영역지향 프로그래밍 기술을 적용한 CBD 방법론: UML 다이어그램의 개선을 중심으로

김 치 수*, 김 태 영**

요 약

최근 소프트웨어 개발 방법론 중 소프트웨어를 부품화하고 이를 조립·합성하여 새로운 애플리케이션을 개발하는 방식의 CBD 방법론이 많이 연구되고 있다. 그러나 CBD 방법론은 시스템의 기능적인 특성을 중심으로 분할하는 방법이 많아 컴퓨터에 대한 문제, 문서화, 포트리 이 해를 어렵게 하는 단점이 있다. 따라서 본 논문에서는 영역지향 프로그래밍 기술을 CBD 방법론과 UML 다이어그램에 적용하여 CBD 방법론의 단점을 보완함으로써 컴퓨터의 재사용을 용이하게 하고 시스템 개발 시, 단축과 개발비용의 감소를 유도하였다.

The Methods of Component-Based Development Using Aspect-Oriented Programming Techniques: Focusing on Improvement in UML Diagram

Chi-su Kim*, Tae-young Kim**

ABSTRACT

Among many recent methods developing software, the method of component-based development (CBD), which refers to the method of treating software as parts of a larger whole, and developing new applications through the assembly and synthesis of existing software, has been thoroughly studied. CBD, however, has demerits that cause difficulty in making inferences and understanding the code of components, and lack adequate documentation because the method of CBD divides systems according to the functional characteristics of these systems. Therefore, this study shows how to reuse components without difficulty and reduce the development time of systems and development costs by compensating for the weak points of the method of CBD. Aspect-Oriented programming technique has been applied to the method of CBD and the UML diagram for this purpose.

키워드: 영역지향프로그래밍(Asspect-Oriented Programming), 컴퓨터개발방법론(CBD), UML.

1. 서 론


본 논문에서는 영역지향 프로그래밍 기술을 CBD 방법론에 적용하여 CBD 방법론이 가지고 있는 상위레벨 체계의 결합된 사항을 개선함으로써 컴퓨터의 재사용을 용이하게

1) 복잡성과 높은 시스템의 수록, 예외/예외 처리, 보안, 영속성, 하드웨어, 네트워크 관리 등이 고려되어야 하는데, 이러한 시스템 작성이 단일한, 주의력과 기술력이 요구되는 특성과 요청을 다루기 위한 시스템을 구성하는 모듈의 여러가지 기계 속에 존재하는 현상.
1436 정보처리학회논문지 D 제11-1권 제7호(2004.12)

하고 시스템 개발 시간 단축과 개발비용의 감소를 유도하고자 하였다.

2. 관련 연구

2.1 영역지향 프로그래밍(AOP: Aspect-Oriented Programming)
영역의 개념은 결국 중심 프로그래밍 기술과 깊게 지향 프로그래밍 기술에서 충분히 처리될 수 없는 문제들에 대응하기 위해 소프트웨어 개발 분야에서 소개되었다. 1999년 5월에 Gregor Kiczales(ZXerox PARC(Palo Alto Research Center)'의 수석 연구원)과 그의 그룹에 의해 정의되었으며, 시스템의 총체적인 레벨 세계를 수직으로 분할함으로써 시스템의 기능과 사용자 서비스에 집중한 새로운 동작적인 모듈(컴포넌트)단위 구조를 수평으로 자동으로 만들기로 정의하였다[8].

영역지향 프로그래밍은 영역이라는 새로운 프로그램 구조를 정의하여 사용하고 있으며, 이 구조는 스토리치, 클래스, 인터페이스 등과 같은 특징한 용도의 구조화된 것이다. 이는 영역 내에 프로그램의 여러 모듈들에 돼져 있는 기능(해당하는 기능이 여러 모듈들에 분해되어 있든 것)을 보다 정리하게 된다. 결국 이들의 관계의 각 클래스는 자신에게 주어진 고유 기능만을 수행하고, 추가된 각 영역들이 총체적인 기능들을 모아 처리함으로써 크로스컷팅 문제를 해결하면서, 총체적인 프로그램을 이루는 형태를 만들게 된다. 여기서 총체적인 기능이라는 사용자 인터페이스, 처리, 지속성, 암호화, 메시지를 관리, 손실 등을 같은 비 기능적인 영역이나 수준 높은 서비스들들을 의미한다. 그리고 영역은 사물의 캡슐화 원칙에 따라 생성된 형태로만 사용되기 때문에 영역은 객체의 Private 영역까지 접근할 수 있지만, 다른 클래스의 존재 사양의 캡슐화에는 영향을 주지 않는다.

