

사례 발표

MSF/CD를 기반으로 한 ERP 솔루션 개발 사례

이은성¹⁾ 박병형²⁾ 이승호³⁾

목 차

1. 서 론
2. 관련연구
3. 결론 및 향후 과제

1. 서 론

20세기 말, 재앙이라던 밀레니엄버그(서기 2000년 표기문제)를 기점으로 IT업계뿐만 아니라 사회 전체적으로 엄청난 변화의 바람을 일으킨 원인이 바로 인터넷의 출현이라 할 수 있을 것이다. 인터넷은 우리의 생활환경은 물론 기업환경을 혁명적으로 바꾸어 놓았고, 고객이라 부르는 사용자 집단들은 급변하는 프로세스에 대응하기 위해서 많은 요구사항을 공급자, 개발자 등 컴퓨터 전문가 집단에 요구하기에 이르렀다. 이들의 수준 높은 요구사항들을 빠르게 지원할 수 있는 방법으로 컴포넌트 기반 개발 방식(CBD : Component Based Development)이 기업환경의 새로운 패러다임으로 등장하게 된다. 과거 전산실에서 개발하던 방식을 1970년대부터 사용하다가 소위 전산전문 기업집단인 SI(System Integration)업체에 의뢰하던 단계를 거쳐, 앞으로는 기존에 잘

개발된 컴포넌트를 조립하거나 재활용하는 CBD 방식에 도달하게 될 것이다. 우리 정부도 2010년 소프트웨어 강국을 이루기 위해서는 훌륭한 인적 자원을 활용하여 다양한 프로세스를 컴포넌트로 만들어 소프트웨어를 수출하여 소프트웨어 강국을 이루고자 노력을 기울이고 있는 실정이다. 실제로 가트너 그룹에서는 2004년까지 CBD를 도입한 기업은 그렇지 못한 기업보다 5~10배의 소프트웨어 생산성 향상을 가져올 것이라는 예측 자료를 내놓았다. 또한 IT선진국들은 이미 소프트웨어를 재활용하는 CBD기법을 급속도로 도입하고 있으며, 국내에서도 빠르게 금융권, 대기업, 공기업 등을 중심으로 CBD기법이 확산되고 있다. 과거 우리나라는 많은 기법과 좋은 도구들, 그리고 다양한 IT패러다임을 충실하게 수행한 나라라고 자부할 수 있을 것이다. 그러나 그 속사정을 살펴보면 아직도 심한 소프트웨어 정체현상과 유지보수에 엄청난 비용을 투자하고 있으며, 기존의 레거시 시스템 문제로 인하여 새롭게 나아가는 길과 그 발목을 잡는 기존의 방법론, 도구, 전산자원들에 많은 어려움을 겪고 있는 것이 우리가 처한 현실이다. 이러한 현실 속에서 소프트웨어 공학에 대한 풍부한 경험과 기반지식을 바탕으로 그 동안

1) (주)케미스 R&D

2) 호서대학교 벤처전문대학원 컴퓨터응용기술학과 박사과정

3) (주)이지시스템 대표이사

다년간 쌓아왔던 CASE Tool 개발기술과 ERP솔루션 개발기술에 CBD방법론을 도입하게 되었다. 그리고 그 결과 물로 CBD기반의 개발도구와, CBD기반의 개발도구에 풍부한 ERP 구축경험을 적용하여 만드는 가장 CBD의 취지에 적합한 CBD기반의 Web ERP를 개발하는 것이 본 개발의 목적이라 할 수 있다. 여러 CBD 플랫폼 중 .NET플랫폼을 선택한 이유는 하나의 통합환경에서 가장 높은 성능을 가질 수 있고 하나로 통합된 공급사의 장점을 가지고 있기 때문이다.

