

XMI로 표현된 UML 모델로부터 생성된 시험 데이터를 이 용한 컴포넌트 기반 소프트웨어 시험 도구

김준산^o 강제성 마유승 권용래

한국과학기술원 전자전산학과

e-mail: {kimjs, jskang, ysm, kwon}@salmosa.kaist.ac.kr

Component-based Software Testing Tool Using Test Data Generated From UML Models in XMI

Jun San Kim^o Jae Sung Kang Yu-seung Ma Yong Rae Kwon

Dept. of Electronic engineering and Computer Science, KAIST

요 약

재사용이 가능한 컴포넌트들을 사용하는 컴포넌트 기반 소프트웨어 개발 기술은 현재 소프트웨어 개발의 핵심 기술로써, 많은 소프트웨어들이 이 가법을 이용해 개발되고 있다. 컴포넌트 기반 소프트웨어를 보다 효율적으로 시험하기 위한 많은 노력이 있었으나, 아직까지 자동으로 시험을 쉽게 수행해 주는 도구는 별로 없으며 이미 개발된 도구들도 그 효율성이 미약하다. 본 논문에서는, 이전에 제안했던 컴포넌트 기반 소프트웨어 자동화 시험 도구인 CTM(Component Testing Manager)의 확장 내용으로써 XMI(XML Metadata Interchange) 형식으로 표현된 UML(Unified Modeling Language) 모델로부터의 시험 데이터 생성 및 수행기법에 대해 제시한다. 본 기법은 기존의 UML 모델을 이용한 다른 시험기법과는 달리 표준화된 XMI를 사용함으로써 UML 모델링 도구와 독립적이고 분산환경에서 쉽게 사용할 수 있는 장점을 제공하여 보다 쉽게 컴포넌트 기반 소프트웨어의 시험을 가능하게 해 준다.

1. 서론

컴포넌트 기반 소프트웨어 공학은 재사용이 가능한 컴포넌트들을 소프트웨어의 기본 구성 단위로 사용함으로써, 소프트웨어의 생산성과 품질을 획기적으로 제고하려는 최근의 시도이다. 그러나 좋은 품질의 소프트웨어를 만들기 위해서는 이에 대한 시험이 선행되어야 하며, 전체 소프트웨어 개발 과정의 많은 부분을 차지하는 이 과정이 자동화 되어야만 보다 효율적으로 소프트웨어를 개발할 수 있다.

컴포넌트 기반 소프트웨어는 컴포넌트와 컴포넌트 기반 기술의 특징으로 인해 이를 시험하고 유지하는 데 있어 기존에 없던 새로운 문제들이 발생하게 된다. 즉, 한 소프트웨어에 사용되는 여러 컴포넌트들은 각각 다른 언어로 쓰여질 수 있어 효율적인 시험이 어려우며, 컴포넌트 제작자와 사용자가 다른 경우가 많기 때문에 제작자는 컴포넌트를 사용자 기준에서 원하는 동작들을 완벽히 시험할 수 없고, 사용자는 소스 코드를 볼 수 없기 때문에 기존의 코드 기반 시험을 수행할 수가 없는 등의 문제가 있다. 보다 안전하고 믿을 수 있는 컴포넌트 기반 소프트웨어를 얻기 위해 이를 보다 효율적으로 시험하기 위한 연구와 노력이 많이 진행되어 왔다. Harrold는 컴포넌트 제공자와 컴포넌트 사용자의 관점에서 컴포넌트 기반 소프트웨어를 분석하여 시험하는 방법을 제시하였다[3]. Buy는 컴포넌트들 사이의 데이터 흐름 분석을 통해 점증적으로 시험을 수행하는 방법을 제시하였으며[4], Ghosh는 컴포넌트의 인터페이스와 예외기준에 기반한 시험 방법을 제시하였다[5]. 국내에서도 단시간에 많은 컴포넌트를 시험하기 위한 시험 환경을 제안하였다[8]. 하지만 많은 연구가 진행되었음에도 불구하고, 실제로 컴포넌트 기반 소프트웨어를 자동으로 시험해주는 도구는 거의 없었다. 이미 개발된 컴포넌트 기반 소프트웨어 시험 도구가 있지만, 시험 과정

의 많은 부분이 수동으로 이루어져 사용하기 번거로우며, 가격도 상당히 높아 실제 컴포넌트 기반 소프트웨어 개발 과정에서 사용하기는 쉽지 않다.

