

저궤도 관측위성에서의 원격 측정 데이터 관련 MIMIC 설계 및 구현

허윤구*, 김영윤***, 조승원***, 최종연****

Design and Development of MIMIC regarding Telemetry in LEO Satellites

Yun-Goo Huh*, Young-Yun Kim**, Seung-Won Cho***, Jong-Yeoun Choi****

Abstract

The telemetry data received from satellite in real-time are used to monitor LEO satellite during the AIT (Assembly, Integration & Test) phase and the mission operation phase after launch. However, it is impossible to check all the incoming telemetry data from satellite in real time in order to detect abnormality of satellite quickly. Especially, the contact time of LEO satellite is limited because of its orbital characteristics. So the anomaly state of the LEO satellite should be detected and resolved during the contact time. Therefore, all incoming spacecraft telemetry data must be selected and manipulated in MIMIC. It is used in order to display summarized information about spacecraft in a visualized way that is quickly and easily understood. That is, it provides essential function to monitor a satellite both in orbit and during testing.

In this paper, the design and development of MIMIC currently used in KOMPSAT, a LEO Earth observation satellite is described in detail. In future work, we plan to enhance MIMIC in order to improve user-friendliness and efficiency.

초 록

지상에서 위성체 총조립 및 시험(AIT; Assembly, Integration & Test) 단계와 발사 후 위성 운용 단계에서 저궤도 관측위성에 대한 다양한 상태 정보들은 위성으로부터 수신한 원격 측정 (Telemetry) 데이터를 통해서 알 수 있다. 하지만 매초 단위로 위성으로부터 수신되는 수많은 원격 측정 데이터들을 모두 일일이 실시간으로 모니터링 하여 위성의 이상 유무를 판별하기는 거의 불가능하다. 특히 저궤도 관측위성은 발사 후 주기적으로 지상국과 교신하는 짧은 시간 내에 위성으로부터 수신한 원격 측정 데이터로부터 위성의 상태를 빠르게 판단해야 하고, 경우에 따라서는 위성의 긴급한 상황을 극복할 수 있는 조치를 위성과의 짧은 교신 시간 내에 완료해야 한다. 따라서 위성으로부터 수신한 원격 측정 데이터들을 선별하고 필요에 따라서 가공하여, 신속하고 요약된 정보를 시각화된 그림이나 도표 등으로 모니터 화면상에 시연해주는 MIMIC은 위성 시험 단계에서 뿐만이 아니라 위성 운용 단계에서도 필수적이다.

본 문서에서는 저궤도 관측위성인 다목적 실용위성에서 현재 개발 완료되어 사용되고 MIMIC에 대한 설계 및 구현 사항에 대해서 자세히 기술한다. 또한 MIMIC의 개발환경과 향후 추가적으로 보완 개발될 MIMIC에 대한 개선 사항들에 대해서도 설명한다.

키워드 : MIMIC, Telemetry(원격 측정), AIT(위성체 총조립 및 시험), MCE(관제 시스템)

접수일(2012년 5월 18일), 수정일(1차 : 2012년 6월 14일, 게재 확정일 : 2012년 7월 1일)

* 전자통합시험팀 / perfect@kari.re.kr ** 전자통합시험팀 / orpheous@kari.re.kr

*** 전자통합시험팀 / y2kim@kari.re.kr **** 전자통합시험팀 / jychoi@kari.re.kr

1. Introduction

저궤도 관측위성에 대한 다양한 상태 정보들은 위성으로부터 수신한 원격 측정 (Telemetry) 데이터를 통해서 알 수 있다. 하지만 매초 마다 위성으로부터 수신되는 수많은 원격 측정 데이터들을 모두 일일이 실시간으로 모니터링 하여 위성의 이상 유무를 판별하기는 거의 불가능하다. 특히 저궤도 관측위성은 발사 후 주기적으로 지상국과 교신하는 짧은 시간 내에 위성으로부터 수신한 원격 측정 데이터로부터 위성의 상태를 빠르게 판단해야 하고, 경우에 따라서는 위성의 긴급한 상황을 극복할 수 있는 조치를 위성과의 짧은 교신 시간 내에 완료해야 한다. 따라서 위성으로부터 수신한 원격 측정 데이터들을 선별하고 필요에 따라서 가공하여, 신속하고 요약된 정보를 시각화된 그림이나 도표 등으로 모니터 화면상에 시연해주는 응용 프로그램이 필수적이다.

