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 어플리케이션을 개발하는 방식의 CBD 방법론이 많이 연구되고 있다. 그러나 CBD 방법론은 시스템의 기능적인 특성을 중심으로 분할하는 경향이 많아 컴포넌트에 대한 추론, 문서화, 코드의 이해를 어렵게 하는 단점이 있다. 따라서 본 논문에서는 영역지향 프로그래밍 기술을 CBD 방법론과 UML 다이어그램에 적용하여 CBD 방법론의 단점을 보완함으로써 컴포넌트의 재사용을 용이하게 하고 시스템 개발 시간 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(Aspect-Oriented Programming), 컴포넌트개발방법론(CBD), UML

1. 서 론

CBD 방법론은 표준화되고 검증된 컴포넌트를 활용하여 소프트웨어 개발 생산성을 향상시키기 위한 방법의 하나로 소프트웨어를 부품화 하고 이를 조립·합성하여 새로운 어플리케이션을 개발한다[4]. 대표적인 컴포넌트 기반 기술의 예가 Java Beans, Enterprise JavaBeans, COM, CORBA, JViews이다[7, 11]. 그러나 대부분의 이 기술들은 시스템 하위 레벨에서 사용자 인증과 명세, 이벤트 흐름, 프로세스 뷰, 프로세스 단계 등과 같은 시스템의 기능적인 서비스에 초점을 두고 있다. 결과적으로 이러한 기술들은 상위 레벨 체계의 특성에 대해 상대적으로 결핍된 컴포넌트를 만들 수 밖에 없다. 이 기술들은 시스템 컴포넌트에 대한 사용자 인터페이스, 처리관리, 영속성, 협력과 보안 영역(aspect)등과 같은 시스템의 비 기능적인 부분과 크로스컷팅(Cross-cutting)[1]에 대한 정보가 부족한 것이 단점이다. 특히 요즈음 대부분의 기업용 비즈니스 컴퓨팅 환경이 분산 환경이고, 그 규모 또한 방대하며 이종의 운영체제와 네트워 환경이 잘 고려되어야 하는 특징을 가지고 있다. 그렇기 때문에 컴포넌트의 개발에 있어서도 특별한 상태로 컴포넌트를 재 사용하기 위한 고려 사항이 충분히 반영되어야 한다.

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

† 종신회원 : 공주대학교 정보통신공학부 교수(공학연구원)
†† 준 회 원 : 공주대학교 대학원 컴퓨터공학과
논문접수 : 2004년 3월 23일, 심사완료 : 2004년 8월 23일

1) 복잡도가 높은 시스템일수록 동기화, 예외/에러 처리, 보안, 영속성, 자원 공유, 배포, 메모리 관리 등이 고려되어야 하는데, 이러한 시스템 속성들이 한, 두개의 클래스에 지역화 되지 않고 시스템을 구성하는 모듈의 여기저기에 흩어져 있는 현상.

하고 시스템 개발 시간 단축과 개발비용의 감소를 유도하고자 하였다.

2. 관련 연구

2.1 영역지향 프로그래밍(AOP : Aspect-Oriented Programming)

영역의 개념은 절차 중심 프로그래밍 기술과 객체 지향 프로그래밍 기술에서 충분히 처리될 수 없는 문제들을 다루기 위해 소프트웨어 개발 분야에서 소개되었다. 1999년 5월에 Gregor Kiczales(Xerox PARC[Palo Alto Research Center]의 수석 연구원)와 그의 그룹에 의해 정의되었으며, 시스템의 총체적인 레벨 체계를 수직으로 분할함으로써 시스템의 기능과 사용자 서비스에 집중하였던 기존의 종단적인 모듈(컴포넌트)단위 구조를 수평으로 자르는 디자인 모듈장치라고 정의하였다[8].

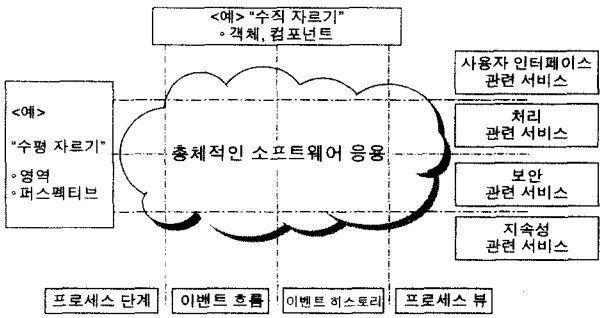
영역지향 프로그래밍은 영역이라는 새로운 프로그램 구조를 정의해 사용하고 있으며, 이 구조는 스트럭처, 클래스, 인터페이스 등과 같은 특정한 용도의 구조와 같다. 또한 영역 내에 프로그램의 여러 모듈들에 흩어져 있는 기능(하나의 기능이 여러 모듈에 흩어져 있었던 것)을 모아서 정의하게 된다. 결국 어플리케이션의 각 클래스는 자신에게 주어진 고유 기능만을 수행하고, 추가된 각 영역들이 횡단적인 기능들을 모아 처리함으로써 크로스컷팅 문제를 해결하면서, 전체적인 프로그램을 이루는 형태를 만들게 된다. 여기서 횡단적인 기능이란 사용자 인터페이스, 처리, 지속성, 암호화, 배치, 메모리 관리, 협력 등과 같은 비 기능적인 제약이나 수준 높은 서비스들을 의미한다. 그리고 영역은 객체의 캡슐화 원칙에 위배되나 제한된 형태로만 사용하기 때문에 영역은 객체의 Private 영역까지 접근할 수 있지만, 다른 클래스의 객체 사이의 캡슐화에는 영향을 주지 않는다.

