

메시지 정의 및 관리시스템(MDMS) 적용의 문제점 및 개선방안 연구

류 재 영
한화탈레스(주)

e-mail : jeyoung.yu@hanwha.com

A Study on Problems and Improvements for Application of MDMS (Message Definition and Management System)

Je-Young Ryu
Hanwha Thales Co., Inc.

요 약

많은 개발자가 참여하는 대형 소프트웨어 시스템 개발의 효율성 증대를 위해서는 컴포넌트 간 인터페이스의 효과적인 관리가 필수적이다. MDMS는 컴포넌트간의 인터페이스 정의 및 변경 이력 관리, 인터페이스 기술서 자동 생성, 소스 코드 자동 생성 등의 기능을 제공하는 도구이다. 한화탈레스는 1998년 MDMS를 처음 개발하여 적용한 이후 다양한 국방 소프트웨어 시스템 개발에 적용하여 소프트웨어 개발 생산성 향상에 효과를 보았다. 하지만, MDMS를 다양한 프로젝트에 적용하는 과정에서 최초 개발시 고려하지 못했던 여러 문제점 및 한계를 발견하였다. 이러한 문제점 및 한계는 다양한 프로젝트의 특성 지원 미흡, 형상관리 및 유지보수 문제, 프로젝트 간 메시지 재사용 미지원, 취약한 보안 등이다. 본 논문에서는 이를 해결하기 위하여 네가지 개선방안을 제시한다. 첫 번째, 다양한 프로젝트에 적용이 가능한 유연한 SW 구조로 개선해야 한다. 두 번째, 통일되고 일관된 형상관리와 함께 전담 개발 및 유지보수 조직이 필요하다. 세 번째, 프로젝트간 메시지의 재사용 지원을 위한 방안으로 프로젝트 별 MDMS 운용이 아닌 통합된 MDMS의 운용이 필요하며, 그를 통하여 다른 프로젝트의 인터페이스 정의를 상호 참조할 수 있는 구조를 적용해야 한다. 마지막으로, 외부 협력업체와의 협업을 위한 보안 대책을 수립하고, 관련 보안 기능을 지원하여 사외의 협력업체의 개발자가 직접 MDMS에 접속하여 개발할 수 있도록 개선해야 한다.

1. 서론

많은 개발자가 참여하는 대형 소프트웨어 시스템 개발의 효율성 증대를 위해서 컴포넌트 간 인터페이스의 효과적인 관리는 필수적이다. MDMS(Message Definition and Management System)는 한화탈레스에서 개발한 소프트웨어 개발 지원도구로써 컴포넌트간 인터페이스 정의 및 변경 이력 관리, 인터페이스 기술서 자동 생성, 소스 코드 자동 생성 등의 기능을 제공한다[1]. 한화탈레스는 1998년 MDMS를 처음 적용한 이후 다양한 국방 소프트웨어 시스템 개발에 적용하여 소프트웨어 개발 생산성 증대에 다음과 같은 효과를 보았다[2].

첫째, 기존의 문서 및 코드 중심의 인터페이스 정의 및 변경 절차가 MDMS 중심의 절차로 단순화되었다. 그에 따라, 개발자 간의 인터페이스 설계와 변경을 위한 커뮤니케이션이 단순화되었으며, 통합과정에서 컴포넌트 간의 인터페이스 불일치로 인한 통합 오류가 현저하게 감소하였다.

둘째, 인터페이스 변경에 따른 형상관리가 원활해졌다. 현실적으로 개발 과정에서 일정에 쫓겨 형상관리가 소홀해지는 경우가 많다. MDMS는 인터페이스 변경 시 자동으로 변경 이력관리를 함으로써 개발자가 별도의 형상관리를 위한 노력을 기울일 필요 없이 형상관리를 가능하게 하였다.

셋째, 인터페이스 요구사항 정의서(IRS) 및 인터페이스 설계 명세서(IDD)의 작성 자동화로 개발자의 문서화 작업을 줄였다. MDMS 적용 전에는 개발 초기 IRS 및 IDD 문서를 작성한 후, 개발 중에는 문서가 일관적으로 업데이트 되지 않는다. 왜냐하면, 개발 중에는 인터페이스가 수시로 변경, 추가, 삭제되므로 실시간으로 문서를 업데이트하는 것은 거의 불가능하기 때문이다. 하지만, MDMS를 적용함으로써 인터페이스 정의, 변경, 추가, 삭제의 행위와 코드 생성, 문서 생성의 모든 활동이 하나의 활동으로 통합되었다. 개발자는 MDMS에서 한번의 인터페이스 정의, 변경 작업만 하면, 항상 최신의 인터페이스 정의가 반영된 문서를 생성 가능하다. 이는 개발자에게 문서화 작업에

대한 부담을 경감시켰을 뿐만 아니라, 문서와 구현간의 불일치도 해소시켰다.

