

## 특집

## MSF/CD 방법론 기반의 분석/설계 틀 설계에 관한 연구

이승호<sup>1)</sup> 박병형<sup>2)</sup> 김영희<sup>3)</sup>

## 목 차

1. 서 론
2. 관련연구
3. 결론 및 향후 과제

## 1. 서 론

1960년대 말 Boem은 이미 소프트웨어 위기를 예고하였다. 즉, 하드웨어 발전에 비해 상대적으로 소프트웨어가 미치지 못할 것이라는 예측은 적중하였다. 수 많은 소프트웨어 개발 방법론과 품질향상 그리고 이를 지원한 수 많은 개발 도구들이 등장하였고 잠시 개발자들에게 관심을 끌긴 하였지만 기대에 미치지 못하였고, 또 다시 수많은 소프트웨어 엔지니어들은 새로운 대안을 찾기에 전전 공공하고 있는 것이다. 21세기에 들어서자 인터넷이라는 획기적인 인프라가 정보통신은 물론 사회 전 분야에 널리 활용되면서 소프트웨어 분야에도 엄청난 변화를 예견하고 있었던 것이다. 이제 수 많은 전문 소프트웨어 개발자가 제한된 공간에서 제한된 개발 방법으로 특수를 누리는 시대가 아닌 일반인들도 누구나 쉽게 소프트웨어를 접근하는 시대가 인터넷을 통해 강하게 점화되고

있다. 다시 말해 소프트웨어는 만드는 것이 아니라 소프트웨어를 유통(인터넷)을 통해 구매를 하고 그 소프트웨어는 엄격한 표준과 품질을 거쳐 판매가 되는 것이다.

또한 소프트웨어도 더 이상의 완성된 제품을 판매하는 것이 아니라 다양한 템플릿 형태를 갖춘 부품 형식의 반제품 형식으로 고객들에게 다가서고 있다. 최근에 국내외에서 조립형 즉, 컴포넌트 소프트웨어가 키워드로 등장하고 있고 “한국 컴포넌트 컨소시엄”을 주축으로 주요 IT공급사, 대형 SI 업체, 소프트웨어 전문 업체들이 활발한 활동을 벌이고 있다.

컴포넌트 소프트웨어 !

이를 개발하기 위해서는 과거 전통적인 개발방법론이 아닌 CBD(Component Based Development) 방법론을 기반으로 소프트웨어를 개발하여야 하며 컴포넌트 형태의 소프트웨어를 조립과정을 거치면서 고객에게 완성된 제품 형태로 제공된다는 것이다.

최근 SUN이 자바를 기반으로 SUNONE을 통해 종합적인 지원체제를 갖추어 일찌기 시장을 선점한 반면에 Microsoft사에는 닷넷이라는 통합환경과 XML 및 C#이라는 강력한 객체지향적인 프로

1) (주)이지시스템 대표이사

2) 호서대학교 벤처전문대학원 컴퓨터응용기술학과 박사과정

3) (주)케미스 R&D

그럼 언어를 통해 웹서비스를 주창하고 있다. 여기에 MSF/CD라는 컴포넌트 개발방법론을 소개하였다.

본 고에서는 MSF/CD 개발방법론을 기반으로 컴포넌트 분석/설계 그리고 이를 조립하여 완성된 Application을 구축하는 개발도구를 소개하고자 한다.

웹 서비스는 인터넷을 기반으로 하여 XML 표준화에 의한 정보통신의 통합을 의미하며 일반 사용자는 게임기, 휴대폰 등에서 기업 활동에 필요한 모든 분야를 모아 이를 컴포넌트화 하여 높은 차원의 서비스를 제공하는 데 있다.

컴포넌트란, 독립적이고 부품화 되어 이를 쉽게 조립할 수 있도록 그 조립성을 유지하여 재사용을 쉽게 하는 대체성을 특징으로 가지고 있어 향후 소프트웨어의 유일한 대안으로 대두되고 선진국은 물론 국내에서도 최근 새로운 프로젝트는 CBD를 적용하고 있는 실정이다.

여기서 마이크로 소프트사의 개발방법론인 MSF/CD(Microsoft Solutions Framework/Component design)를 활용하여 CBD 개발 현장에서 실질적인 컴포넌트를 구현한다.

