

그레이박스를 사용한 컴포넌트의 관심사 분리 보안 모델

김 영 수*, 조 선 구**

Separation of Concerns Security Model of Component using Grey Box

Young Soo Kim *, Sun Goo Jo **

요 약

컴포넌트에 대한 의존도 및 활용도가 증가하면서 컴포넌트의 보안성 강화를 위한 필요성이 증가하고 있다. 컴포넌트는 재사용을 통한 소프트웨어의 개발 생산성을 향상시키는 이점을 제공한다. 이러한 이점에도 불구하고 컴포넌트의 보안 취약성은 재사용에 제한을 한다. 이의 개선을 위해 컴포넌트의 보안성을 높이는 경우에 가장 문제가 되는 부분이 재사용성에 대한 제한이 확대된다는 것이다. 따라서 컴포넌트의 재사용성과 보안성을 동시에 고려하는 컴포넌트의 모델이 제공되어야 한다. 이의 해결책으로 정보오프와 수정의 용이성을 제공하여 보안성과 재사용을 확대할 수 있도록 재사용 모델을 결합하고 포장 및 애스펙트 모델을 통합한 컴포넌트 재사용 확대를 위한 관심사의 분리 보안 모델을 제안하고 응용시스템을 구축하여 모델의 적합성을 검증하였다. 이의 응용은 핵심 및 보안 관심사의 분리를 통한 컴포넌트 기능의 확장과 수정의 용이성을 제공함으로써 보안성을 높이는 동시에 재사용성을 확대한다.

Abstract

As the degree of dependency and application of component increases, the need to strengthen security of component is also increased as well. The component gives an advantage to improve development productivity through its reusable software. Even with this advantage, vulnerability of component security limits its reuse. When the security level of a component is raised in order to improve this problem, the most problematic issue will be that it may extend its limitation on reusability. Therefore, a component model concerning its reusability and security at the same time should be supplied. We suggest a Separation of Concerns Security Model for Extension of Component Reuse which is integrated with a wrapper model and an aspect model and combined with a reuse model in order to extend its security and reusability by supplying information hiding and easy modification, and an appropriate application system to verify the model's compatibility is even constructed. This application model gives the extension of component function and easy modification through the separation of concerns, and it raise its security as well as extends its reusability.

▶ Keyword : 관심사분리(Separation of Concerns), 보안모델(Security Model), 컴포넌트재사용(Component Reuse), 그레이박스(Grey Box),

• 제1저자 : 김영수
• 접수일 : 2008. 8. 13, 심사일 : 2008. 8. 20, 심사완료일 : 2008. 9. 25.
* 충북대학교 컴퓨터공학과 포닥 ** 나사렛대학교 호텔관광경영학과교수

1. 서론

인터넷의 보편화와 더불어 이기종 컴퓨터간 연동기술의 발전으로 컴포넌트기술은 소프트웨어 개발의 적시성 및 생산성을 향상시키고 규격화 및 표준화를 촉진한다. 이러한 장점에도 불구하고 컴포넌트는 기본적으로 버그나 의도적 또는 비의도적인 악성코드가 내재하고 있는 경우에 정보보호에 취약할 수 밖에 없으므로 컴포넌트에 보안성이 요구된다[1]. 컴포넌트로부터 보안성을 제공받기 위해서는 컴포넌트 내부에 보안코드를 추가해야 한다. 이로 인해서 컴포넌트의 기본 기능과 보안기능이 서로 얽혀버리는 문제가 발생한다. 이러한 문제점을 때문에 컴포넌트의 재사용이 제한된다. 따라서 재사용성을 확대하고 보안성을 향상시킬 수 있는 방법이 필요하다. 이의 해결책으로 컴포넌트의 기본 기능과 보안 기능을 분리하여 컴포넌트한 후 통합해서 컴포넌트를 개발하는 방법이 있다. 이 방법은 보안기능을 기본기능의 내부에 주입하는 방식으로 컴포넌트의 세부 내용의 이해와 빌딩블록의 통일된 명명규칙을 필요로 한다. 그러나 컴포넌트는 일반적으로 블랙박스 형태로 제공되기 때문에 재사용이 제한되므로 컴포넌트를 적절히 수정하여 사용하고 컴포넌트의 세부 내용을 자세히 알 수 있는 화이트박스 형태의 컴포넌트로 전환할 수 있는 메커니즘이 필요하다. 컴포넌트는 컴포넌트 내부의 세부 구현사항들을 외부에 노출시키지는 않지만 보안성을 추가할 수 있도록 컴포넌트가 지니는 속성이나 메서드의 동작을 이해하고 변경할 수 있도록 지원되어야 한다[2]. 이를 위하여 본 논문에서는 블랙박스 형태의 컴포넌트 모델에 관심사의 분리 보안모델을 수용한 컴포넌트의 재사용 확대 보안 모델을 제안하고 이를 사용한 컴포넌트 기반 시스템을 구현하여 모델의 실용성을 검증하였다.

