

메시지 지향 소프트웨어 모델링 및 분석 도구 개발

류제영

삼성탈레스(주)

jeyoung.yu@samsung.com

Development of Message Oriented Software Modeling and Analysis Tool

Je-Young Ryu

Samsung Thales Co., Inc.

1. 서론

대형 소프트웨어 시스템은 독립된 기능을 수행하는 수백개 이상의 컴포넌트로 구성된다. 각 컴포넌트는 정해진 인터페이스에 따라 정보를 주고 받으며, 상호작용을 함으로써 전체 시스템이 목적인 기능을 수행한다.

수십에서 수백명의 개발자가 참여하는 대형 소프트웨어 시스템의 컴포넌트 간의 인터페이스의 정의 및 관리는 어려운 작업이다. 컴포넌트 간 인터페이스는 소프트웨어 아키텍처에 의하여 영향을 받으며, 세부적인 인터페이스 내용은 개발 과정에서 수시로 변경된다. 인터페이스의 변경은 그 파급효과가 하나의 컴포넌트에 한정되지 않고, 해당 컴포넌트와 정보를 교환하는 모든 주변 컴포넌트에 영향을 주기 때문에 컴포넌트 개발자들 간의 신속하고 정확한 커뮤니케이션이 이루어지지 않을 경우 소프트웨어의 통합에 지장을 초래한다.

이러한 문제를 해결하기 위하여 MDMS (Message Definition & Management System)[1]가 개발되었으며, 실제 대형 소프트웨어인 함정 전투체계 소프트웨어 개발에 적용되어 그 효용성이 입증되었다. MDMS는 컴포넌트 간의 메시지의 정의 및 정의된 메시지 관리 기능을 제공해준다. 하지만, 여러 컴포넌트 간의 인터페이스 관계를 동시에 파악하기 어려우며, 컴포넌트를 도식적으로 표시하며 인터페이스를 설계할 수 없다.

MOSMAT은 MDMS가 제공하는 기능들을 기반으로 각 컴포넌트 레벨 별로 컴포넌트 간 인터페이스 관계를 도식적으로 표현해 줌으로써 인터페이스 중심의 소프트웨어 구조 분석을 효과적으로 지원해 준다. 또한 MDMS의 단순한 테이블 형태의 인터페이스 정의 및 변경에서 벗어나 그래픽적인 컴포넌트 정의와, 컴포넌트 간의 인터페이스 정의가 가능하다. MOSMAT은 오픈 개발 환경인 이클립스(Eclipse) 플러그인 형태로 개발되었다. 따라서, 이클립스에서 제공하고 있는 다양한 개발도구들과 쉽게 통합이 가능하다.

2. 관련연구

대형 소프트웨어 시스템은 일반적으로 고도의 분산 시스템으로 구성된다. 소프트웨어 시스템을 구성하는 각 컴포넌트는 필요에 따라 분산 시스템의 다양한 노드에 설치되어 실행될 수 있다. 따라서, 컴포넌트 간 인터페이스의 복잡도를 감소시키고 위치 중립성을 지원하는 통신 미들웨어를 필요로 한다.

일반적으로 널리 사용되는 통신 미들웨어는 두가지 계열로 분류된다. 첫번째는 CORBA와 DCOM, Java RMI 로 대표되는 객체 지향 미들웨어 (Object-Oriented Middleware: OOM)이다. 두번째는 웹 서비스, JMS (Java Message Service), DDS (Data Distribution Service)와 같은 메시지 지향 미들웨어(Message-Oriented Middleware: MOM)이다.

MDMS(Message Definition and Management System)는 메시지 지향 미들웨어 기반의 대형 소프트웨어 시스템의 개발 과정에서 컴포넌트 간 메시지 지향 인터페이스 정의 및 관리를 효율적으로 하기 위하여 개발되었으며, 다음과 같은 기능을 제공한다.

