

에이전트 기반의 컴포넌트 테스트 방안

○최신형*, 한판암**

*경남발전연구원, **경남대학교 컴퓨터공학과
e-mail:cshinh@korea.com

Agent Based Component Test Technique

Shin Hyeong Choi*, Pan Am Han*

Kyongnam Development Institute
Department of Computer Science and Engineering, Kyungnam

요약

본 논문에서는 분산객체환경에서 추가와 변경이 반복되는 컴포넌트들을 테스트하기 위해 입출력 메소드 관계 테이블을 포함하는 에이전트 기반의 컴포넌트 테스트 방안을 제시한다.

테스트 에이전트 내에 컴포넌트의 객체들간 입출력 메소드 연결관계 테이블을 포함시키고, 컴포넌트들의 추가와 변경이 발생할 때마다 Sensor로부터 받아들인 컴포넌트에 대해 입출력 메소드 관계 테이블과의 비교분석과정을 거쳐 테스트 여부를 결정하고, 테스트가 필요한 컴포넌트에 대해서는 지식영역 라이브러리에 명시한대로 테스트를 실시함으로써, 불필요한 컴포넌트를 테스트하는 것을 감소시킬 수 있다.

1. 서론

1980년대 중반까지 컴퓨팅 환경은 다수의 사용자가 사용하기 위해서는 대규모, 단일 메인프레임 시스템이 주를 이루었다. 하지만 이런 메인프레임 환경은 다음과 같은 두 가지 큰 문제점을 가지고 있었다. 즉, 중앙 집중식 단일 시스템으로서는 너무 규모가 커지며, 병렬처리, 시스템 확장, 자원공유에는 유연성이 떨어진다는 것이다. 이런 문제를 해결하기 위해 클라이언트-서버 환경이 제안되었으나, 시스템 규모가 더 커짐에 따라 시스템 구조, 유지보수, 확장성에 있어서 한계에 도달했다. 그 대안으로서 분산객체 환경이 제안되었다[3].

특히, 네트워크 환경하의 대규모 시스템이 주류를 이루는 현 상황에서는 컴포넌트 기반 개발방법이 각광받고 있다. 이는 컴포넌트 기반 소프트웨어가 현재로서는 대규모 어플리케이션 소프트웨어를 구축하

기에 최적의 솔루션이며, 소프트웨어의 품질을 보증하고 재사용을 통한 소프트웨어 개발 생산성을 향상시키기 위한 방법으로서 중요하다는 것을 나타낸다.

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 컴포넌트 기반의 분산 객체 환경에서 컴포넌트 추가와 변경에 따른 변화를 인식하여 테스트하는 에이전트 기반의 테스트 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[9].

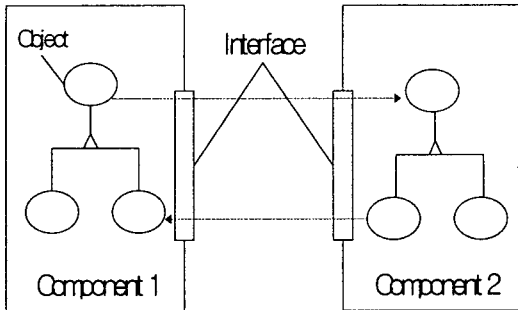


그림 1. 컴포넌트와 인터페이스

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. 에이전트

에이전트는 지속적으로 환경에서 지각된 것과 내부 지식을 바탕으로 추론하며 그 결과 행동하여 환경에 영향을 미치고 또한 사용자를 포함한 다른 에이전트와 의사 소통하는 지속적으로 존재하는 소프트웨어 요소라고 정의할 수 있다[10].

현재 많은 수의 컴퓨터 시스템이 자동화된 에이전트를 도입하고 있고, 튜링머신 보다 현재의 컴퓨팅 환경을 더 현실성 있게 반영하는 새로운 모델로서 각광받고 있으며, 분산 시스템을 위한 차세대 모델

로서도 지지를 받고 있다. 또한 에이전트는 컴포넌트와 함께 인공지능분야를 이끌어갈 프레임워크로 사용되고 있으며, 지능적 엔티티를 설계하고 구축하기 위해 필요하다.