본 논문에서는 영역지향 프로그래밍 기술을 CBD 방법론에 적용하였으며, 영역정보가 UML 다이어그램 내에서 어떻게 표현될 수 있는지를 보았다.

3. 영역지향 프로그래밍 기술을 적용한 CBD 방법론과 UML 다이어그램
영역지향 프로그래밍 기술을 적용한 CBD 방법론은 저작권, 저작권 성립, 동작적인 순응성이 존재하는 소프트웨어 컴포넌트를 개발하기 위해 본 논문에서 제시한 CBD 방법론이다. 영역지향 프로그래밍 기술을 적용한 CBD 방법론은 소프트웨어의 전체적인 개발 과정에 적용되어진다.

본 논문을 통해 제시하고 있는 영역지향 기술은 컴포넌트의 기능적인 특성을 나타내기 위해 수직으로 분할된 소프트웨어 컴포넌트를 다시 수평으로 절라 크로스컷팅 문제와 비 기능적인 특성을 나타낼 수 있도록 영역 표기법을 도입하는 기법을 제안하였으며, 이것은 기존의 CBD 방법론에 적용하였다.

(그림 1) 일반적인 개념의 컴포넌트와 영역 개념이 포함된 컴포넌트
영역지향 프로그래밍 기술을 적용한 CBD 방법론에서의 영역은 컴포넌트의 기능적인 정보들을 감지할 수 있도록 각 영역의 기능을 적절히 동작하도록 설계할 수 있도록 한다. 각 컴포넌트의 크로스컷ting 문제 해결을 위해 더욱 효과적인 특징과 구현 방법을 지원함으로써 개발자와 사용자에게 컴포넌트들에 대한 지식을 공유하게 하는데 초점을 둔다.

보통 일반적인 영역은 사용자 인터페이스, 처리, 지속성, 암호화, 메시지 관리, 클러스 등과 같은 비 기능적인 영역이나 수준 높은 서비스들들을 의미한다. 그리고 영역은 사물의 캡슐화 원칙에 따라 생성된다. 따라서 영역은 객체의 Private 영역까지 접근할 수 있지만, 다른 클래스의 존재 사양의 캡슐화에는 영향을 주지 않는다. 본 논문에서는 영역지향 프로그래밍 기술을 CBD 방법론에 적용하였으며, 영역정보가 UML 다이어그램 내에서 어떻게 표현될 수 있는지를 보았다.
수 있도록 도와준다. 셋째, 영역지향 컴포넌트 개발자들이 더욱더 정확하고 완전하게 컴포넌트를 문서화 할 수 있도록 도와준다. 넷째, 영역을 가지고 있는 컴포넌트는 개발자와 최종사용자에게 컴포넌트의 다양한 관점을 줄 수 있게 한다.

3.1 영역지향 컴포넌트 요구사항
영역지향 컴포넌트 요구사항은 기능적이거나 비 기능적인 요구사항을 입증하고 지정하기 위한 것인데, 이것은 시스템의 중요한 부분과 관련되어진다. 보통 시스템 컴포넌트는 코로스팅이 문제 해결을 위한 서비스를 제공하기 원하거나 혹은 필요로 한다. 예를 들면, 컴포넌트 개발자는 컴포넌트의 기능적이거나 비 기능적인 면과 관련된 사용자 인터페이스, 처리, 지속성, 보안등에 대해 명확하게 하기 위하여, 또한 이를 문서화하기를 원한다. 컴포넌트의 요구사항 명세를 규정하는 데는 개발자는 어떤 컴포넌트들은 많은 영역 서비스들이 갖고 있을 수 있으며 어떤 컴포넌트들은 약간의 영역 서비스들만 갖을 수 있다는 것을 알게 될 것이다. 더욱이 어떤 컴포넌트들은 영역 서비스를 공유할 수 있음으로 알게 될 것이다. 영역지향 컴포넌트의 요구사항 명세는 컴포넌트별로 하는 기술자들에게 관련된 컴포넌트 속성을 이해시키는데 매우 유용하다.

