컴포넌트 개발에서의 검증에 관한 연구

홍동택° 김병기
전남대학교 전산학과
{dthong°, bgkim}@chonnam.ac.kr

Study on Verification of Component Development technique

Dong-Taek Hong° Byung-Ki Kim
Dept. of Computer Science, Chonnam National University

요 약

CORBA나 DCOM과 같은 개발환경을 기반으로 한 컴포넌트 기반 개발(CBD : Component Based Development)은 광범위한 소프트웨어 재사용을 유도한다. 하지만 컴포넌트의 개발 과정 중 명세 및 설계 단계에서 그 컴포넌트가 과연 실행이 가능한 것인지, 오류는 없는지를 명확하게 알 수 있는 방법이 없다. 컴포넌트의 개발 초기 단계에서 그에 대한 검증의 과정을 통하여 논리적 설계 오류를 초기에 피드백할 수 있는 기회를 얻을 수 있을 뿐만 아니라 개발하고자 하는 컴포넌트의 품질을 보장할 수 있게된다.

본 논문에서는 비동기 시스템 컴포넌트 소프트웨어 개발에서 개발의 초기에 발생될 오류를 발견하기 위해 검증 언어인 PROMELA(Process Meta LAnguage)로의 변환과 LTL의 제약사항을 적용하여, SPIN을 통해 명세의 일관성을 검사하고 정확성을 검증하는 방법을 제안함으로써 생산비를 감소시키고 안정성과 정확성 그리고 신뢰성을 가진 컴포넌트 소프트웨어를 개발할 수 있다.

1. 서 론

구조와 행위를 캡슐화한 컴포넌트를 기반으로 한 개발은 상호 작용하는 컴포넌트들을 조립하는 과정으로 볼 수 있다. 그러나 아무리 발전된 기술을 사용한다 할지라도 소프트웨어 컴포넌트를 시스템으로 통합하는 것은 쉬운 일이 아니며, 또한 개발되는 컴포넌트의 안정성과 실용성에 대한 확신이 필요하다는 것이다.

기존의 컴포넌트 개발에 있어서는 컴포넌트 적용이나 사용이 용이하면서도 정형 검증이 가능한 컴포넌트 개발을 위한 방안이 요구되며, 이를 이용하여 개발의 초기 단계에서 논리적 설계 오류를 발견하고 수정하는 것은 무엇보다 중요하다.

컴포넌트의 검증을 위해서 SPIN이라는 검증도구를 이용하여 컴포넌트 개발 초기에 향후의 오류발생을 최대한 줄일 수 있게 된다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 본 논문과 관련되어 살펴본 컴포넌트에 대한 정의와 개발 환경을 소개하고, 3장은 컴포넌트 개발을 위한 명세기법과 검증언어인 SPIN, 그리고 SPIN의 입력언어인 PROMELA를 소개하였으며, 4장에서는 컴포넌트 개발에서 SPIN을 통한 검증을 살펴본 후 5장에서 SPIN 적용이 컴포넌트 개발에 미치는 영향을 다루고, 마지막으로 연구에 대한 결과와 향후 연구 방향에 대해 언급함으로써 마무리한다.

2. 컴포넌트

2.1 정의

컴포넌트에 대한 정의는 다양하다. Rational 사의 Philippe Krutchen은 "컴포넌트는 잘 정의된 아키텍처 상에서 어떠한 기능을 수행하는 시스템의 독립적이면서 대치 가능한 부분이다. 컴포넌트는 인터페이스들의 집합에 대한 물리적인 구현을 제공한다"라고 정의한다[1].

Gartner 그룹은 "컴포넌트는 동적으로 바인드 할 수 있는 하나 이상의 프로그램들을 하나의 단위로 관리하는 패키지로서, 실행시간에 인터페이스를 통해 접근이 가능하다"라고 정의하고 있다[1].

Desmond Francis D'Souza는 "컴포넌트는 소프트웨어 구현을 갖는 패키지이다"라고 정의하였다.[1][2].

2.2 CBSE

컴포넌트 기반 소프트웨어 개발(Component-based Software Development, 이하 CBSD라 함)이란 독립적인 기능을 담당하는 다양한 컴포넌트 소프트웨어의 집합으로부터, 해당 업무의 수행에 필요한 기능을 담당하는 하나 이상의 컴포넌트를 결합하여 해당 업무를 위한 소프트웨어를 개발하는 기술을 말한다[3].

현재 주목받고 있는 소프트웨어 컴포넌트 기술 중 적용 절차, 특히 소프트웨어 생명주기 관점에서의 CBSD 절차는 요구 분석, 컴포넌트 기반 설계, 소요 컴포넌트의 획득, 소프트웨어 조립, 수행 및 평가, 유지 보수로 구성된다.