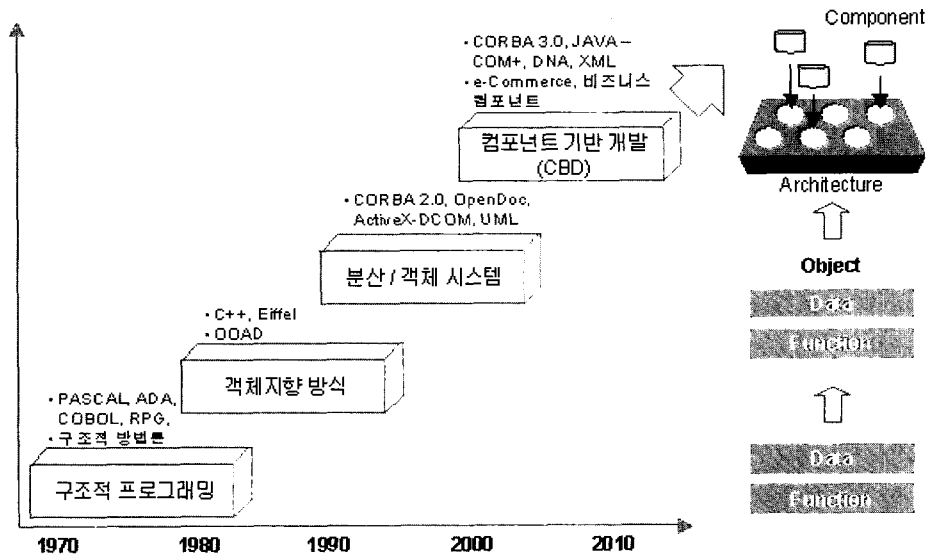
2. 관련연구

2.1 MSF/CD 방법론의 적용

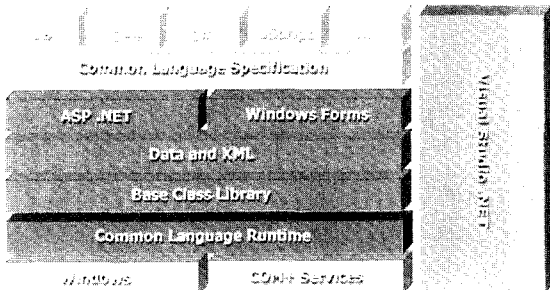
CBD기반의 Web ERP 솔루션이 작동하기 위한 환경으로 Microsoft사의 .NET Platform을 적용하였다. Microsoft의 .NET Framework는 웹 서비스를 지원하는 플랫폼으로 ERP솔루션 및 개발도구의 개발에 적용되는 기술이며 다음과 같

은 특징이 있다. 기술적으로 인터넷 표준 언어로 대두된 확장성 표기 언어인 XML의 직접 지원을 고려하여 MS사가 XML 웹 서비스를 위해 개발한 플랫폼으로서 분산 애플리케이션 개발을 위한 차세대 프레임워크이며 생산성 증가의 효과를 기대할 수 있다. 또한 플랫폼 중립(Platform-neutral)적이고 언어독립적(language-Independent)이며 강력한 상호연동성(interoperability)을 지닌다. 그리고 웹 표준 기술인 HTTP, XML, SOAP, UDDI 등을 활용한다. 분산 컴퍼넌트 개발 방법론을 따라서 구축할 수 있고 'MSF/CD' 방법론이 바로 그것이다.

.NET Platform 을 구성하는 프로그램 언어로는 C#.NET이 대표적이다. C#.NET은 C++의 Next Generation 언어로 표현되며, 다양한 플랫폼으로 포팅(특히 다양한 모바일 디바이스)할 수 있다는 특징이 있다. 따라서 본 MSF/CD기반의 ERP 솔루션 개발에서는 C#언어를 주 개발언어로 선택하였다.



(그림 1) 소프트웨어 개발 패러다임의 변화



(그림 2) .NET Platform 구성도

2.2 NET기반의 ERP개발

앞서 설명한 CBD기술과 .NET 관련기술, Web 구축기술을 기업의 업무를 전산화 하는 ERP에 접목시키는 것이 연구 및 개발의 목적이라 하겠다. 따라서 MSF/CD기반 개발, .NET Platform, Web 기술이 적용된 시스템이 갖는 모든 장점을 수용하는 ERP솔루션을 의미한다. 개발 생산성을 향상시키고, 시스템 적용 후의 유지보수가 용이한 시스템을 구축하고 표준 프로세스의 경영관리 업무의 전산화를 전사적 자원관리차원에서 접근하였다. 향후의 기술 Trend 변화에 대처 가능한 시스템으로 구축하기위해 .NET플랫폼을 선택하였다.