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

3. 영역지향 프로그래밍 기술을 적용한 CBD 방법론과 UML 다이어그램

영역지향 프로그래밍 기술을 적용한 CBD 방법론은 현저히 높은 재사용성과, 뛰어난 신장성, 동적인 순응성이 존재하는 소프트웨어 컴포넌트를 개발하기 위해 본 논문에서 제시한 CBD 방법론이다. 영역지향 프로그래밍 기술을 적용한 CBD 방법론은 소프트웨어의 전체적인 개발 과정에 적용되어 진다.

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



(그림 1) 일반적인 개념의 컴포넌트와 영역 개념이 포함된 컴포넌트

영역지향 프로그래밍 기술을 적용한 CBD 방법론에서의 영역은 컴포넌트의 기능적인 정보들을 담을 수 있을 뿐만 아니라, 컴포넌트들의 비 기능적인 제약 사항도 담을 수 있다. 또한 컴포넌트의 크로스컷팅 문제 해결을 위해 더욱 효과적인 특징과 구현 방법을 지원함으로써 개발자와 사용자들에게 컴포넌트들에 대한 지식을 풍부하게 제공하는데 초점을 둔다.

보통 일반적인 영역은 사용자 인터페이스, 처리, 지속성, 암호화, 배치, 메모리 관리, 협력 등을 포함한다. 높은 수준의 보안을 요하는 시스템은 여러 가지 추가 보안에 관한 영역을 가지며, 특히 중복과 인증에 높은 영역을 가진다. 시스템 내의 각 컴포넌트에 대한 영역 정보는 별도의 표를 이용하여 상세하게 기록되어지며, 이것은 제3의 컴포넌트나 시스템 개발자에게 컴포넌트의 재사용을 위한 명확한 정보를 제공해 주게 된다.

영역지향 프로그래밍 기술을 적용한 CBD 방법론은 컴포넌트 개발자에게 다양한 체계의 영역정보로부터 컴포넌트간의 상호작용에 대한 추론을 풍부하게 하고, 위의 (그림 1)에서와 같이 시스템을 수평으로 자를 수 있도록 해 준다. 이를 위해 컴포넌트 개발자는 개발 시 응용 컴포넌트가 가질 수 있는 크로스컷팅과 관련된 주요 시스템 레벨을 자세하게 기술해야 하고, 적절한 영역 세부 사항과 높은 수준의 컴포넌트 정보, 컴포넌트의 복구와 내부 검색 자료 교환을 위한 영역 특성들을 자세히 기술하여야 한다.

영역지향 프로그래밍 기술이 적용된 CBD 방법론의 장점으로는 첫째, 컴포넌트들이 다양한 전망(영역), 구조적 컴포넌트 요구사항과 설계의 풍부함을 제공할 수 있게 한다. 둘째, 동적인 구성과 독립된 컴포넌트 상호작용을 더욱더 잘 실행할

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 요구들을 입증하고 지정하기 위한 것인데, 이것은 시스템의 중요한 부분과 관련되어 진다. 보통 시스템 컴포넌트는 크로스컷팅 문제 해결을 위한 서비스를 제공하기 원하거나 혹은 필요로 한다. 예를 들면, 컴포넌트 개발자는 컴포넌트의 기능적이거나 비 기능적인 면과 관계된 사용자 인터페이스, 처리, 지속성, 보안등에 관해 명확히 하기를 원하며, 또한 이를 문서화하기를 원할 것이다. 컴포넌트의 요구사항 명세를 규정하는 동안 개발자는 어떤 컴포넌트들은 많은 영역 서비스들을 갖고 있을 수 있고 어떤 컴포넌트들은 약간의 영역 서비스들을 갖질 수 있다는 것을 알게 될 것이다. 더욱이 어떤 컴포넌트들은 영역 서비스를 공유할 수 있음도 알게 될 것이다. 영역지향 컴포넌트의 요구사항 명세는 컴포넌트를 필요로 하는 기술자들에게 관련된 컴포넌트 속성들을 이해시키는데 매우 유용하다.

