UML기반의 테스트 데이타 자동생성 도구 : AUTEG

(Automatic UML-based Test Data Generating Tool: AUTEG)

김 청 아 † 최 병 주 ††
(Cheongah Kim) (Byoungju Choi)

요 약 본 논문에서는 UML 개발도를 이용하여 테스트 데이타를 자동 생성하는 방안을 제안하고, XML 기술을 이용하여 개발한 "테스트 데이타 자동화 도구인 AUTEG(Automatic UML-Based Test Data Generation)"를 "Insurance System"의 사례에 적용한 결과를 분석 기술한다. AUTEG는 전체 시스템을 구성하는 모듈 사이의 인터페이스(interface)영역에 존재하는 오류 추출이 가능한 테스트도(test diagram)와 기존의 화이트 박스 테스트(white-box test)기법을 테스트도에 적용하여 테스트 데이타를 자동 생성한다. 또한 AUTEG는 통합 테스트와 시스템 테스트에 적용할 수 있으며, 사용자가 통합 테스트의 단위 모듈을 자유롭게 그룹화 할 수 있다.

키워드 : 통합 테스트, 테스트 데이타 자동 생성, UML

Abstract In this paper we suggest a method to produce automatically test data using UML development diagrams, and analytically describe the application of a tool, Automatic UML-based Test Data Generation (AUTEG) developed using XML technology, to the examples of insurance system. Our AUTEG automatically generates test diagrams that enable to detect errors existing at the interface area between modules composing the whole system, along with test data by applying the existing white-box test technique to the test diagram. Our AUTEG can be applied to the integration test as well as the system test and using the tool, users may make the unit modules of the integration test into several groups.

Key words : Integration test, automatic test data generation, UML

1. 서 론

소프트웨어 테스트는 크게 단위 테스트(unit test)와 각 단위들의 통합을 목적으로 수행하는 통합 테스트(integration test), 전체 시스템을 대상으로 하는 시스템 테스트(system test)로 테스트 레벨을 구분할 수 있다. 단위 테스트의 경우 해당 단위를 구현한 개발자 자신이 테스터가 되어서 코드를 완전히 이해한 후 테스트를 진행하기 때문에 테스트가 용이하고, 현재까지 단위 테스트를 위한 기법은 많이 연구되었다. 이에 반해, 통합 테스트의 경우에는 소프트웨어의 요구사항이 늘어나고 규모가 커지면서 여러 사람이 팀을 이루어 개발하는 것이 일반적이므로 서로 다른 개발자들이 구현한 코드를 통합해야한다. 또한 시스템이 언어나 운영체제 등의 개발 환경의 한계를 벗어나 다른 여건에서 구현되는 추세이기 때문에 이들의 통합이 중요한 이슈가 되고 있으나 통합을 위한 테스트는 단위 테스트나 시스템 테스트에 비해 연구가 미흡한 상태이다.

현재 객체지향 개발 프로세스에 표준으로 이용되고 있는 UML(Unified Modeling Language)을 다양한 개발환경과 특성을 고려하여 확장한 연구가 많이 진행되고 있다. 실시간 시스템을 위한 UML[1], 비즈니스 모델링을 위한 UML[2], 프로세스를 위한 UML[3], 컴포넌트와 프레임워크를 위한 UML[4] 등이 있다. 그러나, 이들은 주로 개발 프로세스에서 요구되는 모델들을 제

· 본 연구는 한국과학재단 목적기초연구(과제번호 : 2000-0-303-02-3) 지원으로 수행되었음
† 비 회 원 : 이화여자대학교 컴퓨터학과
992COG27@ewha.ac.kr
†† 종신회원 : 이화여자대학교 컴퓨터학과 교수
bjchoi@ewha.ac.kr
논문접수 : 2001년 8월 30일
심사완료 : 2002년 1월 18일

공할 뿐, 테스트 프로세스에 대한 고려는 미약한 상황이다. 따라서, 본 논문에서는 'UML 기반 테스트도(test diagram)'(이후 'UML 기반 테스트도'는 '테스트도'로 기술한다.)와 테스트도에 의한 테스트 자동화 방안을 제안하고, XML 기술을 적용하여 개발한 UML기반의 테스트 데이타 자동화 도구인 AUTEG(Automatic UML-Based Test Generation)를 기술한다.

AUTEG는 테스트 데이타를 추출하기 위해서 프로그램의 원시 코드만으로는 필요한 정보를 추출하기 어렵기 때문에 명세의 대표적인 형태인 시나리오(scenarios)를 이용한다. 즉, 시스템의 통합 테스트를 수행하기 위해서 UML의 순서도(sequence diagram)와 클래스도(class diagram)를 이용하여 통합 대상 사이의 메시지를 중심으로 테스트 항목 추출이 가능한 테스트도(test diagram)를 작성하며, 이를 이용하여 통합 대상 사이의 인터페이스(interface)를 중심으로 테스트 데이타를 생성한다.

본 논문의 구성은 다음과 같다. 2장에서는 UML기반의 테스트 데이타 생성 기법과 도구에 대한 기존 연구를 살펴보고, 3장에서는 UML을 이용한 테스트 자동화 방안을 기술한다. 4장에서는 사례를 이용하여 UML기반의 자동화 알고리즘에 적용하고, 본 논문에서 개발한 AUTEG에 실행 결과를 기술한다. 5장은 2장에서 열거한 기존 UML기반의 테스트 데이타 생성 도구를 AUTEG와 비교 분석하고, 마지막 6장에서는 결론 및 향후 연구 과제를 기술한다.