현재 이러한 역할을 하는 MIMIC이란 응용 프로그램이 개발 완료되어 저궤도 관측위성인 다목적 실용위성의 지상에서 위성체 총조립 및 시험(AIT; Assembly, Integration & Test) 단계와 발사 후 위성 운용 단계에서 실제로 사용되고 있다.

본 논문에서는 이러한 위성 상태 모니터링을 위해서 필수적인 MIMIC을 실제 구현하기 위해 필요한 개발 환경에 대한 설명에서부터, 현재 개발 완료되어 사용되고 있는 MIMIC에 대한 설계 및 구현 사항에 대해서 자세히 기술한다. 그리고 향후 추가적으로 보완 개발될 MIMIC에 대한 개선 사항들에 대해서도 설명한다.

2. MIMIC 개발 환경

일반적으로 소프트웨어 개발시 발생하는 개발 반복 횟수는 비용의 증가를 담보하므로 이를 최소화하기 위한 방안으로 개발 관리 환경과 개발들을 하나로 묶어서 통합적인 개발 환경을 구축하며, 이러한 개발 환경을 통하여 개발 생산성을 최대화할 수 있다.

MIMIC의 경우에는 Visual Studio .NET의 Team 개발 환경을 통하여 호환성 및 개발 생산성을 최대화할 수 있도록 했다.

- 개발 자료 관리 : SharePoint Service를 이용하는 Team Portal 활용
- 개발 소스 관리 : Team Project의 소스 제어 기능 이용
- 개발 언어 : 높은 코드 가독성을 제공하는 C# 이용

즉 MIMIC 개발을 위하여 Microsoft의 Team Foundation Server (이하 TFS)를 이용하여 서버 환경을 구축하고 Visual Studio .NET 2005 Team Edition의 개발 환경에 TFS 클라이언트 환경인 Team Explorer를 Plug-In하여 개발 관리 환경으로 활용하였다 (그림 1 참조).