넷째, 자동 코드 생성을 통하여 코드 구현의 효율화 및 인터페이스 관련 코드의 표준화가 가능해졌다. 수십 명의 소프트웨어 개발자가 참여하는 대형 소프트웨어 시스템 개발의 경우, 경험 많은 개발자와 경험이 부족한 개발자들이 함께 개발을 진행하는 경우가 많다. 그 경우, 코드의 품질의 격차가 상당히 벌어지게 된다. MDMS의 코드 자동생성 기능을 통하여 우수한 품질의 표준화된 인터페이스 처리 코드를 모든 개발자가 사용할 수 있도록 함으로써 부분적인 코드 표준화를 지원하였다.

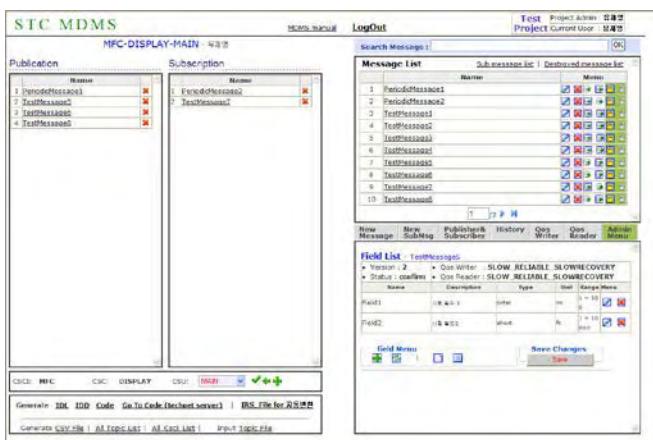


그림 1. MDMS 실행화면

이처럼 한화탈레스는 그동안 다양한 국방 소프트웨어 시스템 개발에 MDMS를 적용함으로써 소프트웨어 개발 생산성에 많은 효과를 보았으나, 그 과정에서 MDMS의 여러 한계와 문제점들을 발견하였다. 여러 문제점들 중 전체 소프트웨어 시스템의 인터페이스 관계를 파악하기 곤란하다는 문제는 컴포넌트 관계의 도식화를 및 구조 분석 기능을 추가한 MOSMAT(Message Oriented Software Modeling and Analysis Tool)[3]의 개발을 통하여 해결 방안을 마련하였다.

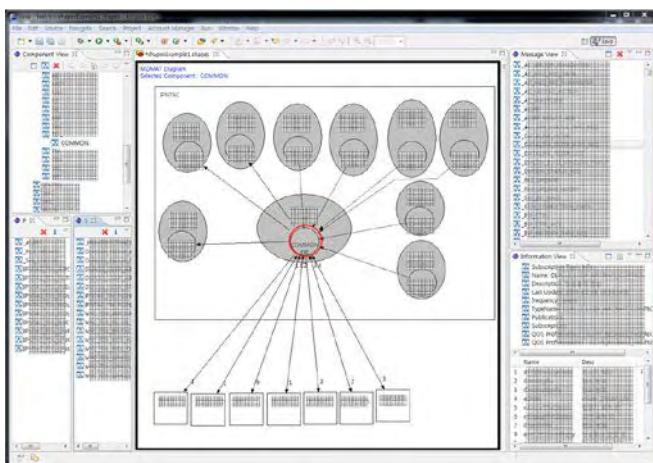


그림 2. MOSMAT 실행화면

이 논문에서는 한화탈레스가 MDMS를 적용하면서 발견한 MDMS의 한계와 문제점을 소개한다. 또한, 그 한계와 문제점을 극복하기 위한 개선 방안을 제시한다.

2. MDMS 적용의 문제점

한화탈레스는 MDMS를 다양한 프로젝트에 적용하는 과정에서 최초 개발시 고려하지 못했던 다음과 같은 문제들을 발견하였다.

첫째, MDMS의 소스코드 자동 생성 기능 및 문서 자동 생성 기능은 MDMS에 직접 구현되어 있어 코드 및 문서 생성 기능을 변경하기 위해서는 MDMS 자체의 코드 수정이 필요하다. MDMS를 다양한 프로젝트에 적용하는데 있어 각 프로젝트의 특성에 따라 자동생성 코드 및 문서가 달라진다. 이러한 MDMS 코드 수정으로 인하여 프로젝트 별로 상이한 MDMS 버전이 나타나게 되었으며, 이는 MDMS의 형상관리에 심각한 문제를 초래하였다.

