다음과 같이 기술하고자 한다.

2.8.1 내장 데이터 훈련 유닛(Embedded Data Training Unit, EDTU)

EDTU는 조종사에게 가상훈련 상황의 기반이 되는 각종 가상 비행 정보를 제공함으로써 평상시 조종사로 하여금 모의 훈련의 성숙도를 높일 수 있도록 하는 유닛이다. MC와 EDTU는 이더넷(Ethernet) 기반으로 통신한다. 세부적으로는 FC OFP가 EDTU와 이더넷 통신을 수행하여 얻은 각종 가상 비행 데이터를 각 OFP로 전송하는 구조로 구성되어 있다. MFD OFP는 레이더(Radar) 및 기타 가상 비행 정보를 바탕으로 표적 모의 시뮬레이션 기능을 수행한다.

2.8.2 전자지도 (Digital Moving Map DMM) 제어

DMM은 항공기 운용 지역이 미리 저장된 2D/3D 전자지도를 이용하여 지형/지물을 현실감 있게 조종사에게 제공한다. 이는 야간, 악천후 상황 또는 저고도 비행 시 지형 및 장애물에 대한 탐색, 추적, 분류, 인식 등의 능력을 향상시키는 유닛이다. MFD OFP는 비디오 제어 모듈을 통하여 DMM으로부터 수신된 비디오 신호를 MFD에 시현하도록 제어한다.

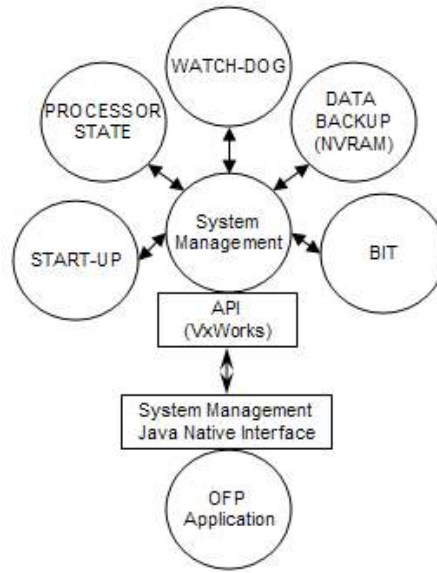


Fig. 10. System Management Structure

Table 3. Detailed design of System Management

III. 미들웨어 (Middleware)

3.1 미들웨어 (Middleware) 설계

미들웨어의 개발은 MC 개발 과정에 있어 기본이 되는 과정으로, 이의 주요 목적은 프로세서 간의 통신, 시스템 제어 및 진단을 위함이다. 이는 C 언어로 구현되어 라이브러리 형태로 제공되며, 자바 응용프로그램 레벨(OFP)에서는 JNI를 통하여 해당 API에 접근 할 수 있다. 미들웨어는 크게 시스템 관리 부분과 비디오 및 통신 제어 부분으로 나눌 수 있다.

3.1.1 시스템 관리 (System Management)

다중 프로세스를 제어하기위한 시스템 관리 모듈은 Fig. 10과 같이 5개 영역의 관리 모듈로 구성되어 있다. 5개의 관리 모듈은 다음과 같이 1) 시스템 초기 관리(START UP), 2) 프로세스 상태 관리(PROCESSOR STATE), 3) 시스템 결함 관리(FAILURE), 4) 데이터 백업 관리(DATA BACKUP), 5) 자체 점검 관리(Built In Test BIT)로 나눌 수 있다. Table 3은 각 항목에 대한 세부적으로 구현된 기능을 나열한 것이다.

3.1.2 비디오 및 통신 제어

심볼 및 비디오 제어를 담당하는 VPM, HPM 모듈과 아날로그 및 디스크리트 통신을 담당하는

API	Function Description
START UP	1. REQUEST RESTART 2. QUERY STARTUP TYPE 3. SYSTEM RESET 4. OFP RESET 5. QUERY TIME SINCE STARTUP
PROCESSOR STATE	1. SET SYSTEM STATE 2. QUERY SYSTEM STATE 3. QUERY PROCESSING
FAILURE	1. WATCHDOG 1-1. ENABLE WATCHDOG 1-2. RELOAD WATCHDOG 1-3. DISABLE WATCHDOG 1-4. ISR WATCHDOG 1-5. SET WATCHDOG DURATION 2. HARDWARE EXCEPTION 2-1. NOTIFY HARDWARE EXCEPTION
DATA BACKUP	1. WRITE TO MEMORY 2. READ FROM MEMORY
BIT	1. EXECUTE BIT 1-1. EXECUTE IBIT 1-2. EXECUTE SBIT 1-3. EXECUTE CBIT 2. QUERY BIT RESULT 2-1. IBIT RESULT 2-2. SBIT RESULT 2-3. CBIT RESULT

ADID(Analog Discrete Interface Driver) 모듈, 1553B MUX 통신을 담당하는 MUX 모듈, TCP/IP통신을 담당하는 이더넷 모듈, 그리고 시리얼 통신을 담당하는 UART 모듈은 각기 단일화된 라이브러리 형태로 미들웨어에 포함된다.

IV. OFP 시험환경 및 결과

4.1 OFP 시험환경

본 논문에서는 비행운용프로그램 검증 및 항공전자 장비 고장탐구에 사용되는 항공전자 시스템 통합시험 장비 (System Integration laboratory, SIL)(11)를 사용한 소프트웨어 통합 시험 및 시스템 통합 시험을 통하여 임무 컴퓨터의 안정성 및 OFP의 신뢰성 테스트를 수행하였으며, 최종적으로 시스템 통합 시험에 사용된 테스트 장비의 형상은 Fig. 11과 같다.