그림 1. MIMIC 개발 TFS 운영 환경

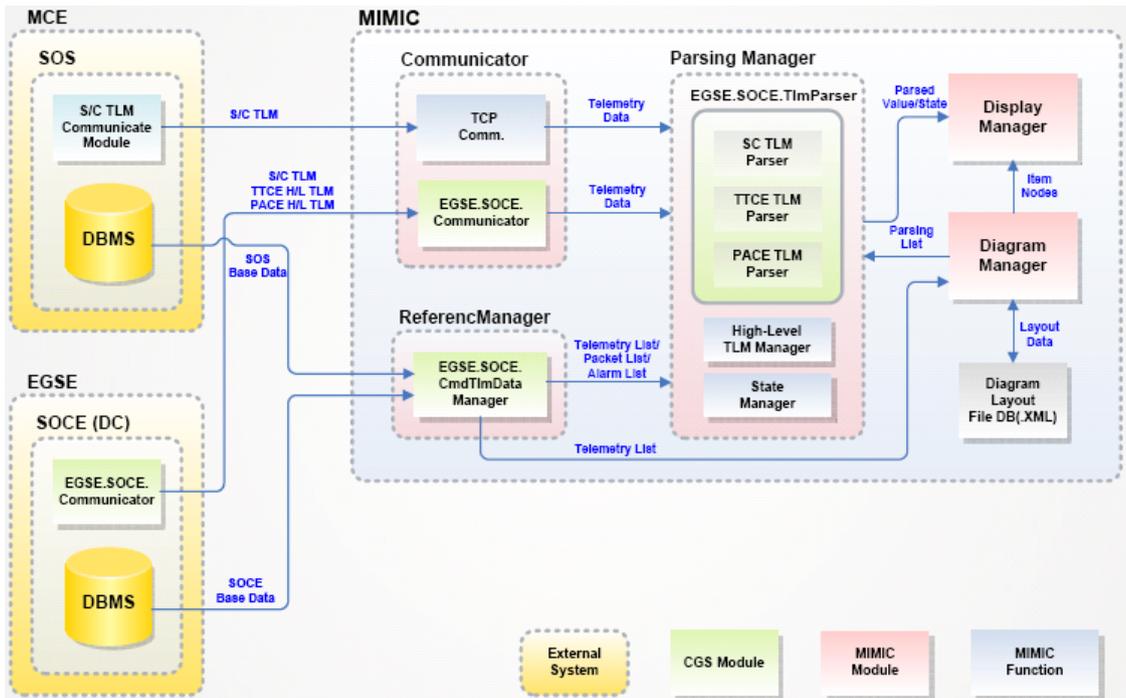


그림 2. MIMIC의 Data flow Diagram

3. MIMIC 설계 및 구현

MIMIC은 간단하게 정의하면 위성으로부터 수신한 원격 측정 데이터를 문자 또는 그래픽으로 표현하여 보여주며, 이를 이용하여 위성의 상태를 모니터링 하는 기능을 제공하는 응용 프로그램이다[1]. 즉 MIMIC은 위성으로부터 수신한 원격 측정 데이터를 파싱하고 MIMIC을 구성하고 있는 각 화면들에 사용되는 값들을 추출한 후 필요에 따라 가공하여 시각화된 그림이나 도표, 문자 등으로 화면상에 표현해 준다.

여기서 MIMIC은 위성으로부터 원격 측정 데이터를 직접 수신하지는 않고, 그림 2에서 보는 바와 같이 위성과 직접 통신을 담당하는 EGSE (Electrical Ground Support Equipment: 전기지상지원장비)나 지상국의 SOS (Satellite Operation Subsystem: 위성운영서브시스템)를 통해서 위성

의 원격 측정 데이터를 받아 오게 되는데, 이때 MIMIC내에 EGSE나 SOS로부터 위성의 원격 측정 데이터를 전달받는 역할을 하는 Communicator라는 모듈이 설계 및 구현되어 있다.

또한 ReferenceManager는 위성의 시험이나 운용에 실제 사용되는 위성 Database를 참조하여 MIMIC의 화면 구성에 필요한 원격 측정 데이터의 Mnemonic 정보를 Database로부터 검색해서 추출해오는 역할을 하는 모듈이다.

ParsingManager는 위성으로부터 수신한 원격 측정 데이터를 ReferenceManager를 통해서 전달 받은 위성 Database 관련 정보에 근거하여 파싱하여 원하는 값들을 추출하는 역할을 하는 클래스이다. 여기서 High level TLM manager는 위성으로부터 수신한 원격 측정 데이터들을 연산하여 MIMIC 화면의 특정 diagram 항목에 연결될 경우에 이러한 정보들을 저장하고 실제로 연산하는 역할을 한다.