3.2 영역지향 컴포넌트 설계
요구사항을 명시 하는 동안 개발자들은 영역 체계를부터 기능적이며 비 기능적이고 균일한 체계를 만드는 것이며 이로운 설계들은 영역을 갖는 컴포넌트 설계로 정립 되게끔 사용된다. 요구사항 수준과 세부적인 영역들은 소프트웨어로 제작된 디자인 손준의 컴포넌트 영역들로 정립 될 수 있는데 이들은 디자인 손준의 영역의 서비스들을 특정화한다. 예를 들면, 세분화된 디자인 단계의 영역 명세들은 사용자 인터페이스, 컴포넌트 지속성, 분배, 보안과 처리 모델, 컴포넌트 구조 등이다.

설계 단계에서 영역의 세부 명세는 컴포넌트 구현 설계 기술을 연장하하게 한다. 설계 단계에서 컴포넌트 서비스들은 관련된 영역을 문서화시키고, 설계자가 다양한 추론과 설계 의 방향을 쉽게 수립할 수 있도록 하기 위해 다른 것들은 엮이진 영역들까지도 문서화한다.

3.3 영역지향 컴포넌트 구현과 테스트
영역지향 프로그래밍 기술은 적용할 CBD 방법론을 사용해서 설계된 컴포넌트들은 일반적인 컴포넌트 기반 구현 기술이나 Enterprise JavaBeans™, JViews 등과 같은 컴포넌트 기반 프레임워크를 사용함으로써 실행될 수 있다. 엔터프라이즈 자바베이스는 시스템 레벨 서비스들을 제공한다. 컴포넌트 영역은 인터페이스, 어떤 브라우저 또는 디자인 패널을 통해 실행 될 수 있다. 영역을 가진 컴포넌트는 각각의 다른 기능을 접근하고, 컴포넌트 상호간의 인터페이스 정의를 인터페이스 정의를 안내받을 수 있게 하기 위해 관련된 컴포넌트들에 대한 테스트를 제공할 수 있다. 하나의 컴포넌트 안에서 실행되는 영역 서비스들은 다른 컴포넌트들이나 최종사용자에 의해 라이브러리에서 사용될 수 있으며, 라이브러리 동안 컴포넌트의 능력을 높이는데 사용된다.

3.4 영역지향 컴포넌트 테스트
영역지향 CBD 방법론은 컴포넌트 요구사항과 명세, 설계, 구현과 테스트 단계에서 적용되었다. 그러나 영역 서비스를 가지고 실행되는 컴포넌트가 다른 문제로 개발되거나 한 시스템의 부분으로 사용되었을 때 컴포넌트의 요구사항이 얼마나 잘 정의되었는지, 명확되었는지를 테스트해 볼 필요가 있다. 그러므로 컴포넌트에 대한 시험 계획과 진단을 만들어 볼 필요성이 있다. 그것은 컴포넌트 영역 기술자들이 관련하고, 복잡한 테스트 에이전트를 만들어 사용함으로써 가능할 것이다.

3.5 UML 다이어그램에서의 영역정보 표현
기존의 UML 다이어그램은 시스템의 기능적인 특성에만 초점을 맞추어 작성되었기 때문에 시스템에 대한 높은 수준의 세부 정보와 개발 사항이 나타나 있지 않는다. 그러나 영역지향 프로그래밍 기술은 포함된 UML 다이어그램에서는 시스템에 대한 높은 수준의 세부 정보와 개발 사항을 UML 확장 수준의 스테이지마크와 주석을 사용하여 표현하고 있는 것이 기존의 CBD 방법론을 적용한 UML 다이어그램과 다르다.

4. 적용 사례
기존까지의 일반적인 CBD 기반의 컴포넌트는 기능 중심적인 수준의 일반적인 시스템의 개발을 위한 기존의 CBD 방법론을 사용하여 대형 시스템을 개발하고자 할 때에는 높은 수준의 시스템 정보와 개발 사항들이 나타나 있지 않아 시스템 개발에 어려움이 많았다. 따라서 본 논문에서는 영역 지향 프로그래밍 기술을 추가한 CBD 방법론을 적용하여 시스템의 비 기능적인 정보와 개발사항들을 포함하여 통합적인 시스템을 설계와 수편으로 동시에 활발하였다. 영역은 사용자 인터페이스, 지
속성, 처리, 실행과 보안 같은 시스템의 비 기능적인 문맥에 의해 식별되는 많은 정보가 컴퓨터에 전형적으로 영향을 미친다.