이 컴포넌트 설계는 Object와 Service를 찾아내고 그에 따른 절차와 과정 그리고 구현 방법에 대한 지침을 쉽게 정의하는데 이것은 다른 CBD 방법론과 유사하게 구성되어 있는데, 개념설계, 논리설계 그리고 물리설계로 구성되어 있다.

MSF/CD는 윈도우 DNA(인터넷 분산 아키텍처, Distributed iNternet Architecture)에 기초한 3 tier환경하에서 크게 두가지 요소기술을 적용하여 구현된다.

객체지향 분석설계 부분은 MSF/CD방법론의 Object Oriented Analysis and Design 방법론이며 공동 객체모델 부분의 가동 환경인 COM+이다.

이러한 마이크로 소프트의 CBD가 실제 프로젝트

에 적용되는 단계를 살펴보면

첫째, MSF/CD에 기초하여 오브젝트를 설계 및 정의를 한다.

둘째, Visual Basic, Visual C++, C# 등의 언어를 활용하여 오브젝트(COM 컴포넌트)를 물리적으로 구현한다.

셋째, 물리적으로 구현된 COM 컴포넌트가 실행되기 위한 최적화된 환경으로서의 어플리케이션 서버인 COM+와 연결을 한다.

여기서 COM+에 대하여 알아보기로 하자.

COM 컴포넌트의 가동 환경에서 이는 윈도우 2000서버에 하나의 서비스로 내장되어 있다. 3 tier 중에서 비즈니스 서비스 계층인 Middle tier에 해당하는 어플리케이션 서버이며 MSF/CD에서 설계된 비즈니스 로직을 담은 비즈니스 오브젝트 및 데이터베이스 수행을 위한 데이터 오브젝트가 등록되어 동작하게 된다.

이는 마치 IBM CICS(Customer Information Control System), Entera, Tuxedo, Tmax 등과 같은 미들웨어에 해당하는 것으로 트랜잭션의 안정적인 처리, 데이터의 정합성 유지를 위함이다.

지금부터 마이크로 소프트사의 CBD 개발방법론인 MSF/CD를 활용한 분석/설계 도구에 대하여 알아보기로 하자.

## 2. 관련연구

### 2.1 컴포넌트 개발도구 개발배경

지금까지 우리는 컴포넌트에 대한 정의, 기술적 배경 그리고 시장 현황 등에 대하여 1-3장에서 확인을 하였다.

본 장에서는 컴포넌트에 대한 활용분야를 실제로 운영하는 개발도구를 중심으로 소개하여 독자로서 하여금 다양한 컴포넌트 개발도구에 대하여 피부로 느끼는 장으로 소개하고자 한다.

기업의 환경과 프로세스의 모델이 매우 복잡해지고 있는 것이 일반적이라 할 수 있다. 사람의 능력에는 한계가 있어 생각하는 것을 말로 혹은 글로 표현하기란 무척이나 어려운 일이 아닐 수 없다. 그래서 일찌기 사람들은 의사소통의 원활화를 위해 그림을 그려 대화를 하였고 나아가 컴퓨터 시스템 구축도 그림(모델링)을 통해 시스템의 전반적인 모습을 한눈에 볼 수 있도록 하는 모델링 기법을 개발하게 되었다.

CBD(Component Based Development) 방법론 이전에 IE(Information Engineering) 방법론에서도 DFD(Data Flow Diagram)를 통해 분석된 정보를 체계적으로 도식화 하여 의사소통을 원활하게 하였다.

CBD기반의 모델링 기법은 개발자가 사용자의 요구사항을 시스템으로 설계해 가는 과정을 체계적으로 만들어 주고 시스템의 아키텍처를 견고하게 설계하도록 지원해준다. 이렇게 하여 설계된 아키텍처는 개발절차의 효율성을 높여주고 사용자의 요구에 능동적이고 탄력적으로 대응할 수 있도록 한다. 또한 모델링을 통하여 개발자와 사용자간의 의사소통의 연결고리가 만들어지게 된다. 이러한 표기법은 표준화된 규약에 의해서 그려져야 하며 단순히 그림을 그리는 기능을 확대하여 시스템 개발에 다양한 프로젝트 관리를 할 수 있는 기법을 제공하여야 한다. 이러한 종합적인 기능을 제공할 때 비로서 시스템 개발에 생산성향상과 신뢰가 높은 시스템 개발 그리고 유연한 시스템을 가지게 된다.

그러면 Microsoft사의 MSF/CD를 중심으로 컴포넌트 분석/설계도구(Aurora.net)에 대하여 알아보기로 하자.