3.2 영역지향 컴포넌트 설계

요구사항을 명세 하는 동안 개발자들은 영역 체계들로부터 기능적이고 비 기능적인 제약들을 확인할 수 있는데 이러한 제약들과 영역들은 영역을 갖는 컴포넌트 설계로 정련되는데 사용된다. 요구사항 수준의 세부적인 영역들은 소프트웨어로 세분화된 디자인 수준의 컴포넌트 영역들로 정련될 수 있는데 이것들은 디자인 수준의 컴포넌트 영역 서비스들을 특징짓는다. 예를 들면, 세분화된 디자인 단계의 영역 명세들은 사용자 인터페이스, 컴포넌트 지속성, 분배, 보안과 처리 모델, 컴포넌트 구성 등이다.

설계 단계에서 영역의 세부 명세는 컴포넌트 구현 설계 기술을 연상하게 한다. 설계 단계에서 컴포넌트 서비스들은 관련된 영역을 문서화시키고, 설계자가 다양한 추론과 설계의 방향을 쉽게 수립할 수 있도록 하기 위해 다른 것들과 겹쳐진 영역들까지도 문서화한다.

3.3 영역지향 컴포넌트 구현과 런타임

영역지향 프로그래밍 기술을 적용한 CBD 방법론을 사용해서 설계된 컴포넌트들은 일반적인 컴포넌트 기반 구현 기술이나 Enterprise JavaBeans™, JViews 등과 같은 컴포넌트 기반 프레임웍을 사용함으로써 실행될 수 있다. 엔터프라이즈 자바빈즈는 시스템 레벨 서비스들을 제공한다. 컴포넌트 영역은 인터페이스, 언어 반사 또는 디자인 패턴을 통해 실행 될 수 있다. 영역을 가지고 실행되는 컴포넌트는 각각의 다른 기능을 접근하고, 컴포넌트 상호간의 인터페이스 정의를 안내할 수 있게 하기 위해 관련된 컴포넌트들에 대한 테크닉을 제공할 수 있다. 하나의 컴포넌트 안에서 실행되는 영역 서비스들은 다른 컴포넌트들이나 최종사용자에 의해 런 타임에서 사용될 수 있으며, 런 타임 동안 컴포넌트의 능력을 높이는데 사용된다.

3.4 영역지향 컴포넌트 테스팅

영역지향 CBD 방법론은 컴포넌트 요구사항과 명세, 설계, 구현과 런 타임 전개 단계에서 적용되었다. 그러나 영역 서비스를 가지고 실행되는 컴포넌트가 다른 응용으로 개발되거나 한 시스템의 부품으로 사용되었을 때 컴포넌트 요소의 영역이 얼마나 잘 정의되고, 명세 되었으며, 설계되었는지를 테스트해 볼 필요가 있다. 그러므로 컴포넌트 영역들에 대한 시험 계획과 전략들을 만들어 낼 필요성이 있다. 그것은 컴포넌트 영역 기술자를 규정하고, 특별한 테스팅 에이전트를 만들어 사용함으로써 가능할 것이다.

3.5 UML 다이어그램에서의 영역정보 표현

기존의 UML 다이어그램은 시스템의 기능적인 특성에만 초점을 맞추어 작성하였기 때문에 시스템에 대한 높은 수준의 세부 정보와 제약 사항들이 나타나 있지 않다. 그러나 영역지향 프로그래밍 기술이 포함된 UML 다이어그램에서는 시스템에 대한 높은 수준의 세부 정보와 제약 사항을 UML 확장 수단인 스테레오타입과 주석을 사용하여 표현하고 있는 것이 기존의 CBD 방법론을 적용한 UML 다이어그램과 다르다.

4. 적용 사례

지금까지의 일반적인 CBD 기반의 컴포넌트는 기능 중심적인 수직 분할에 중점을 두었기 때문에 특히 분산환경에서 기존 컴포넌트를 사용해서 대형 시스템을 개발하고자 할 때에는 높은 수준의 시스템 정보와 제약 사항들이 나타나 있지 않아 시스템 개발에 어려움이 많았다.

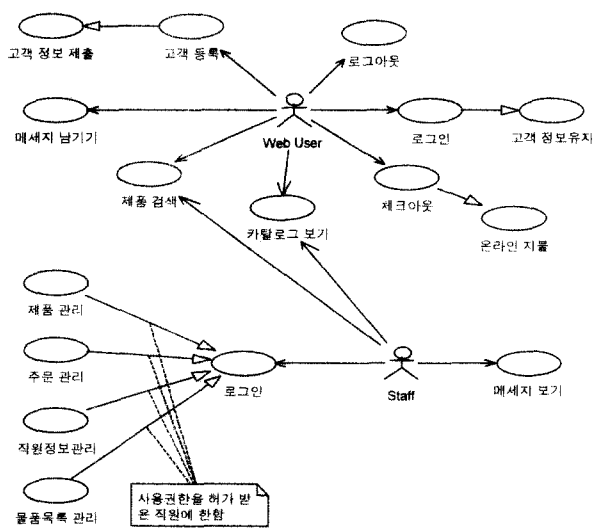
따라서 본 논문에서는 영역 지향 프로그래밍 기술을 추가한 CBD 방법론을 적용하여 시스템의 비 기능적인 정보와 제약사항들을 포함하여 총체적인 시스템을 수직과 수평으로 동시에 분할하였다. 영역은 사용자 인터페이스, 지

속성, 처리, 실행과 보안 같은 시스템의 비 기능적인 분해에 의해 식별되는 많은 컴포넌트에 전형적으로 영향을 미친다.