2.2.1 개발시스템 내용

1 SDM - 영업관리 (Sales & Distribution Module)	7 EIS - 경영층 정보 (E) (Executive Information System)
2 MMM - 구매/자재관리 (Material Management Module)	8 BIM - 기준정보관리 (B) (Base Information Module)
3 PDM - 생산관리 (Production & Distribution Module)	9 SYS - 시스템관리 (Y) (System Management)
4 QCM - 품질관리 (Quality Control Module)	10 HRM - 인사/급여관리 (Human Resource Module)
5 TMS - 수출입(무역)관리 (T) (Trade Management)	11 FAM - 고정자산관리 (Fixed Asset Module)
6 RDM - 연구개발 관리 (Research Development Management)	12 GAM - 재무회계관리 (General Account Module)

(그림 3) ERP 표준 프로세스 구축

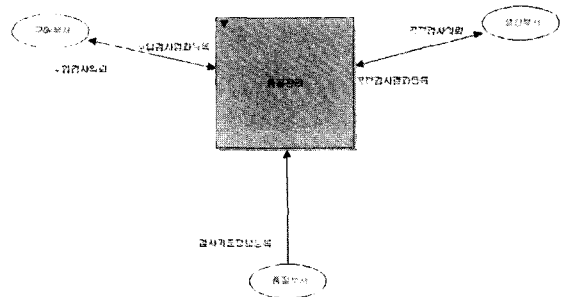
위와 같은 일반적인 표준 프로세스를 MSF/CD 방법론에 따라 CBD기반의 표준 어플리케이션으로 개발하였다. 각 모듈에 대한 설명은 생략하고 다음 장에서 하나의 모듈을 MSF/CD를 통하여

어떻게 개발하였는지를 알아보자.

2.2.2 개발절차

이번 장에서는 '품질 관리' 모듈의 개발 절차를 설명하기로 한다. 일부 모듈에 대한 설명이지만 다른 업무의 개발절차와 동일한 사항이기 때문에 MSF/CD기반의 ERP의 전체 개발절차를 충분히 예측해 볼 수 있다. 품질검사는 수입검사관리, 공정검사관리, CLAIM검사관리의 기능이 있다. 수입검사는 수입 검사일지등록, 수입 검사결과통보, 불량조회, 실적조회를 하고 공정검사에서는 공정 검사일지를 등록하고 검사결과통보, 완제품검사, 실적조회, 공정검사 불량조회로 이루어져있다. CLAIM검사는 영업부서에서 접수 후 검사결과를 등록한다. 전체 검사결과 등록 불량별 유형별, 실적조회 등 분석기능도 제공한다.

품질 관리 업무는 다음과 같은 세부 프로세스로 구성하였다.



(그림 4) 품질관리 모듈구성(Business Context Diagram)

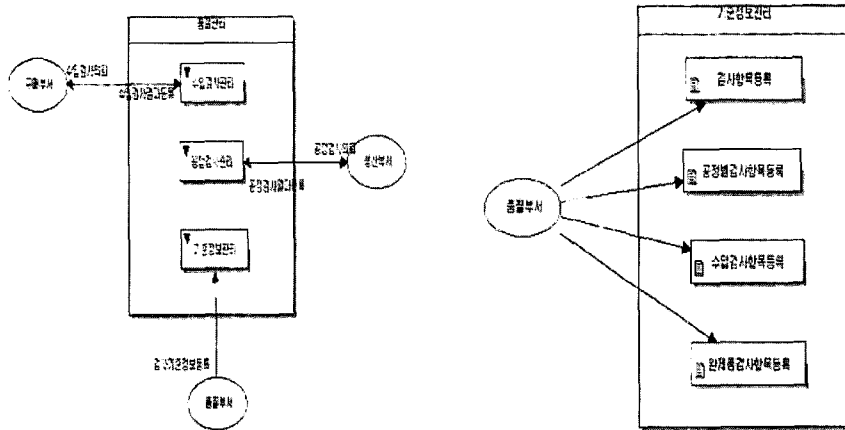
위의 (그림 4)는 MSF/CD의 개념설계(Conceptual Design)에 해당하는 단계며 업무 배경도 (Business Context Diagram)를 설계하는 절차에 해당한다. 품질관리라는 프로세스에서 구매부서, 생산부서, 품질부서라는 외부요소가 있고 그러한 요소들과 수입검사의뢰, 검사기준정보 등록 등의 데이터의 흐름 따위를 표현한다.