본 논문에서는 컴포넌트 기반 소프트웨어의 시험 과정에서 시험 데이터의 자동 수행과 수행 결과의 자동 비교를 가능하게 해 주는 도구인 CTM[8,9]에 추가된 시험 데이터 생성기의 시험 데이터 자동 생성 기법에 대해 설명한다. 시험 데이터를 생성하기 위해서 컴포넌트 명세의 일반적인 표준으로 널리 사용되고 있는 UML 모델을 이용하여 실제 산업체에서 적용하기 쉽게 하였으며, UML 모델을 메타데이터 교환 표준인 XMI로 변환한 형태를 이용하여 실제 시험 수행이 UML 모델링 도구와 독립적인 동시에 분산환경에서 용이하게 이루어질 수 있도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안하는 시험 데이터 생성과 관련된 UML 다이어그램과 XMI에 대한 기본개념에 대해 설명하고, 3장에서는 시험 데이터 생성기를 추가한 CTM 도구에 대해 설명한다. 4장에서는 XMI로부터의 시험 데이터 생성 방법에 대해 설명한다. 5장에서는 결론과 향후 과제를 제시한다.

2 UML 다이어그램과 XMI

UML은 소프트웨어 시스템의 여러 요소들을 명세하고, 시각화하고, 구축하고, 문서화하는 데 쓰이는 국제 표준화된 모델링 언어이다. UML은 객체지향 소프트웨어를 만드는 데 있어서 가장 널리 쓰이는 모델링 언어이며, 객체지향과 컴포넌트 기반 기술의 적용이 확대됨에 따라 앞으로 더 많은 개발자들이 이 언어를 사용하게 될 것이다. UML 모델을 이용하여 소프트웨어 시험 데이터를 뽑아 내는 기법에 대해서는 UML 상태 다이어그램에서 컨트롤과 데이터 흐름을 바탕으로 시험 데이터를 생성하는 연구와[6], UML 상태 다이어그램에서 상태전이 기준을 사용하여

통계적 기능검사를 수행하는 연구[7]와 같이 기존에 많은 연구가 진행되어 왔다.

XMI는 분산 이종 환경에서 UML에 기반한 모델링 도구와 MOF(Meta Object Facility)에 기반한 메타데이터 저장고 사이에 메타데이터의 교환을 쉽게 하기 위한 XML(eXtended Markup Language), UML, MOF에 대한 OMG(Object Management Group)의 표준 규약이다. UML 모델과 같이 MOF에 맞는 메타모델들을 바탕으로, XMI 표준은 이 메타모델에 해당하는 문법을 기술하고 이 문법을 통해 모델을 저장하고 불러올 수 있다. 이 문법을 기술하기 위해서 XMI 표준은 XML DTD(Document Type Definition)를 사용한다. 즉, XMI는 MOF에 맞는 메타모델을 특정 DTD에 매핑하기 위한 규칙들의 집합을 명세할 뿐 아니라 이 DTD에 맞는 XML 파일로 모델을 매핑하기 위한 방법을 명세한다. 이를 이용하여 분산 시스템 개발자들은 오브젝트 모델이나 다른 여러 메타데이터들을 인터넷을 통해 손쉽게 주고받을 수 있다[2]. 즉 여러 다른 모델링 도구에 의해 만들어진 UML 모델을 XMI 라는 표준형식으로 변환시켜 인터넷을 통해 서로 주고받을 수 있는 것이다. XML 문서를 쉽게 제작할 수 있도록 지원해 주는 최근의 모델링 툴은 UML 모델을 XMI 형식으로 변환하는 기능을 제공하고 있기 때문에, 개발자들은 쉽게 UML 모델을 XMI 형식으로 나타낼 수 있다. UML 모델을 XMI로 표현하는 것과 관련하여 여러 연구가 진행되었는데, Keienburg는 UML 모델을 직접 확장하는 대신 모델을 XMI로 표현한 다음 이를 확장하는 것에 대해 연구하였으며 [10], Damm은 XMI를 이용하여 여러 모델링 도구를 통합하는 것에 대해 연구하였다[11].