이 장에서는 컴포넌트에 기반을 둔 온라인 가전제품 판매 시스템을 하나의 적용사례로 영역지향 프로그래밍 기술을 추가한 CBD 방법론을 적용하여 설계하였다.

4.1 유즈케이스 모델링

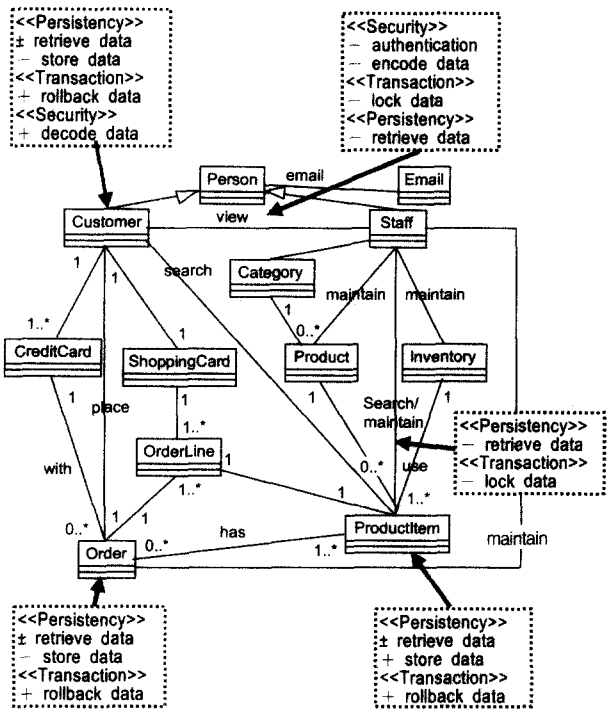
유즈케이스 다이어그램 내에서는 시스템 사용의 사례들을 그려 놓은 것이기 때문에 별도로 영역정보에 대한 특별한 표현은 필요로 하지 않는다. (그림 2)는 온라인 가전제품 판매시스템의 유즈케이스 다이어그램을 나타낸 것이다.



(그림 2) 온라인 가전제품 판매시스템의 유즈케이스 다이어그램

4.2 클래스 모델링

기존의 클래스 다이어그램은 크로스컷팅 상위레벨 체계즉, 사용자 인터페이스, 보안, 처리, 지속성 영역 정보와 같은 정보들이 결여되어 있다. 그리고 이 영역들이 클래스 다이어그램이나 다른 UML 다이어그램에서 문서화되어지지 않았다. 더욱이 이 영역 정보가 때로는 더 많은 중요한 시스템 구조와 관련 특성을 가지고 있을 수 있다. (그림 3)은 영역 정보를 포함한 클래스 다이어그램인데 개발자로 하여금 주석을 달아 고객 클래스가 영속성, 처리, 보안에 관한 영역을 필요로 하고 있음을 훨씬 분명하게 알려주고 있다. 주석으로 영역정보를 표현해 주기 위해서는 먼저 영역 세부사항들을 정의하여 4.3장의 <표 1>에서 보는 바와 같이 문서화가 선행되어야 하나, 이 부분에서는 생략하였다.



(그림 3) 영역지향 개념이 포함된 클래스 다이어그램

〈표 1〉 체크아웃에 대한 시퀀스 다이어그램에서의 영역 정보

영 역	영역 세부 사항 및 속성과 간단한 추론
<<User Interface>>	<<+ provided>> 고객 정보/신용카드 정보 뷰 현재 처리 정보와 피드백 정보를 웹 사용자에게 보여준다. 이 서비스는 온라인 쇼핑 고객 같은 웹 사용자에 의해 요구되어 진다.
	<<– required>> 프레임/폼 세부 사항 ,속성은 GUI 컴포넌트나 HTML 태그일 것이며, HTML과 JSP, 스윙 컴포넌트에 의해 제공될 수 있다.
	<<+ provided>> 응답 시간 5초 미만, 웹 사용자에 의해 요구되어 진다.
<<Transaction>>	<<+ provided>> 데이터 송신/수신 운반 데이터에 트랜스포트 서비스가 필요하며, 이것은 코바/RMI와 같은 미들웨어 시설을 통하여 웹/랜 접속에 의해 제공될 것이고, 전송 속도가 제한되어질 것이다.
	<<– required>> 락/롤백 데이터 병렬 접근을 피하고 에러 핸들링을 위해 사용한다.
<<Persistency>>	<<+ provided>> 저장/회복 데이터 지속성과 회복에 객체들의 정보를 필요로 한다. 한꺼번에 저장할 수 있는 최대 저장 크기는 최고 10MB로 제한한다.
	<<– required>> 저장 매체 파일이나 데이터베이스이다.
<<Security>>	<<+ provided>> 접근 인증 사용자 체크아웃을 위해서는 로그인이나 올바른 접근을 획득하기 위한 등록을 필요로 한다. 세부사항 속성은 패스워드 인증 마스크이다.
	<<– required>> 부호화/복호화 데이터 64 베이스 암호화/복호화 또는 암호화 알고리즘을 사용한다. 서비스는 암호화/복호화 컴포넌트 또는 암호화 패키지 프로바이더에 의해 제공될 수 있다.