3. 검증

3.1 정형명세기법

최근 들어 컴포넌트 및 아키텍처 명세의 표기법으로 OMG(Object Management Group)에서 제정한 UML(Unified Modeling Language)를 사용하려는 연구가 활발히 일어나고 있다. 이는 UML이 제공하는 시각적 표기법을 통한 명세의 가독성과 표준화 등의 장점에 기인한 것이다. UML 표기법을 사용하고 있는 컴포넌트 기반 개발 방법론으로는 대표적으로 미국의 Icon Computing 사의 Catalysis와 Select Software사의 Select 등을 들 수 있다[4].

3.2 SPIN

SPIN은 AT&T Bell 연구소에서 1995년에 발표한 LTL model checker이다. SPIN은 비동기 프로세스 시스템을 설계하고 검증하는데 사용되는 모델체커이며, 프로세스 상호작용(Process interaction)의 정확성에 초점을 둔다[5].

(그림 1)은 SPIN의 인터페이스를 나타낸다. 입력언어를 불러들여 검증과 시뮬레이션을 수행한다.

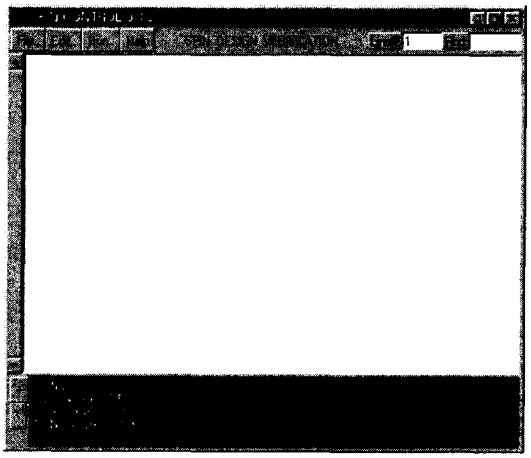

그림 1 SPIN Control

SPIN은 분산 시스템에서의 논리적 설계 오류를 찾는데 사용될 수 있다. 특히 운영체제, 데이터통신 프로토콜, 교환시스템, 동시성 알고리즘, 기차선로 신호 프로토콜 등 실제적인 시스템의 디자인에서의 검증에 매우 적합한 것으로 평가되고 있다. SPIN을 이용해서 명세의 일관성을 검사할 수 있는데 데드락, 명세 되지 않은 응답, 경쟁조건 등을 잘 보여준다[5]. 실제적인 시스템의 검증에 SPIN이 쓰이는 이유는 시스템의 오동작 등으로 인하여 인명과 재산이 치명적인 피해를 입힐 수 있는 시스템을 검증하여 이러한 오류를 개발 초기에 발견하며, 치명적인 오류를 수정하는 것이다.

SPIN은 PROMELA(Process Meta LAnguage)라는 검증 언어를 사용하여 시스템 명세를 작성하고, PROMELA는 LTL(Linear Temporal Logic)의 문법으로 명세화하여 정확도를 증명한다[6]. SPIN은 LTL의 형태로 받아들여진 어떤 특성들을 부치 오토마타의 형태로 변환시킨 후 이를 다시 PROMELA의 형태로 변환시켜 명세된 시스템과 병렬로 수행을 시킨다. 이때 검증기를 생성하여 그 결과를 분석하면 명세된 시스템이 요구되는 특성을 만족하는지를 분석할 수 있다.

3.3 PROMELA(PROcess MEta LAnguage)

PROMELA는 SPIN에 사용되는 검증 모델 언어이다. 분산 시스템이나 프로토콜의 추상모델을 제공하고 프로세스, 메시지 채널, 그리고 변수로 구성되어 있다.

PROMELA는 식별자, 키워드, 상수, 연산자로 구성되어 있으며, 상수정의는 정수 형태만이 가능하며 C에서와 같이 macro를 사용하여 선언하는 방법과 PROMELA에서의 키워드인 mtype을 사용하여 선언하는 방법이 있다. 그 구조는 C와 매우 유사하여 구현과 오류수정이 매우 간단한 것이 장점이다[6].

프로세스, 채널, 변수 등은 사용되기 이전에 선언되어야 한다. 변수와 채널은 프로세스 내에서 지역적으로 선언되거나, 전역적으로 선언 가능하다. 그러나 프로세스는 proctype 키워드를 이용해서 전역적으로 선언되어야 한다.

3.4 LTL(Linear Temporal Logic)

SPIN은 PROMELA(PROcess MEta LAnguage)라는 검증 언어를 사용하여 설계를 하고 PROMELA는 LTL의 구문으로 명세화하여 정확성을 검증한다.

PROMELA는 전체적인 시스템을 논리식을 이용하여 프로세스들을 감시하는 유한 상태 기계(Finite State Machine)를 기술할 수 있다. 이는 LTL로 시스템의 제약사항이나 불변의 원리, 그리고 선행조건등을 기술하여 교착상태나 기아상태 등을 검사할 수 있다.