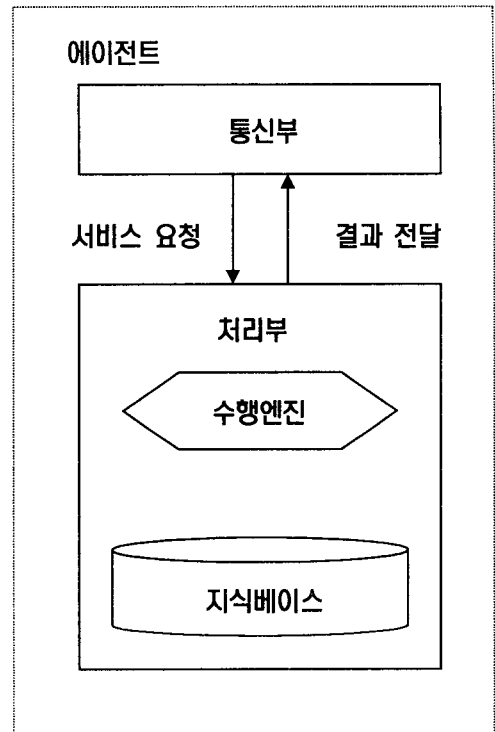


그림 2. 에이전트 구조

에이전트는 여러 가지 특징이 있지만 Autonomy, Adaptation, Cooperation 이 대표적인 특징이라 할 수 있다. 즉, Autonomy(자율성)는 에이전트가 인간의 직접적 간섭 없이 자신의 상태에 기초하여 행동한다는 것이고, Adaptation(적응성)은 에이전트가 환경을 인지하여 그 환경의 변화에 반응하는 것이며, Cooperation(협력성)은 에이전트가 다른 에이전트들과 상호 작용할 수 있다는 것을 말한다.

4. 컴포넌트 기반 소프트웨어

컴포넌트 기반 소프트웨어는 대규모 응용 소프트웨어를 개발하는데 적합한 방법이며, 응용소프트웨어

어를 개발하기 위해 잘 정의된 소프트웨어 부품을 재사용함으로써 소프트웨어 개발 생산성은 매우 향상될 수 있다.

그러나, 분산객체환경에서는 언제, 어디에서 컴포넌트가 변경 혹은 추가·삭제될 지 예측이 불가능할 뿐 아니라 수많은 컴포넌트를 관리한다는 것은 상당히 어려운 작업이다. 이와 같이 분산시스템에서는 이질적인 소프트웨어 실행 환경과 하드웨어 플랫폼으로 인해 다양한 프로그래밍 언어, 운영체제, 통신 매커니즘, 인터페이스 등의 여러 장벽이 개발에 어려움을 주며, 또한 다양한 변경 혹은 이로 인해 발생하는 오류를 발견하는 것은 쉽지 않다.

4.1 컴포넌트 테스트

분산객체환경에서 각 컴포넌트가 병렬 실행하는 동안 시스템 상태는 비결정적으로 변경될 수 있다. 즉, 각종 어플리케이션이 다수의 분산 객체들을 인스턴스화하며, 각각의 객체는 독립적, 동시에 실행되므로 전통적 분산 프로그램과 같이 비결정적으로 실행된다[11].

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[4,6].

본 논문에서는 분산객체환경에서 끊임없이 변화, 추가, 삭제되는 컴포넌트를 테스트하기 위해 Autonomy, Adaptation, Cooperation이라는 특징을 가지는 에이전트에 의한 테스트 방안을 제시한다.