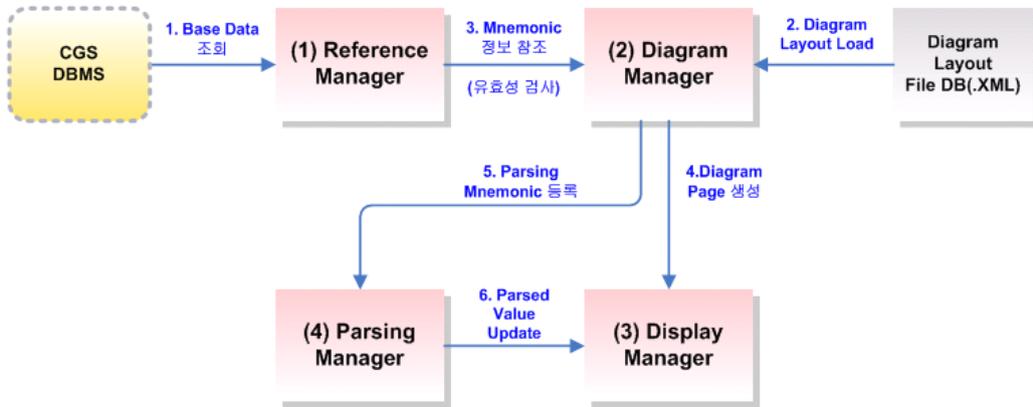


그림 3. MIMIC 초기화 과정

또한 State Manager는 MIMIC 화면에서 참조하는 원격 측정 데이터들의 값이 갱신되는 시점에서 값의 변동 여부를 확인하고, 변동 사항이 있을 시 이를 감지하고, 이 값을 DisplayManager에 전달하는 역할을 하며, 더불어 일정 시간 동안 갱신되지 않은 원격 측정 데이터의 Mnemonic은 Expired 상태로 변경한다.

DiagramManager는 XML 형태의 Diagram Layout 파일을 관리하는 클래스로, LayoutParser 객체를 통해 각종 리소스 및 배치정보를 로드한다. 이때 Layout 파일로부터 얻어진 배치 정보를 기반으로 Diagram 항목의 특성에 따라 버튼, 라벨, 텍스트박스과 같은 여러가지 WPF (WindowsPresentation Foundation) control을 생성하고 배치하여 화면을 구성한다.

마지막으로 DisplayManager는 Diagram Manager로부터 전달받은 MIMIC 화면 구성에 MIMIC 내부에서 처리된 각종 원격 측정 데이터를 연결하고 시각화하여 화면상에 시연하고 사용자로부터 입력을 받아 여러 가지 동작을 수행하는 하는 클래스이다.

부가적으로 설명을 하자면, MIMIC을 처음 실행시키면 그림 3과 같은 초기화 과정을 거치게 된다. 우선 Reference Manager가 관련 위성 Database 정보를 가지고 있는 DBMS (Database Management System)를 통해서 MIMIC에서 이용되는 위성 Database 정보를 입수하게 되고,

DiagramManager는 화면 배치 정보 및 연관된 Mnemonic 정보들을 담고 있는 diagram layout file을 로드하여 MIMIC의 화면 구성에 필요한 원격 측정 데이터의 Mnemonic들에 대해서 ReferenceManager를 통해서 유효성 검사를 실시한다. 그 이후 Diagram Manager는 파싱되어야 할 Mnemonic들을 ParsingManager에 등록하게 된다. 마지막으로 DisplayManager는 DiagramManager로부터 받은 MIMIC 화면 구성에 필요한 정보들과 ParsingManager가 보내주는 원격 측정 데이터를 파싱한 Mnemonic들의 값들을 조합하여 MIMIC 화면상에 시각적으로 시연하게 된다. 앞서 언급했듯이 MIMIC은 지상에서 수행되는 위성에 대한 시험에서도 사용되지만, 위성 발사후 지상국에서 위성 운영시에도 이용된다. 하지만 위성 운영시에는 지상 시험에서만 사용되는 EGSE들을 모니터링 하기 위한 MIMIC 화면은 생략되며, 위성의 위치를 세계지도상에 표시 해주는 부분도 위성의 위치만 추적해서 화면상에 표시해주는 지상국에서 이용중인 전용 응용프로그램이 있기에 지상국에서 사용시 MIMIC 화면에서는 빠지게 된다. 즉 MIMIC의 초기화 과정중 현재 MIMIC의 용도가 위성 시험용인지 아님 지상국에서 사용될지를 판단하여 각각 다른 MIMIC 화면으로 구성되는데, 이것에 관한 판단은 위성 시험에서만 사용되는 MIMIC과 연계된 EGSE 장비들에 할당된 IP와 호스트네임을 이

용하여 MIMIC의 사용 용도가 위성 시험용인 지상국에서 사용할지 구분한다.