본 논문은 다음과 같이 구성한다. 2장에서는 본 논문의 모델과 응용의 이해를 위해 컴포넌트의 일반 모델과 정보은폐 모델을 간략하게 소개한다. 3장에서 컴포넌트에 보안성을 추가하기 위한 컴포넌트 보안 모델로 관심사의 분리 보안 모델과 보안 컴포넌트의 수정성 모델을 분석하고 4장에서는 보안성과 재사용성 향상을 위한 수정 가능한 컴포넌트의 관심사 분리 보안모델을 도출하고 모델의 적합성을 검증하기 위하여 컴포넌트 기반 응용시스템을 구현한다. 5장에서는 결론과 시사점을 기술한다.

II. 컴포넌트 모델

2.1 컴포넌트의 일반 모델

컴포넌트란 독립적인 단위기능을 수행하는 소프트웨어 모듈을 의미한다. 즉 특정기능을 수행하기 위해 독립적으로 개발 보급되고 잘 정의된 인터페이스를 가지며 다른 부품과 조립되어 응용시스템을 구축하기 위해 사용되는 소프트웨어 단위부품이다[3]. 컴포넌트는 재사용하면서도 교체 가능한 컴포넌트를 개발하기 위하여 일정한 표준을 따른다.

컴포넌트 모델은 컴포넌트를 조립하고 수행할 수 있는 컴포넌트의 상호작용과 구성에 대한 표준을 제공하고 분산기반 기능을 처리한다. 컴포넌트 모델은 컴포넌트 기반시스템을 가시화하고 명세화하고 그리고 문서화하는데 중요하고 실행시스템을 만들고 조립하는 지침이다[4]. 일반적인 컴포넌트 모델의 구조는 <그림 1>과 같다. 서버는 멀티프로세싱이나 부하 조절 그리고 트랜잭션서비스와 같은 시스템 서비스를 제공하는 프로세스이다. 컨테이너는 컴포넌트 구현을 위한 서버측 런타임 환경으로 컴포넌트의 인스턴스를 생성하고 소멸하는 관리를 수행한다. 컴포넌트는 데이터베이스를 참조해서 클라이언트에게 서비스를 제공한다.

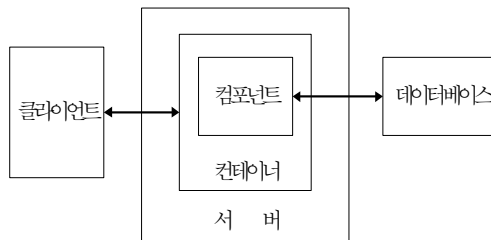


그림 1. 컴포넌트 모델의 일반적 구조
Fig 1. General Architecture of Component Model

컴포넌트는 컴포넌트 모델을 따르는 빌딩블록이다. 컴포넌트는 객체와 인터페이스로 구성되는 독립된 기능단위로 서비스 제공을 위해 필요한 데이터와 동작의 분리로서 내부의 객체와 외부의 인터페이스를 갖는다. 컴포넌트의 구성요소는 <그림 2>와 같고 하나 이상의 클래스와 하나 이상의 인터페이스가 될 수 있다. 컴포넌트는 특정서비스를 제공하는 소프트웨어 단위로서 객체지향시스템의 클래스들의 상호작용을 기반으로 클래스보다 큰 재사용을 위한 독립적인 소프트웨어 단위이다. 객체지향시스템은 인터페이스를 구현하는 객체간의 관계가 복잡하고 연관성이 깊은 반면 컴포넌트 기반시스템은 인터페이스를 구현하는 객체간의 관계가 명확히 구분된다[5].