2.2.2.1 추진전략

- 기준정보관리는 공정별 검사품에 검사계획 및 기준자료를 설정하는 것에 대한 점을 중요시한다.
- 품질검사 시 발생하는 불량유형과 불량원인에 대한 기준을 관리하는 것을 중요시한다.
- 수입검사 시 거래처에서 입고 되는 모든 제품의 품질상태를 검사하고 수입검사상세에서 전체적인 내용을 입력하고 불량유형을 기입해 분석자료로 활용하는 점을 중요시한다.

- 공정검사는 제품공정에 따른 공정을 검사하고 불량공정에 한에 내역관리를 하고 불량공정에 대한 다양한 패턴별 관리 기능을 제공해준다.
- 검사결과에 대한 문제해결을 위한 다양한 분석도구를 활용해 품질현황을 분석할 수 있도록 한다.
- 검사운영관리에 대한 수입검사, 공정검사, 최종 입고검사에 중점을 둔다.

품질관리 시스템(프로세스)과 외부와의 데이터 흐름을 정의한 다음 비즈니스 프로세스 내의, 그



(그림 5) 품질관리 세부 프로세스와 부서들간의 정보흐름(Workflow Process Diagram)

Title	품질관리	기준정보관리	ETC
Use Case Name	수입검사항목등록		
Pro Condition	1. 품질관리자는 구매부서로부터 수입검사 의뢰를 받는다.		
Business Process	<p>[등록]</p> <p>1. 품질관리자는 해당 날짜의 입고입력정보에서 입고상태가 검사기초인것을 조회한다. 2. 품질관리자는 수입검사하는 검사입력번호, 품목코드, 거래처코드, 입고수량, 입고일자, 입고입력지 등의 정보를 가져온다. 3. 검사일수지, 검사입력지의 시일정보를 인사시스템에서 가져온다. 4. 품질관리자는 공통코드에서 구분코드, 작업일자, 작업순번을 가져온다. 5. 품질관리자는 해당 수입검사 상세에서 수입검사항목을 등록한다. 6. 품질관리자는 해당 항목자세코드인 검사항목정보(검사명, 검사구분, 품목코드, 변경기준, 변경기준, 기준코드)를 가져오고 상세입력서 상세정보를 조회후 입력한다. 7. 품질관리자는 수입검사결과를 등록한다.</p> <p>검사항목ID: 구분코드, 작업일자, 작업순번, 상세순번, 검사코드, 검사기준, 검사기준E, 기준코드, 품목, 품목사명, 검사수량, 품목유일코드, 품목가수명, 품목명칭, 비고 검사항목사명: 검사일, 작업일자, 작업순번, 검사구분, 품목코드, 거래처코드, 검사일수지, 검사일수지, 검사입력지, 검사방법, 검사항목, 검사항목수명, 품목수명, 품목명칭수명, 품목명칭, 품목명칭, 품목명칭, 품목사명, 검사항목번호, 품목명칭, 품목명칭</p> <p>8. 품질관리자는 수입검사후 검사결과를 검사항목에 입력한다. 제품명칭사미적 등록사명 품목코드, 검사일자, 순번, 검사구분, 검사수량, 품목명칭수명, 품목유일(제품유일) 검사방법 거래처코드 검사항목지</p>		
Post Condition	1. 품질관리자는 수입검사결과를 등록하고 결과를 분석자료로 활용한다.		

(그림 6) Use case Scenario

리고 부서들 사이의, 정보의 흐름을 보여준다. Workflow Process Diagram을 통해 관련된 정보들과 부서들, 내부 및 외부 고객, 외부 연계와 기존의 프로세스들과 시스템, 규칙들의 성능, 거대한 프로세스들의 하위프로세스들로의 분해와 같은 구성요소를 정의한다.

다음으로 오브젝트간 상호 작용들과 오브젝트와 actor사이의 상호 작용들의 하나의 연속된 흐름으로 Use Case의 특정 예를 예시하며 과업 순서로 그려진다. MSF/CD 개념설계단계의 마지막 절차로 (그림 6)과 같이 Use Case Scenario를 통해 작업의 현 상태와 작업의 미래 상태를 보여줄 수 있다.