2. 관련연구

AUTEG는 UML의 순서도를 이용하여 통합 대상들 사이의 인터페이스를 중심으로 테스트하는 명세 기반의 테스트이다. 따라서, 기존의 UML기반의 도구인 ATTOL Test Suit™, Prosa/om UML Case, Stp/T을 기술하고, 본 논문에서 제안한 AUTEG의 필요성을 나타낸다.

ATTOL Test Suit™[5]는 ATTOL Coverage™에 의해 코드의 모든 커버리지(coverage)를 분석해서 테스트 데이타를 생성하고, ATTOL UniTest™ 와 ATTOL SystemTest™를 이용하여 단위 테스트와 시스템 테스트를 수행한다. UML 상태도를 이용하여 테스트 데이타를 자동 생성하고, 테스트가 수행되는 사항을 모니터링 할 수 있다. 상태도의 객체가 상태 전이를 유발할 때마다 그 객체와 이벤트(event)를 테스트 데이타로 추출하며, 실시간 시스템에 적용한다.

Prosa의 Prosa/om UML Case[6]는 Prosasim이라는 소프트웨어 테스트 기준을 적용하여 이벤트 경로를 추적하면서 테스트 데이타를 추출한다. UML의 클래스도를 이용하여 메소드의 경로를 추적하여, 수행되는 모든 절차를 모니터링 할 수 있을 뿐만 아니라 병렬적인 작업을 동시에 테스트하고, 검증할 수 있다. 그러나, 단위 테스트 레벨에 한정되어 진행된다.

Stp/T[7]는 Stp/UML로 작성된 명세와 클래스도를 이용하여 테스트를 수행한다. 시스템 모델로부터 제공하는 명세를 이용하여 테스트를 수행하기 전에 UML에 관한 분석과 설계에 대한 정보를 저장소(repository)에 저장하여 테스트 데이타를 생성한다. 클래스를 테스트 단위로 하는 단위 테스트와 시스템 테스트에서 수행된다.

위에 기술한 기존 UML기반의 테스트 도구들은 상태도나 클래스도를 이용한다. 또한, 단위 테스트나 시스템 테스트 레벨에만 적용하도록 개발한 도구들이다. 본 연구에서 개발한 AUTEG는 UML의 순서도와 클래스도를 이용하여 테스트도와 테스트 데이타를 자동 생성한다. 테스트도는 테스트를 위한 UML기반 모델로서 통합 대상들 사이의 인터페이스를 중심으로 테스트할 때 효과적으로 사용 될 수 있다. 또한, UML 기반의 소프트웨어 개발 프로세스의 전 단계와 연계되어 체계 적인 테스트 데이타를 선정할 수 있을 뿐만 아니라 사용자가 통합테스트 대상의 단위를 자유롭게 그룹화할 수 있도록 하였다.

3. AUTEG의 설계 및 개발

3.1 테스트도

본 논문에서 개발한 AUTEG는 UML 기반의 개발 프로세스에 따라 생성된 클래스도와 순서도 등의 개발 산출물을 이용하여 테스트도를 자동 생성하고, 이 테스트도로부터 테스트 데이타를 자동 생성하는 테스트 도구이다. 테스트도는 통합 대상들 사이의 인터페이스 영역에서의 오류를 찾을 수 있는 테스트 항목 추출을 위하여 논문에서 제안한 '통합 테스트 모형'이다. 본 논문에서는 이 통합 테스트 모형을 테스트도라 칭하고, UML기반의 객체지향 개발 프로세스에 따라 생성되는 순서도와 클래스도에 직접 연계하여 테스트도를 자동생성하고, 이로부터 통합 테스트데이타를 자동 생성 할 수 있도록 한 AUTEG 테스트 도구의 설계 및 구현 내용을 기술한다.

테스트도는 통합 대상을 나타내는 노드(node)와 이들 사이의 인터페이스를 표현하는 메시지 흐름(message flow) 즉, 에지로 구성한다. 그림 1은 이들에 대한 명세를 나타낸다. 여기에서 ASF(Atomic System Function)란 일반적인 의미로는 입력에서 시작하여 출력까지의 단

위를 말하는데[8], 본 논문에서는 통합테스트 대상 단위로 전이를 일으키는 메시지를 기준으로 ASF를 나눈다.

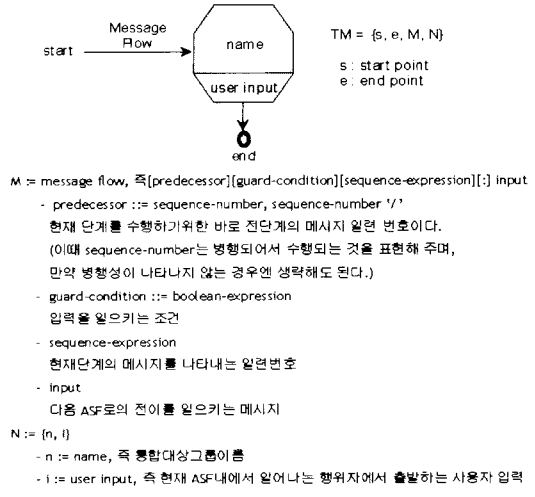


그림 1 테스트도

3.2 AUTEG의 구성

AUTEG의 구조는 그림 2와 같다. 특정항목을 추출하고 저장하는 모듈, 테스트도를 작성하기 위한 대상을 정의하고 저장하는 통합 대상 정의 모듈, 테스트도와 테스트 데이타를 생성하는 모듈로 구성된다. AUTEG는 Windows NT 4.0 환경에서 개발하였으며, 개발 언어로는 Java2 SDK 1.2.2 (JDK1.2.2)으로 구현하였다. AUTEG는 XML 1.1 및 Unisys Rose/XML Interchange package [9,10]를 이용하여 UML을 XML으로 변환하였으며, XMI 포맷을 준수하는 "uml.dtd"를 그대로 수용한다. AUTEG에 사용된 데이타베이스는 JDBC이다.

