

사례 발표

ERP Component 개발사례(지엔텍 WINIZ)

노상석¹⁾

목 차

1. 개발배경
2. 개발과정
3. WINIZ*Framework
4. 기대효과

1. 개발배경

지엔텍은 지난 1991년 설립하여 MIS와 SI 프로젝트 사업을 수행하면서 6년 동안의 프로젝트 구축경험과 1994년부터 1996년까지 ERP의 대명사라고 할 수 있는 SAP/R3의 국내 Localization 및 컨설팅의 경험을 살려서 1997년에 국내 자체 기술로 VISION21이라고 하는 클라이언트/서버 버전의 ERP를 개발하여 지금까지 벤처기업으로서 성공적인 비즈니스를 수행하고 있다는 평가를 받아오고 있다[13].

그러나 많은 ERP 프로젝트를 수행하면서 다양한 모델과 기술환경 별로 약 15개 정도의 ERP 솔루션이 존재하게 되었다. 하나의 솔루션이 개발되기 위해서는 약 5~10명 정도의 개발자가 6개월 정도의 시간을 투자해야 한다. 또 하나의 솔루션이 보존되기 위해서는 약 3명의 지속적인 투자가 이루어질 수밖에 없다. 따라서, 지엔텍이 보유한 모든 솔루션을 유지하기 위해서는 최소 45명 정도의 순수 연구 개발인력이 고정 투자되어야 하는

것이 현실이다. 그러나 시장은 이를 용인해 주지 않았으며 빠른 비즈니스 모델의 변화는 솔루션의 생명 주기를 더욱 짧게 만들어 상황을 더욱 어렵게 하고 있다.

더우기 하나의 솔루션이 패키지로서 사이트에 적용되면 고객의 요구를 수용하기 위해 대개 50%에 육박하는 커스터마이징이 필요했다. 이에 따라 솔루션들의 경계는 점차 모호해져 가고 있어서 ERP 솔루션 개발 및 유지보수에 점점 많은 어려움을 겪고 있다[4,7].

이러한 어려운 상황을 극복하기 위해 지엔텍은 2000년 초부터 “언제 어디서나 손쉽게 이기종의 하드웨어와 IT자원을 재활용할 수 있고 소프트웨어 재활용, 프로세스 확장성이 용이한 솔루션” 개발의 필요성을 느끼게 되었다. 이러한 요구사항을 해결할 수 있는 방법에 대해 고민하다가 CBD(Component-Based Development) 방법론에 관심을 갖기 시작하여 2년 동안의 연구개발을 통해 2002년 10월에 CBD방법론을 이용한 J2EE 기반의 Component ERP인 WINIZ라는 제품을 개발완료 하였다. 이러한 WINIZ는 인터넷망을 통해서 웹브라우저만으로 접근이 가능하게 되어 “언제 어디서나 손쉽게”라는 요구사항을 만족할

1) 지엔텍 정보기술연구소 연구소장

수 있었다. 또한, J2EE(Java 2 Enterprise Edition)과 WAS(Web Application Server)를 통해 이미 보유하고 있는 다양한 이기종의 IT자원(Windows, Unix, Linux등)을 재활용 가능하게 되었으며, CBD방법론을 이용하여 Technical Component, Business Infrastructure Component, Business Component, Application Component를 개발하여 서로 필요한 컴포넌트를 조합하여 ERP를 만들으로써 “소프트웨어 재활용, 프로세스 확장성의 용이함”을 충족하게 되었다[1,5,7,8,9,10,13].

2. 개발과정

ERP Component를 개발하기 위해서 Rational의 소프트웨어 개발 방법론인 RUP(Rational Unified Process)를 사용하였다. 소프트웨어의 개발은 여러 번의 반복(Iteration)을 거치며 각각의 반복은 요구사항 분석, 분석 & 설계, 구현, 테스트 & 평가 과정을 포함하고 있어 자체로서도 하나의 개발주기를 이룬다. 이러한 반복적인 개발 방법에서는 반복마다 실행 가능한 제품이 산출되고 이는 반복이 거듭될 수록 제품의 품질이 향상되어 결국 최종 시스템으로 발전되었다[11,12].