4.3 동적 객체 상호작용 모델링

객체 상호작용 모델링은 실행, 지속성, 보안, 처리, 배포 영역 정보 등과 같은 비 기능적인 영역에 관한 정보의 결여로부터 클래스, 오퍼레이션, 서비스와 그룹 컴포넌트를 쉽게 발견하는데 매우 큰 도움을 준다. 다음은 영역 정보를 포함한 객체 상호작용 다이어그램에 대해 설명한 것인데 여기에서는 <표 1>과 같이 체크아웃 시퀀스 다이어그램에 대한 영역 정보와 (그림 4)와 같이 다이어그램만을 세부적으로 설명하였다.

- 체크아웃의 유즈케이스 진행 단계 -

전제조건 - 고객이 체크아웃 버튼을 누른다.

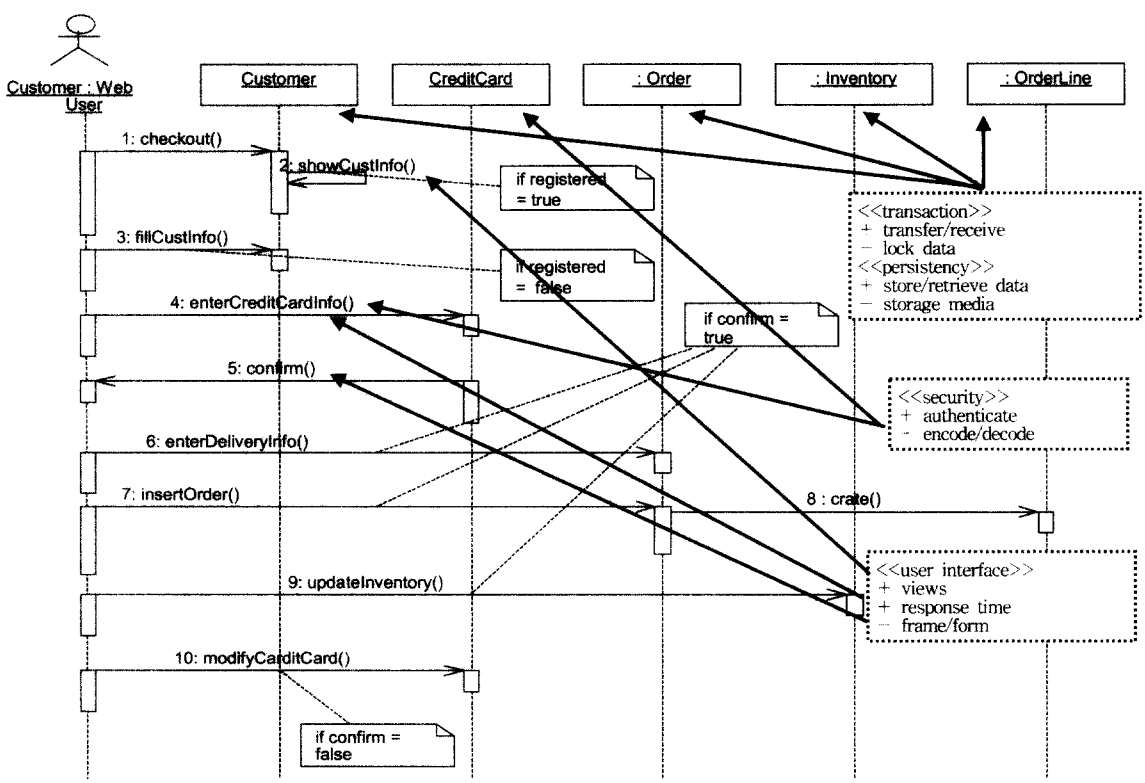
1. 고객 정보를 보여준다.
   1.1 만일, 고객이 등록되어져 있다면, 등록된 정보를 표시한다.
   1.2 만일, 고객이 등록되어져 있지 않다면, 고객 정보 서식을 채운다.
2. 이미 제출되어진 고객 정보가 있다면, 신용카드 정보를 묻고 신용카드를 확인하기 위해 기다린다.
   2.1 만일, 카드가 유효한 것이라면 신용카드 정보를 채운다.
      2.1.1 데이터베이스에 주문을 등록하라.
      2.1.2 데이터베이스의 목록 정보를 갱신한다.
   2.2 만일, 신용카드 확인에 실패하면, 2.1로가 신용카드 정보를 수정한다.