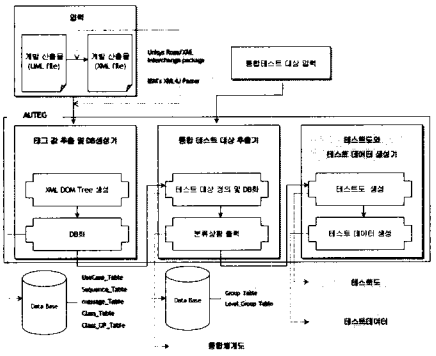


그림 2 AUTEG의 구조

3.2.1 태그값 추출 및 DB 생성기

AUTEG의 입력은 테스트 대상 시스템의 UML 개발 산출물이다. 이 UML은 XML 파일로 변환하여 파싱 작업을 통해 사용사례도, 패키지도, 클래스도, 순서도, 협력도 등의 해당 태그값(element)을 추출하여 데이타 베이스에 저장한다.

(1) XML DOM 트리 생성

UML로부터 변환된 XML 파일로부터 IBM's XML4J parser를 이용하여 XML DOM 트리를 생성한다.

(2) 태그값 추출 및 DB화

XML DOM Tree로부터 사용사례도, 패키지도, 클래스도, 순서도, 협력도의 해당 태그 값을 추출하여 데이타 베이스에 저장한다. 그림 3은 추출할 태그값과 그림 4는 DB의 구조를 나타낸다.

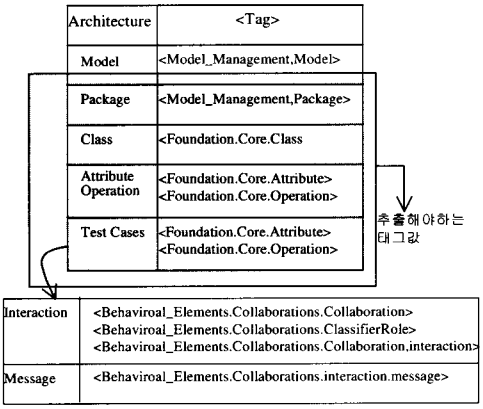


그림 3 추출할 태그값

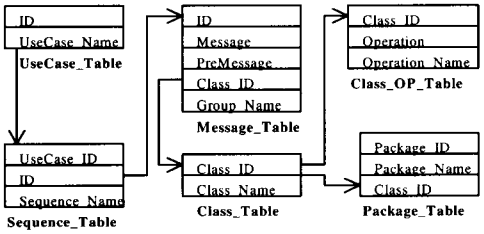


그림 4 DB 구조

3.2.2 통합테스트대상 추출기

AUTEG는 통합테스트 대상의 범위 및 통합 체계를 사용자로부터 자유로이 결정할 수 있도록 하였다. 사용자가 테스트 대상 시스템의 개발 산출물인 클래스도에 통합테스트 대상과 통합 테스트 대상 사이의 통합 체계

를 표시하면, 이 통합테스트 대상에 관한 정보를 데이타베이스에 저장한다.

(1) 통합 대상 정의 및 DB화

사용자로부터 입력된 통합테스트대상 범위와 통합체계의 정보를 그림 5에서처럼 Level_Group_Table과 Group_Table로 데이타베이스에 저장한다.

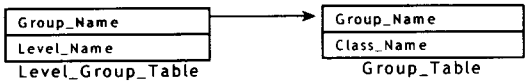


그림 5 통합대상범위 정보를 포함한 DB구조

(2) 분류 상황 출력

각 통합 단위의 범위와 이들 사이의 통합체계도를 출력한다. 예를 들어 4절의 사례연구에서 출력한 통합체계도는 그림 10과 같다.

3.2.3 테스트도 생성기

테스트도는 통합 대상을 나타내는 노드(node)와 이들 사이의 인터페이스를 표현하는 메시지 흐름(message flow)으로 구성한다. 테스트도는 태그 값 추출 및 DB생성기와 통합테스트대상 추출기로부터 저장된 데이타 베이스로부터 자동 생성된다. 테스트도 생성단계는 1)순서도와 협력도 추출, 2) 통합 테스트 대상 범위 표시, 3) 테스트도 생성의 세 단계로 구성한다.

(1) 순서도와 협력도 추출

그림 3의 순서도와 협력도에 해당하는 태그값은 그림 4의 데이타베이스의 message_Table에 저장되어 있으므로, 이로부터 통합 테스트 대상에 관련된 순서도와 협력도를 추출한다.

(2) 통합테스트 대상 범위 표시

그림 5의 데이타베이스 정보로부터 각 순서도의 객체들을 통합 테스트 대상 범위에 따라 그룹핑하여 그 정보를 그림 4의 message_Table의 Group_Name에 추가한다. 이 과정의 이해를 돕기 위하여 XML이 아닌 UML로써 예를 들어 기술하면 그림 6과 같다. 그림 6은 4절의 사례연구의 순서도 중 하나로 그림 11의 통합테스트체계도의 내용에 따라 통합 테스트 대상 그룹, TM1-1, TM1-2, TM1-3 등을 표시한 것이다. 이 순서도의 정보는 앞에 기술하였듯이 해당 클래스명이 있는 message_Table의 Class_ID를 참고하여 검색하며 통합테스트 대상 그룹명을 message_Table의 Group_Name에 추가한다. 본 논문의 통합테스트대상의 범위 그룹체계는 UML의 패키지를 이용하여 레벨별로 정의하였다. 사용자가 자유롭게 지정할 수 있다.