전통적인 waterfall 프로세스와 비교했을 때 반복적인 개발 방법이 갖는 장점은 다음과 같다.

- 요구사항 변경에 대한 관리가 용이하다.
- 매우 높은 수준의 재사용이 가능하다.
- 전체적으로 높은 품질을 얻을 수 있다.
- 프로세스가 진행됨에 따라 프로젝트 팀원의 기술이 향상된다.

RUP에서 소프트웨어 생명주기(lifecycle)는 여러 주기(cycle)로 나누어지며 각각의 주기는 각기

한단계 진보된 제품을 완성하였다. RUP에서는 하나의 개발 주기를 다음과 같은 4개의 연속적인 단계(phase)로 나눈다.

- 착수 단계(Inception phase)
- 상세화 단계(Elaboration phase)
- 구축 단계(Construction phase)
- 전이 단계(Transition phase)

2.1 착수(Inception) 단계

착수 단계에서는 시스템을 위한 비즈니스 케이스를 수립하고 프로젝트의 범위를 정하였다. 이를 위해 먼저 시스템과 상호작용을 할 외부의 개체(entity : Use Case에서 actor에 해당)를 파악하고 상호작용의 특성에 대한 개념적인 수준의 파악이 이루어졌다. 이 과정에서 모든 유스 케이스가 파악되고 중요한 유스 케이스에 대한 간략한 기술이 행해졌다.

이러한 착수 단계의 산출물은 다음과 같다 [11,12].

- 비전(Vision) 문서 : ERP의 핵심이 되는 요구사항, 주요 기능, 제한 사항을 기술한 문서
- 초기 버전의 유스 케이스 모델 (10%-20%의 완성도) : Use Case Diagram, Use Case Scenario, Use Case Specification등
- 초기 버전의 위험요소 평가
- 프로젝트 계획 : 수행 단계와 반복을 표현
- WINIZ*Framework 설계

2.2 상세화(Elaboration) 단계

상세화 단계의 목적은 문제 영역을 분석하여 아키텍처의 기반을 설립하고 프로젝트 계획을 수립하며 이를 통해 프로젝트에서 가장 큰 위험요소를 제거하는 데 있다. 이러한 목적을 달성하기 위해서는 시스템에 대한 폭 넓고 동시에 심도 있는 이해가 수반되어야 한다. 아키텍처에 관련된 결정사항은 ERP의 범위, 주요한 기능, 성능에 대한 요

구사항과 같은 전체 시스템에 대한 사항의 이해를 바탕으로 이루어졌다[11,12].

상세화 단계가 앞서 언급한 네 단계 중에서 가장 중요하다고 볼 수도 있다. 이 단계의 끝에서는 개발할 시스템이 갖춰야 할 핵심 동작 기능을 제공하는 완전한 공학적 시스템이 완성되어야 한다. 또한 전체 프로젝트 수행 중 중요한 결정이라 할 수 있는 ERP에 대한 완성과 사용자에게 배포하는 문제 등이 다루어진다. 대부분의 프로젝트에서 이러한 결정은 비용과 위험요소가 작게 드는 단계에서 시작하여 많이 드는 단계로의 발전 추세를 일치한다.

개발 프로세스가 항상 변화를 수용할 수 있어야 하지만, 상세화 단계를 거치고 나면 시스템의 아키텍처, 요구사항, 계획은 충분히 안정화되며 대부분의 위험요소는 제거된다. 따라서 이후의 개발을 위한 비용과 일정을 예측하는 것이 용이해진다. 상세화 단계에서는 반복(iteration)마다 실행 가능한 컴포넌트 아키텍처인 WINIZ* Framework이 구축되었다[1,2,13].

상세화 단계의 산출물은 다음과 같다.

