2001년 한국정보처리학회 춘계 학술발표논문집 제8권 제1호

메소드 매핑 관계와 상속 트리 분석을 통한 테스팅

∪최신형*, 한판암**
*경남발전연구원,**경남대학교 컴퓨터공학과
e-mail:cshinh@netian.com

Testing by the Method Mapping Relation and Analysis of Inheritance tree

Shin Hyeong Choi*, Pan Am Han*
Kyongnam Development Institute
Department of Computer Science and Engineering, Kyungnam

요 약

분산환경에서 CORBA는 컴포넌트 기반의 소프트웨어 개발을 위한 보다 쉬운 방법을 제공한다. 그러나, 컴포넌트 기반의 프로그램 테스트는 전통적인 분산 프로그램의 테스트보다 상당히 난해하다. 서버와 클라이언트 모두 다양한 컴포넌트들로 구성되어 있으며, 이들 대부분은 대응되는 컴포넌트에게 요구하거나, 상대 컴포넌트 요구를 받아 공급하는 역할을 한다.

본 논문은 클라이언트와 서버에 위치한 컴포넌트에 대한 테스트 방안을 제시한다. 즉, 클라이언트와 서버 측에 있는 컴포넌트들을 테스트하기 위해 컴포넌트들간 상호작용을 테스트하는데는 입출력 메소드 연결관계 정보 테이블을 포함하는 컴포넌트 어댑터를 이용하며, 컴포넌트 내에서는 상속 트리 분석을 통해 테스트한다. 이러 방법을 사용함으로써 불필요한 컴포넌트 테스트를 줄일 수 있다.

1. 서론

지난 수십 년 간 소프트웨어에 대한 사용자의 요구사항이 다양해짐에 따라 소프트웨어의 규모와 복잡성도 기하급수적으로 증가하였다. 그에 대한 대안으로 1980년대 이후 객체 기술 및 컴포넌트 기술이 제시되면서 이러한 문제들을 해결하고자 하였다[1,2]. 하지만 네트워크 환경하의 대규모 시스템이 주류를 이루는 현 상황에서는 컴포넌트 기반 개발방법이 각광받고 있다. 이는 컴포넌트 기반 소프트웨어가 현재로서는 대규모 어플리케이션 소프트웨어를 구축하기에 최적의 솔루션이며, 소프트웨어의 품질을 보증하고 재사용을 통한 소프트웨어 개발 생산성을 향상시키기 위한 방법으로서 중요하다는 것을 나타낸다.

컴포넌트 기반 개발방법론을 사용함으로써 소프트웨어를 부품화하고 이를 조립, 합성하여 어플리케이션을 개발하는 사이클이 단축될 수 있다[1,2]. 또한, 기존 소프트웨어 시스템을 업그레이드하고 다시 완전하게 하기 위해서는 시스템의 여러 부분들을 유연하고 호환성 있는 인터페이스를 가진 새로운 컴포넌트들로 대체함으로써 가능하다.

이와 같은 분산 시스템에서는 이질적인 소프트웨어 실행 환경과 하드웨어 플랫폼으로 인해 다양한 프로그래밍 언어, 운영체제, 통신 메커니즘, 인터페이스 등과 같은 여러 장벽이 개발에 어려움을 주며, 또한 다양한 변경 혹은 그에 따른 오류가 발생할 가능성이 더욱더 높아지므로 소프트웨어 신뢰성 향상과 품질 측정을 위한 테스트 작업이 필요하다.

본 논문에서는 인터페이스를 통해 통신하는 컴

포넌트 기반의 분산 객체 환경에서 컴포넌트 추가와 변경에 대한 테스트 방법으로서 컴포넌트간 테스트에는 컴포넌트 어댑터에 입출력 메소드 관계 테이블을 이용하며, 컴포넌트 내부에서는 상속 관계 테이블을 작성하여 테스트하는 방안을 제시한다.