컴포넌트는 인터페이스의 구성요소인 오퍼레이션에 의한 클래스들 사이의 상호작용을 통해서 서비스가 제공된다. 외부에서 컴포넌트의 서비스를 이용하기 위해서는 오퍼레이션을 호출하고 오퍼레이션은 컴포넌트 내부의 클래스나 다른 컴포넌트의 클래스와의 상호작용을 통해서 서비스를 제공한다.

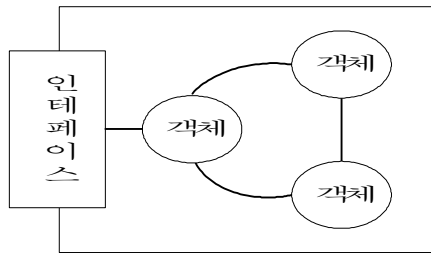


그림 2. 컴포넌트 모델의 일반적 구조
Fig 2. The Building Block of Component

2.2 컴포넌트의 정보은폐 모델

컴포넌트의 재사용성을 높이기 위한 방안의 하나로 컴포넌트의 접근은 인터페이스를 통해서만 가능하게 하는 것이다. 인터페이스는 컴포넌트의 기능을 추상화시켜 소프트웨어의 단위인 컴포넌트를 외부로 보이게 하는 역할과 외부에서 컴포넌트의 서비스를 이용하기 위해서 접근하는 매개자 역할을 한다. 인터페이스는 컴포넌트가 제공하는 서비스가 무엇인지를 나타낼 뿐 컴포넌트가 어떤 방법으로 서비스를 제공하는지의 컴포넌트의 내부를 보이지 않는데 이는 캡슐화와 정보은폐 개념으로 대표되는 추상화의 실체이다. 컴포넌트의 캡슐화는 정보와 이 정보를 조작하는 오퍼레이션을 하나의 컴포넌트로 결합하는 것으로 정보은폐의 이점을 제공한다(6).

캡슐화된 컴포넌트의 외부 인터페이스를 명확하게 정의함으로써 독립적으로 작성된 컴포넌트간의 상호 종속성을 극소화하여 캡슐화 컴포넌트는 외부 인터페이스만을 통하여 접근될 수 있도록 한다면 세부적인 구현사항에 대해서는 컴포넌트 내부에 은폐시킬 수 있다. 캡슐화와 정보은폐는 컴포넌트의 내부구현사항을 외부의 접근으로 보호할 수 있고 오류는 컴포넌트 내부로 국지화 할 수 있다. 캡슐화와 정보은폐는 또한 각각의 컴포넌트간의 인터페이스를 적게 하고 한 컴포넌트는 다른 컴포넌트의 수행에 무관하게 사용함으로써 다른 컴포넌트의 실행의 변화에 거의 영향을 받지 않는다. 따라서 유지보수가 좋고 새로운 프로그램의 확장이 용이하다. 컴포넌트의 재사용은 컴포넌트를 블록과 같이 조립하여 새로운 시스템을 구성하는 방법으로 컴포넌트 내부에서 일어나는 세부 내용의 변경없이 사용하는 블랙박스 재사용이 일반적이다. 블랙박스

재사용의 경우 가장 중요한 개념은 정보은폐이다. 블랙박스 재사용에서 정보은폐가 핵심개념이 되는 이유는 정보은폐가 잘 되어 있을수록 그만큼 추상화가 잘되어 있게 되는데 이는 그 컴포넌트의 구현상의 세부사항을 알 필요 없이 사용할 수 있게 되어 컴포넌트의 재사용을 확대한다.