이 논서에서는 컴퓨터에 기반을 두 온라인 가전제품 판매시스템을 하나의 적용례로 영역지향 프로그래밍 기술을 추가한 CBD 방법론을 적용하여 설계하였다.

4.1 유즈케이스 모델링
유즈케이스 다이어그램 내에서는 시스템 사용의 사례들을 그려 놓은 것이기 때문에 별도로 영역정보에 대한 특별한 표현은 필요로 하지 않는다. (그림 2)는 온라인 가전제품 판매시스템의 유즈케이스 다이어그램을 나타낸 것이다.

(그림 2) 온라인 가전제품 판매시스템의 유즈케이스 다이어그램

4.2 클래스 모델링
기존의 클래스 다이어그램은 크로스 컨텍스트 상위레벨 계계즉, 사용자 인터페이스, 보안, 처리, 지속성 영역 정보와 같은 정보들이 결합되어 있다. 그리고 이 영역들이 클래스 다이어그램이나 다른 UML 다이어그램에서 문서화되지 않았다. 더욱이 이 영역 정보가 따로는 더 많은 중요한 시스템 구조와 관련 특성을 가지고 있을 수 있다. (그림 3)은 영역 정보를 포함한 클래스 다이어그램인데 개발자로 하여금 주석을 달아 고유 클래스가 영속성, 처리, 보안에 관한 영역을 필요로 하고 있음을 적절한 분명하게 알려 주고 있다. 주석으로 영역정보를 표현해 주기 위해서는 먼저 영역 세부사항들을 정의하여 4.3장의 (표 1)에서 보는 바와 같이 문서화가 진행되어야 하며, 이 부분에는 설명하였다.

(표 1)チェック아웃에 대한 시스템 다이어그램에서의 영역 정보

<table>
<thead>
<tr>
<th>영역</th>
<th>영역 세부 사항 및 성립과 간단한 주주</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt;+ provided&gt;&gt; 고유 정보/상호감 정보, 본문, 처리 정보와 관련 정보를 사용자에게 보여준다. 이 서비스는 온라인 쇼핑 고유 관련 정보와 사용자에 의해 요구되어 진다.</td>
<td></td>
</tr>
<tr>
<td>&lt;&lt;User Interface&gt;&gt;</td>
<td>프레임/옵션 사용 정보: 상호감/HTML 사용 정보: 온라인 쇼핑 정보를 사용하려면, HTML과 JS와 스위퍼 센터에 의해 제공될 수 있다.</td>
</tr>
<tr>
<td>&lt;&lt;Transaction&gt;&gt;</td>
<td>매장 정보 전송 안내 화면의 사용을 위해 사용한다.</td>
</tr>
<tr>
<td>&lt;&lt;Persistency&gt;&gt;</td>
<td>저장/복제 데이터</td>
</tr>
<tr>
<td>&lt;&lt;Security&gt;&gt;</td>
<td>접근 인증 사용자 접근을 위해서는 프로세스나 온라인 접속을 획득하기 위한 정보를 필요로 한다. 사용자 정보는 프로세스 인증 마스터이다.</td>
</tr>
<tr>
<td>&quot;&quot;+ required&quot;&quot;</td>
<td>게시판/메이커 데이터</td>
</tr>
</tbody>
</table>
4.3 동적 객체 상호작용 모델링

 객체 상호작용 모델링은 선택, 지속성, 보안, 처리, 배포 영역 정보 등과 같은 비 기능적인 영역에 관한 정보의 결합으로부터 실제, 오프라인, 서비스화된 컴퓨티드를 쉽게 발전하는데 매우 큰 도움을 준다. 다음은 영역 정보를 포함한 객체 상호작용 다이어그램에 대해 설명한 것인데 여기에서는 <표 1>과 같이 체크아웃 시스템 다이어그램에 대한 영역 정보와 <표 2>와 같이 다이어그램을 세부적으로 설명하였다.

- 체크아웃의 유즈 케이스 진행 단계
  전체조건 : 고객이 체크아웃 버튼을 누른다.

  1. 고객 정보를 보여준다.
  1.1 만일, 고객이 등록되어 있다면, 등록된 정보를 표시한다.
  1.2 만일, 고객이 등록되어 있지 않다면, 고객 정보 서식을 제공한다.

