

메시지 정의 및 관리시스템(MDMS) 적용 사례 연구

류제영
삼성탈레스(주)

e-mail : jeyoung.yu@samsung.com

A Study on Application of Message Definition and Management System

Je-Young Ryu
Samsung Thales Co., Inc.

요 약

효율적인 인터페이스 정의 및 관리를 위하여 개발된 MDMS (Message Definition and Management System) 는 인터페이스 요구사항명세서(IRS: Interface Requirement Specification)과 인터페이스 설계기술서 (IDD: Interface Design Description), 인터페이스 구현 및 통합의 단계로 이루어진 인터페이스 개발 과정을 통합 관리할 수 있도록 해준다. 이러한 통합관리는 인터페이스 개발에 따른 IRS, IDD, 구현 코드에 걸친 일관성 유지를 쉽게 해주며, 소프트웨어 개발 과정에서 가장 문제가 많이 발생하는 단계라고 할 수 있는 소프트웨어 통합에 있어서 인터페이스 불일치로 인한 문제의 대부분을 제거해 준다. MDMS 기반의 인터페이스 관리는 한번의 인터페이스 정의로 IRS, IDD, 송수신 코드까지 생성함으로써 인터페이스 정의 및 개발의 단계를 한단계로 축소하여 각 단계 사이의 불일치 발생이 미연에 방지되며, 일관성 유지 노력이 필요없다. 삼성탈레스에서는 2008 년 이후 10 여개의 함정전투체계 프로젝트의 소프트웨어 개발에 MDMS 를 적용하여 소프트웨어 개발 생산성의 향상을 이루었다. 특히 인터페이스 정의, 설계, 문서 작성, 코드 구현, 형상관리가 통합은 소프트웨어 인터페이스 개발 과정에서의 불필요한 중복 작업을 최소화하고, 그에 따른 문서와 코드간의 불일치를 근본적으로 제거할 수 있었다. 또한, 소프트웨어 개발 관리 측면에서도 인터페이스의 복잡도를 평가하여, 향후 유사 시스템의 소프트웨어 아키텍처 구성에 활용할 수 있다.

1. 서론

대형 소프트웨어 시스템은 그 성격과 기능에 따라 소프트웨어 아키텍처를 결정하고, 그에 따라 모듈들이 나뉘어 진다. 이러한 모듈들은 그 규모에 따라 개발자들이 할당된다. 전체 소프트웨어 시스템은 이렇게 각 개발자에게 할당된 최소단위 모듈들이 정해진 인터페이스에 따른 상호 작용을 함으로써 정해진 목적을 달성한다. 모듈간의 인터페이스는 개발 초기에 정의되고, 개발 과정에서 수없이 변경된다. 소프트웨어 모듈 간의 인터페이스 관리는 수십명 이상의 소프트웨어 개발자들이 참여하는 대형 소프트웨어 시스템의 개발 효율에 큰 영향을 미친다.

효율적인 인터페이스 정의 및 관리를 위하여 개발된 MDMS (Message Definition and Management System) [1]는 인터페이스 요구사항명세서(IRS: Interface Requirement Specification)과 인터페이스 설계기술서 (IDD: Interface Design Description), 인터페이스 구현 및 통합의 단계로 이루어진 인터페이스 개발 과정을 통합 관리할 수 있도록 해준다. 이러한 통합관리는 인터페이스 개발에 따른 IRS, IDD, 구현 코드에 걸친 일관성 유지를 쉽게 해주며, 소프트웨어 개발 과정에서

가장 문제가 많이 발생하는 단계라고 할 수 있는 소프트웨어 통합에 있어서 인터페이스 불일치로 인한 문제의 대부분을 제거해 준다.

이 논문에서는 MDMS 의 개발 전의 소프트웨어 인터페이스 개발 프로세스와 함께, MDMS 를 실제 프로젝트에 적용할 때의 소프트웨어 인터페이스 개발 프로세스를 소개한다. 그리고, MDMS 를 적용한 소프트웨어 인터페이스 개발 프로세스를 적용했을 때의 장점과 함께 MDMS 의 적용 사례를 소개하여 그 장점이 실제 프로젝트의 개발효율을 어떻게 높여주는 지를 보여준다.

2. MDMS 소개

MDMS 는 대형 소프트웨어 시스템의 메시지 기반 인터페이스 설계 및 관리를 위한 소프트웨어 개발 지원도구이다. 각 모듈의 개발자들은 MDMS 를 통하여 해당 모듈이 전송하는 메시지를 정의하고, 다른 모듈로부터 정의된 메시지 중 수신에 필요한 것들을 선택한다. MDMS 는 송수신 메시지를 정의하는 기능과 함께 메시지의 변경 내역을 관리하는 형상관리 기능, 정의된 메시지에서부터 메시지 송수신을 위한 코드를

자동으로 생성하거나 인터페이스 정의 문서의 생성을 지원하는 기능 등을 가진다.