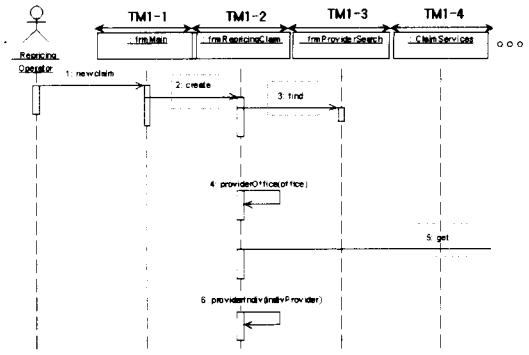


그림 6 통합테스트체계도에 따른 순서도에 통합테스트 대상표시

(3) 테스트도 생성

통합테스트 체계도에 따라 통합테스트 대상이 표시된 순서도 및 협력도로부터 테스트도를 작성한다. 테스트도는 3.1절에 기술하였듯이 노드와 메시지흐름을 나타내는 에지로 구성하는 그래프이다. 노드는 ASF, 즉 통합테스트 대상의 단위를 함축하고 있으며, 에지는 통합테스트 대상 단위로 흐르는 메시지를 나타낸다. 예를 들어 그림 6에서 통합테스트대상 그룹 TM1-1과 TM1-2 및 TM1-3로 흐르는 메시지가 있는 부분은 박스로 표시된 create 및 find이다. 이것은 4절의 사례연구의 테스트도인 표 1의 TM2-7 테스트도의 에지 가운데 frmRepricingClaim::create()와 frmProviderSearch ::find()에 해당된다. 테스트도를 생성하는 알고리즘은 아래와 같다.

알고리즘 1 테스트도 작성

```
Input :  DB의 Message_Table, Class_Table,
         Class_OP_Table //그림 4 DB
Output : 테스트도
// x:y 표기 의미는 x의 필드 y
begin
npre = Message_Table의 Predecessor 필드값;
n = Message_Table의 첫째 레코드;
while (n is not null) do
   if (n:Group_Message is not
                        npre:Group_Name)  then
      DrawTestDiagram(n_Group_Name,
                        n:Class_ID, n:Message);
   else
      n = n:Next;
   endif
endwhile
end;
```

3.2.4 테스트데이타 생성기

테스트도는 노드와 에지로 구성하는 그래프이다. 이것에 화이트박스 테스트 기법을 적용하여 통합테스트 데이타를 생성한다. 현재 AUTEG는 테스트도에 all-edge를 적용하여 테스트 데이타를 자동 생성하였다.

4. AUTEG의 실행 사례

본 연구에서는 연수원관리시스템, 물품관리시스템, Insurance System에 AUTEG를 적용하여 사례연구를 수행하였다. 이들에 대한 UML 개발 산출물과 통합테스트체계도 및 AUTEG로부터 생성되는 테스트도와 테스트데이타는 홈페이지[11]에 기록하였다. 본 논문에서는 Insurance System의 통합테스트의 테스트데이타를 AUTEG로부터 생성한 사례결과를 간략히 기술하겠다.

4.1 UML 개발 산출물

사례 대상인 Insurance System의 사용사례도와 AUTEG 생성과정 설명에 필요한 패키지도를 그림 7과 그림 8에 나타내었다. 클래스도 등 전체 개발산출물의 내용은 본 프로젝트 결과 홈페이지 [11]를 참고하기로 한다.

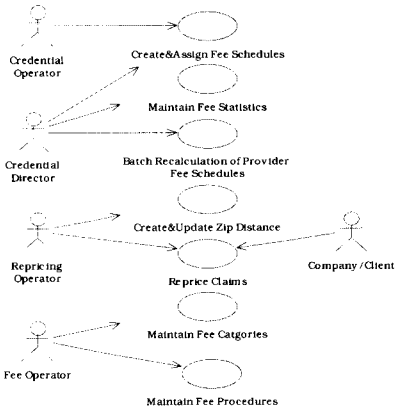


그림 7 Insurance System의 사용사례도

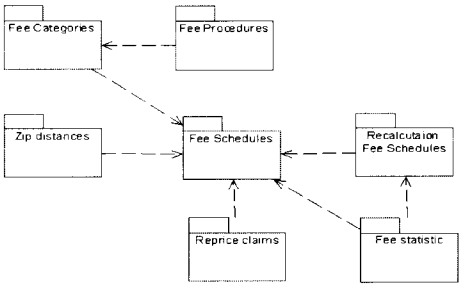


그림 8 Insurance System의 패키지도

4.2 AUTEG의 실행

본 절에서는 AUTEG를 이용하여 테스트도와 테스트데이타를 생성하는 실행사례 결과를 (1)태그 값 추출 및 데이타 베이스 생성, (2)통합 테스트 대상 추출, (3) 테스트도 및 테스트 데이타 생성의 순서로 기술하겠다.

4.2.1 태그값 추출 및 데이타베이스 생성

그림 9는 AUTEG의 초기화면으로부터 사용자가 사례를 선택해서 태그값 추출 및 DB를 자동 생성하는 과정이다. 그림 9의 화면 1은 AUTEG의 초기화면이며, 화면 2는 초기화면의 "View"메뉴를 선택하여 Insurance System의 UML 개발 산출물을 XML DOM 트리로 변환하여 화면에 출력한 결과이다. 또한, 데이타베이스를 생성하기 위해서 Insurance System의 파일을 선택하여 태그값을 추출하여 DB화하는 과정이 화면 4이다.