2. 소프트웨어 컴포넌트

컴포넌트란 분산객체환경에서 시스템을 구성하는 기본 단위이다. 소프트웨어 컴포넌트는 완전한 형태로 존재하며, 하나 혹은 그 이상의 잘 정의된 인터페이스들을 가지는 소프트웨어의 부품이라 할 수 있다[3].

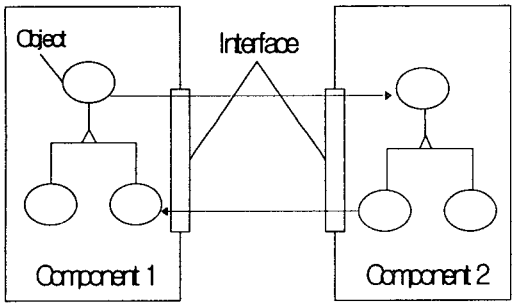

그림 1. 컴포넌트와 인터페이스

그림 1과 같이 외부에서 컴포넌트에 접근할 수 있는 유일한 방법은 인터페이스를 통하는 방법이며, 컴포넌트도 인터페이스를 거쳐야만 자신의 서비스를 외부에 공개할 수 있다. 그리고 실제 컴포넌트의 구현 부분은 한 개의 객체가 될 필요는 없다. 결국 여러 개의 객체가 하나의 컴포넌트를 이루고 이들 서비스가 인터페이스를 통해 외부에 공개될 수 있다.

3. 컴포넌트 테스트

이질적인 분산 환경 즉, 상이한 언어로 작성된 프로그램, 다른 운영체제와 하드웨어 플랫폼 사이의 호환성 문제에서 자유로운 CORBA는 사용되는 프로그래밍 언어, 운영체제와 네트워크 상에서 일관성 있는 분산 프로그래밍과 실시간 환경을 제공한다.

하지만, CORBA 환경에서 각 컴포넌트가 병렬 실행하는 동안 시스템 상태는 비결정적으로 변경될 수 있다. 또한, CORBA 어플리케이션이 다수의 분산 객체들을 인스턴스화하며, 각각의 객체는 독립적, 동시에 실행되므로 전통적 분산 프로그램과 같이 비결정적으로 실행된다[6].

이런 비결정성 때문에 분산 프로그램의 테스트는 순차 프로그램의 테스트보다 난해하다.

반면에, 컴포넌트는 외부와 인터페이스만 통해 동작하므로 외부와 철저히 단절되고 결과적으로 시스템의 단순성이 높아진다. 즉, 복잡성이 감소한다. 또한 컴포넌트는 인터페이스와 내부 구현이 엄격히 단절되므로 내부 구현이 바뀐다 하더라도 기존 인터페이스를 사용하던 클라이언트는 그대로 사용할 수 있다[1,4].

본 논문에서는 인터페이스를 통해 통신하는 클라이언트와 서버 측에 있는 컴포넌트의 테스트를 두 가지 측면, 컴포넌트간 테스트와 컴포넌트 내부 테스트로 구분한다.

3.1 컴포넌트간 어댑터

상호 작용하는 컴포넌트간 테스트를 위해 컴포넌트 기반의 분산 객체 환경에서 컴포넌트 추가와 변경에 대해 테스트할 수 있도록 컴포넌트 어댑터에 입출력 메소드 연결관계 정보 테이블을 포함시켜서 테스트하는 방안을 제시한다.

소프트웨어 컴포넌트들은 입력과 출력 인터페이스들의 연결로 구성되며, 컴포넌트 내부에 있는 객체의 입력 인터페이스는 다른 컴포넌트 내부의 객체들에 의해 호출될 수 있고, 출력 인터페이스는 다른 컴포넌트 내부의 객체 인터페이스를 호출한다.

그러나, 분산 컴포넌트에 대해서는, 컴포넌트를 구현할 때 서버들의 이름을 아는 것은 쉽지 않다. 보통 이런 정보는 소프트웨어 개발자가 컴포넌트들의 출력 메소드를 서버 컴포넌트의 입력 인터페이스에 연결할 때 이용 가능하다.