## 2.2 Aurora.net이란 ?

### 2.2.1 정의

Aurora.net는 객체지향적(Object Oriented)

개발 방법론을 적용하여 시스템을 개발할 때 소프트웨어 시스템을 분석/설계하는 데 사용하는 비주얼 모델링 도구라고 정의할 수 있다.

기본적으로 모델/뷰(Model/View) 구조에 근거하여 소프트웨어 시스템을 설계하고 프로그램코드를 자동으로 생성해 가는 반복개발(Iterative Process)을 위한 제반 환경을 제공한다.

기존의 UML 모델링 솔루션들인 Rational Rose나 Together가 J2EE 환경의 EJB를 만드는데 그 기능을 하였다면, Aurora.net Designer와 같은 제품의 경우에는 MS의 .NET환경에서 운영되는 컴포넌트인 COM+를 설계, 생성하는 유일한 제품이라고 할 수 있다.

### 2.2.2 특징

첫째, CBD(Component Based Development)를 지원한다. 소프트웨어 개발 방법론의 신기술인 CBD 공학 기법을 지원하여 IT 최대의 이슈인 컴포넌트와 웹 서비스, e-Business가 가능하다.

둘째, UML(Unified Modeling Language) 표준 모델링 기능을 지원한다. 전 세계 표준 방법론인 UML 방법론의 표준 모델링 기능을 지원하기 때문에 기업 표준 모델링 툴로서 적합하다.

셋째, .NET환경의 컴포넌트인 COM+를 지원한다. 현재 컴포넌트 기술을 이끌어 가고 있는 .NET과 J2EE 양쪽 모두를 지원하기 때문에 CBD를 활용하는 어떤 분야에서도 적용이 가능하다.

넷째, XML 기능 지원으로 표준화된 Document를 제공한다. 전 세계 표준 데이터 전송 언어인 XML을 지원하여 표준화 되어 있는 Document를 제공한다.

다섯째, MSF/CD 방법론을 지원한다. .NET 환경에서 COM+를 생성하고 업무를 분석/설계하는 방법론인 MSF/CD를 지원한다.

여섯째, 주요 기능 중심으로 제품이 구성되어있기

때문에 사용하기 간편하다. 기존의 모델링 도구의 많은 기능들을 습득하기는 매우 어려운 일이었다. Aurora.net는 모델링 기능을 주요기능 중심으로 구성하였기 때문에 사용자가 사용하기에 매우 간편하다.

일곱째, 컴포넌트 생산성 확보는 물론 자동 작성된 컴포넌트를 기반으로 UI 컴포넌트 추가를 통해 C# ASP.NET 어플리케이션을 자동으로 생성한다.

여덟째, Forward Engineering, Reverse Engineering 모듈을 지원한다. 순공학(Forward Engineering) 및 역공학 (Reverse Engineering)을 지원하기 때문에 SRT (Simultaneous Round Trip) 기능이 지원된다.

아홉째, 소프트웨어 개발 전 단계를 지원한다. 기존의 모델링 도구들이 소프트웨어를 분석/설계까지만 지원하는 반면 Aurora.net는 분석/모델링/구현/테스트/배치의 소프트웨어 개발의 전 과정을 지원할 뿐만 아니라, 실제 구현된 업무를 컴포넌트화 시키는 단계까지 지원한다.

### 2.2.3 기대효과

첫째, 체계적이고 표준화된 방법론 적용으로 인한 시스템 분석, 설계, 구축의 안정화다. 업무 분석 및 설계단계에서 체계화되고 표준화 된 방법론을 장착한 Aurora.net를 통해 분석 및 설계를 할 수 있기 때문에 시스템을 체계적이고 안정적으로 구축할 수 있으며 신뢰도를 향상 시킬 수 있다.

둘째, 프로젝트 개발 전단계를 하나의 플랫폼에서 제공함으로써 생산성 향상 업무 분석/설계/구현 모든 프로젝트 개발의 전 단계가 하나의 플랫폼 안에서 모두 지원되므로 생산성이 향상된다.

셋째, 개발 비용 및 개발 기간 위험도를 최소한으로 줄일 수 있다. 체계화 되고 구체적인 방법론을 따라 업무를 분석하고 설계하기 때문에 개발비용 및 개발 기간을 최소화 할 수 있고 안정적인 방법

론의 절차를 따라 진행하면 되므로 시스템 개발의 위험도를 줄일 수 있다.