III. 컴포넌트의 보안 모델

3.1 관심사의 분리 보안 모델

현재 정보보호 기능을 필요로 하는 컴포넌트에서 기존의 다양한 보안 API가 사용되고 있다. 보안 API를 이용하여 컴포넌트에 보안기능을 제공할 경우에 보안 코드가 컴포넌트 내부에 산재하기 때문에 컴포넌트의 재사용성을 저하시키는 요인이 된다. 컴포넌트 기반 시스템에서 이러한 문제가 발생하지 않도록 하기 위하여 컴포넌트 기반에서 표준화된 보안 서비스를 제공하는 컴포넌트의 보안 모델로 관심사의 분리 보안 모델이 있다(7). 관심사의 분리 보안 모델은 컴포넌트의 기본 기능을 다루는 핵심관심사와 보안 기능을 포함하는 횡단관심사로 관심사를 명명해서 사용한다. 본 논문에서는 핵심관심사와 횡단관심사의 동일 개념으로 기능관심사와 보안관심사라는 용어를 사용하고 보안코드에 대한 직조규칙을 정의하고 있는 애스펙트 컴포넌트를 직조컴포넌트라는 용어로도 사용한다.

관심사의 분리 보안모델은 <그림 3>과 같고 기능관심사의 핵심 컴포넌트가 자신이 필요한 보안관심사의 보안 컴포넌트를 찾아 사용하는 대신에 직조 규칙을 정의하고 있는 직조컴포넌트를 통해서 핵심관심사의 컴포넌트 내부에 필요한 보안관심사의 컴포넌트가 동작하도록 엮여지게 만든다(8). 핵심관심사의 컴포넌트는 더 이상 보안관심사의 컴포넌트를 직접 포함하지 않으며 보안 관심사의 복잡성은 애스펙트라는 직조컴포넌트로 분리된다. 컴포넌트가 보안컴포넌트와 핵심컴포넌트를 분리하고 직조 규칙을 정의하고 있는 애스펙트 컴포넌트를 통한 보안성이 강화된 컴포넌트를 쉽게 개발할 수 있다(9).

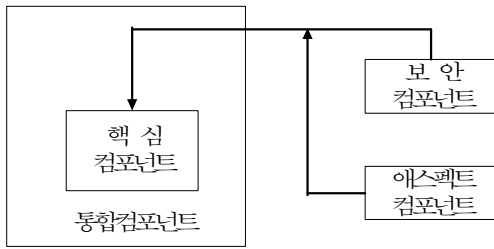


그림 3. 관심사의 분리 보안 모델
Fig 3. Separation of Concern Security Model

3.2 컴포넌트의 수정성 모델

핵심컴포넌트에 보안코드를 직조해서 보안성을 갖도록 하는 관심사의 분리 보안모델의 특성으로 화이트박스 형태의 컴포넌트 재사용이 요청된다. 화이트박스 재사용의 경우에는 수정성이 가장 중요한 품질이 된다(10). 화이트박스 재사용에서 수정성이 핵심이 되는 이유는 수정에 드는 노력과 비용이 적으면 기존의 컴포넌트의 재사용이 확대되기 때문이다. 컴포넌트는 컴포넌트 사용자들이 자신의 목적에 맞도록 컴포넌트가 지니는 속성이나 메서드를 변경 할 수 있도록 지원되어야 한다. 컴포넌트의 수정을 위해서는 컴포넌트에 대한 이해에 상당히 많은 시간을 소모하므로 높은 이해성은 용이한 수정을 위해서 필수적이라 할 수 있다. 화이트박스 재사용은 컴포넌트를 재사용하고자 하는 요구사항에 맞게 수정한 후 재사용함으로써 수정성이 높은 컴포넌트가 재사용성을 확대한다. 보안 컴포넌트의 특정 기능을 핵심 컴포넌트에 끼워넣기 위해서는 화이트박스 재사용이 적합하다.

IV. 컴포넌트 재사용 확대를 위한 관심사의 분리 보안 모델

4.1 컴포넌트의 포장 모델

블랙박스 형태의 컴포넌트에 관심사 분리 보안 모델을 이용해 보안기능을 삽입하기 위해서는 화이트박스 형태의 컴포넌트로 전환할 수 있는 방법이 요청된다. 이를 위해서 블랙박스 형태의 컴포넌트를 포장해서 수정가능성을 제공하는 포장 모델이 있다(11). 직조컴포넌트에 의해서 보안컴포넌트를 핵심컴포넌트에 합성하는 경우에 블랙박스 형태의 핵심컴포넌트를 재정의한 포장 컴포넌트를 개발해 보안코드가 끼워지도록 해야 한다.