- 단계별 컴포넌트 정의 기능 (MIL-STD-498 및 무기체계 내장형 소프트웨어 개발 지침에서 정의된

CSCI-CSC-CSU 단계 정의)

- 최하위 컴포넌트 간 출판-구독 (Publish-Subscribe) 형식의 인터페이스 정의, 수정 및 삭제 기능
- 인터페이스 정의 및 수정에 대한 이력 추적 기능
- 입력된 인터페이스 정보를 기반한 인터페이스 처리 템플릿 코드 및 인터페이스 설계 문서 자동 생성 기능
- 웹 기반의 사용자 인터페이스
- 프로젝트 및 사용자 관리 기능

이러한 MDMS가 가진 컴포넌트 간 인터페이스 정보는 소프트웨어 아키텍처를 효과적으로 표현할 수 있으며, 소프트웨어 개발 과정에서 소프트웨어 아키텍처의 모델링 및 분석을 위한 정보로써 활용될 수 있다.

3. 설계 고려 사항

3.1. MDMS와의 호환성

MOSMAT은 MDMS가 제공하고 있는 원래의 기능들을 효과적으로 계승하고, 추가적으로 그래픽 환경에서 컴포넌트 간의 인터페이스의 모델링과 분석이 가능하도록 설계되었다. MOSMAT은 MDMS가 제공하고 있는 기능들을 리치 클라이언트 어플리케이션 (RCP: Rich Client Application) 형태로 구현하여 효율적인 사용자 인터페이스와 빠른 응답 속도를 제공한다. 또한, 서버와의 통신을 최소화함으로써 서버의 부하를 줄임으로써 더 많은 수의 개발자가 동시에 작업을 할 수 있다.

3.2. 그래픽 기반의 컴포넌트 간 인터페이스 모델링

MOSMAT은 그래픽 화면에서 드래그 앤 드롭 형식으로 컴포넌트를 정의하거나 컴포넌트 트리에서 정의한다. 인터페이스 정의는 화면에 표시된 컴포넌트 사이에 선을 연결한 후, 생성된 선을 선택하여 컴포넌트 사이에 출판/구독하는 메시지를 정의함으로써 수행된다. 이러한 그래픽적 인터페이스 모델링은 컴포넌트 사이의 인터페이스 관계를 효율적으로 표현해줌으로써 개발자가 소프트웨어의 구조를 쉽게 파악할 수 있도록 해준다. 또한 인터페이스 관계의 복잡도를 도식적으로 보여주어, 소프트웨어 구조 개선에 도구로써 사용할 수 있다.

3.3. 이클립스 기반의 오픈 개발 환경

이클립스는 자바를 비롯한 다양한 언어를 지원하는 프로그래밍 통합 개발환경임과 동시에 OSGi 기반의 범용 응용 소프트웨어 플랫폼이다. 이클립스 기반으로 MOSMAT을 개발함으로써 다음과 같은 이점을 얻을 수 있다.

- 이클립스의 소프트웨어 개발 플랫폼을 그대로 사용
- 다양한 이클립스 플러그인(컴포넌트)들을 재활용 하여 MOSMAT 개발 노력 절감
- MOSMAT을 확장하여 이클립스가 제공하는 다양한 통합 개발환경 컴포넌트들과 결합 가능
- MS 윈도우즈 계열, 리눅스 계열, 맥 OSX 등 다양한 플랫폼을 지원

4. 결론 및 향후 연구

MOSMAT은 다수의 개발자가 메시지 지향 미들웨어를 사용하여 대형 소프트웨어를 설계를 지원하기 위한 도구이다. MOSMAT은 여러 대형 국방 프로젝트에 적용되어 그 효과가 검증된 MDMS를 더욱 발전시키고 개량하였으며, MDMS와의 하위 호환성을 유지하였다. 특히 그래픽적으로 컴포넌트와 컴포넌트간의 인터페이스를 효과적으로 도식화해 줌으로써 대형 소프트웨어를 구성하는 많은 수의 컴포넌트들 간의 인터페이스를 효과적으로 분석할 수 있다.

5. 참고 문헌

- [1] 류제영, 박진희, “ 메시지 기반 인터페이스 공동 개발을 위한 메시지 관리 시스템”, 정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제6호 p.609~p.613, 2008.8