### 2.2.4 Aurora.net 절차도

Aurora.net의 특징적인 기능을 열거하고 각각의 기능에 대하여 설명하기로 하자.

- UML(Unified Modeling Language) 기반의 모델링
- 시스템 구성에 사용할 클래스 라이브러리나 코드 작성 지침에 따라 알맞게 조정 할 수 있는 코드 생성 기능
- 데이터베이스 모델링 기능 제공(Aurora.net Modeler)
- COM Component 소스 코드 생성
- 모델을 관찰하고 검색할 수 있는 시각적인 Browser 기능을 제공
- 모델과 코드의 지속적인 일관성 및 연관성 보장 모델링을 Template화 시켜 재사용
- 다이어그램을 마이크로소프트 워드로 변환
- 자사 컴포넌트 조립 툴인 Aurora.net Developer 등과의 연계를 통한 손쉬운 S/W 개발 환경 제공

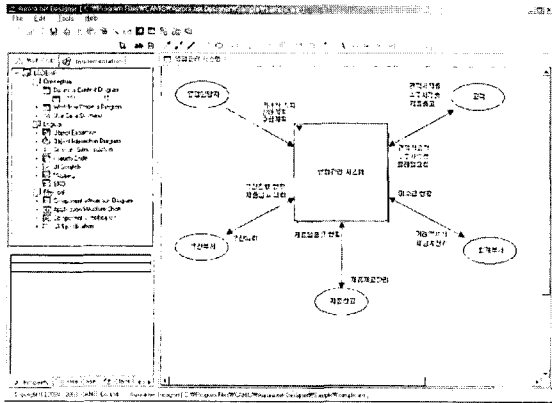
Aurora.net은 다음과 같은 방법으로 구현되었다. 첫째, 표기법 지원 : Aurora.net는 MSF/CD 개발방법론의 기본 모델링 기법을 지원한다.

둘째, 모델링 기반 : Aurora.net Designer는 다이어그램 기반의 모델링을 제공한다.

그러면 실제적인 주요화면을 중심으로 정리하여 보기로 하자.

#### 2.2.4.1 업무 배경도(Business Context Diagram)

시스템 또는 비즈니스 영역을 중심으로 외부 관련사항(External Interactions)을 기술하며 외부 구성요소 (External)는 User 또는 시스템의 외부에 표기된다. Business Context Diagram은 DFD(Data Flow Diagram)의 0 Level에 해

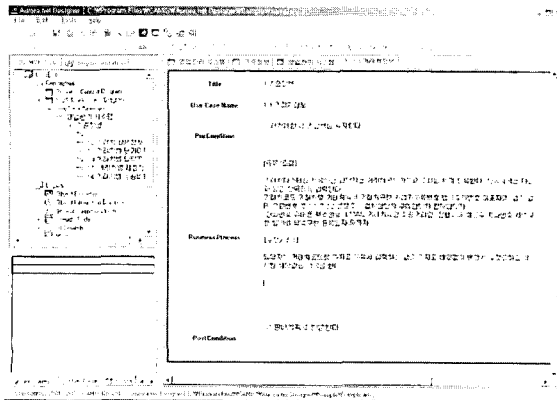


(그림 1) Business Context Diagram

당하는 Diagram으로 개발범위를 명확히 하고, 개발 이후의 관련된 외부인터페이스를 정의한다.

2.2.4.2 업무 절차도(Workflow Process Diagram)

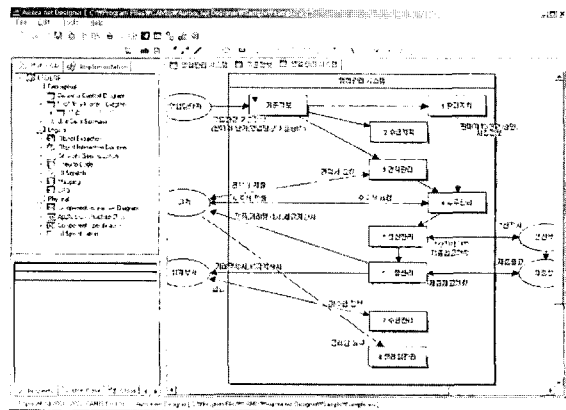
워크플로우 프로세스는 비즈니스 프로세스 내의, 그리고 부서들과 고객들 사이의 제품 및 정보의 흐름을 보여준다. Workflow Process Diagram은 관련된 정보들과 부서들, 내부 및 외부 고객, 외부 연계와 기존의 프로세스들과 시스템, 규칙들의 성능, 거대한 프로세스들의 하위프로세스들로의 분해와 같은 구성요소로 구성되어진다.