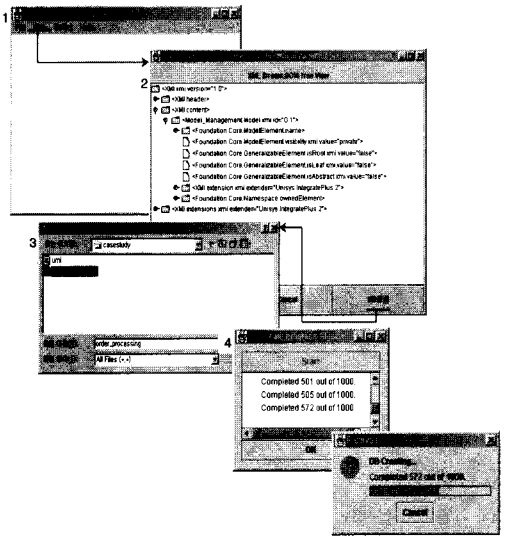


그림 9 데이타베이스 생성과정

4.2.2 통합 테스트 대상 추출

그림 10은 AUTEG의 초기화면에서 통합테스트대상의 범위를 정의하는 화면과 통합테스트체계도를 나타내는 화면이다. 화면 2는 사용자가 클래스를 그룹화 것으로 다이얼로그 박스(dialog box)에 해당 레벨과 레벨을 구성하는 그룹명을 입력하고, 통합대상정의화면에 출력된 항목을 사용자가 직접 선택한다. 선택 된 입력 사항들을 데이타 베이스에 각각 저장해서 테스트도를 작성할 때 이용된다. 화면 3은 사용자가 레벨 1에서 구성된 TM1-1, TM1-2, TM1-3 등과 같은 통합테스트 대상 그룹들을 상위 단계에서 다시 그룹화하는 사항을 나타

낸 것이다. 이처럼 상위 레벨로 올라갈수록 하위 레벨에서 정의된 그룹을 이용하여 새로운 그룹을 추가하게 된다. Insurance System의 통합테스트를 위하여 사용자로부터 입력된 통합테스트 대상의 범위와 통합테스트 체계에 따라 AUTEG로부터 출력되는 통합테스트 체계도는 그림 11과 같다

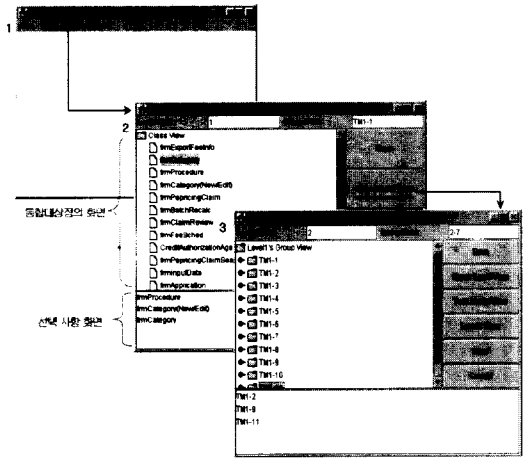


그림 10 통합테스트대상의 범위를 정의하는 화면

4.2.3 테스트도 및 테스트데이타 생성

그림 12는 Insurance System을 AUTEG에 적용해서 얻은 테스트도의 일부이다. 즉, 그림 11의 화면 3의 Test Model View 메뉴를 선택하면, 그림 12과 같이 테스트도와 테스트 데이타가 생성된다. 그림 12의 화면 2는 통합테스트체계도의 TM2-7에 해당하는 테스트도이다. Insurance System를 AUTEG로 실행한 통합테스트 레벨2의 TM2-1, TM2-2, TM2-3, TM2-4, TM2-5, TM2-6, TM2-7의 해당 테스트도와 생성된 테스트 데이타는 표 1과 같다. 그 외 전체 결과는 홈페이지[11]를 참조하자.

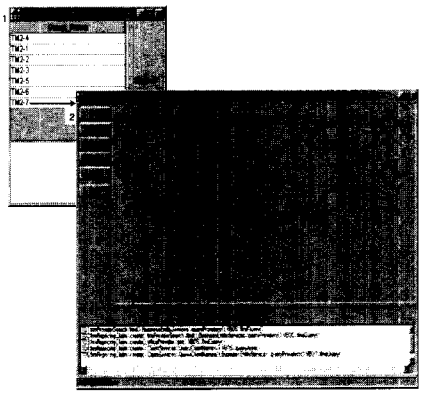


그림 12 테스트도 생성(TM2-7)