MIMIC에서는 앞서 설명한대로 경우에 따라서 위성으로 부터 수신한 원격 측정 데이터들을 조합하여 연산한 새로운 형태의 값이 필요하다. 실제 예를 들면, 위성의 Fuel Mass 값을 아래 그림 4와 같이 MIMIC 화면에 표현하기 위해서는 Propulsion Tank 온도 (TANK_A)값과 Propulsion Tank 압력 (TANK_B) 값을 이용한 다음 수식을 연산한 결과 값으로 표현될수 있다.

$$\text{Fuel Mass} = 1004.0 * (0.09073353 - 350.0 * 6894.757 * (0.09073353 - 72.5 / 1004.0) / (8314.3 / 28.0 * 311.15)) * (\text{TANK_A} + 273.15) * 8314.3 / 28.0 / (\text{TANK_B} * 6894.757);$$



그림 5. MIMIC 화면 구성

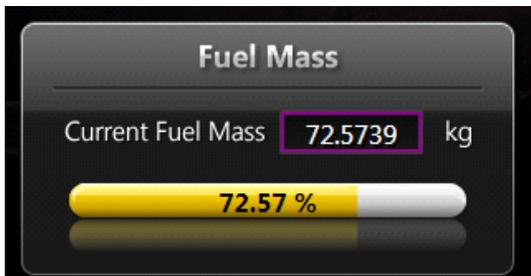


그림 4. MIMIC에서 Fuel Mass 표현

3.1 MIMIC 화면 구성

3.1에서 언급한 MIMIC의 초기화가 완료되고 나면 그림 5와 같은 MIMIC 화면 (지상국에서 사용하는 화면)이 나타난다. 그리고 그림에 표시된 각 번호에 대한 설명은 아래와 같다.

- ① 대상 위성
- ② 연결 및 원격 측정 데이터 수신 상태 (Disconnected/Ready/Active)
- ③ MIMIC 설치 시스템의 Local System 시간
- ④ 위성의 Onboard Time
- ⑤ 현재 화면 페이지 이름
- ⑥ Subsystem Diagram 메뉴
- ⑦ Option

3.2 XML 기반의 유연한 구성 파일

MIMIC에서는 다음과 같은 XML (eXtensible Markup Language) 기반의 구성 파일을 이용하여, MIMIC 화면을 구성하도록 설계 및 구현하였다. 즉 MIMIC이 사용하는 XML 파일에는 MIMIC에서 필요로 하는 위성으로부터 수신 받은 원격 측정 데이터들 각각에 대해서 MIMIC 화면상에 표현되어야 하는 위치와 표현될 때 폰트 크기, 관련된 그림이나 도표의 크기 등을 사용자가 자유롭게 설정 가능하며, 또한 수식을 이용한 원격 측정 데이터의 가공이 필요로 할 경우에는 그것에 대응되는 수식에 대한 정보를 포함하고 있다 (표 1 참조).

MIMIC의 화면 구성에 대한 다양한 설정을 변경하고자 할 경우에는 MIMIC의 소스 코드를 수정할 필요 없이 XML로 작성된 구성 파일을 수정하기만 하면 된다. 즉 MIMIC 관련된 다양한 설정 가능한 사항들은 XML 기반의 구성 파일로 작성되어 있다. 따라서 이러한 설정들을 변경할 때 하드코딩(Hard coding) 되어 있지 않기 때문에 MIMIC의 관련 소스 코드를 매번 수정하여 다시 컴파일(compile) 할 필요가 없다.