컴포넌트 기반 소프트웨어의 명세를 위해서 UML이 업계에서 널리 사용되고 있으며, 이 UML 모델들을 통해서 컴포넌트 기반 소프트웨어를 시험할 수 있는 정보를 얻을 수 있다. XMI 형식으로 변환된 UML 모델을 사용하면, 컴포넌트 사용자가 인터넷을 통해 컴포넌트 개발자로부터 XMI로 표현된 UML 모델을 쉽게 전송 받을 수 있으므로 보다 용이하게 시험 데이터를 자동으로 생성할 수 있다

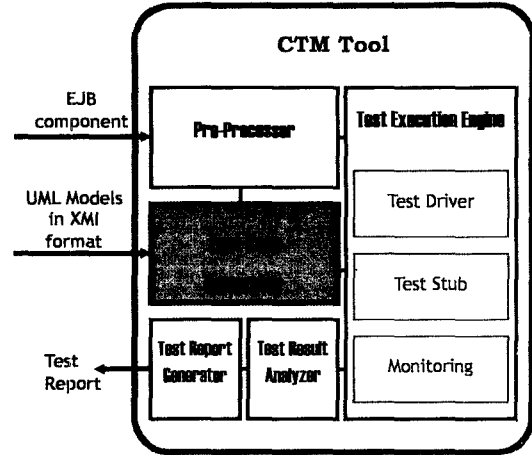
3. CTM 도구 구현

우리가 이미 제안한 시험 도구인 CTM은 컴포넌트 기반 소프트웨어의 시험 수행, 시험 결과 분석, 시험 결과 보고서 생성까지의 과정을 자동으로 처리하며, 시험 예상 결과 추출도 부분적으로 자동 처리한다. 자바 언어로 작성된 CTM은 EJB(Enterprise JavaBeans) 컴포넌트를 시험 대상으로 하며, Weblogic과 J2EE 서버를 지원한다. 블랙 박스 시험 기법을 사용하는 CTM은 EJB 컴포넌트를 자동으로 설치해 주며, 대부분의 시험 과정을 자동 처리함으로써 용이하게 컴포넌트 기반 소프트웨어를 시험할 수 있도록 해 준다. <그림1>은 본 연구에서 제시한 시험생성 기법을 추가한 CTM의 전체적인 구성을 보여 준다. CTM은 크게 Pre-Processor, Test Case Generator, Test Execution Engine, Test Result Analyzer, Test Report Generator로 구성되는데 이 중 본 논문에서 제시한 시험 데이터 생성 기법은 Test Case Generator부분에 속한다. Pre-Processor가 제공해 놓은 환경 하에서 Test Case Generator는 XMI 형식의 UML 파일을 받아 들여 시험에 필요한 데이터를 생성한 후, 이를 Test Execution Engine에 전달한다. 다음은 CTM 수행 단계에 대한 설명이다.

- Pre-Processor

시험 대상 컴포넌트를 응용프로그램 서버에 설치한다. 시험 대상 컴포넌트에 관한 정보를 읽고 시험수행이 가능한 환경을 제공한다.

- Test Case Generator



<그림1 : CTM 수행 단계>

UML 모델로부터 시험 데이터를 생성한다. UML 모델에 나타나지 않는 파라미터 값은 CTM에 이미 저장된 대표값들을 이용하며, 이 대표값들과 UML 모델과의 조합을 통해 시험 데이터들이 얻어진다. 이 외에도 사용자가 시험 데이터를 새로 정의하여 추가할 수 있다.

- Test Execution Engine

시험 드라이버와 스텝(stub)을 생성한다. 시험 수행과정을 모니터링 하면서 오류가 발생했을 경우 주적이 가능하게 한다.

- Test Result Analyzer

시험 대상 컴포넌트의 외부 출력값과 데이터베이스의 변경이 제대로 이루어졌는지를 분석한다.