위에서 작성한 Use case scenario에서 객체를 나열하고 그 객체들이 어떠한 Class에 속할 것인지를 정의하며 Attribute를 정의하고 프로그램상의 Values를 정의한다. 이것은 MSF/CD의 논리설계(Logical Design)의 Object Extraction 절차를 따라서 진행되며 개발도구에 의해서 자동 추출한다. 다음 (그림 6)과 같이 Use Case Scenario 상의 명사들을 추출하고, 명사들 간의 Affinity (High Cohesion, Loosely Coupling)

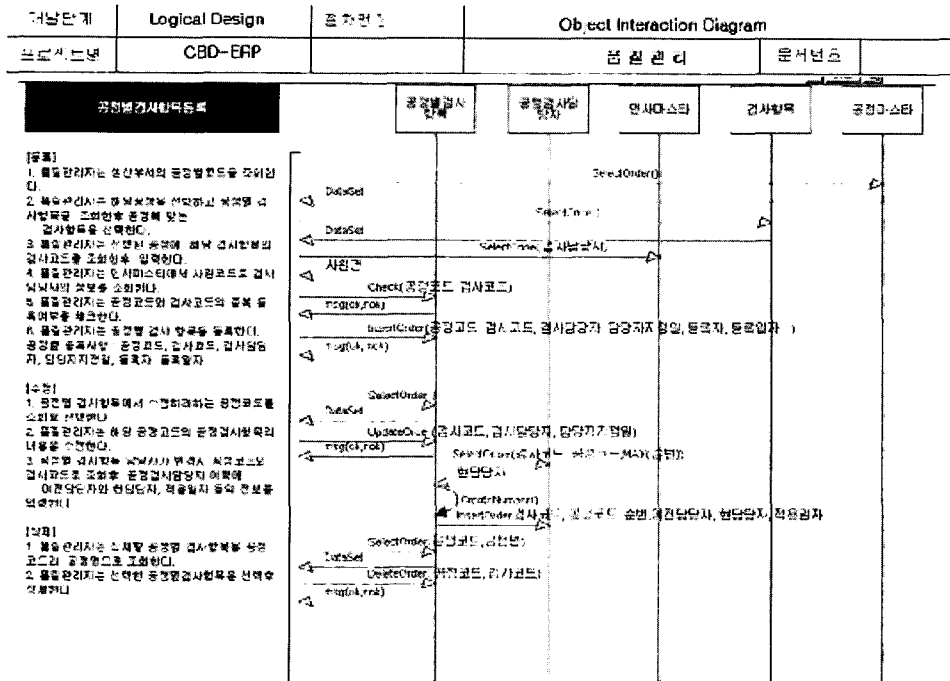
를 고려하여 Clustering 수행하며, Class 와 Attribute 들을 도출해 낸다.

품질관리 모듈의 각 Use Case Scenario에 대하여 Class 간의 Interaction(Class 간 서비스 요청/응답) 을 도식화한다. Object Interaction Diagram에서 추출해내는 서비스들은 User Services, Business Services, Data Services 로 나눌 수 있다. 품질관리 모듈의 공정별 검사항목등록 프로세스에 대한 Object Interaction Diagram을 예로 들면 다음 (그림 8)과 같다.

OID 상에 나타난 각 Class별 서비스명 들을 종합 정리하여 서비스의 성격을 파악한 후, 유사한 서비스들에 대해 통합된 하나의 Public Service 를 도출해내는 과정이다. 일반화된 서비스들은 이후에 각 Public Service에 대한 상세 절차 및 관련 로직을 Pseudo code 형식을 빌어 기술하게 된다. 그 후에 서비스 일반화(Service Generalization)를 거치고 Pseudo Code를 작성한다. UI에 대한 대략적인 정의를 내리고 Database modeling을 하고 논리, 물리적 데이터베이스를 구성한다. 그리고 물리적 설계 (Physical Design)단계의 여러 절차들을 거쳐서