(그림 2) Workflow Process Diagram

2.2.4.3 유즈케이스 시나리오(Use Case Scenario)

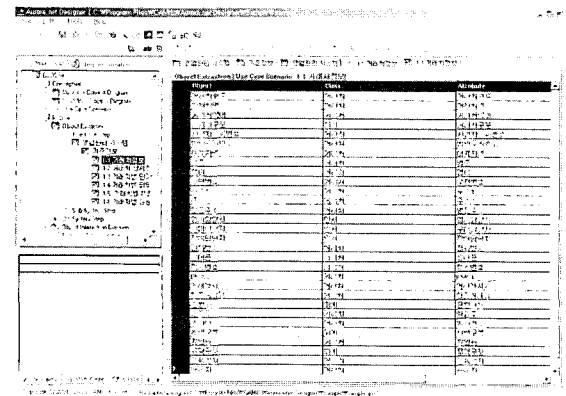
오브젝트간 상호 작용들과 오브젝트와 actor사이의 상호 작용들의 하나의 연속된 흐름으로 Use Case의 특징 예를 예시하며 과업 순서로 그려진다. Use Case Scenario는 작업의 현 상태와 작업의 미래 상태를 보여줄 수 있다.



(그림 3) Use Case Scenario

2.2.4.4 오브젝트 추출(Object Extraction)

Use Case Scenario 상의 명사들을 추출하고, 명사들 간의 Affinity (High Cohesion, Loosely Coupling)를 고려하여 Clustering 수행하며, Class 와 Attribute 들을 도출해 낸다.



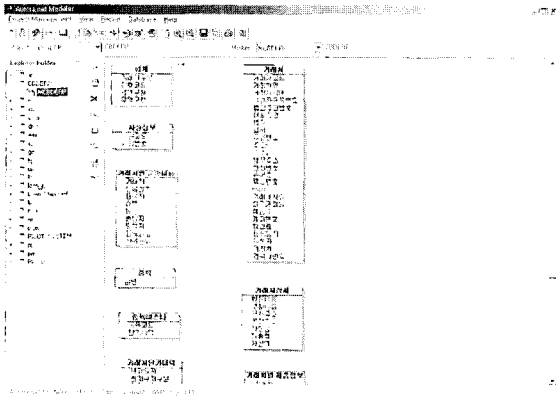
(그림 4) Object Extraction



실제로 구현된 프로그램의 UI에 대하여 간단한 화면 템플릿과 이벤트 등에 대하여 작성한다. 이후에 컴포넌트 상관도에서 UI Event에 해당하는 부분을 활용하게 된다.

2.2.4.9 ERD(Entity Relation Diagram)

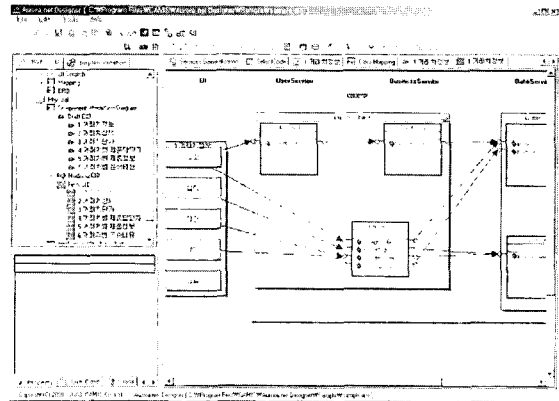
Logical단계에서의 Logical ERD와 Physical 단계에서의 Physical ERD를 지원하는 Database Modeling 기능으로 Aurora.net Modeler를 활용한다.