4.4 컴포넌트 디자인 명세

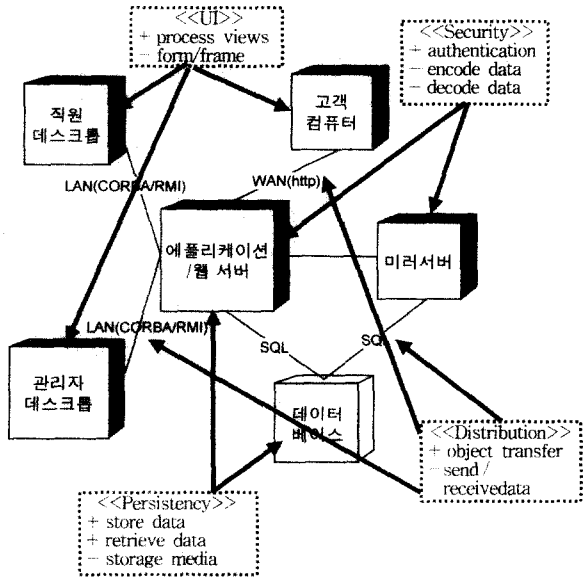
일반적인 컴포넌트 명세와 디자인은 컴포넌트를 낮은 수준의 기능적인 특성들로 분해하였기 때문에 컴포넌트의 높은 수준의 체계와 비 기능적인 영역 정보는 전혀 확인할 수 없었다. 그러나, 영역정보의 제공은 컴포넌트에 대한 일반적인 특성을 부여하고, 색인을 붙이는 것보다 더 유용하다.

4.4.1 배치 모델링

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



(그림 4) 영역 분석을 포함한 체크아웃 시퀀스 다이어그램



(그림 5) 영역지향 개념이 포함된 배치 다이어그램

〈표 2〉 영역 분석을 포함한 배치 모델링의 영역 정보

영 역	영역 세부 사항 및 속성과 간단한 추론
<<Security>>	<<+ provided>> 접근 인증 세부 사항 특성은 패스워드 인증 마스크이다. 웹 GUI 컴포넌트나 어플리케이션 컴포넌트는 고객 또는 직원 로그인을 식별하는 서비스를 요구한다.
	<<- required>> 부호화/복호화 데이터 64 베이스 부호화/복호화 스키마 또는 암호화 알고리즘을 사용한다. 서비스는 부호화/복호화 컴포넌트 또는 암호화 패키지에 의해 제공된다.
<<User Interface>>	<<+ provided>> 처리/피드백 뷰, 사용자 정보 표시 어플리케이션 GUI와 웹 고객 GUI 컴포넌트에 의해 요구되어 진다.
	<<+ provided>> 시스템 응답 세부 사항 속성은 사용자 뷰이며, 다양한 영역에 의해 공유할 수 있다. 강제적인 응답 시간은 5초 미만이다.
	<<- required>> 폼/프레임 스윙 컴포넌트와 같은 HTML 태그 또는 GUI 컴포넌트에 의해 공급되어 진다.
<<Distribution>>	<<+ provided>> 객체 이동 배포는 웹/랜 접속, 코바/RMI/SQL 프로토콜을 사용한다. 서비스는 미들웨어에 의해 제공된다.
	<<- required>> 송/수신 데이터 데이터 처리 능력은 SQL문을 사용한 JDBC 같은 데이터베이스 연결에 의해 제공되어 진다.
Persistency	<<+ provided>> 저장/회복 데이터 서비스는 GUI 컴포넌트에 의해 정보를 전달하거나, 최신의 것으로 갱신하기 위해 사용된다. 서버 컴포넌트는 데이터베이스가 같은 서비스를 제공하는 것을 요구할 것이다.
	<<- required>> 저장 매체 세부 사항 특성은 파일이나 데이터베이스이다. 최소한 1GB보다 큰공간을 필요로 한다.

4.4.2 고객 컴포넌트 설계

<표 3>은 고객 컴포넌트의 영역 정보를 상세하게 설명하고 있으며, (그림 6)은 고객 컴포넌트 클래스 다이어그램에서 영역정보가 어떻게 표현되었는지를 보여주고 있다.

〈표 3〉 고객 컴포넌트의 영역 정보

영 역	영역 세부 사항 및 속성과 간단한 추론
<<Security>>	<<+ provided>> 접근 인증 세부 사항 특성은 패스워드 인증 마스크이다. 웹 GUI 컴포넌트는 고객 로그인을 확인하기 위한 서비스를 필요로 한다.
	<<- required>> 부호화/복호화 데이터 64 베이스 부호화/복호화 또는 암호화 알고리즘을 사용한다. 서비스는 부호기/복호기 컴포넌트 또는 암호화 패키지 프로바이더에 의해 제공될 수 있다.
<<Distribution>>	<<+ provided>> 객체 이동 배포는 웹/랜 접속, 코바/RMI/COM 프로토콜을 사용한다.
	<<- required>> 송/수신 데이터 데이터 처리 능력은 JDBC와 같은 데이터베이스 접속에 의해 제공 된다.
	<<- required>> 고객 정보 업데이트에 대한 처리
<<Persistency>>	<<+ provided>> 저장/회복 데이터 세부 사항 특성은 삽입, 갱신, 선택, 삭제이다. 서비스는 웹 GUI 컴포넌트에 의해 정보를 전달하거나, 최신의 것으로 갱신하기 위해 사용된다. 고객 컴포넌트는 데이터베이스가 같은 서비스를 제공하는 것을 요구 할 것이다. 서비스는 10MB까지 데이터 크기를 제한 한다.
	<<- required>> 저장 매체 세부 사항 특성은 파일 또는 데이터베이스이며, 최소 1GB보다 큰 공간을 필요로 할지도 모른다.