포장 컴포넌트는 블랙박스 형태의 컴포넌트 기능을 확장하고 내부적인 요소들의 간단한 변경으로 재사용성을 확대하는 메커니즘을 제공한다(12). 본 논문에서 제안하는 포장기술은 보안성과 재사용성을 확대하기 위한 컴포넌트의 관심사분리 보안 모델을 위하여 사용한다. 기존 컴포넌트에 대한 잘 정의된 인터페이스를 제공함으로써 다른 컴포넌트에서 기존 컴포넌트의 서비스를 이용할 수 있도록 해주는 컴포넌트의 포장모델을 연결하여 사용함으로써 컴포넌트 재사용 확대를 위한 관심사 분리 보안 모델은 수정의 유연성을 확대하고 기능을 확장한다. 컴포넌트의 기능을 확장하는 컴포넌트의 포장 모델은 <그림 4>와 같다. 이 모델에서 포장 컴포넌트는 기존 컴포넌트의 모든 인터페이스에 대한 호출을 포함하고 있다. 컴포넌트에 대한 입력 서비스 x, y와 출력서비스 a, b가 대응되어 있다. 컴포넌트의 포장 모델은 기존 컴포넌트의 모든 서비스를 포장하는 대신 유용한 서비스만을 선별하여 일괄 포장하는 메커니즘을 사용할 수 있다.

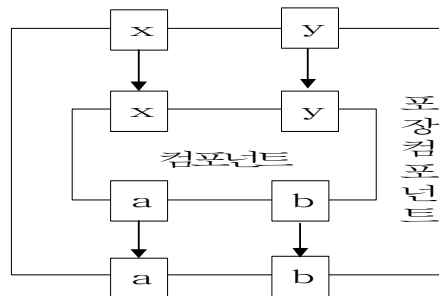


그림 4. 컴포넌트의 포장 모델
Fig 4. Wrapper Model of Component

4.2 컴포넌트 재사용 확대 보안 모델

보안컴포넌트는 단독으로 수행되는 응용이라기보다는 컴포넌트와 통합되거나 컴포넌트간의 조립과 커스터마이징으로써 비즈니스 로직을 수행한다. 따라서 컴포넌트의 재사용성은 컴포넌트의 정보은익과 수정성을 높임으로써 확대된다(13). 따라서 블랙박스 재사용과 화이트 박스 모두를 고려한 그레이박스 형태의 컴포넌트가 요청된다. 블랙박스 재사용은 컴포넌트 내부에서 일어나는 세부내용의 변경 없이 사용하는 반면에 화이트박스 재사용은 컴포넌트의 내용을 적절히 변경하여 재사용하는 방법으로 부품의 내용을 상세히 알아야 한다. 재사용은 소프트웨어 개발의 생산성과 품질을 향상시키기 위한 핵심기술이다. 커스터마이징하기 어려운 블랙박스 재사용 컴포넌트 재사용이 저하되는 결과를 가져온다. 블랙박스 컴포넌트는 정보은익을 제공함으로써 컴포넌트 기반 시스템

의 재사용성에 기여하지만 기능추가나 수정이 필요한 경우에 재사용성을 떨어뜨린다. 컴포넌트는 컴포넌트 내부의 세부 구현 사항을 외부에 노출시키지는 않지만 컴포넌트 사용자들이 자신의 목적에 맞도록 컴포넌트 기능을 지니는 속성이나 메시지를 변경할 수 있도록 지원해야 한다.

컴포넌트 재사용 확대 보안 모델은 <그림 5>와 같이 블랙박스의 정보은폐와 화이트박스의 수정용이성의 장점을 얻도록 컴포넌트의 재사용모형을 결합한다. 컴포넌트의 인터페이스를 사용하는 블랙박스 재사용과 컴포넌트의 소스코드를 사용하는 화이트박스 재사용의 이점을 공유할 수 있는 그레이박스 형태의 컴포넌트에 대한 재사용 확장 모델을 제안한다.