표 1. XML 기반 MIMIC 구성 파일

```

<label itemID="Main_PowSa3" location="683
525" size="60 20" fontSize="10" format="0.00">
  <dExpression>
    <![CDATA[
      return ((@TESAR1INI1F * @TESAR1INV1F
+ @TESAR2INI1F * @TESAR2INV1F) +
              (@TESAR1INI2F *
@TESAR1INV2F + @TESAR2INI2F *
@TESAR2INV2F)) / 2;
    ]]>
  </dExpression>
</label>
<image
  itemID="Main_MassMemBackgroundP"
valueKey="TCMMPUSE" location="848 695"
size="180 30">
  <numericStatus
    resource="MainMemoryBackground">
    <state value="0" index="0" />
    <state value="1" index="1" />
  </numericStatus>
</image>

```

참고로 XML은 W3C(World Wide Web Consortium)에서 제안한 규격으로 최신 규격은 1.1 이다. 이 XML은 누구나 간단하게 작성 가능한 단순 명료한 구조로 가독성이 뛰어나고, 운영 체제나 프로그래밍 언어에 독립적으로 호환성이 높다. 또한 이름 그대로 사용자가 새로운 태그를 임의로 정의해서 사용할 수 있어 확장성이 뛰어나다 [2].

3.3 MIMIC 부가 기능

3.3.1 돋보기 기능

MIMIC을 display하기 위해서 일반 PC 모니터 화면이 아닌 대형 스크린을 사용할 경우 MIMIC 화면의 특정 부분을 확대하여 멀리서도 자세히

볼수 있는 돋보기 기능이 그림 6와 같이 구현되어 있다.

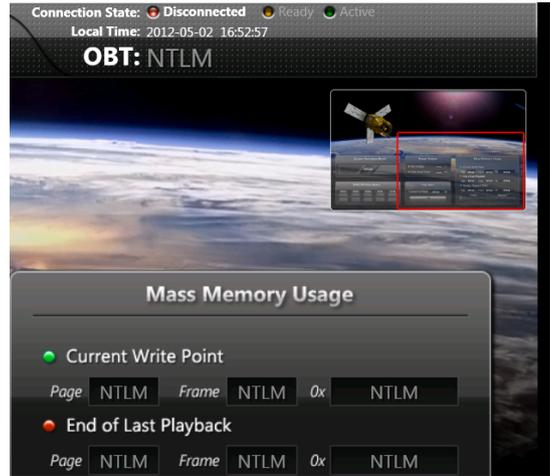


그림 6. MIMIC의 돋보기 기능

또한 사용자가 관심있게 보고자 하는 MIMIC 화면의 특정 부분이 있을 경우, MIMIC의 돋보기 기능을 이용하여 그 부분만 크게 확대 하여 지속적으로 자세히 모니터링 할 수 있다.

3.3.2 디버깅 모드

MIMIC은 실제 위성으로부터 수신한 원격 측정 데이터를 선별하고 필요에 따라서 가공한 정보를 모니터 화면에 보여주게 되는데, MIMIC 설계 및 구현 단계에서는 실제 위성으로부터 원격 측정 데이터를 직접 수신할 수 없기 때문에 이 디버깅 모드를 이용하여, 그림 7 처럼 특정 MIMIC 화면에 예상되는 원격 측정 데이터 값들을 입력해보고 이에 대해 MIMIC 화면상에 올바르게 표현되는지 미리 확인해 볼 수 있다.

또한 MIMIC 구현이 완료되어 위성 시험이나 위성 운용에서 실제 MIMIC을 이용할 때에도 이 디버깅 모드를 이용하여 사용자가 직접 예상되는 원격 측정 데이터 값들을 입력하여 MIMIC 화면을 미리 만들어 예측해 볼수 도 있다.