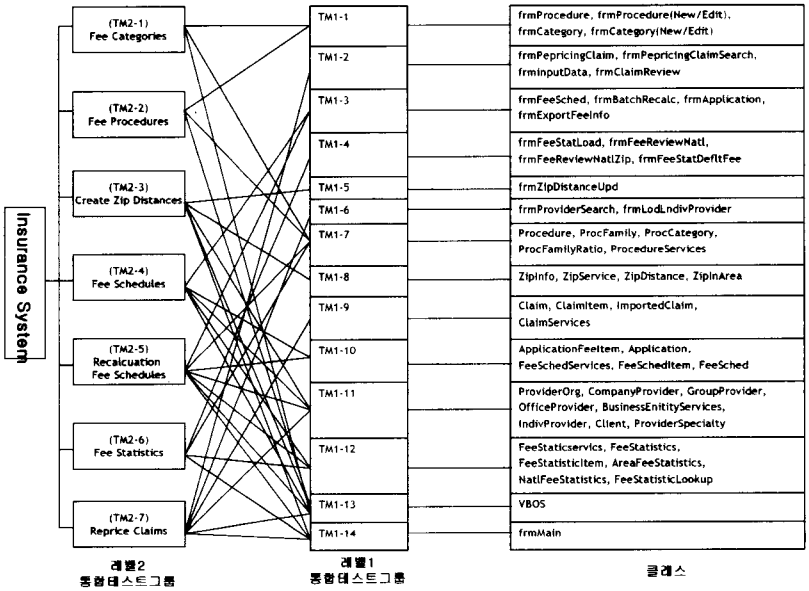
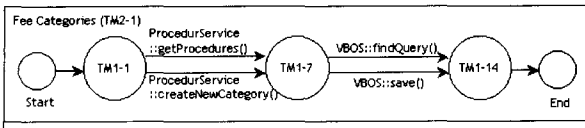
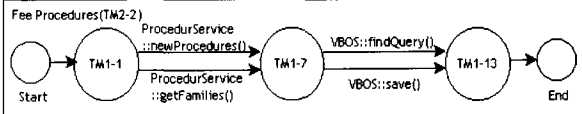
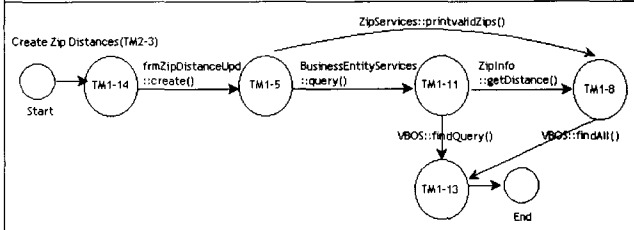
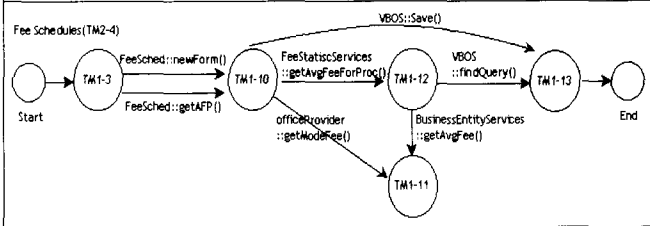
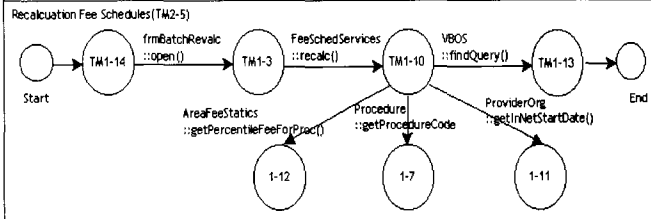
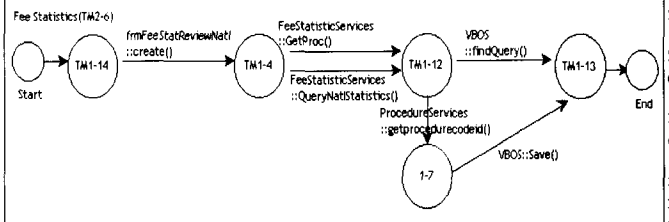
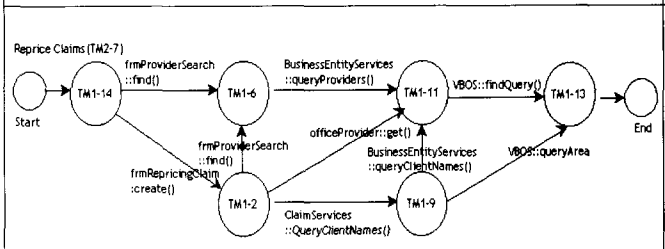


그림 11 통합테스트체계도

표 1 레벨 2의 테스트도 및 테스트데이타

	-ProcedurService::getProcedures(),VBOS::findQuery() -ProcedurService::createNewCategory(),VBOS::save()
	-ProcedurService::newProcedures(),VBOS::findQuery() -ProcedurService::getFamilies(),VBOS::save()
	-frmZipDistanceUpd::create(),BusinessEntityServices::query(),ZipInfo::getDistance(),VBOS::findAll() -frmZipDistanceUpd::create(),ZipServices::printvalidZips(),VBOS::findAll() -frmZipDistanceUpd::create(),BusinessEntityServices::query(),VBOS::findQuery()
	-FeeSched::newForm(),FeeStatiscServices::getAvgFeeForProc(),VBOS::findQuery() -FeeSched::newForm(),VBOS::save() -FeeSched::getAFP(),FeeStatiscServices::getAvgFeeForProc(),VBOS::findQuery() - FeeSched::getAFP(),VBOS::save()
	-frmBatchRevalc::open(),FeeSchedServices::recalc(),VBOS::findQuery()
	-frmFeeStatReviewNatl::create(),FeeStatisticServices::GetProc(),VBOS::findQuery() -frmFeeStatReviewNatl::create(),FeeStatisticServices::GetProc(),ProcedureServices::getprocedurecodeid(),VBOS::save() -frmFeeStatReviewNatl::create(),FeeStatisticServices::QueryNatlStatistics(),VBOS::findQuery() -frmFeeStatReviewNatl::create(),FeeStatisticServices::QueryNatlStatistics(),ProcedureServices::getprocedurecodeid(),VBOS::save()
	-frmProviderSearch::find(),BusinessEntityServices::queryProviders(),VBOS::findQuery() -frmRepricingClaim::create(),frmProviderSearch::find(),BusinessEntityServices::queryProviders(),VBOS::findQuery() -frmRepricingClaim::create(),officeProvider::get(),VBOS::findQuery() -frmRepricingClaim::create(),ClaimServices::QueryClientNames(),VBOS::queryArea() -frmRepricingClaim::create(),ClaimServices::QueryClientNames(),BusinessEntityServices::queryClientNames(),VBOS::findQuery()