5. 결론 및 향후 연구

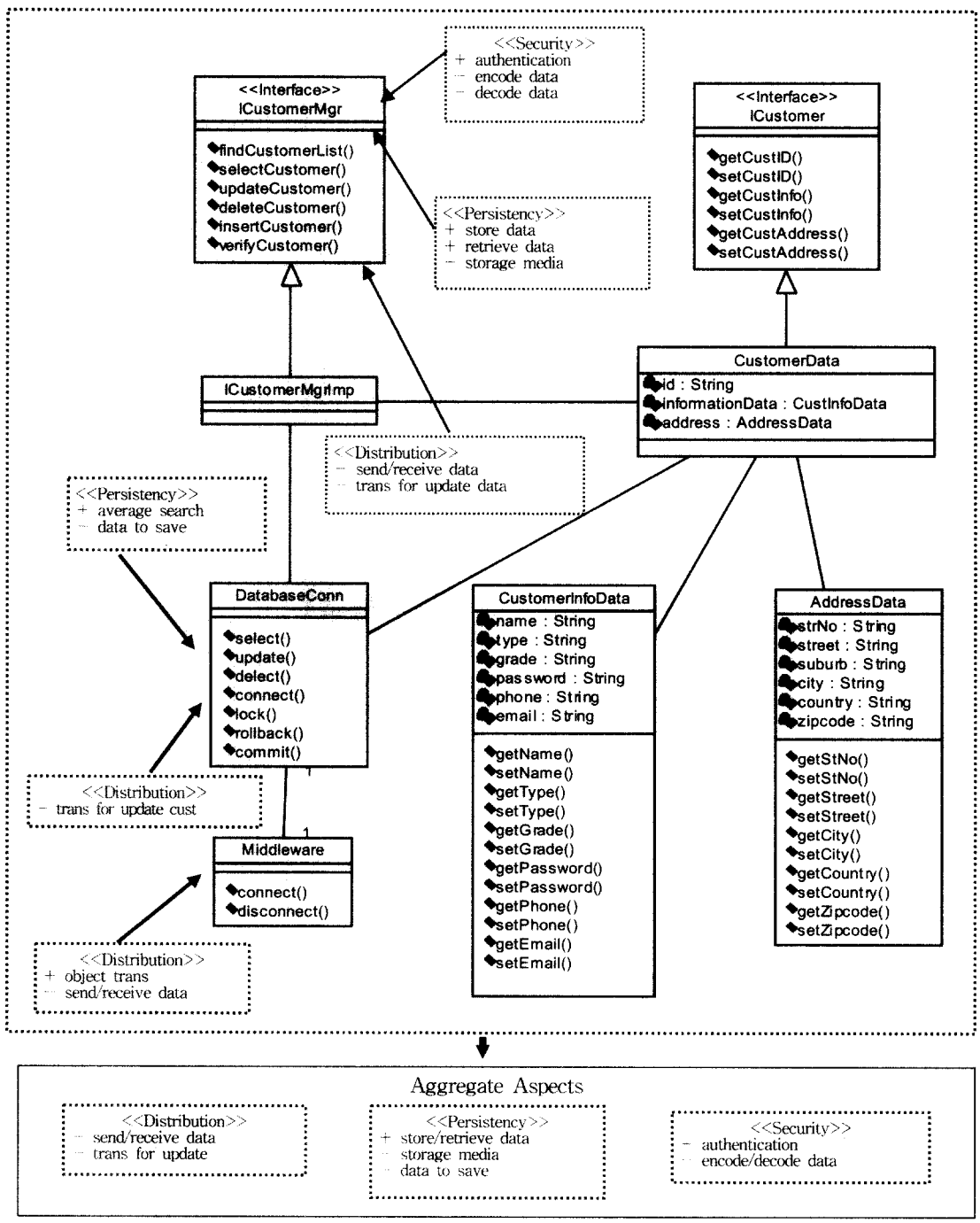
기존의 컴포넌트는 시스템 설계 영역에서 컴포넌트의 기능적인 설계와 서비스 실행에 집중되어 짐으로 인해 높은 수준의 컴포넌트 특성과 비 기능적인 크로스컷팅 정보가 부족하였다. 따라서 제 3의 개발자가 시스템 환경을 확장하거나, 기존 컴포넌트를 사용한 시스템 개발을 추진할 경우 컴포넌트를 이해하고 사용 가능한 컴포넌트로 만드는 데는 많은 어려움이 있었다.