이와 같이 컴포넌트의 객체들간 호출관계로 구성된 컴포넌트 어댑터의 기능에 객체들의 입출력 메소드 연결관계 테이블을 포함시켜 컴포넌트간 테스트에 이용한다. 컴포넌트 어댑터를 사용함으로써, 분산 컴포넌트의 모든 출력 메소드는 이들 메소드들이 입력 메소드나 이벤트의 동일한 시그니처를 가지는 한, 다른 인터페이스를 가진 다른 서버 컴포넌트에 연결된다. 이것은 컴포넌트 설계자들로 하여금 더 많은 융통성을 주며, 기존 컴포넌트의 재사용성을 늘릴 수 있는 점이다[5].

본 논문에서는 분산 객체 환경에서 클라이언트의 컴포넌트 요구를 서버 측 컴포넌트에 매핑 시키는 역할을 하는 컴포넌트 어댑터에 입출력 메소드 연결관계 테이블을 정의함으로써 컴포넌트 변경이 발생하는 환경에서 클라이언트와 서버 측의 컴포넌트에 대한 테스트를 실시할 때 연결관계가 있는 컴포넌트들을 함께 테스트할 수 있고, 불필요한 컴포넌트 테스트를 감소시킬 수 있다.

입출력 메소드 연결관계 테이블은 클라이언트 측의 출력 메소드가 특정 서버의 입력 메소드에 연결된다는 것을 레코드 형식으로 나타낸다.

컴포넌트 어댑터는 그림 2에서 보이듯이 클라이언트와 서버 컴포넌트 사이를 연결시켜주며, 대리자로서의 역할을 한다. 또한 매번 컴포넌트 어댑터에 등록된 모든 컴포넌트를 테스트하는 것은 부하가 많이 걸리므로 기존에 테스트되고 변경되지 않은 컴포넌트에 대해서는 하나의 필드에 체크를 함으로써 불필요하게 모든 컴포넌트를 테스트하는 것을 줄일 수 있다. 즉, 입출력 메소드 연결관계 테이블은 다음과 같은 포맷을 따른다.

i_mod	incoming_ method	target_ server	o_mod	outgoing_ method
0/1			0/1	

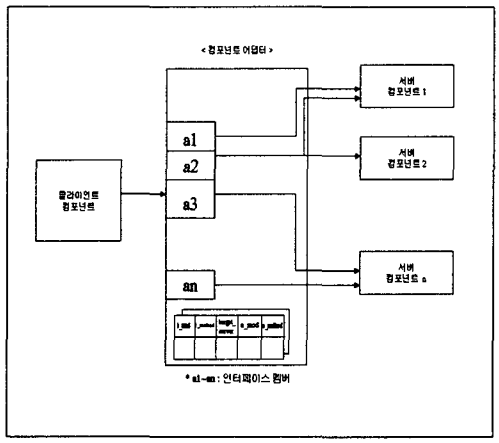

그림 2. 컴포넌트 어댑터

이와 같이 컴포넌트 어댑터에 입출력 메소드 연결관계 테이블을 정의함으로써 서로 관련이 있는 컴포넌트를 선별해서 테스트할 뿐 아니라, 변경이 발생하지 않은 컴포넌트에 대해서는 테스트를 하지 않아도 된다. 즉, 컴포넌트 어댑터를 사용하면 컴포넌트의 변경이 발생할 수 있는 환경에서 클라이언트와 서버 측의 컴포넌트에 대한 테스트를 따로 실시할 필요 없이 한 곳에서 테스트를 할 수 있으므로 일관성 있고 전체적인 테스트를 실시할 수 있다.

그림 2에서 보이듯이 컴포넌트 어댑터는 stub와 달리 입력 인터페이스의 각 멤버가 하나 혹은 그 이상의 서버 컴포넌트에 사상된다. 이런 사상은 변경이 발생할 때마다 클라이언트를 변경하지 않고 동적으로 입출력 메소드 사이의 관계를 레코드 형식으로 구성함으로써 가능하다.