- 유스 케이스 모델 (최소 80%의 완성도) - Use Case Diagram, Use Case Scenario, Use Case Specification 등
- 실행 가능한 컴포넌트 아키텍처를 반영한 WINIZ*Framework 구현
- 개선된 위험요소 목록과 비즈니스 케이스
- 프로세스의 흐름을 파악할 수 있는 Sequence Diagram, Activity Diagram, Class Diagram, Component Diagram, Deploy Diagram 등
- 실제관계도(ERD)

2.3 구축(Construction) 단계

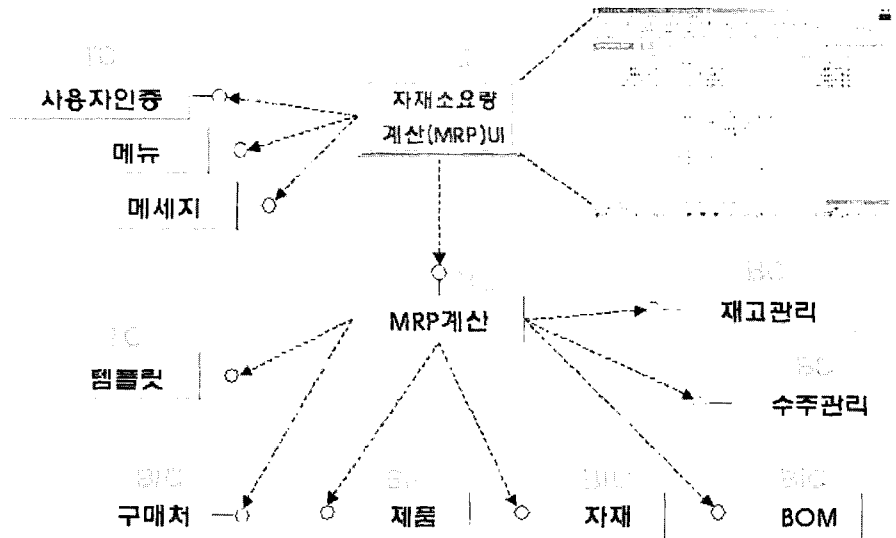
구축 단계에서는 모든 컴포넌트와 어플리케이션의 기능이 개발되어 제품으로 통합되고 철저한 테

스트가 이루어졌다. 구축 단계는 다르게 표현하면 비용과 일정, 그리고 품질을 최적화 하기 위해 자원관리와 공정제어에 중점을 둔 생산 과정이라고 말할 수도 있다. 프로젝트 관리 측면에서 래쇼날의 개발 프로세스는 착수 단계와 상세화단계 동안 소프트웨어 시스템의 지적 자산에 대한 개발에 초점을, 그리고 구축 단계 및 전이 단계 동안에는 배포 가능한 제품의 개발에 초점을 맞추었다 [11,12].

구축단계에서는 기술적컴포넌트(TC : Technical Component), 비즈니스 인프라스트럭처 컴포넌트 (BIC : Business Infrastructure Component), 비즈니스 컴포넌트(BC : Business Component), 애플리케이션 컴포넌트(AC : Application Component) 등 4 종류의 컴포넌트를 개발하였다. TC는 사용자인증, 에러처리 등과 같이 소프트웨어 시스템의 기술적인 서비스를 제공하는 컴포넌트며, BIC는 제품, 계정, 거래처 등과 같이 공통적인 비즈니스 서비스를 제공하는 컴포넌트로 여러 비즈니스 컴포넌트에서 공통으로 사용될 수 있다. BC는 ERP의 핵심 컴포넌트로 수주관리, 전표관리 등과 같이 특정 비즈니스 서비스를 제공하는 컴포넌트며, AC는 수주 등록UI, 전표관리UI 등과 같이 특정 애플리케이션과 관련된 컴포넌트다[11,12,13]. 이러한 4가지 종류의 컴포넌트의 적용사례를 살펴보면 아래 그림과 같다.

구축 단계의 산출물은 최종사용자에게 전달할 준비가 완료된 제품이다. 이러한 구축 단계의 산출물은 다음과 같은 것들을 포함하고 있다.