(그림 9) Entity Relation Diagram

2.2.4.10 컴포넌트 상관도(Component Interaction Diagram)

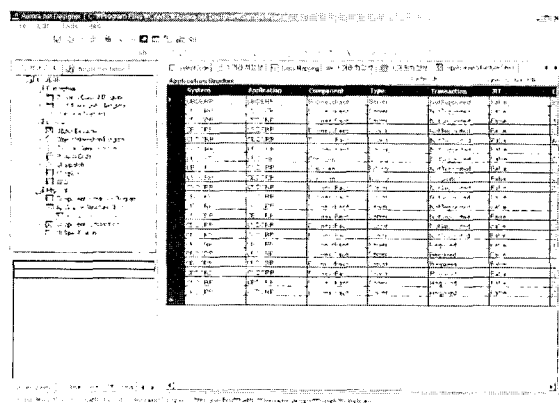
UI 상의 조회/신규/저장/취소 등의 이벤트를 발생시켰을 때, 이에 대한 서비스를 제공하기 위해 어떤 컴포넌트들이 어떻게 상호 작용하는지를 도식화한다. 1차, 2차, 3차로 나누어 작성하는데 1차 CID는 시스템 제한 사항에 대해 고려하지 않고 단순히 컴포넌트 간의 interaction을 있는 그대로 도식화하는 과정이고, 2차 CID는 1차 CID에 대해 Network Topology, Component Topology 관점의 최적화 방안을 반영하여 재작성하는 단계다. 특히 2차 CID는 Client와 Server 간 Round trip을 최소화하도록 작성한다. 3차 CID는 실제 Application에 배치될 Deployment 사항을 반영하여 작성한다.



(그림 10) Component Interaction Diagram

2.2.4.11 어플리케이션 구조도(Application Structure Chart)

어플리케이션 패키징 전략(Application Packaging Strategy)에 따라서 작성하며, 성능(Process Isolation and performance degradation, 컴포넌트 간 Out-of-process 통신 최소화)에 따른 성능 향상 등)과 업무 유관성(High Cohesion .vs. Loose Coupling), 관리의 용이성, 재사용성 등을 고려하여 작성한다.

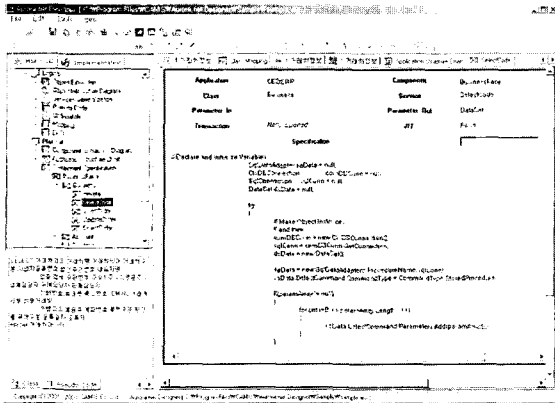


(그림 11) Application Structure Chart

2.2.4.12 컴포넌트 명세서(Component Specification)

컴포넌트의 속성(Method, Property, Parameter, Return Value 등)에 대하여 상세히

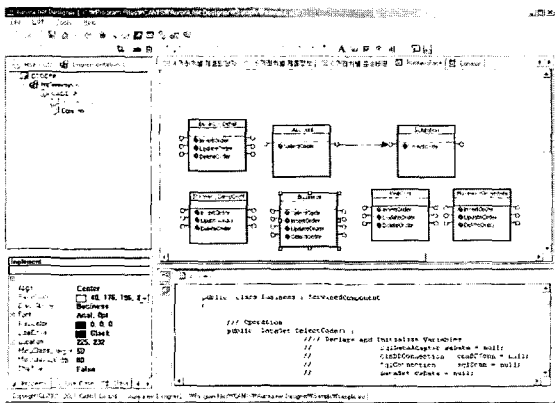
작성하고, 이후에 컴포넌트가 재사용되기 위하여 Black Box 형태로 지원될 수 있도록 내부 처리 내용에 대하여 상세하게 기술한다.



(그림 12) Component Specification

2.2.4.13 소스코드 자동 생성 (Implementation)

컴포넌트를 구성하는 프로그램 소스코드를 Physical 단계에서 설계한 Component 명세서를 기본으로 하여 자동으로 생성하여 준다.