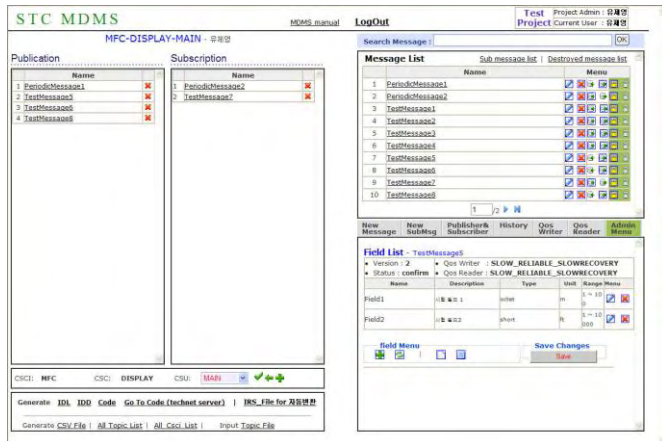


그림 1. MDMS 실행화면

MDMS 는 그림 1과 같이 웹 기반으로 개발되어 다수의 개발자가 별도의 소프트웨어의 설치없이 바로 접속하여 사용할 수 있다. 또한, 모든 소프트웨어 개발자가 항상 최신의 인터페이스 정의를 유지할 수 있으므로, 인터페이스 불일치로 인한 소프트웨어 통합 문제를 미연에 방지한다. 또한, MDMS를 통한 인터페이스 정의 및 변경 작업만으로 최신의 인터페이스 관련 코드를 자동으로 생성하여 사용할 수 있으며, 인터페이스 정의 문서의 최신화를 쉽게 수행할 수 있다.

3. MDMS 적용 소프트웨어 개발 프로세스

MDMS 의 개발 및 적용 전에는 MIL-STD-498[2]에 정의된 개발문서인 IRS 및 IDD 와 IEEE/EIA 12207[3]의 소프트웨어 산출물인 Software Interface Design Description 을 기반으로 한 인터페이스 요구사항 명세서 및 인터페이스 설계기술서를 이용하여 문서 기반의 모듈간 인터페이스의 개발 및 관리를 수행하였다. 이러한 방식의 인터페이스 개발 및 관리 프로세스는 그림 2와 같다.

각 모듈의 개발자는 인터페이스 요구사항 명세서 (IRS)를 작성하고, 관련 모듈의 개발자들에게 IRS 를 배포한다. 관련 모듈 개발자들이 IRS 를 검토한 후 그 결과를 적용하여 IRS 가 갱신 된다. 수차례의 IRS 의 배포와 갱신을 통하여 IRS 가 완성되면 IRS 를 토대로 인터페이스 설계기술서(IDD)를 작성한다. IDD 역시 IRS 와 같이 관련 모듈 개발자들에게 배포와 검토를 통하여 완성된다. IDD 의 검토와 업데이트 동안, IRS 의 변경이 필요하면 IRS 를 추가적으로 검토하고 업데이트하게 된다. 완성된 IDD 를 사용하여 모듈 개발자는 구현을 시작하게 된다. 모듈을 구현하는 동안에도 IDD 나 IRS 의 변경이 빈번히 발생한다. 특히, 구현 초기에는 IDD 및 IRS 의 변경의 발생률이 매우 높으며, 구현이 중반을 넘어서는 시점에서 인터페이스는 안정화된다. 각 개발자의 모듈 구현이 완료된 후에는 모듈간의 통합이 시작된다.