Object	Class	Attribute	Values
수입검사	수입검사클래스	검사일자	
수입검사번호	수입검사클래스	검사번호	
수입검사내역	수입검사클래스	검사내역	
수입검사항목	수입검사클래스	검사항목	
수입검사결과	수입검사클래스	검사결과	
수입검사담당자	수입검사클래스	담당자명	
수입검사대상	수입검사클래스	대상명	
수입검사위치	수입검사클래스	위치명	
수입검사상태	수입검사클래스	상태명	
수입검사종류	수입검사클래스	종류명	
수입검사등급	수입검사클래스	등급명	
수입검사비율	수입검사클래스	비율명	
수입검사비율	수입검사클래스	비율명	
수입검사비율	수입검사클래스	비율명	
수입검사비율	수입검사클래스	비율명	
수입검사비율	수입검사클래스	비율명	
수입검사비율	수입검사클래스	비율명	
수입검사비율	수입검사클래스	비율명	

(그림 7) Object Extraction Diagram



(그림 8) 객체와 객체간의 서비스 상관관계(Object Interaction Diagram)

품질관리 모듈의 설계를 마치게 된다. 그리고 개발도구에서 지원하는 Implementation 단계의 컴포넌트 소스코드 생성기능으로 COM+에 가장 적합한 C#소스코드를 자동으로 생성하여 준다. 이와 같은 절차를 거쳐서 만들어진 COM+컴포넌트가 Aurora.net Developer를 통해 UI, Business Component, Database가 조합되어 '품질관리' 모듈이 컴포넌트 기반으로 만들어진 다. 이와 같은 방법을 생산, 구매자재, 인사, 회계 등의 전체 모듈에 적용하여 전체 ERP 솔루션이 만들어진다.

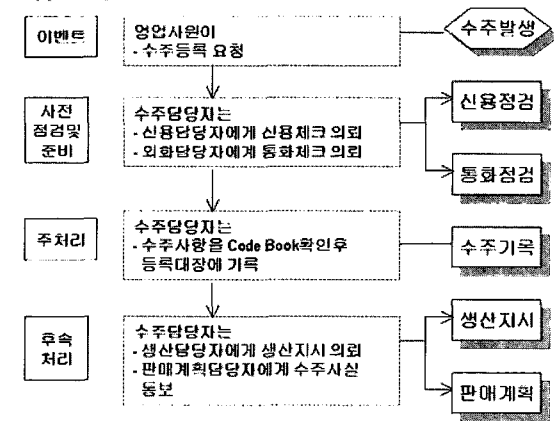
2.3 컴포넌트 재사용성 입증

2.3.1 모델링 방식에 따른 재사용성

MSF/CD를 통해서 개발하게 되면 (그림 9~11)과 같이 같은 업무프로세스를 가지고도 전통적인 C/S 어플리케이션과 CBD 어플리케이션

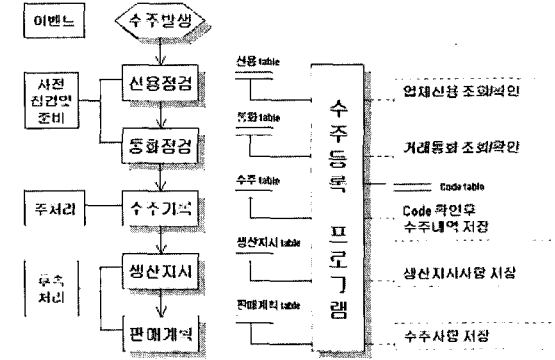
은 서로 다른 모델링 접근방식을 갖는다.

Supposing a scenario : 수주등록



(그림 9) 실제 업무처리 Flow

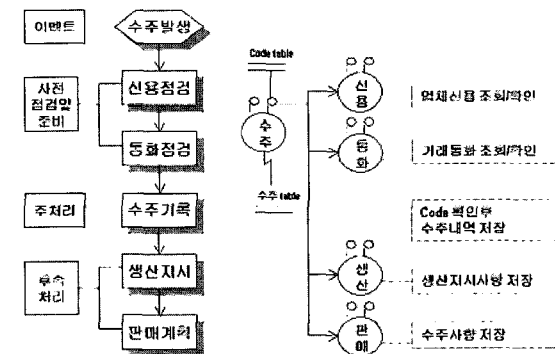
Supposing a scenario : 수주등록



(그림 10) Traditional C/S Application

위의 (그림 10)은 전통적인 C/S 어플리케이션의 모델링 구조도다. 이벤트별 테이블을 가지고 있으며 각각의 행위는 하나의 단위 프로그램으로 구성되며 이들은 모두 예시에서 보듯이 '수주등록 프로그램'이라는 하나의 단위 프로그램으로 묶여있다. 이와 같은 프로그램은 'Top-Down' 방식으로 구성되며, 이벤트에 종속적이다. 또한 데이터모델 중심으로 설계되고 Database, UI, Logic이 서로 종속적인 구조가 된다.