4. 컴포넌트 개발에서의 SPIN 적용

기존의 컴포넌트 개발에서는 컴포넌트 적용이 용이하면서도 개발의 초기단계에서 논리적 설계 오류를 발견하고 수정하는 것이 개발의 효율성을 높일 수 있을 뿐만 아니라 고품질의 컴포넌트 개발을 위해서도 중요하다.

초기단계인 분석,설계단계에서의 산출물중의 하나인 UML Diagram에서 표기의 형태와 의미를 추출하여 같은

의미를 갖는 PROMELA 문법으로 표기하면 <표1>과 같다. PROMELA 문법 표기를 적용하여 코드를 작성한 후,

표 1 PROMELA 표기법

●	{	시작을 의미
Activity	run pname(*parameters*)	클래스의 메소드는 함수와 같은 의미로 수행
◇	if [:: *statements*]* fi	선택에 의한 분기
━	do [:: *statements*]* od	병렬성을 나타내는 동기화 막대로 이 지점이후로는 병렬적으로 수행되는 것을 의미
→	*step* [; *step*]*	명령의 순서를 나타낸다. 단, 동기화막대에서 나오는 표기는 순서의 의미가 없다
◉	}	종료를 의미

전체적인 시스템이 가지고 있는 제약사항등을 기술한 LTL을 작성한 후, LTL이 삽입된 PROMELA 코드를 SPIN 인터페이스에 입력하여 검증하게 된다. (그림 2)는 SPIN Control에 PROMELA 코드를 불러와 검증하는 화면이다. 검증과정을 거쳐 검증의 결과를 가지고 다시 문제영역을 통하여 전체적인 개발에서의 치명적 오류가 있는지 검증하게 된다.

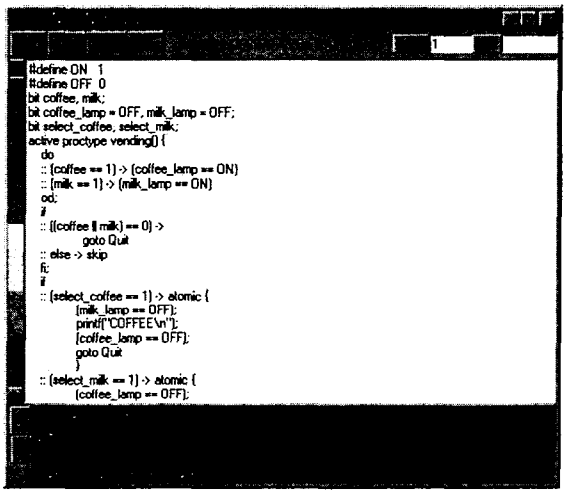

그림 2 SPIN Control 적용

5. SPIN 적용이 컴포넌트 개발에 미치는 영향

컴포넌트 개발의 초기단계에서 SPIN을 적용한 치명적 오류 검증은 피드백과 다음 단계로의 전이 결정을 통하여 보다 빠르고 정확한 소프트웨어를 개발할 수 있다. 개발의 마지막 단계인 테스트 단계에서 오류가 발견된 경우 지금까지 들였던 개발비용과 인력의 비용은 허사가 된다. 그러므로, 검증을 통한 컴포넌트 개발은 안정성과 정확성에서 보다 나은 결과물을 기대할 수 있다.

SPIN을 적용한 오류 검증은 통하여 개발된 컴포넌트는 정상적으로 동작함을 보증하게 되므로 보다 안정된 시스템 구현이 가능하게 된다.

6. 결론

본 논문에서는 컴포넌트 소프트웨어 개발에서 개발의 초기에 발생될 오류를 발견하여 생산비를 감소시키고 안정성과 정확성 그리고 신뢰성을 가진 컴포넌트 소프트웨어를 개발하기 위하여 SPIN의 입력언어인 PROMELA의 문법을 서로 변환시키고, 시스템의 제약사항을 기술한 LTL을 만들어 PROMELA source code로 변환하고, 변환된 code를 입력언어로 사용하는 SPIN을 이용하여 명세의 일관성을 검사하고 정확성을 검증하는 과정을 연구하였다.

7. 참고문헌

[1] Alan Brown, "From Component Infrastructure to Component- Based Development," ICSE, 1998.

[2] Desmond Francis D'Souza, Alan Cameron Wills, 'Objects, component, and frameworks with UML: the Catalysis approach,' Addison Wesley Longman, Inc, 1999.

[3] Butler Group, "Component-Based Development : Application Delivery and Integration Using Component Software," 1998.

[4] Desmond Francis D'Souza, Alan Cameron Wills, 'Objects, component, and frameworks with UML: the Catalysis approach,' Addison Wesley Longman, Inc, 1999.

[5] G.J.Holzmann, "The Model Checker SPIN," IEEE Transactions on Software Engineering, VOL.23, NO 5, MAY, 1997.

[6] 최진영, 방기석, 유혁, 허윤정, "Holzmann의 "The Model Checker SPIN"에 대하여," 한국 정보처리학회 추계 학술발표 논문집 제25권 제11호, 1998.