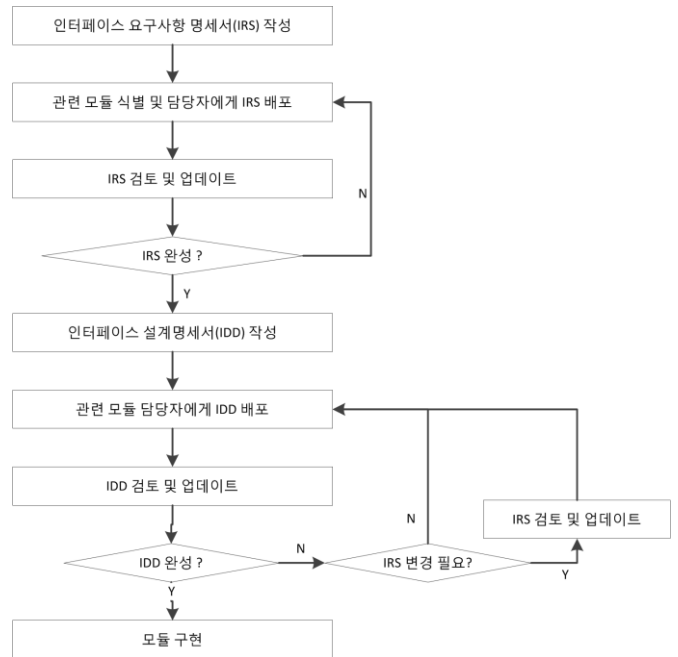


그림 2. 문서기반 인터페이스 개발 프로세스

IRS 및 IDD 를 이용한 문서 기반의 인터페이스 관리는 여러가지 문제점을 가진다. 우선, 문서의 작성과 검토에 많은 시간이 소요된다. 또한, 문서 기반의 관리는 버전관리에 많은 노력이 소요된다. 인터페이스 관리가 IRS 작성, IDD 작성, 구현, 통합의 네단계로 분리되어, 각 단계 사이에 오류가 발생할 가능성이 높다. 구현 초기에 인터페이스의 변경이 많이 발생할 때, 구현과 IRS, IDD 의 일관성을 유지하기가 어렵다.

MDMS 는 이러한 문서기반의 인터페이스 관리를 웹 환경의 시스템 기반 인터페이스 관리로 바꿔준다.

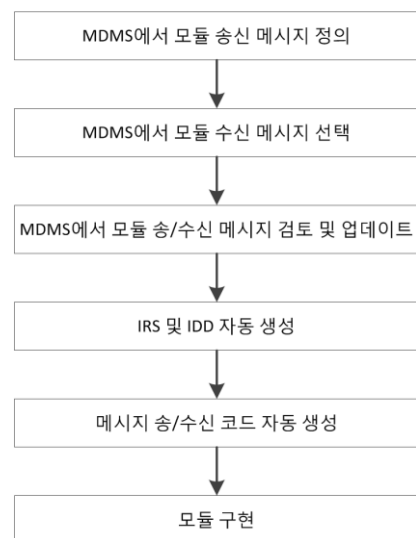


그림 3. MDMS 기반 인터페이스 개발 프로세스

그림 3과 같이 각 모듈의 개발자는 MDMS 에서 모듈이 생성하는 메시지를 정의한다. 그 후 다른 모듈에서 정의된 메시지 중 자신의 모듈이 필요하는 메시

지를 수신 메시지로 선택한다. 송신 메시지의 정의와 수신 메시지의 선택을 수행한 후 각 모듈의 개발자는 송/수신 메시지를 검토하고, 필요에 따라 수정한다. 송/수신 메시지의 수정 내역은 시스템에 기록되어 자동으로 형상관리가 이루어진다. 이렇게 송/수신 메시지가 결정되면, 그 정보를 바탕으로 IRS 와 IDD 를 생성한다. 또한, 메시지 송/수신 코드를 자동으로 생성하여, 모듈 구현에 이용한다.

MDMS 기반의 인터페이스 관리는 문서 기반의 인터페이스 관리와 비교하여 많은 장점을 가진다. 한번의 인터페이스 정의로 IRS, IDD, 송수신 코드까지 생성함으로써 인터페이스 정의 및 개발의 단계가 네단계에서 한단계로 축소된다. 따라서 각 단계 사이의 불일치 발생이 미연에 방지되며, 일관성 유지 노력이 필요없다. 각 개발자는 IRS, IDD 문서를 작성하고, 버전을 관리하고 유지할 필요없이 문서가 필요한 때 최신의 정보를 반영한 IRS 와 IDD 문서를 MDMS 시스템에서 생성하면 된다. 구현을 위한 메시지 송/수신 코드도 MDMS 에서 자동으로 생성하여 구현에 편의성과 인터페이스 구현 코드의 표준화가 가능하다. 이러한, 통합 관리는 구현 초기 인터페이스 변경이 잦을 때 더욱 큰 힘을 발휘한다. 각 개발자는 인터페이스 변경이 필요할 때 MDMS 에서 인터페이스를 변경한다. 인터페이스가 변경되면 변경 내역이 저장되며, 관련 모듈에 변경 여부가 통보된다. 각 개발자는 MDMS 에서 인터페이스 수정 외의 문서 수정이 별도로 필요 없으며, 메시지 송/수신 코드 역시 자동 생성하여 사용함으로써 수정 노력을 최소화한다.

4. MDMS 적용 사례

삼성탈레스에서는 2007 년 MDMS 를 개발한 후 함정전투체계의 소프트웨어 개발에 MDMS 를 적용하고 있다. 대형 함정전투체계 소프트웨어 개발에는 규모에 따라 수십명 이상의 소프트웨어 개발자가 참여하여 각 담당 모듈들을 개발한다.