  2. 이미 저장되어 있는 고객 정보가 있다면, 신용카드 정보를 묻고 신용카드를 확인하기 위해 기다린다.
  2.1 만일, 카드가 유효하다면 신용카드 정보를 제공한다.

  2.2 만일, 카드가 유효하지 않다면 신용카드 정보를 제공한다.

  2.3 만일, 카드가 유효하지 않다면 신용카드 정보를 제공한다.

  2.4 만일, 카드가 유효하지 않다면 신용카드 정보를 제공한다.

  2.5 만일, 카드가 유효하지 않다면 신용카드 정보를 제공한다.

  2.6 만일, 카드가 유효하지 않다면 신용카드 정보를 제공한다.

  2.7 만일, 카드가 유효하지 않다면 신용카드 정보를 제공한다.

  2.8 만일, 카드가 유효하지 않다면 신용카드 정보를 제공한다.

  2.9 만일, 카드가 유효하지 않다면 신용카드 정보를 제공한다.

  2.10 만일, 카드가 유효하지 않다면 신용카드 정보를 제공한다.

2.1.1 데이터베이스에 주문을 등록하라.
2.1.2 데이터베이스의 목록 정보를 검색하라.
2.2 만일, 신용카드 확인에 실패하면, 2.1로가 신용카드 정보를 수정한다.

4.4 컴퓨트 디자인 협세

 일반적인 컴퓨트 디자인과 디자인은 컴퓨트를 낮은 수준의 기능적인 특성을 분해하기 때문에 컴퓨트의 높은 수준의 계약과 비 기능적인 영역 정보는 정리 확인할 수 없었다. 그러나, 영역정보의 제공은 컴퓨트에 대한 일반적인 특성을 부여하고, 책임을 붙이는 것보다 더 유용하다.

4.4.1 배치 모델링

시스템 배치 구조 다이어그램은 시스템 레이아웃 처리 요소, 컴퓨트의 정적인 배치 상태와 시스템이 사용한 프로토콜들을 그림으로 그린 것이다. 그러나 이 배치 다이어그램에서도 보면 영역이나 지속성 영역 등의 영역 정보가 들어 있지 않다. <표 2>는 배치 다이어그램에서의 상세한 영역 정보를 설명하고 있으며, (표 5)는 배치 다이어그램에서 영역 정보가 어떻게 표현되었는지를 보여 주고 있다.

(그림 4) 영역 분석을 포함한 체크아웃 시스템 다이어그램
4.4.2 교체 핸들링 설계

(표 3)은 교체 핸들링의 영역 정보를 상세하게 설명하고 있으며, (그림 6)은 교체 핸들링 클래식 데이터 디자인에서 영역정보가 어떻게 표현되었는지를 보여주고 있다.

### (표 3) 교체 핸들링의 영역 정보

<table>
<thead>
<tr>
<th>영 역</th>
<th>영역 세부 사항 및 속성과 간단한 주문</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>&lt;Security&gt;</strong></td>
<td>접근 인증, 세부 사항 특정은 매스토드 인증 마크이다. 웹 GUI 핸들링을 사용한 경우, GUI 핸들링은 접근 로그인을 확인하기 위한 서비스를 필요로 한다.</td>
</tr>
<tr>
<td><strong>&lt;Distribution&gt;</strong></td>
<td>후보/서희 데이터, 64 데이터 평가/서희 혹은 암호화 암호화를 사용한다. 서비스는 웹 GUI/64 데이터 핸들링 또는 암호화 해킹에 의해 제공될 수 있다.</td>
</tr>
<tr>
<td><strong>&lt;User Interface&gt;</strong></td>
<td>풀/프레임 스위치, 디비 또는 HTML 배포를 위한 HTML 배포 또는 GUI 핸들링에 의해 제공될 수 있다.</td>
</tr>
<tr>
<td><strong>&lt;Distribution&gt;</strong></td>
<td>풀 또는 데이터베이스를 위한 SQL을 사용한 JDBC와 같은 데이터베이스 접속에 의해 제공된다.</td>
</tr>
<tr>
<td><strong>&lt;User Interface&gt;</strong></td>
<td>풀 또는 GUI 핸들링에 의해 제공된다.</td>
</tr>
</tbody>
</table>

5. 결론 및 향후 연구

기존의 핸들링은 시스템 설계 영역에서 핸들링의 기능적인 설계와 서비스 실행에 집중되어 왔으나, 본 논문에서는 수준의 핸들링 특성과 비 기능적인 코스프레임 정보가 분석되었다. 따라서, 계 3의 개별자가 시스템 환경을 확장하기 나, 기존 핸들링을 사용한 시스템 개발을 추진할 경우 핸들링을 이해하고 사용 가능한 핸들링을 만드는 데는 많은 어려움이 있었다.