- 기술적 컴포넌트, 비즈니스 인프라스트럭처 컴포넌트
- 비즈니스 컴포넌트, 애플리케이션 컴포넌트
- ERP 제품(WINIZ)
- 사용자 매뉴얼, 운영자 매뉴얼, 개발자 매뉴얼



2.4 전이(Transition) 단계

전이 단계의 목적은 소프트웨어 제품을 사용자 환경으로의 전이(transition)다. 제품이 최종사용자에게 주어지면 새로운 제품에 대한 요청이나 발견된 문제점의 교정, 개발이 지연된 기능의 완성에 대한 요구가 발생한다.

제품이 최종 사용자 환경에 배포될 수 있을 만큼 완성도가 높아질 때 전이 단계가 시작되었다고 말할 수 있다. 일반적으로 전이 단계에서는 사용자 환경으로의 배포가 모든 프로젝트 구성원에게 득이 될 수 있을 만큼 시스템의 기능과 품질이 받아들일 수 있을 정도의 수준에 이르고 사용자를 위한 문서화가 이루어진다(11,12).

전이 단계에서는 해당 소프트웨어를 최종 사용자에게 인도하기 위해 요구되는 활동에 중점을 둔다. 일반적으로 전이 단계에서는 베타 릴리즈, 정식 릴리즈, 오류 수정 릴리즈, 기능 향상 릴리즈를 위한 여러 번의 반복(iteration)을 거친다. 또한 사용자 지향의 문서 작성, 사용자 교육, 초기 사용 단계에서 사용자 지원, 사용자 feedback에의 대응 등에 많은 노력이 소요된다. 전이 단계에서는 사용자 feedback은 주로 제품 튜닝, 사양 설정,

설치, 사용편이성과 같은 문제에 국한된다.

전이 단계의 산출물은 다음과 같다.

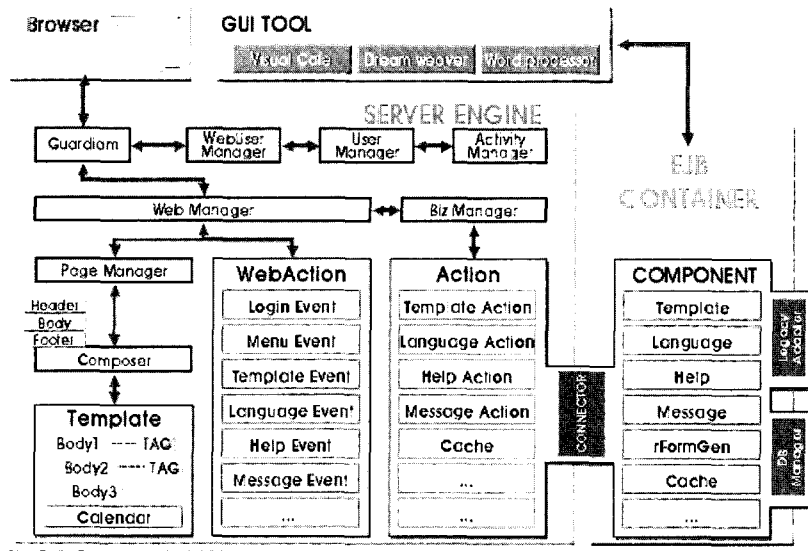
- 제품의 릴리즈 계획 : 베타, 정식, 오류 수정, 기능 향상 릴리즈
- 제품소개서
- 제품의 Customizing & Add on 방법 및 절차
- 제품의 컨설팅방법론

3. WINIZ*Framework

WINIZ는 모든 애플리케이션이 Framework을 기반으로 되어 있으므로 이에 전반적인 흐름을 살펴보기로 한다(2,13).

아래 그림은 WINIZ*Framework의 전체적인 구성도이며 지엔텍의 ERP제품인 WINIZ가 작동되는 절차를 살펴보면 다음과 같다.