Supposing a scenario : 수주등록



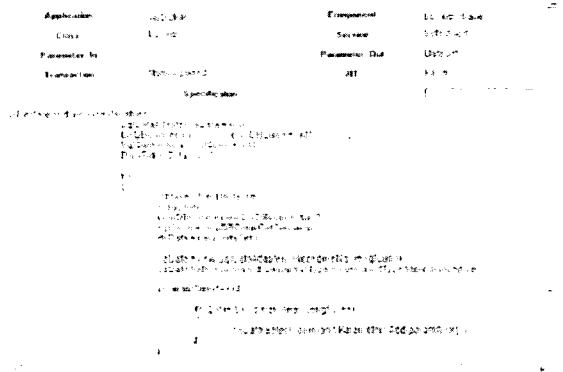
(그림 11) CBD Application

위의 (그림 11)은 MSF/CD를 따른 모델링 구조다. 객체간의 서비스 흐름 중심으로 설계되어 사용자 중심의 구현이 용이하다. 또한 Business

logic과 UI, Database 간의 종속성이 낮으므로 향후 재활용이 용이하다.

2.3.2 컴포넌트에 대한 상세명세

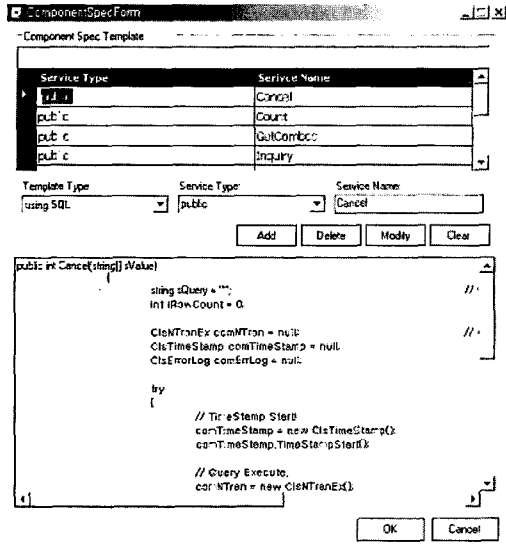
설계과정에서 만들어 지는 모든 컴포넌트들에 대해 (그림 12)와 같은 컴포넌트 명세서를 작성한다.



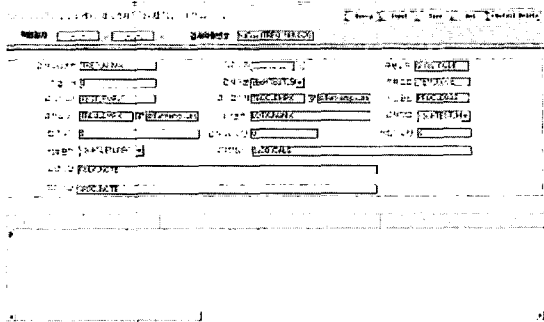
(그림 12) 컴포넌트 명세서(Component Specification)

위의 (그림 12)는 특정 서비스를 수행하는 컴포넌트의 명세서를 작성하는 화면이다. 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.

개발도구(Aurora.net Designer)에서 지원하는 Component Specification 기능을 통해서 쉽게 컴포넌트 명세서를 작성할 수 있고 향후 Component Repository 에서 컴포넌트의 명확한 분류가 가능하다.



(그림 13) Component Specification Template



(그림 14) 품질관리 화면(수입검사결과등록)

3. 결론 및 향후 과제

지금까지 MSF/CD를 기반으로 ERP솔루션 개발과정을 간략히 설명하였다. 최근 .net플랫폼 기반으로 소프트웨어를 개발하는 사례가 많이 있으나 MSF/CD를 적용하여 프로젝트를 진행하는 사례는 그다지 많지 않다. 일반적으로 CBD방법론을 적용하여 개발하는 것이 상당히 복잡하고 까다롭다는 선입견을 가지고 있는 사람이 대다수이기 때문이기도 하다. 또 전통적인 개발방법론으로 개발하는 것이 CBD방법론 기반으로 개발하는 것

보다 어렵지 않다는 것을 부정할 수도 없는 실정이다.