MDMS 개발 이전에는 3 장에서 설명한 IRS 및 IDD 문서 기반의 인터페이스 관리 방식을 사용하여 모듈간 인터페이스를 개발하고 관리했다. 각 개발자는 요구사항 정의와 설계 단계에서 문서를 주고 받으며 인터페이스를 설계하고 결정한다. 설계단계에서 IDD 작성이 완료되면, 개발자들은 구현을 시작하고, 각자의 구현 방식에 따라 인터페이스 코드를 작성한다. 구현 초기 단계에서 인터페이스 변경이 필요한 경우 각 개발자는 관련 개발자들과 해당 내용을 검토하여 인터페이스를 업데이트한다. 이 과정에서 IRS 와 IDD 의 업데이트가 함께 이루어지지 않는 경우가 많으며, 그에 따라 인터페이스의 변경 내역과 변경 이유가 불명확해지는 경우가 많았다.

각 개발자의 담당 모듈의 구현이 끝나면 소프트웨어 모듈 통합이 시작된다. 구현 과정에서 발생한 인터페이스 변경에 의해서 통합 과정에서 다수의 인터페이스 불일치가 발생했다. 인터페이스 변경에 대한 형상관리의 부재로 인하여 이러한 인터페이스 불일치는 불일치의 원인과 해결에 있어서 많은 시간을 필요

로 하고, 때에 따라 논쟁의 대상이 되기도 했다. 통합이 완료된 후에는 보통 IRS 와 IDD 의 내용과 통합이 완료된 인터페이스 구현 결과가 일치하지 않게 된다. 그에 따라 최종의 인터페이스 구현 결과에 따라 IRS 와 IDD 의 업데이트가 필요했다.

최초로 MDMS 가 적용된 과제인 울산(I)급 전투체계 개발 과제에서는 앞에 기술한 대다수의 문제들이 해결되었다. 인터페이스 개발 절차가 3 장의 그림 3과 같이 이루어짐에 따라, 인터페이스 정의, 설계, 문서 작성, 코드 구현, 형상관리가 통합되었다. 모든 인터페이스의 정의 및 갱신은 MDMS 에서 이루어지며, MDMS 에서 관리된다. 문서와 구현된 코드의 불일치가 발생할 여지가 근본적으로 사라졌다. 통합과정에서의 인터페이스 불일치는 근본적으로 제거되었으며, 통합 과정에서 불일치가 발생하더라도 MDMS 에 정의된 최신의 MDMS 인터페이스를 적용하는 것으로, 간단하게 해결되었다.

소프트웨어 아키텍처 차원에서도 전체 소프트웨어 모듈간의 인터페이스를 MDMS 라는 하나의 인터페이스 관리 시스템에서 관리함으로써 인터페이스의 복잡도를 모니터링하고, 관리할 수 있게 되었다. 이를 통하여, 모듈간의 인터페이스 복잡도를 평가하여 전체 소프트웨어 아키텍처 구성에도 활용이 가능하다.

5. 결론 및 향후 연구

삼성탈레스에서는 2008 년 이후 10 여개의 함정전투체계 프로젝트의 소프트웨어 개발에 MDMS 를 적용하여 소프트웨어 개발 생산성의 향상을 이루었다. 특히 인터페이스 정의, 설계, 문서 작성, 코드 구현, 형상관리가 통합은 소프트웨어 인터페이스 개발 과정에서의 불필요한 중복 작업을 최소화하고, 그에 따른 문서와 코드간의 불일치를 근본적으로 제거할 수 있었다. 또한, 소프트웨어 개발 관리 측면에서도 인터페이스의 복잡도를 평가하여, 향후 유사 시스템의 소프트웨어 아키텍처 구성에 활용할 수 있다. 현재 삼성탈레스에서는 MDMS 를 더욱 발전시킨 MOSMAT[4]을 기반으로한 차세대 메시지 관리 시스템을 개발하고 있다.

6. 참고문헌

- [1] 류제영, 박진희, “메시지 기반 인터페이스 공동 개발을 위한 메시지 관리 시스템”, 정보과학회논문지: 컴퓨팅의 실제 및 레터 제 14 권 제 6 호 p.609~p.613, 2008.8
- [2] Military Standard Software Development and Documentation (MIL-STD-498), Department of Defense US, Dec, 1994
- [3] (ISO/IEC 12207) Standard for Information Technology - Software life cycle processes (IEEE/EIA 12207), Mar, 1998
- [4] 류제영, 박진희, “메시지 지향 소프트웨어 모델링 및 분석 도구 개발”, 2010 한국정보과학회 2010 한국 컴퓨터종합학술대회 논문집 제 37 권 제 2 호 (A) 2010.11, page(s): 42-43