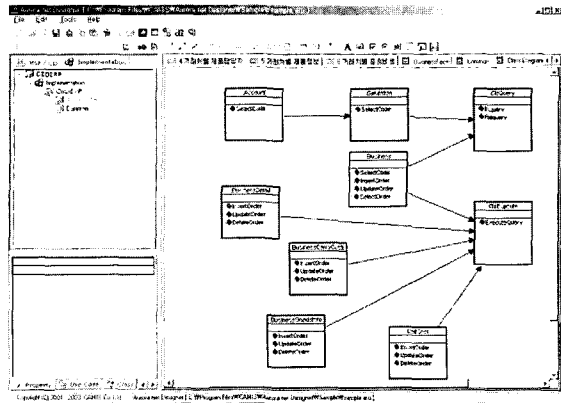


(그림 13) Implementation

2.2.4.14 클래스도 (Class Diagram)

Implementation 단계에서는 Application 또는 Component 단위로 클래스 간의 연관관계를

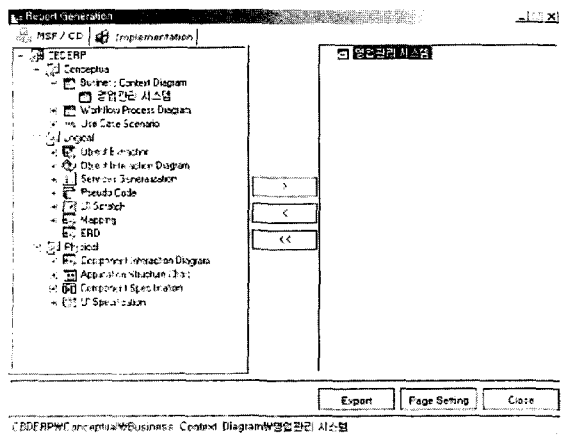
작성한데 반하여 Class Diagram은 전체 시스템의 클래스를 기술하고 클래스간 연관관계를 작성한다. UML Spec 1.3 의 Class Diagram과 유사한 Diagram이다.



(그림 14) Class Diagram

2.2.4.15 문서화 지원 (Report Generation)

시스템에서 작성된 모든 형태의 Document를 Microsoft Word 문서로 Export해 준다.



(그림 15) Report Generation



### 3. 결론 및 향후 과제

본 연구에서는 MSF/CD 개발 방법론을 기반으로 분석/설계에 관하여 자동화 개발도구를 소개하였다.

MSF/CD 방법론 자체가 초기 단계라 다양성이나 많은 Reference Site가 부족한 것은 사실이나 기존 윈도우 시장의 방대함과 Microsoft사의 닷넷 확장성은 미래지향적이고 나아가 IT 공급자들이 기존 하드웨어 판매에서 소프트웨어 컨설팅 사업에 힘을 집중시킨 점은 플랫폼 판매 못지 않게 소프트웨어 컨설팅 사업도 치열한 각축전을 벌일 전망이다.

따라서 MSF/CD 개발방법론을 톨로 개발한 Aurora.net(Designer, Modeler, Developer)은 많은 보완 작업과 MSF/CD 개발방법론의 발전을 직시하여 보다 나은 UML 분석/설계/구축 톨로 발전해 갈 것이다.

### 참고문헌

- [1] 한국SW컴포넌트컨소시엄, 컴퍼넌트란 무엇인가?, 2001.12
- [2] 박준성, E-Enterprise & CBD, 2001.2
- [3] CBDi, Component-Based Development Fundamentals, 2002
- [4] 정희원, 컴포넌트 소프트웨어 산업동향, 2000.7
- [5] 배준수, 컴포넌트웨어의 전망과 발전방향, 2002-07-31
- [6] Paul Allen, CBD를 사용하는 e-Business 통합, 2002.7
- [7] 배두환, CBD for e-Business, 2002
- [8] 임춘봉 DACOM ST 품질보증팀, 컴포넌트 중심 개발, "Component Based Development" - A Roadmap to eBusiness Success, White Paper, Princeton Softech, Jan., 10, 2000, <http://www.princetonsoftech.com/index.>, 2000.8. 7
- [9] 한국SW컴포넌트컨소시엄, 소프트웨어 컴퍼넌트 창간호, 2002.7
- [10] 야곱슨, Component Based Software Engineering, 2002
- [11] 참조 사이트 : <http://selab.cataegu.ac.kr/>
- [12] 박준성, "CBD개론", 2001.12
- [13] 박준성, "CBD 시장 바로보기," 디지털 타임즈, 2001. 8. 3.
- [14] 박준성, "CBD에 대한 소고," 정보통신연구진흥지 제3권 제3호, 2001. pp. 42-55.
- [15] 오영배, 박준성, "CBD 적용 사례 연구," 한국정보과학회 소프트웨어공학회지 제12권 제3호, 1999. 9. pp. 75-85.
- [16] 성공적인 소프트웨어 개발 방법론 (상권 - 요구사항 분석 및 설계 중심으로) 충남 대학교 윤청 지음. - 생능 출판사. 1996년.
- [17] 월간 마이크로 소프트웨어 (1996년 1~6월호)
- [18] 컴포넌트비전(주), EJB기반 컴포넌트 소프트웨어 개발기술소개
- [19] 전진욱, 해외 및 국내 CBD 시장동향과 활용현황, 2001
- [20] ETRI, 공용컴포넌트 개발 및 기술개발전략
- [21] 홍은주, Web Service, 2001
- [22] 윤태권, 정한일, "컴포넌트 소프트웨어 산업동향", (정보처리 제7권 제4호, 2000.7)
- [23] 정한일, CBD를 활용한 SW개발·활용현황 파악 및 유통활성화를 위한 제도적 연구수행계획서, 2000.6
- [24] 양영중, ETRI 컴퓨터소프트웨어기술연구소 컴포넌트 시장 분류 및 전망, 2000.5