본 논문에서는 이러한 어려움을 극복하기 위한 방법으로 기존의 CBD 방법론에 기초하여 만들어진 컴포넌트에 영역 정보를 추가하는 영역지향 프로그래밍 기술이 추가된 CBD 방법론을 제안하였다. 영역 정보의 표현은 UML 다이어그램에서 주석과 스테레오 타입을 정의하여 상세하게 기록할 수 있도록 제시하였다.

영역지향 개념이 추가된 CBD 방법론을 적용하여 컴포넌트를 개발할 경우 그 이점은 첫째, 프로그램 코드의 이해와 문서화를 쉽게 하고 높은 수준의 추론을 제공한다. 둘째는 컴포넌트 내부의 관계와 그룹핑, 색인에 대한 추론을 위해 사용될 수 있다. 따라서 컴포넌트의 재사용과 재가공을 쉽게 하는데 많은 도움이 될 것이며, 시스템 개발에 드는 노력과 비용의 경감을 가져오는 결과를 얻을 수 있다.

현재 대부분의 시스템들이 대규모적이면서 복잡도가 증가하고 있는 것이 사실이며, 분산환경으로의 전환이 증가하고 있는 추세이다. 이런 시점에서 본 논문을 통해 제안한 영역지향 프로그래밍 기술이 추가된 CBD 방법론이 소프트웨어 시스템 개발자에게 시스템의 기능적인 특성과 함께 유익하고 풍부한 영역 정보를 제공하여 컴포넌트의 조립과 재사용성을 높이는데 많은 유익을 가져다 줄 수 있을 것으로 예상된다.

향후 과제로는 이미 만들어져 사용되고 있는 재사용 가능 컴포넌트들도 영역지향 프로그래밍 개념이 추가된 CBD 방



(그림 6) 영역지향 개념이 포함된 고객 컴포넌트 클래스 다이어그램

법론을 적용하여 영역 정보를 추가해 나감으로써 기성 컴포넌트의 재사용성을 증대시키고 효율적이며 경제적인 시스템 개발을 도모하고자 한다.

참 고 문 헌

[1] 배두환, "컴포넌트 기술발전 동향과 전망", 소프트웨어 컴포넌트, p.4, 2002.

[2] 시사컴퓨터, "조립식 개발방법론 CBD에 대하여", Available at http://www.sisait.co.kr/column/200105/buyers/tech-han.htm.

[3] *Alan W. Brown*, "Large-Scale," Component-based Development, 2000.

[4] *Booch G., Kozaczynski Wojtek*, "Component-Based Software Engineering," IEEE Software, pp.34-36, October, 1998.

[5] *Choi, Jung pil*, "Aspect-Oriented Programming with Enterprise JavaBeans," Enterprise Distributed Object Computing Conference. EDOC 2000. Proceedings Fourth International, pp.252-261, 2000.

[6] *C. R. Guareis 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 Computing System, 2001.

[7] *Desmond Francis D'Souza, Alan Cameron Wills*, "Objects, Components and Frameworks with UML, The Catalysis™ Approach," Addison Wesley Longman Inc., 1999.

[8] *Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Videira Lopes, Jean-Marc Loingtier, John Irwin*, "Aspect-Oriented Programming," In proceedings of the European Conference on Object-Oriented Programming (ECOOP), Finland. Springer-Verlag LNCS 1241, June. 1997.

[9] *Ho, W. M., Pennaneach, F., Jezequel*, J. M. and Plouzeau, N., "Aspect-oriented Design with the UML," Proceedings of the ICSE2000 Workshop on Multi-Dimessional Separation of Concerns in Software Engineering, Jun, 2000 (Limerick, Ireland).

[10] *Karl J. Lieberherr*, "Early Definition of Aspect-Oriented Programming," Available at http://www.ccs.neu.edu/research/demeter/AOP/early-def/AP-AOP.html.

[11] *Karl Lieberherr*, Connections between Demeter/Adaptive Programming and Aspect-oriented Programming(AOP), Available at http://www.ccs.neu.edu/home/lieber/connection-to-aop.html, 1999.

[12] *Rashid, A., Blair, L.*, "Editorial : Aspect-oriented Programming and Separation of Crosscutting Concerns," The Computer Journal, Vol.46, No.5, pp.529-541, 2003.

김 치 수

e-mail : cskim@kongju.ac.kr

1984년 중앙대학교 전자계산학과(학사)

1986년 중앙대학교 일반대학원 전자계산학과(석사)

1990년 중앙대학교 일반대학원 컴퓨터공학과(공학박사)

1990년~1992 공주교육대학교 전임강사

1992년~현재 공주대학교 정보통신공학부 교수

관심분야 : 객체지향 분석 및 설계, CBD 방법론

김 태 영

e-mail : youngimk@gjue.ac.kr

2002년 한밭대학교 컴퓨터공학과(학사)

2004년 공주대학교 대학원 컴퓨터공학과(공학석사)

관심분야 : CBD 방법론, 정보보안