둘째, MDMS를 지속적으로 개발하고, 유지보수하기 위한 전담조직의 부재로 인하여 일관된 기능 개선 및 유지보수가 이루어지지 않았다. 또한, MDMS 최초 개발자의 부서 이동 및 퇴사로 인하여 전체적인 설계지식이 단절되었다. 현재 MDMS의 적용 방식은 A 프로젝트에서 MDMS를 A 프로젝트에 필요한 기능을 추가하여 A 프로젝트용 MDMS를 만들어 사용하고, B 프로젝트에서는 B 프로젝트에 필요한 기능을 추가하여 B 프로젝트용 MDMS를 만들어 사용하는 방식이다. A 프로젝트에서 MDMS의 공통적인 버그를 발견하거나 개선사항을 추가하더라도 이는 B 프로젝트용 MDMS는 적용되지 않는다. 이로 인하여 B 프로젝트 개발자들이 A 프로젝트용 MDMS에서 수정한 버그나 기능 개선을 중복해서 수행하는 일이 발생한다. 이러한 상황이 반복되면서 현재 MDMS는 지금까지 나왔던 모든 오류가 수정되고 개선사항들이 적용된 최종 버전이 존재하지 않게 되었다. 이는 앞에 첫번째 문제로 제시한 형상관리 문제가 더욱 심각하게 하였다.

셋째, MDMS는 정의된 인터페이스를 프로젝트 간에 재사용하기 위한 기능을 제공하지 않는다. MDMS는 소프트웨어 컴포넌트 간의 메시지 정의 및 관리의 문제를 해결하기 위하여 개발되었다. 새로운 프로젝트에서 컴포넌트 간의 메시지를 새로 정의할 때, 기존의 유사한 프로젝트의 메시지를 상당수 재사용하여 효율성을 높일 수 있다. 하지만, 현재 MDMS는 이러한 기능을 직접 제공하지 않아, 개발자는 기존 프로젝트의 MDMS에 정의된 메시지를 참고하여 새로운 프로젝트의 MDMS에 새로 메시지를 정의하고 입력해야 한다.

넷째, MDMS는 최초에 한화탈레스 내부 개발자들 간의 협업을 위하여 개발되어 사내 네트워크 내에서 사용하는 것을 가정하고 개발되었다. 따라서, 시스템 보안 및 네트워크 보안 기능이 충분히 고려되지 않았다. 그런데, 소프트웨어 개발에 외부 전문 협력업체들과의 협업이 증가됨에 따라 사외의 협력업체들도 MDMS를 사용해야 할 필요성이 발생하였다. 하지만,

MDMS 의 부족한 보안 기능으로 인하여, 협력업체들이 한화탈레스 사외에서 한화탈레스 내부의 MDMS 에 접근하여 협업을 하는 것이 불가능하다. 이러한 제약사항은 협력업체와 인터페이스를 정의하고, 관리 할 때 상당한 비효율성을 가져왔다.

3. MDMS 개선 방안

MDMS 는 최초 개발 당시의 목표인 대형 소프트웨어 시스템에서 컴포넌트간의 메시지 정의 및 관리는 탈성하였다. 하지만, 2 장에서 제시한 것과 같이 현재의 MDMS 는 다양한 프로젝트의 특성 지원 부족, MDMS 자체의 형상관리 및 유지보수 문제, 프로젝트 간 인터페이스 재사용 미지원, 취약한 보안으로 인한 협력업체와의 협업 곤란이라는 문제점 및 한계를 가진다. 이러한 MDMS 의 문제점 및 한계를 극복하기 위해서는 다음과 같은 개선이 필요하다.

우선, MDMS 를 다양한 프로젝트에 적용이 가능한 유연한 SW 구조로 개선하는 것이 필요하다. MDMS 는 다양한 과제에 어떻게 MDMS 가 적용될 수 있을지에 대한 명확한 이해가 없는 상태에서 개발이 되었다. 지금까지 MDMS 를 적용한 경험을 바탕으로 다양한 프로젝트에 적용되기 위해서 특히 코드 및 문서 자동생성 기능의 분리가 필요하다. 메시지 정의 및 관리 기능은 동일한 P/S 방식의 미들웨어를 사용하는 경우 프로젝트 별로 기능의 차이가 거의 없다. 하지만, 프로젝트 별 요구사항 및 특성에 따라 자동 생성 코드 및 문서는 차이가 발생한다. 따라서 코드 및 문서 자동생성 기능만 프로젝트 별로 구현하고 컴포넌트화하여 MDMS 와 연동하여 사용할 수 있도록 함으로써, 최소한의 수정만으로 다양한 프로젝트를 지원 할 수 있다.