4.2 테스트 에이전트

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

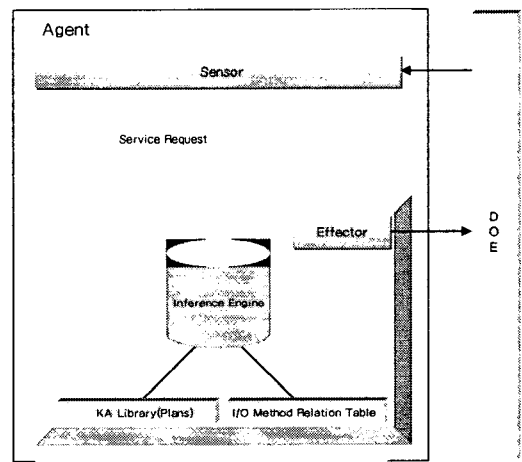


그림 3. 테스트 에이전트 구조

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.

이와 같이 테스트 에이전트 내에 컴포넌트의 객체들의 입출력 메소드 연결관계 테이블을 포함시키고, 분산객체환경에서 컴포넌트들의 추가와 변경이 발생할 때마다 Sensor로부터 받아들인 컴포넌트에 대해 입출력 메소드 관계 테이블과의 비교분석과정을 거쳐 테스트 여부를 결정하고, 테스트가 필요한 컴포넌트에 대해서는 KA(지식영역) Library에 명시한대로 테스트를 실시한다. 즉, 매번 입력받은 컴포넌트들을 테스트한다는 것은 시스템 전체에 부하를 많이 줌으로 기존에 테스트되고 변경되지 않은 컴포넌트

에 대해서는 하나의 필드에 체크를 함으로써 불필요하게 모든 컴포넌트를 테스트하는 것을 줄일 수 있다. 즉, 입출력 메소드 연결관계 테이블은 다음과 같은 포맷을 따르며, 클라이언트 측의 출력 메소드가 특정 서버의 입력 메소드에 연결된다는 것을 레코드 형식으로 나타낸다.

i_mod	incoming_m ethod	target_ server	o_mod	outgoing_m ethod
0/1			0/1	

본 논문에서는 분산재채환경에서 클라이언트의 컴포넌트 요구와 그에 대응하는 서버 측 컴포넌트들의 입출력 메소드 연결관계 테이블을 정의하여 테스트 에이전트에 포함시킴으로써 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.

5. 결 론

본 논문에서는 분산재채환경에서 추가와 변경이 반복되는 컴포넌트들을 테스트하기 위해 입출력 메소드 관계 테이블을 포함하는 에이전트 기반의 컴포넌트 테스트 방안을 제시하였다.

테스트 에이전트 내에 컴포넌트의 객체들간 입출력 메소드 연결관계 테이블을 포함시키고, 컴포넌트들의 추가와 변경이 발생할 때마다 Sensor로부터 받아들인 컴포넌트에 대해 입출력 메소드 관계 테이블과의 비교분석과정을 거쳐 테스트 여부를 결정하고, 테스트가 필요한 컴포넌트에 대해서는 KA(지식영역) Library에 명시한대로 테스트를 실시함으로써, 불필요한 컴포넌트를 테스트하는 것을 감소시킬 수 있다.

참고문헌

- [1] Nicholas R. Jennings, "On agent-based software engineering", Artificial Intelligence, 1999
- [2] Stephen S. Yau and Bing Xia, "Object-Oriented Distributed Component Software Development based on CORBA", IEEE, 1998
- [3] Nwana, H.S. Software Agents: An Overview, Knowledge Engineering Review, 11(3), pp. 205-244, 1996
- [4] 이상구의 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] Kozaczynski, W. and Booch, G., "Component-Based Software Engineering", IEEE Software, 1998
- [7] 최정은, 최병주, "에이전트 기반의 객체지향 소프트웨어 테스트 방안", 정보과학회논문지 제27권 11호, 2000.11
- [8] Sooyoung Park 외 4, "Agent-based Software Analysis Method in Distributed Environment", IEEE, 1999
- [9] Stephen S. Yau and Bing Xia, "Object-Oriented Distributed Component Software Development based on CORBA", IEEE, 1998
- [10] Stuart Russel and Peter Norving. "Artificial Intelligence: A Modern Approach", Prentice-Hall, 1995
- [11] Hwan Wook Sohn, David C.Kung, and Pei Hsia, "State-based Reproducible Testing for CORBA Applications", IEEE, 1999