- (1) 사용자가 Browser를 통해 Web Server에게 서비스를 요청한다.
- (2) 발생된 모든 요청은 Guardian이라는 Filter 클래스에 전달된다.(관련된 XML 파일에 등록된 정보를 활용해서 관련된 요청 처리)
- (3) Guardian은 초기에 사용자가 사용할 수 있는



- Action 및 Event의 사용 가능 여부를 확인한다.
- (4) 이 때 내부적으로 WebUserManager, UserManger, ActivityManager를 활용하여 확인기능에 활용된다.
 - (5) WebUserManager 클래스는 사용자 인증, 사용자 권한, 로그 온 상태에 대한 처리를 한다. 이런 기능을 수행하기 위해서 UserManager 클래스와 상호 동작 한다.
 - (6) UserManager 클래스에서는 사용자 테이블에 있는 정보를 읽어서 UserMap에 등록하고 사용자가 실행할 수 있는 Activity 정보를 읽어서 사용자별 ActivityMap에 저장한다.
 - (7) ActivityManager는 모든 기능에 대한 ActivityMap을 생성한다. 향후의 프로그램에서 활용한다.
 - (8) (5),(6),(7)의 작업은 초기에 한번 수행 되어서 ServletContext 나 메모리 상에 필요한 정보를 로드하는 작업을 수행한다.
 - (9) 사용자의 요청 중에 xxx.action의 요청이 있으면 다음으로 WebManager에게 전달된다.
 - (10) WebManager는 Business 기능을 수행하기 위한 BizManager 클래스 객체를 생성하고

- 요청한 Action명과 Event 명을 얻는다. 이 정보를 이용해서 실제적인 동작객체인 WebAction 객체를 얻고 실행한다. 그 결과로 Event 객체(일반적인 Entity 정보)가 리턴된다. 결과로 받은 Event 객체를 BizManager에 전달하고 수행을 요청한다.
- (11) BizManager에서는 event명이 포함되어 있는 Action 객체를 얻어서 실제적인 Action 객체의 해당 기능을 수행시킨다. 수행결과로서 Result 객체가 반환된다.
 - (12) WebManager는 반환된 결과를 받아서 RequestDispatcher에 의해서 다음에 보여 주어야 할 페이지로 forward (이때 정보로 xxx.page가 전달 된다.)되며 이것에 의해서 servlet 매핑 정보에 의해서 PageManager로 전달된다.
 - (13) PageManager는 처음 수행될 때 페이지에 관련된 정보(pageDefs.xml)를 모두 읽어서 Map에 저장하고 Composer.jsp로 Forwarding한다.
 - (14) Composer.jsp는 CurrentPage 정보를 이용해서 해당 페이지를 화면상에 나타낸다. 화면

상의 정보를 나타내기 위해서는 Composer Tag Library를 활용한다.

3.1 WINIZ*Framework의 특징(13)

3.1.1 유연한 운영 환경

J2EE 기반으로 개발되어 운영체제, 플랫폼에 상관없이 운영이 가능하다. 시스템의 환경에 따라 Servlet Version 과 EJB Version 중 선택이 가능하다.

3.1.2 Rule Base에 따라 개발 생산성 향상

Model - View - Control 의 구조를 가지므로 개발자간의 명확한 역할 분담이 가능하므로 생산성에 있어서 다른 Framework 보다 매우 우수하다.

3.1.3 Business Component 개발의 최적 환경 제공

WINIZ*Framework은 Framework내에서 기술적컴포넌트(TC), 비즈니스 인프라스트럭처 컴포넌트(BIC), 비즈니스 컴포넌트(BC), 애플리케이션 컴포넌트(AC)를 분리하고 있으므로 ERP의 핵심인 비즈니스 컴포넌트 개발에 집중할 수 있다.

3.1.4 Application 성능 향상

WINIZ*Framework는 Cache 기능을 통해 ERP에서 미리 필요한 데이터를 메모리나 하드디스크에 캐싱함으로써 데이터베이스로의 접근을 최소화 하여 속도가 우수하다.

3.1.5 ASP 환경 지원(6)

동일한 Application으로 여러 회사를 동시에 지원이 가능하다. 이때에 하드웨어는 서비스하는 기업의 수와 데이터의 처리용량에 따라 가변적으로 쉽게 증설이나 감축이 가능하다.