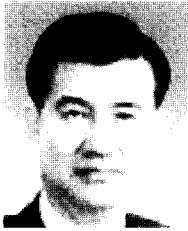
- [25] 정보통신부, 컴포넌트산업 육성 실천계획안, 2000.4
- [26] 한국SW컴포넌트컨소시엄, 한국SW컴포넌트컨소시엄 창립 기념 세미나, 1999.11
- [27] 데이브 토머스, Object Technology International, 1995년 3월
- [28] Paul Allen, "Realizing e-Business with Components", Addison-Wesley, 2000
- [29] Alan W. Brown, "Large-Scale Component Based Development", Prentice Hall, 2000
- [30] George T. Heineman & William T. Council, "Component-Based Software Engineering - ", Addison-Wesley, 2001
- [31] Peter Herzum & Oliver Sims, "Business Component Factory: A Comprehensive Overview of Component-Based Development for the Enterprise", John Willey & Sons, Inc, 1999
- [32] John Cheesman & John Daniels, "UML Components : A simple process for specifying component-based software", Addison-Wesley, 2000
- [33] Center for Software Engineering, <http://sunset.usc.edu>
- [34] Software Engineering Institute, <http://www.sei.cmu.edu>
- [35] 박병형, 블록놀리와 CBD, 태영출판사, 2002
- [36] 김연홍, 마이크로소프트의 웹서비스, 소프트웨어 컴포넌트 창간호, 2002.7
- [37] 손중모 외 6인, ".NET 시스템 구축 방법론", 정보문화사, 2002

## 저자약력



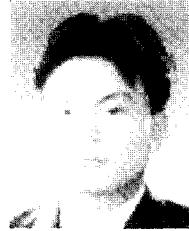
**이승호**

1978년 한국과학기술연구소 소형전산기 국산화 연구실  
1984년 한국전자통신 연구원(ETRI)  
1990년 (주)한국에이아이소프트 개발부 차장  
1992년 (주)씨엔아이 개발 담당이사  
1992년 한국이지시스템 창업  
현 재 (주)이지시스템 대표이사, (주)한국이지정보 대표이사  
EVA 회원사(ETRI 벤처기업협회), 한국정보처리학회  
학회지 이사 겸 편집위원, KTB n-Club 회장단  
(2002/5/31 취임)  
관심분야 : 데이터베이스, CRM, EIP, ERP, CBD  
이 메 일 : shlee@ezsxs.co.kr



**박병형**

1975년~80년 삼성그룹 종합 전산실 (신세계 백화점) 대리,  
군복무 전산화 (크레디트 카드 업무, 관리회계  
업무)  
1980년~81년 숭실대학교 전산실 연구원 (학사관리, 인사급  
여시스템, 도서관리)  
1982년~89년 (주)유공(현재 SK) 정보시스템부 전문직 과장,  
전산화(구매자제, SW QA, 개발센터장, 케이  
스틀개발)  
1989년~ (주)케미스 대표이사 / CEO, 미국현지법인  
NexxIT(회장)  
2002년~호서대학교 벤처전문대학원 컴퓨터응용기술학과 박  
사과정  
관심분야 : LM, CBD, ERP, CASE  
이 메 일 : pph@camis.co.kr



**김영희**

1998년 전북대학교 화학공학과 졸업  
1998년~98년 동양시멘트  
1999년~(주)케미스 R&D 재직중