본 논문에서는 이러한 어려움을 극복하기 위한 방법으로 기존의 CBD 범위론에 기초하여 만들어진 핸들링에 영역 정보를 추가하는 영역지원 프로그래밍 기술이 추가된 CBD 범위론을 제안하였다. 영역 정보의 표현은 UML 디자인에 주석과 스테레오 타입을 정의하여 상세하게 기술할 수 있도록 제시하였다.
영역지향 개념이 포함된 객체 지향 프로그래밍의 개념을 중심으로 141

영역지향 개념이 포함된 CBD 방법인을 통해 객체지향 프로그래밍의 개념을 실제 적용할 수 있다. 이는 설계도의 구조를 통해 복잡한 시스템을 간단히 관리할 수 있게 한다. 

그림 6은 영역지향 개념이 포함된 객체 지향 클래스 다이어그램을 나타낸다. 

(그림 6) 영역지향 개념이 포함된 객체 지향 클래스 다이어그램
본문을 작성하여 영역 정보를 추가해 나갈 것으로서 기존 웹포
넌트의 재사용성을 중대시키고 효율적이며 경제적인 시스템
개발을 도모하고자 한다.

참고 문헌

[1] 박두환, "웹포넌트 기술발전 동향과 전망", 소프트웨어 웹포
넌트, p.4, 2002.
[2] 시사과정, "조립식 개발방법론 CBD에 대하여", Available at
http://www.sisait.co.kr/column/200105/buyers/tech-ha
n.htm.
[3] Alan W. Brown, "Large-Scale," Component-based Devel-
[4] Booch G., Kozaczynski Wojtek, "Component-Based Soft-
[5] Choi, Jung pil, "Aspect-Oriented Programming with En-
terprise JavaBeans," Enterprise Distributed Object Com-
puting Conference, EDOC 2000. Proceedings Fourth Inter-
[6] C. R. Guardes De Farias, L. Ferreira Pires, M. van Sinderen,
D. Quartel, A combined Component-Based Approach for the
Design of Distributed Software Systems, Proceedings of the
Eighth IEEE Workshop on Future Trends of Distributed
[7] Desmond Francis D’Souza, Alan Cameron Wills, "Objects,
Components and Frameworks with UML, The Catalysis™
[8] Gregor Kiczales, John Lampring, Anurag Mendhekar,
Chris Maeda, Cristina Videira Lopes, Jean-Marc Loing-
tier, John Irwin, "Aspect-Oriented Programming," In pro-
cedings of the European Conference on Object-Oriented
Programming (ECOOP), Finland. Springer-Verlag LNCS
1241, June, 1997.
[9] Ho, W. M., Pennnance, F., Jezequel, J. M. and Plouzeau,
N., “Aspect-oriented Design with the UML,” Proceedings
of the ICSE2000 Workshop on Multi-Dimensional Separ-
ation of Concerns in Software Engineering, Jun, 2000
(Limerick, Ireland).
Programming," Available at http://www.ccs.neu.edu/research/
demeter/AOP/early-def/AOP-AOP.html.
Programming and Aspect-oriented Programming(AOP),
Available at http://www.ccs.neu.edu/home/lieber/connec-
ing and Separation of Concerns," The Computer Journal, Vol.46,

김치수

e-mail : cskim@kongju.ac.kr
1984년 중앙대학교 전자계산학과(학사)
1986년 중앙대학교 일반대학원 전자계산
학과(석사)
1990년 중앙대학교 일반대학원 컴퓨터공
학과(박사)
1990년-1992 공주교육대학교 전임강사
1992년-현재 공주대학교 정보통신공학부 교수가

김태영

e-mail : youngm@gue.ac.kr
2002년 한밭대학교 컴퓨터공학과(학사)
2004년 공주대학교 대학원 컴퓨터공학과
(공학석사)

김치수 : CBD 방법론, 정보보안

김태영 : CBD 방법론, 정보보안