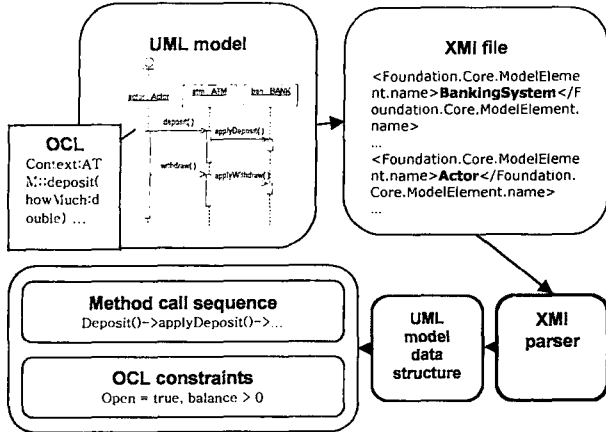
- Test Report Generator

시험수행 결과를 보고한다.

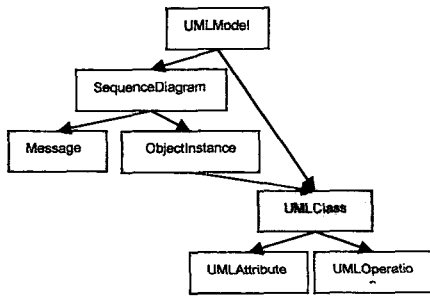
4. XMI로부터 시험 데이터 생성 방법

컴포넌트 기반 소프트웨어는 대부분의 경우 UML을 사용하여 명세하게 된다. UML 모델의 여러 다이어그램 중에서, 시퀀스 다이어그램으로부터 메소드 호출 순서를 뽑아 낼 수 있고, 클래스 다이어그램으로부터 메소드와 멤버 변수와 같은 클래스 정보를 뽑아낼 수 있다. 또한, UML OCL(Object Constraint Language)을 통해서 메소드가 실행되기 전의 전제조건을 알아낼 수 있다. 그 외에도 유즈 케이스 다이어그램을 통해서 전체 시스템에서 액터와 각 컴포넌트의 역할을 알 수 있으며, 컴포넌트 다이어그램을 통해서 전체 시스템과 컴포넌트 구성을 알 수 있다. 이렇게 UML 모델에서 얻은 정보를 컴포넌트 기반 소프트웨어의 시험에 이용할 수 있다.

이전에 제안한 CTM에 추가된 시험 데이터 생성기는 UML 시퀀스 다이어그램과 UML OCL을 이용하여 시험 데이터를 생성한다. 이 논문에서는 CTM의 시험 데이터 생성기가 UML 모델 대신에, 이 모델을 XMI 형식으로 바꾼 형태를 이용하는 방법을 제시한다<그림2>. 모델링 툴을 이용하여 그런 UML 시퀀스 다이어그램과 UML OCL을 모델링 툴을 통해 XMI 형식의 파일로 변환한다. 시험 데이터 생성기는 이 XMI 파일을 입력으로 받아들이고 XMI 파서를 통해 시험에 필요한 정보인 메소드 호출 순서와 OCL에 기록된 전조건, 후조건 정보를 알아 낸다. XMI 파서는 XMI 파일에 기록된 XMI 태그들을 이용하여 이에 맞는 XMI 오브젝트들을 생성하고, 이렇게 생성된 오브젝트들은 UML 모델을 나타내는 자료 구조에 저장된다<그림3>. 시험 데