CBD(MSF/CD)를 적용하여 .NET기반의 ERP 솔루션을 개발하는데 성공할 수 있었던 것은 .NET에 대한 기반지식, CBD에 대한 이해, ERP 개발의 풍부한 경험 등 여러 가지 요인이 있었다. 하지만 그 중에서 가장 크게 작용한 요소의 하나로 CBD지원 개발도구가 있다. MSF/CD를 Tool 내에 탑재하여 자연스럽게 간편한 컴포넌트 기반 설계가 가능하였고 프로젝트 전반적으로 Tool을 사용함으로써 프로젝트 관리 및 동기화, 생산성 측면에서 사용하지 않는 프로젝트에 비해 성공할 확률이 높다. 지금도 CBD기반의 컴포넌트형 ERP의 개발이 활발히 이루어지고 있다. 또한, 앞으로도 많은 솔루션들이 .NET 플랫폼 기반으로 개발될 것이라 예상된다. 이번 개발사례와 같이 해당 플랫폼에 가장 적절한 CBD개발도구를 사용하면 ERP솔루션뿐만 아니라 CRM, EIP, EKP 등의 다양한 솔루션을 보다 쉽고 안정적으로 개발할 수 있을 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] 박준성, "CBD개론", 2001.12
- [9] 박준성, "CBD 시장 바로보기," 디지털 타임즈, 2001. 8. 3.
- [10] 박준성, "CBD에 대한 소고," 정보통신연구진흥지 제3권 제3호, 2001. pp. 42-55.
- [11] 오영배, 박준성, "CBD 적용 사례 연구," 한국정보과학회 소프트웨어공학회지 제12권 제3호, 1999. 9. pp. 75-85.
- [12] 전진욱, 해외 및 국내 CBD 시장동향과 활용현황, 2001
- [13] 홍은주, Web Service, 2001
- [14] 양영중, ETRI 컴퓨터소프트웨어기술연구소 컴포넌트 시장 분류 및 전망, 2000.5
- [15] Paul Allen, "Realizing e-Business with Components", Addison-Wesley, 2000
- [16] Alan W. Brown, "Large-Scale Component Based Development", Prentice Hall, 2000
- [17] George T. Heineman & William T. Council, "Component-Based Software Engineering - ", Addison-Wesley, 2001
- [18] Peter Herzum & Oliver Sims, "Business Component Factory: A Comprehensive Overview of Component-Based Development for the Enterprise", John Willey & Sons, Inc, 1999
- [19] John Cheesman & John Daniels, "UML Components : A simple process for specifying component-based software", Addison-Wesley, 2000
- [20] Center for Software Engineering, <http://sunset.usc.edu>
- [21] CIO communications, <http://www.ciokorea.com>
- [22] 박병형, 블록놀이와 CBD, 태영출판사,

- 2002
- [23] 박병형, "한권으로 끝내는 e-ERP", 2001.
- [24] 김연홍, 마이크로소프트의 웹서비스, 소프트웨어 컴포넌트 창간호, 2002.7
- [25] 손중모 외 6인, ".NET 시스템 구축 방법론", 정보문화사, 2002.

저자약력



이 은 성

1993년~95년 중림건설 주식회사
 1999년~(주)케미스 R&D 재직중
 2000년 인덕대학 소프트웨어 개발과 졸업
 관심분야 : LM, CBD, ERP, CASE
 이 메 일 : les@camis.co.kr



박 병 형

1975년~80년 삼성그룹 종합 전산실 (신세계 백화점) 대리, 군복무 전산화 (크레디트 카드 업무, 관리회계 업무)
 1980년~81년 숭실대학교 전산실 연구원 (학사관리, 인사급여시스템, 도서관리)
 1982년~89년 (주)유공(현재 SK) 정보시스템부 전문직 과장, 전산화(구매자체, SW QA, 개발센터장, 케이스툴개발)

1989년~ (주)케미스 대표이사 / CEO, 미국현지법인
NexxIT(회장)

2002년~호서대학교 벤처전문대학원 컴퓨터응용기술학과
박사과정

관심분야 : LM, CBD, ERP, CASE

이 메 일 : pph@camis.co.kr



이 승 호

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@easy.co.kr