또한, MDMS 의 통일되고 일관된 형상관리가 필요하다. 이러한 문제를 개선하기 위해서는 기능 개선 및 형상통제를 위한 전담조직이 필요하다. 모든 MDMS 의 기능 개선 및 오류수정은 전담조직의 형상통제를 통하여 이루어져야 하며, 전담조직은 항상 최신의 MDMS 를 각 프로젝트에 제공해야한다. 각 프로젝트에 필요한 전용 기능들은 프로젝트 내에서 컴포넌트 형태 개발된 후 MDMS 에 플러그인되어 수행될 수 있도록 MDMS 의 구조도 수정되어야 한다.

프로젝트 간 메시지의 재사용을 위해서는 프로젝트 별 MDMS 운용이 아닌 통합된 MDMS 의 운용이 필요하며, 그를 통하여 다른 프로젝트의 인터페이스 정의를 상호 참조할 수 있는 구조를 적용해야 한다. 현재 MDMS 는 다수 프로젝트를 지원할 수 있도록 개발되어 있으나, 언급한 바와 같이 프로젝트 별 특성의 반영 문제로 인하여 프로젝트 별로 별도로 운용되고 있다. 더구나, MDMS 는 프로젝트 간의 메시지의 재사용에 대한 고려 없이 개발되어 메시지 재사용을 지원할 수 있는 기능이 제공되고 있지 않다.

마지막으로 보안 문제에 대한 개선 방안이다. 보안이 강조되는 국방 소프트웨어 개발 프로젝트에서 협력업체들과 협업을 하기 위한 고려해야하는 보안 기능은 다음과 같다. 첫째, 사용자 인증 기능을 포함한

접근 통제 기능이다. 허가된 협력업체의 허가된 담당자만 MDMS 에 허가된 영역에 접근하여 허가된 작업만을 할 수 있도록 효과적인 접근 통제기능을 제공해야한다. 둘째, 네트워크 보안 기능을 제공해야한다. MDMS 시스템을 사외 협력업체의 접근을 허가하기 위해서는 트래픽 도청이나 서비스 방해 공격, 메시지변조 공격 등에 대비할 수 있도록 적절한 네트워크 보안 기능이 제공되어야 한다. 셋째, 데이터 백업 및 복원을 위한 재난 복구 기능이 포함되어야 한다. MDMS 는 중요한 소프트웨어 시스템 설계 정보를 포함하고 있다. 따라서, 서버의 고장이나 자연재해, 혹은 외부의 공격으로 인한 데이터 파괴 등에 대비한 재난 복구 대책 수립이 필요하며, 그를 위한 기능이 제공되어야 한다.

4. 결론 및 향후 연구

대형 소프트웨어 시스템 개발을 위한 인터페이스 설계 및 관리 지원도구인 MDMS 는 1998 년에 최초 개발되어 적용된 이후 8 년여 동안 소프트웨어 개발 생산성 향상에 많은 도움이 되었다. 특히 MDMS 는 컴포넌트간 인터페이스 정의 및 수정 작업을 일원화하여 한번의 작업으로 인터페이스 설계, 인터페이스 형상관리, 인터페이스 설계 문서 생성, 인터페이스 코드 생성까지를 종합적으로 지원함으로써 인터페이스 개발 작업을 단순화시켰다. 하지만, MDMS 는 다양한 프로젝트의 특정 지원 부족, MDMS 자체의 형상관리 및 유지보수 문제, 프로젝트간 인터페이스 재사용 미지원 및 취약한 보안이라는 한계를 드러내었다.

MDMS 의 이러한 문제점 및 한계는 다양한 프로젝트의 효과적인 지원을 위한 컴포넌트 기반의 구조 적용, 일관된 형상관리 및 유지보수를 위한 전담 조직의 구성, 인터페이스 재사용 지원 기능의 추가, 그리고 인증, 접근통제, 네트워크 보안, 재난 복구 기능을 포함한 보안 기능의 제공을 통하여 극복할 수 있다.

한화탈레스는 단순한 인터페이스 설계 및 관리 지원도구를 넘어 소프트웨어 아키텍처 및 구조설계 전반을 지원하는 차세대 MDMS 인 MOMAT(Message Oriented Modeling and Analysis Tool)을 개발하고 있다. 향후 연구로 본 논문에서 제시된 MDMS 개선 방안을 MOMAT 에 적용하기 위한 연구가 필요하다.

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] 류제영, “메시지 정의 및 관리시스템(MDMS) 적용 사례 연구”, 한국정보처리학회 2014 년 추계학술발표 대회 논문집 제 21 권 제 2 호 p.639-631, 2014.11
- [3] 류제영, 박진희, “메시지 지향 소프트웨어 모델링 및 분석 도구 개발”, 한국정보과학회 2010 한국컴퓨터종합학술대회 논문집 제 37 권 제 2 호 (A) 2010.11, page(s): 42-43