컴포넌트 어댑터에 클라이언트와 서버 컴포넌트들 사이의 인터페이스 사상 관계 즉, 입출력 메소드 관계를 나타낸 레코드를 주기적으로 구성해 놓으면 어느 한 쪽의 컴포넌트에 변경 내지 오류가 발생하더라도 전체 컴포넌트를 테스트하지 않아도 되고, 또한 한 곳에서 동적으로 테스트를 제

어할 수 있다.

컴포넌트 어댑터의 기능을 이용하여 컴포넌트 테스트를 할 때 선행되어야 할 것이 바로 클라이언트 컴포넌트와 서버 컴포넌트 사이의 입, 출력 인터페이스 관계를 주기적으로 구성해 놓는 것이다.

3.2 컴포넌트 내부 객체 테스트

그림 1에서처럼 컴포넌트 내부에는 각 객체들의 메소드가 존재하며, 객체간의 상속관계와 더불어 메소드 간에도 순서 형태 즉, 상속관계가 존재한다.

이와 같이 컴포넌트 내부를 테스트할 때 상속트리분석을 통해 상속관계 테이블을 구성해놓음으로써 컴포넌트가 테스트되어야 할 때 그 컴포넌트 내부에 있는 객체의 상속관계를 고려하여 해당 객체가 테스트되어야 하는지를 파악하여 테스트된 객체로부터 상속받은 메소드를 제외하고 테스트함으로써 불필요한 객체의 테스트를 줄일 수 있다.

4. 결 론

본 논문에서는 인터페이스를 통해 통신하는 클라이언트와 서버 측에 있는 컴포넌트의 테스트를 두 가지 측면으로 구분한다.

첫째, 상호 작용하는 컴포넌트간 테스트를 위해 컴포넌트 기반의 분산 객체 환경 하에서 컴포넌트 추가와 변경에 대해 테스트할 수 있는 방법으로서 컴포넌트 어댑터에 입출력 메소드 연결관계 정보 테이블을 포함시켜서 테스트한다.

컴포넌트 어댑터를 사용하면 컴포넌트의 변경이 발생할 수 있는 환경에서 클라이언트와 서버 측의 컴포넌트에 대한 테스트를 따로 실시할 필요없이 한 곳에서 테스트를 할 수 있으므로 일관성있고 전체적인 테스트를 실시할 수 있다.

둘째, 컴포넌트 내부를 테스트할 때는 상속관계 테이블을 구성해 놓음으로써 해당 객체의 테스트가 필요할 때 상속관계를 파악하여 테스트된 객체로부터 상속받은 메소드에 대해서는 테스트를 피함으로써 불필요한 객체의 테스트를 줄일 수 있다.

이와 같이 클라이언트와 서버 측에 있는 컴포넌트들을 테스트하기 위해 컴포넌트들간 상호작용을 테스트하는데는 입출력 메소드 연결관계 정보 테이블을 포함하는 컴포넌트 어댑터를 이용하며, 컴포넌트 내에서는 상속 트리 분석을 통해 테스트하는 방법을 제시한다. 이런 방법을 사용함으로써 불필요한 컴포넌트와 내부의 객체 테스트를 감소시킬 수 있다.

참고문헌

[1] Kozaczynski, W. and Booch, G., "Component-Based Software Engineering", IEEE Software, 1998

[2] Stephen S. Yau and Bing Xia,"Object-Oriented Distributed Component Software Development based on CORBA", IEEE, 1998

[3] Sarah Brown, Richard Vallas, "Object Design of a Distributed Client/Server System", IEEE, 1998

[4] 이상구외 3인 편역, "CORBA & Java 분산객체 기술", 교학사, 1998

[5] Holger D. Hofmann, Jeanne Stynes, "Implementation Reuse and Inheritance in Distributed Component Systems", IEEE, 1998

[6] Hwan Wook Sohn, David C.Kung, and Pei Hsia, "State-based Reproducible Testing for CORBA Applications", IEEE, 1999

[7] 최정은, 최병주, "에이전트 기반의 객체지향 소프트웨어 테스트 방안", 정보과학회논문지 제27권 11호, 2000.11