4.2 OFP 시험 결과

GPM에 탑재된 MFD OFP와 IUFC OFP는 총 25HZ의 수행주기를 가지게 되는데, 이 중 MFD OFP에 할당된 수행 시간인 20ms 안에 정상적인 동작을 수행하는지를 지속적으로 모니터링 합과 동시에 조종사 필요 정보를 정상적으로 시현하는지를 외부 출력장치(MFD)를 통하여 MFD OFP에 설계된 각종 페이지(FCR, HSD, EFL, WEAPON, WOV, SMC, ENGINE, FCS, TEST, BIT 등)에 대해서 4차의 MFD OFP에 대한 CSCI(Computer Software Configuration Items) 테스트를 통하여 OFP 시험을 마무리 하였고, 테스트 결과는 Fig. 12와 같다.

V. 결 론

본 논문은 T-50(훈련기, 한국항공우주산업)에



Fig. 11. System Integration laboratory(SIL)

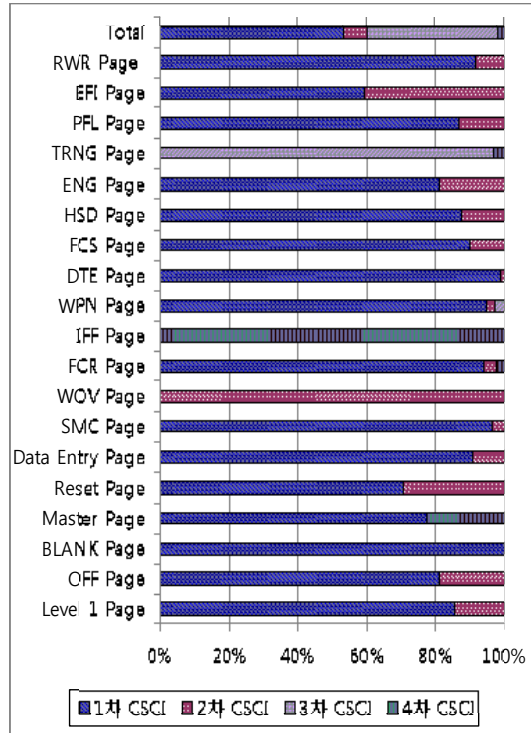


Fig. 12. Test Report of MFD OFP

탑재된 국산화 MC의 비행운용프로그램과 미들웨어 개발에 대한 내용을 기술하였다. 실시간 임베디드 시스템용 자바 가상 머신인 Jamaica VM을 이용한 객체 지향적 OFP 개발을 통하여 자바를 이용한 OFP 개발 가능성을 확인하였다.

자바(Java)는 플랫폼 독립적 과 신뢰성이라는 두 가지 특성을 가지고 있다. 먼저 자바 가상 머신은 다양한 RTOS에 대한 인터페이스를 지원함으로써 플랫폼 독립적인 OFP 개발환경을 제공하며, 또한 객체지향 소프트웨어 개발에 있어 신뢰성의 근간이 되는 메모리 최적화(Garbage Collection)와 예외 처리(Exception Handler)를 지원함으로써 수준 높은 OFP의 안정성을 보장한다. 본 논문은 이러한 자바(Java) 특성을 기반으로 OFP를 설계 및 구현하였고, 최선의 시간 내에 개발을 완료하였다.

또한, 다양한 페이지에 시현되는 각종 심볼을 OpenGL을 이용함으로써 기존 T-50 MFD OFP 대비하여 시현 정보에 대한 가독성을 높일 수 있었으며, 실제 임무컴퓨터에 OFP 탑재 전 PC상에서 가상 시뮬레이션을 통한 선행 개발을 통하여 소프트웨어 개발 기간을 단축 할 수 있었다.

현재 국산화 임무 컴퓨터에 탑재된 OFP는 T-50

항공기에 탑재되어 성공적으로 초도 비행을 마쳤으며, 이를 바탕으로 향후 보다 향상된 신뢰성 및 안전필수(Safety Critical)를 만족하는 시스템 및 OFP 개발을 위하여 지속적인 연구 개발을 하고 있다.

후 기

본 논문은 지식경제부 “산업융합기술산업원정기술개발사업(IT융합) 항공기 임베디드 시스템 개발과제”의 일환으로 개발 되었습니다. 과제를 지원해주신 지식경제부 분들과 한국항공우주산업 임직원 여러분께 감사를 포함합니다.

참고문헌

- 1) 한국항공우주산업, 지경부 Air-BEST 5차년도 수행계획서, 한국항공우주산업(주), 2011
- 2) 한국항공우주산업, 임무컴퓨터 요구규격서, 한국항공우주산업(주), 2011

3) Fridtjof Siebert, "Realtime Garbage Collection in the JamaicaVM 3.0", 2007, JTRES'07 September 26-28, 2007 Vienna, AUSTRIA

4) Fridtjof Siebert, JamaicaVM 6.0 User Manual, 2010, p.36

5) 박종국 외, 내장형 시스템을 위한 자바 AOTC의 설계와 구현, 한국정보과학회, 2005, p.772

6) 이희우 외, 실전 시스템 엔지니어링, 청문각, 2006, p.56

7) Handbook for Object-Oriented Technology in Aviation(OOTiA), Volume 3: Best Practices, October 26, 2004

8) Telelogic, Rhapsody Java Tutorial, 2008

9) Bruce Eckel, Thinking in Java, Fourth Edition, PRENTIC HALL, 2006.

<http://www.prenhallprofessionoal.com>

10) Aaftab Munshi, OpenGL ES 2.0 Programming Guide, 2009

11) 김진혁 외, 항공전자 시스템 통합시험장비 개발, 한국항공우주산업(주), 2008