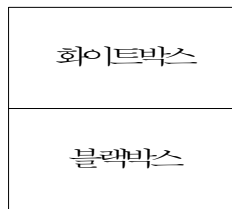


그림 5. 컴포넌트의 포장 모델
Fig 5. Wrapper Model of Component

컴포넌트 재사용 확대 보안 모델은 <그림 6>과 같다. 기본 기능을 제공하는 블랙박스 형태의 핵심컴포넌트가 보안성을 제공하도록 보안컴포넌트의 인터페이스 호출을 추가하기 위해서 핵심 컴포넌트의 인터페이스를 캡슐화한 포장 컴포넌트를 개발한다. 포장 컴포넌트에 보안컴포넌트의 인터페이스를 추가하기 위해서 직조 규칙을 기술한 애스펙트 컴포넌트를 제작한다. 애스펙트 컴포넌트의 직조 규칙은 통합 컴포넌트를 만들기 위해 핵심 컴포넌트와 보안 컴포넌트를 통합하는 방법을 표현한다. 따라서 핵심컴포넌트의 기능을 확장하거나 보안 규칙의 변화에 효과적으로 대응할 수 있다.

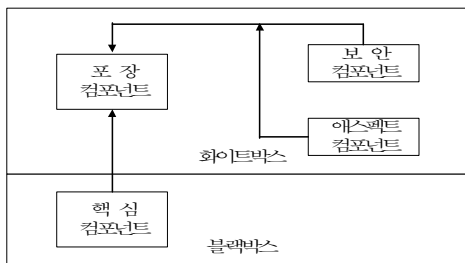


그림 6. 컴포넌트 재사용 확대 보안 모델
Fig 6. Security Model for Extension of Component Reuse

4.3 모델의 적합성 검증

컴포넌트 재사용 확대 보안 모델에 대한 기존 시스템의 적용 사례로서 메시지 보안 시스템의 인증 프로토콜 설계 및 구현[14]에 관한 연구 개발 시스템인 NMAP(New Message Authentication Protocol)을 선정하였다. NMAP 시스템의 기본 구조는 <그림 7>과 같이 인증 컴포넌트를 사용하여 클라이언트는 은행계좌컴포넌트에 접근하기 위해서 사용자의 인증정보가 탑재된 메시지를 서버에 전송하여 인증을 실현하고 있다.

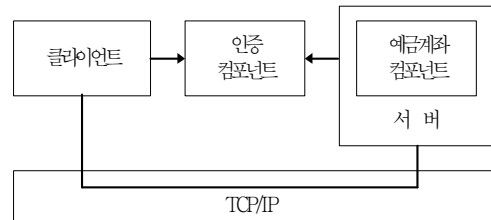


그림 7. NMAP의 기본 구조
Fig 7. Basic Architecture of NMAP

<그림 8>은 기존 시스템의 서비스 제공을 위한 인터페이스와 설계 내용을 토대로 제안 모델에 따라 재구성한 시스템의 구조이다. 클라이언트는 은행계좌 컴포넌트의 잔액조회서비스를 이용하기 위해서는 인증을 필요로 한다. 따라서 은행계좌 컴포넌트에 인증기능을 추가하기 위해서 블랙박스형태의 은행계좌 컴포넌트를 포장해서 화이트박스 형태의 포장 컴포넌트를 개발하고 인증컴포넌트의 사용자인증 기능을 포장 컴포넌트에 삽입하기 위해서 직조규칙을 정의한 애스펙트 컴포넌트를 개발해서 시스템을 구축하였다.

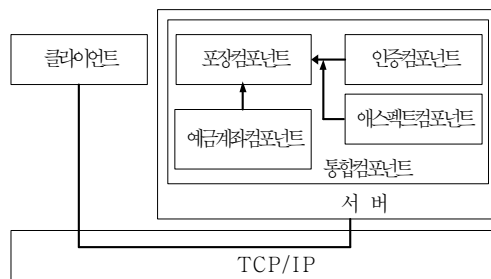


그림 8. 컴포넌트의 보안모델을 사용한 응용시스템의 구조
Fig 8. The Application Architecture for Security Model of Component

본 연구의 실증적 검증을 위하여 정보보안공학이라는 과목을 수강하는 학생을 대상으로 인증서비스를 제공하는 은행계좌컴포넌트에 대한 응용시스템의 완성에 소요되는 시간과 재사용을 측정하였다. 은행계좌 컴포넌트와 인증 컴포넌트를 제공하고 보안 API의 응용 모델과 재사용 확대를 위한 관심사 분리 보안 모델에 대한 실험을 행한 결과는 <표 1>과 같고 응용시스템의 완성을 위해서 오류를 수정하고 테스트하는데 소요된 평균 시간은 보안API의 응용모델을 적용한 방식이 관심사의 분리보안 모델을 사용한 경우보다 오래 걸렸다. 또한 보안API를 적용한 학생들은 예금계좌 컴포넌트를 수정해서 사용하는 경향이 많은 반면 관심사분리보안 모델을 사용한 학생들은 예금계좌컴포넌트를 블랙박스 형태로 사용하는 경향이 있음을 확인하였다.