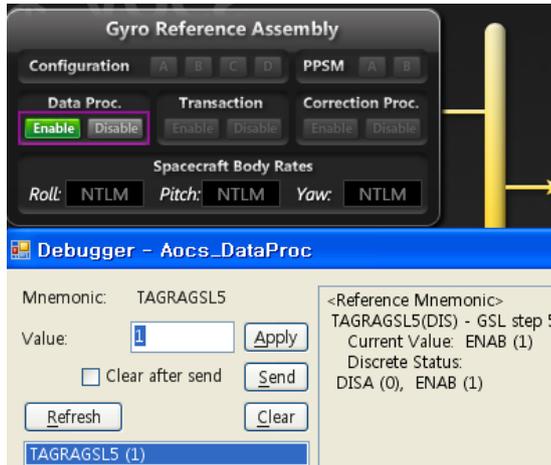


그림 7. MIMIC의 디버깅 모드

3.3.3 전시 모드

모니터나 스크린에 display되는 MIMIC 화면은 공간적 제약 때문에 메인화면과 여러 서브시스템화면 중에서 사용자가 선택한 화면만 보여지게 된다. 하지만 필요에 따라서 일정 시간 간격을 두고 자동으로 메인화면과 여러 서브시스템 화면들이 번갈아 가며 보여 주는 전시모드를 구현하였다(그림 8 참조).

위성 시험이나 운용시 MIMIC의 특정 화면만을 계속해서 집중적으로 살펴볼 필요가 없을 경우, 즉 대부분의 일반적인 상황에서는 MIMIC의 전시 모드를 이용하여 위성의 상태를 모니터링하게 된다.

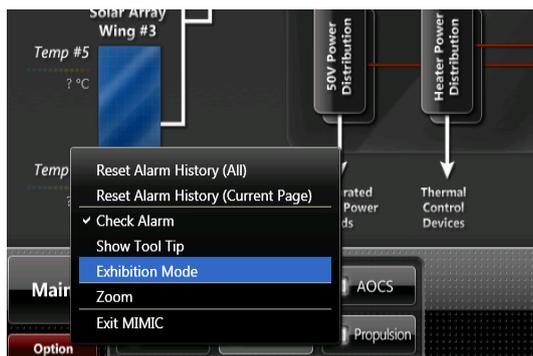


그림 8. MIMIC의 전시 모드

4. 결 론

본 논문에서는 저궤도 관측위성인 다목적 실용위성에서 현재 개발 완료되어 사용되고 MIMIC에 대한 설계 및 구현 사항에 대해서 자세히 살펴보았다. 현재 MIMIC에서는 위성으로부터 수신한 원격 측정 데이터를 문자 또는 그래픽으로 시각화하여 화면상에 시연되기에 위성의 상태를 빠르게 모니터링 하는 기능으로 유용하게 사용되고 있다.

하지만 현재 개발 완료된 MIMIC의 화면을 구성하는 그림이나 도표 등은 고정적으로 사용자가 임의로 모양을 변경하거나 새로운 형태의 그림이나 도표 등을 등록하여 화면을 구성하는 것이 불가능하다. 따라서 향후 추가 보완되어 개발될 MIMIC에서는 사용자가 원하는 그림이나 도표등을 직접 작성하고, 그것과 연관된 원격 측정 데이터와의 관계를 기술하여 등록한 후, 사용자가 원하는 화면구성을 설정 및 저장하여 자유롭게 이용할 수 있도록 할 예정이다.

참 고 문 헌

1. 허윤구, 권재욱, 최종연, “공통 지상시스템 지원모듈에 관한 요구도 분석”, 한국항공우주연구원 Technical Memo, KARI-SFT-TM-2008-001, 2008, pp. 5-6
2. “Extensible Markup Language (XML) 1.1 (Second Edition)”, W3C Recommendation 16 August 2006, edited in place 29 September 2006