5. AUTEG의 분석

AUTEG는 UML의 개발산출물을 이용하여 테스트 데이타를 생성한다. UML기반의 테스트 데이타 도구는 2절에서 기술하였듯이 ATTOL Testware의 ATTOL Test SuitTM, Prosa의 Prosa/om UML Case, Aonix의 Stp/T(Software through Pictures for Testing)가 있다. AUTEG와 이들 도구들을 적용되는 테스트레벨, 테스트 데이타 선정근거, 테스트 데이타 선정기준을 포함한 특징 및 장점을 비교함으로써 AUTEG를 분석하고자 한다. 표 2는 이들을 비교한 내용이다.

ATTOL Testware의 ATTOL Test SuitTM은 클래스를 단위로 하는 단위 테스트나 시스템 테스트 레벨에 적용된다. UML의 상태도(state diagram)에 메시지 기반 테스트(message passing-based test)기법을 이용하여 객체의 상태 전이가 발생할 때마다 전이를 일으키는 이벤트와 해당 객체를 테스트 데이타로 추출한다. 메시지 기반 테스트 기법을 통해 자동차 설계, 항공우주공학, 휴대 전화, 네트워크 관리와 같은 원거리 통신과 네트워크 장비를 포함한 복잡한 내장형 시스템을 테스트할 경우에 효과적이다. 또한, UML이나 MSC 기반의 모델링 도구로부터 테스트 스크립트를 자동 생성할 수 있으며 테스트 수행 과정을 모니터링 할 수 있다.

Prosa의 Prosa/om UML Case는 클래스를 단위로 하는 단위 테스트 레벨에 적용된다. 도구 자체에서 제공되는 Prosa/om Modeling Editor는 소프트웨어 테스트를 수행하기 위한 상호 교환적인 UML을 작성하는 편집기이다. 이 편집기로 작성 된 UML의 클래스도(class diagram)에 화이트 박스 테스트(white-box test)기법을 응용하여 개발한 Prosasim이라는 방법을 이용하여 테스트 데이타를 추출한다. 사용자가 클래스도를 구성하고 있는 클래스를 선택해서 상호작용 하는 메소드의 경로를 다이어그램상에서 시각적으로 추적할 수 있으며 실시간 시스템환경에 적합하다.

Aonix의 Stp/T는 클래스를 단위로 하는 단위 테스트와 시스템 테스트 레벨에 적용된다. Stp/T도 Prosa/om UML Case와 마찬가지로 도구 자체에 UML을 작성하는 편집기가 존재하며 Stp/UML이라고 한다. 이 편집기로 작성 된 클래스도를 명세화하여 관련 된 클래스간의 매핑(mapping)관계를 테스트 데이타로 추출하는 명세기반 테스트(specification-based test)기법을 사용한다. 또한, 초당 1,000개의 테스트 데이타를 생성하므로 실시간 시스템에 적합하며 광고방송용 시스템 장비를 테스트하는데 주로 사용된다.

AUTEG의 경우 기존 도구들과는 달리 단위 테스트나 통합 테스트, 시스템 테스트 레벨에 모두 적용된다. UML의 순서도와 클래스도를 이용하여 시스템을 구성하는 모듈 사이의 인터페이스(interface)에 존재하는 오류 추출이 가능한 테스트도를 작성하고, 화이트 박스 테스트(white-box test)기법을 적용하여 시스템에서 일어나는 이벤트와 관련한 전이사항들을 테스트 데이타로 추출한다. 또한, 컴포넌트 기반 소프트웨어 개발(Component-Based Software Development: CBSD) 개념을 이용한 테스트 프로세스에 활용할 수 있기 때문에 이미 개발 된 컴포넌트를 새로운 컴포넌트에 맞추는 통합 작업을 진행하면서 동시에 일관되고, 안정된 테스트 데이타를 생성할 수 있으므로 시스템의 신뢰도를 높일 수 있다.

표 2 기존 UML 기반의 테스트데이타 도구와 AUTEG의 특징

Tool / 특징	ATTOL, Test SuitTM	Prosa/om, UML Case	Stp/T	AUTEG
테스트레벨	단위,시스템테스트	단위테스트	단위,시스템테스트	단위,통합,시스템테스트
테스트 데이타 선정근거	상태도로부터 객체의 상태전이를 발생시키는 이벤트가 발생 할 때마다 그 객체와 이벤트를 테스트 데이타로 생성	클래스도로부터 사용자가 선택한 클래스들간의 메소드 경로를 테스트 데이타로 생성	클래스도로부터 서로의 매핑관계를 명세화해서 테스트 데이타를 생성	클래스도, 순서도로부터 사용자가 정의한 테스트 대상의 그룹별로 테스트도 생성
테스트 데이타 선정기준	상태도와 메시지 기반 테스트기법을 적용	클래스도와 화이트 박스 테스트기법을 적용	클래스도와 명세 기반 테스트 기법을 적용	테스트도에 화이트 박스 테스트기법을 적용
특징/ 장점	· 테스트 스크립트를 자동 생성 · 테스트 수행 과정을 모니터링 · 내장형 시스템적합	· UML 다이어그램상에서 테스트 수행과정을 모니터링 · 실시간 환경에 적합	· 클래스들 간의 인터페이스 영역 오류 추출 · 초당 1,000의 테스트 데이타를 생성 · 실시간환경에 적합	· 정의한 테스트 대상의 인터페이스 영역 오류 추출 · 컴포넌트 기반 개발 소프트웨어 테스트에 적합