표 1. 컴포넌트의 테스트 시간과 재사용률
Table 1. Test Time and Reuse Rate of Component

구 분	보안API 모델	관심사분리 보안모델	차이
테스트시간	4.2	2.5	1.7
재사용율	60	87	27

분류된 컴포넌트 기반의 소프트웨어 규모 산정을 위해서 코코모II의 재사용 모델을 변형한 식 <4.1>를 사용하여 모델의 적합성을 평가한다[15]. 식 <4.1>에서 상수 K를 구하기 위해서 31개의 프로젝트에 대한 회귀분석을 수행하였다. 이 식<4.1>을 회귀분석에 적용하기 위하여 재사용이 없을 때의 규모를 나타내는 ASIZE를 100으로 놓고 식 <4.2>와 같은 형태로 변환하였다.

$$SIZE = (1 - \frac{K - RUSED}{100}) * ASIZE$$

SIZE : 재사용부분을 고려한 규모(원도우수) (4.1)
 K : 재사용에 대한 초과투입율(상수)
 ASIZE : 재사용부분을 고려하지 않는 규모
 RUSED : 재사용율(%)

$$100 - SIZE = K * RUSED \dots\dots\dots (4.2)$$

회귀분석결과는 회귀계수가 0.74로 나타났고 식 <4.3>과 <표 1>를 사용하여 규모를 측정된 결과는 <표 2>와 같다.

$$SIZE = (1 - 0.74 * \frac{RUSED}{100}) * 100 \dots\dots\dots (4.3)$$

표 2. 컴포넌트의 재사용률과 컴포넌트 기반 시스템의 개발 규모
Table 2. Test Reuse Rate and Size of Component Based System

구분	보안 API모델	관심사분리 보안모델	차이
재사용율	60	87	27
규모	56.5	35.6	-20.9

보안 API모델과 비교해 컴포넌트 재사용 확대를 위한 관심사의 분리 보안 모델을 사용한 응용 시스템의 규모가 줄어들어 개선되었음을 확인할 수 있다.

보안 API를 사용한 경우에는 은행계좌 컴포넌트에 보안 코드가 산재되어 있어서 기본 기능의 파악이 어렵고 기본기능 코드와 보안 코드가 서로 뒤섞여 있어서 재사용을 어렵게 한다. 반면에 본 논문에서 제안한 통합컴포넌트의 보안 모델을 적용한 응용시스템의 경우에는 보안코드가 은행계좌를 표현하고 있는 핵심컴포넌트로부터 분리되어 별도의 에스펙트 컴포넌트로 정의되어 있어서 컴포넌트의 복잡성이 감소하고 재사용이 용이하다.

V. 결론

컴포넌트는 재사용을 통한 소프트웨어 개발 생산성의 향상과 개발에 소요되는 비용을 감소시킬 수 있는 이점을 제공한다. 이러한 이점에도 불구하고 컴포넌트는 내부의 보안 취약성으로 인해서 재사용을 제한한다. 따라서 이의 해결책으로 보안성과 재사용성 향상을 위한 수정 가능한 컴포넌트의 관심사 분리 보안 모델을 제안한다. 컴포넌트는 재사용을 위해서 수정이 필요하거나 또는 필요 없을 수도 있다. 따라서 제안 모델은 컴포넌트의 재사용을 높이기 위해서 컴포넌트 안에서 일어나는 세부내용의 변경 없이 재사용하는 블랙박스 재사용과 재사용할 컴포넌트를 적절히 변경하여 재사용하는 방법 즉 부품의 내용을 자세히 알아야 하는 화이트박스 재사용의 두 모델을 수용하고 포장 컴포넌트와 에스펙트 컴포넌트를 도입해서 보안코드의 분리를 통한 복잡성의 감소와 수정성을 부여해서 재사용성을 확대한다. 응용 모델은 기본기능을 포함하고 있는 핵심 컴포넌트로부터 보안코드를 분리하는 형태로 보안성을 제공함으로써 개발 구조를 개선하고 컴포넌트의 재사용을 향상시킬 수 있다. 또한 컴포넌트에 대한 의존도 및 활용도가 증가하고 있는 상황에서 컴포넌트에 보안성을 제공함으로써 컴포넌트의 재사용의 확대에 기여한다. 향후 포장 컴포