<그림 2 : XMI로부터 시험 데이터 생성>



<그림 3 : UML 모델을 나타내는 자료 구조>

이러한 생성기가 사용 가능한 자료구조에 저장된 UML 모델 정보를 통해서 UML 시퀀스 다이어그램에 나타난 각 메시지들이 어느 곳에서 어느 곳으로 향하며, 각 메시지가 일어날 수 있는 순서는 어떤 것이 있으며, 메시지에 대한 리턴값은 어떤 형태로 주어질 것인지 등에 대한 구체적인 정보를 알아내며, 이를 통해 실제 시험에 사용하는 메소드 호출 순서를 알아낸다. OCL에서 알아낸 전조건, 후조건 정보를 이용하여 검사 과정에서 호출된 메소드들이 과연 사용자가 원하는 대로 수행되는지를 알아보는데 사용한다. 시험 과정에서 메소드 호출시 적용하는 파라미터 값은 이미 CTM 도구에 사용자에게 의해 입력되어 있는 대표값들을 사용한다. 요약하면, 시험 데이터 생성기는 XMI로 표현된 UML 모델을 받아 들여 시험에 필요한 데이터인 메소드 호출 순서와 OCL 정보를 얻어 낸다. UML 모델을 XMI 형식으로 바꾸어 주는 기능은 대부분의 모델링 도구에 의해 지원되고 있으며, 컴포넌트 사용자는 인터넷을 통해 다른 개발자에게서 XMI로 표현된 UML 모델을 쉽게 전송 받을 수 있다. 이렇게 XMI를 이용하면 시험할 프로그램에 대한 UML 모델을 쉽게 얻을 수 있으므로 보다 용이하게 시험 데이터를 자동으로 생성할 수 있다.

5. 결론 및 향후 연구방향

본 논문에서는, UML 모델을 메타데이터 교환 표준인 XMI로 변환한 형태로부터 시험 데이터를 자동 생성하고 수행하는 기법을 제안하였다. 이 기법은 기존의 CTM 도구에 추가되어 컴포넌트 기반 소프트웨어의 시험을 보다 용이하게 하였다. 현재

는 시험 데이터의 파라미터 값으로 CTM에서 제공하는 대표값들을 사용하였으나, 각 메소드의 기능을 충분히 수행하게 해주는 파라미터 값을 뽑는 것에 대한 연구 계획이 있다. 또한 현재 CTM 도구는 WebLogic과 J2EE 서버만을 지원하는데, 앞으로 다양한 서버를 지원하도록 확장할 계획이다. 그리고 UML 모델의 유즈 케이스 다이어그램이나 상태자트 등에서 뽑아낸 정보를 활용하여 시험하는 방법에 대한 연구도 필요하다.

참고문헌

[1] Ye Wu, Dai Pan and Mei-Hwa Chen "Techniques for testing component-based software" In Proceedings of the 7th IEEE International Conference on Engineering of Complex Computer Systems, 2001.

[2] OMG. "XML Metadata Interchange Specification" (version 1.0) <http://www.omg.org>, document number : ad/98-10-05, 1998

[3] M.J. Harrold, D. Liang, and S. Sinha "An approach to analyzing and testing component-based systems" In Proceedings of the 1st International ICSE Workshop on Testing Distributed Component-Based Systems, 1999.

[4] S. Ghosh and A.P. Mathur "Issues in testing distributed component-based systems" In Proceedings of the 1st International ICSE Workshop on Testing Distributed Component-Based Systems, 1999.

[5] U. Buy, C. Ghezzi, A. Orso, M.Pezze and M. Valsasna. "A framework for testing object-oriented components" In Proceedings of the 1st International ICSE Workshop on Testing Distributed Component-Based Systems, 1999.

[6] Y. G. Kim, H. S. Hong, D. H. Bae and S. D. Cha "Test cases generation from UML state diagrams" In IEE Proceedings of Software, 1999.

[7] P. Chevalley and P. Thevenod-Fosse "Automated generation of statistical test cases from UML state diagrams" In Proceedings of Computer Software and Applications Conference, 2001.

[8] Y. S. Ma, S. U. Oh, D. H. Bae and Y. R. Kwon "Framework for Third Party Testing of Component Software" In the Proceedings of 8th Asian-Pacific Software Engineering Conference, 2001.

[9] 김상운, 강제성, 마유승, 권용래 "UML/OCL을 이용한 기열형 컴포넌트 자동화 시험 환경의 구축", 한국정보과학학회 2001 춘계학술대회 논문지, 2001.

[10] F. Keienburg and A. Rausch "Using XML/XMI for tool supported evolution of UML models" In Proceedings of the 34th Annual Hawaii International Conference on System Sciences, 2001.

[11] C. H. Damm, K. M. Hansen, M. Thomsen, M and Tyrsted " Tool integration: experiences and issues in using XMI and component technology" In Proceedings of Technology of Object-Oriented Languages, 2000.