6. 결 론

소프트웨어의 확장성과 재 사용성 등의 객체지향 패러다임을 추구하고, 개발 환경이나 구현 언어의 한계를 극복하려는 현재 소프트웨어의 경향은 컴포넌트 기반의 소프트웨어 개발과 더 나아가 프로덕트라인의 개념이 등장하게 되었다. 이때 개발자가 다르고 개발환경이 다른 컴포넌트(통합 대상)들을 통합하는 것은 쉬운 일이 아니며, 이것은 통합 테스트의 어려움을 말한다. 또한 UML의 등장으로 UML기반의 개발 프로세스는 실세계에서 받아들여지고 있는 추세이나, UML의 개발 산출물을 직접 연결하여 테스트 데이타를 자동 생성할 수 있도록 테스트 프로세스를 고려하는 경우는 드물었다.

본 논문에서는 통합 단위들 사이의 인터페이스 영역에 존재하는 오류들을 추출하기 위해서 테스트도 및 테스트데이타 생성 자동화 방안을 제시하고, UML 개발 산출물인 순서도와 클래스도를 XML로 변환하여 이로부터 테스트도를 자동 생성하고, 이 테스트도에 화이트박스 테스트 기법을 적용하여 테스트 데이타를 자동 생성할 수 있도록 AUTEG 테스트 도구를 개발하였다. UML의 순서도, 클래스도로 부터 직접 테스트도를 자동 생성함으로 개발 분석 산출물과 직접 연계하여 테스트를 위한 분석과정 및 테스트 데이타 생성 과정을 체계화하고 일관성 있게 자동화할 수 있었다.

AUTEG에서 생성하는 테스트도는 노드와 메시지 흐름으로 구성하는 그래프이다. 이때 노드는 통합 대상을 나타내며, 이들 사이의 인터페이스, 즉 통합테스트 대상 단위로 전이를 일으키는 메시지는 메시지 흐름으로써 나타내진다. 따라서, 테스트도에 기존의 화이트박스 테스트 기법을 적용함으로써 통합테스트 테스트 데이타를 생성할 수 있도록 하였다. 노드는 통합 단위를 함축하고 있으므로 통합 단위의 그룹 범위를 자유로이 설정할 수 있는 장점을 갖는다. 또한 노드를 사용자 입력에서 시작하여 출력까지의 범위로 설정할 경우 시스템 테스트를 위한 테스트 데이타 생성이 가능하다.

향후에는 본 논문의 테스트도를 이용하여 보다 효율적인 테스트 데이타를 선정하기 위한 다양한 테스트 데이타 선정 기법의 적용이 요구된다. 본 논문에서 개발한 테스트 데이타의 선정 기준은 all-edge만을 적용했지만, 그래프 상의 노드와 에지를 응용하여 다양한 테스트 데이타 선정 기준을 개발할 필요가 있다. 현재 AUTEG는 클래스도 및 순서도 등의 개발 산출물을 입력으로 받게 되어 있다. AUTEG를 기존의 UML기반 개발 분석 및 설계 지원 도구에 통합하면 테스트 프로세스가 연계된 UML 기반 CASE 도구의 개발이 가능할 수 있다.

참 고 문 헌

[1] Bruce Powel Douglass, *Real time UML*, Addison-Wesley, 1998.

[2] Rational Software, "UML extension for Business Modeling," Sep, 1998.

[3] Rationa Software, "UML extension for objectory process for software engineering," sep, 1997.

[4] Desmond Francis D'Souza, *Objects, Components, and Frameworks with UML*, Addison-Wesley, 1998.

[5] ATTOL Testing Suite, http://www.wrs.com/products/html/attol_ds.html

[6] prosa/om UML Case, http://www.prosa.fi/prosausim.html

[7] The Process for Developing Software in UML with Stp/UML, http://www.aonix.com/content/ index.html

[8] Hoijin Yoon, Byoungju Choi, "UML-based Test Model for Component Integration Test," Workshop on Software Architecture and Component(WSAC), Japan, pp.63-70, Dec, 1999.

[9] Unisys/Rational XML Interchange Package v.4.0.1, ftp://ftp.rational.com/public/rose_extras/RoseXmi4.0.1.zip

[10] XML Parser for Java, http://www.alphaworks.ibm.com/tech/xml4j

[11] ftp://selab.ewha.ac.kr/

김 청 아

1991년 ~ 1995년 인천대 통신학과 학사. 1995년 ~ 1997년 LG소프트웨어. 1997년 ~ 1999년 순천향대 전산학과 학사. 1999년 ~ 2001년 이화여대 컴퓨터학과 석사. 관심분야는 소프트웨어공학, 소프트웨어테스트

최 병 주

1979년 ~ 1983년 이화여대 수학과 학사. 1984년 ~ 1985년 Purdue Univ. Computer Science 학사수료. 1986년 ~ 1987년 Purdue Univ. Computer Science 석사. 1987년 ~ 1990년 Purdue Univ. Computer Science 박사. 1991년 ~ 1992년 삼성종합기술원 1992년 ~ 1995년 용인대 전산통계학과 조교수. 1995년 ~ 현재 이화여대 컴퓨터학과 부교수. 관심분야는 소프트웨어공학, 소프트웨어 테스트, 소프트웨어 및 데이타 품질 측정