넌트와 에스펙트 컴포넌트의 개발을 지원하는 자동화 도구에 대한 연구가 필요하다.

참고문헌

- [1] Llewellyn-Jones, D. and M. Merabti, Askwith, "An Extensible Framework for Practical Secure Component Composition in a Ubiquitous Computing Environment," in International Conference on Information Technology, Las Vegas, USA, April, 2004.
- [2] Mantel, H., "On the composition of secure systems," in Proceedings 2002 IEEE Symposium on Security and Privacy, 2002.
- [3] Lan, K. and Z. Wang. A taxonomy of software component models. In Proc. 31st Euromicro Conference, pp. 88-95, IEEE Computer Society Press, 2005.
- [4] Bruneton, E. and Coupaye, An open component model and its support in Java. In Proceedings of 7th CBSE, pp. 7-22, Springer-Verlag, 2004.
- [5] Beydeda, S. and V. Gruhn. Integrating white- and blackbox techniques for class-level testing object-oriented prototypes. In SEA Software Engineering and Applications Conference (Las Vegas, Nevada, November 6-9), pp. 23-28, IASTED/ACTA Press, 2000.
- [6] Tesanovic, A., et. al.. "Aspect-level WCET analyzer: a tool for automated WCET analysis of a real-time software composed using aspects and components," in Proceedings of the 3rd International Workshop on Worst-Case Execution Time Analysis (WCET 2003), Porto, Portugal, July, 2003.
- [7] Kiczales, G. and M. Mezini. Separation of Concerns with Procedures, Annotations, Advice and Pointcuts. European Conference on Object-Oriented Programming (ECOOP), 2005.
- [8] Suvee, D., et. al., March 2003. JAsCo: an Aspect-Oriented approach tailored for Component Based Software Development, pp. 21-29, Proceeding of AOSD 2003.
- [9] Sipma, H. "A formal model for cross-cutting modular transition systems," in In Proceedings of the Workshop on Foundations of Aspect-Oriented Languages (FOAL 2003), Boston, USA, March, 2003.
- [10] Christine Mingins and Chee Chan, Building trust in third-party components using component wrappers in the .net frameworks, Proceedings of the Fortieth International Conference on Tools Pacific, Australian Computer Society, pp. 153-157, 2002.
- [11] Sewell, P. and J. Vitek, Secure Composition of Untrusted Code: Wrappers and Causality Types, 13th IEEE Computer Security Foundations Workshop (CSFW'00) July 03 - 05, 2000.
- [12] Dean, J.C., Security Wrapper Technology for COTS Software Products, 13th Annual Software Technology Conference, Salt Lake City, Utah, May, 2001.
- [13] Martin Böhuchi and Wolfgang Weck. A plea for grey-box components. Technical Report 122, Turku Centre for Computer Science, Presented at the Workshop on Foundations of Component-Based Systems, Zürich, September, 1997.
- [14] 김영수, "메시지보안시스템의 인증 프로토콜설계 및 검증," 박사학위논문, 국민대학 교대학원, 2003.
- [15] 서정석, "컴포넌트 개발 틀을 사용한 소프트웨어 개발 노력 산정에 관한 연구," 박사학위논문, 국민대학교대학원, 2000.

저 자 소 개



김 영 수

2003년 2월 : 국민대학교 정보관리
학박사

2008년 ~ 현재 : 충북대학교 포 닥
관심분야 : 인터넷, 정보보안, 보안공학



조 선 구

2000년 2월 : 국민대학교 정보관리
학박사

1997년 ~ 현재 : 나사렛대학교 교 수
관심분야 : 인터넷, 정보보안, 보안공학