3.1.6 다양한 형태로의 User Interface 적용 지원

웹 페이지마다 개발자가 개발하던 시대는 지났다. WINIZ*Framework는 레이아웃 정보를 데이터베이스로 관리하여(Meta-Data) 유지보수 및 재사용성이 우수하다.

3.1.7 WINIZ를 개발하는 과정에서 주로 적용한 기술

분석설계에서는 RUP방법론과 Rational Rose, Rational Soda, ER-Win와 같은 CASE Tool이 활용되었으며, 구현단계에서는 Html, Xml, Java, JSP, JavaScripts등이 사용되었고 미들웨어로 Appache, WebToB와 같은 Web Server와 Jeus, WebLogic, Tomcat, WebSphere와 같은 WAS를 이용하였고, EJB, Servlet, JDBC와 같은 J2EE기술등을 활용하였다(13).

3.2 WINIZ*Framework의 주요기능(13)

WINIZ*Framework은 J2EE 기반의 MVC 모델로 이루어진 컴포넌트 아키텍처다. WINIZ*Framework에서는 데이터를 포함한 외부로부터의 자극을 EVENT로 정의하고 그 EVENT를 처리한 결과를 RESULT로 반환하는 흐름을 가지고 있다. 이때 EVENT를 처리하는 것은 ACTION이며 이런 ACTION들을 계층적으로 구성한 것이 ACTIVITY로 정의되어 있다.

다시 말해서 비즈니스를 MVC 모델로 구분할 때 request(EVENT)는 Controller를 통해 비즈니스 즉, Model로 접근하며 비즈니스에서 처리된 결과(RESULT)를 View를 통해 응답하는 구조로 이루어져 있다.

비즈니스는 Activity로 구성되어 있으며 비즈니스를 실제로 처리하는 주체는 각각의 EVENT정보를 가지고 있는 ACTION이 되며 비즈니스에서 Persistency를 요구하는 부분은 데이터베이스를 통해 처리하고 있다. WINIZ*Framework 의 주요 기능은 다음과 같다.

3.2.1 Flexible Environment

J2EE 기반이므로 운영체제 및 플랫폼에 독립적이며 중소형 시스템을 위한 Servlet Version 과 대형 시스템을 위한 EJB Version 이 존재하여 시스템 운영환경에 따라 적절한 버전을 사용하면 된다. 물론 Servlet Version에서 EJB Version으로의 확장이나 EJB Version에서 Servlet Version으로의 전이가 완벽하게 보장된다. 운영 환경을 별도의 설정파일로 관리하므로 유지보수가 용이하다.

3.2.2 최소한의 시간으로 최대한의 생산성 보장

MVC(Model - View - Control) 모델로 이루어진 WINIZ*Framework은 개발자에 대한 역할이 명확히 나누어져 있다. View 를 담당하는 개발자는 JSP, Java Script, HTML, Meta-Data 등을 이용하여 User Interface를 개발한다. Control을 담당하는 개발자는 Servlet, Java Beans, Enterprise Java Beans, Stored Procedure 등을 이용하여 Server Side Program 즉, 비즈니스 처리 과정을 개발한다. Model을 담당하는 개발자는 Java Class, Meta-Data를 통해 비즈니스 로직을 구현한다. 개발자 간의 역할이 명확하므로 독립적인 작업이 가능하여 개발 생산성이 매우 우수하다.

3.2.3 Business Component 개발을 위한 최적 환경 제공

WINIZ*Framework는 기술적컴포넌트(TC), 비즈니스 인프라스트럭처 컴포넌트(BIC), 비즈니스 컴포넌트(BC), 애플리케이션 컴포넌트(AC)들을 layer상으로 분리하여 비즈니스 컴포넌트의 독립성을 완벽하게 보장한다. 따라서 비즈니스 컴포넌트의 재사용성 및 유지보수성이 탁월하다. WINIZ*Framework Component에 대한 이해 없이도 손쉽게 비즈니스 컴포넌트의 개발이

가능하다.

3.2.4 Cache를 통한 Application의 성능 향상

WINIZ*Framework는 공통적으로 사용하는 환경 정보나 자주 사용하는 Business Component를 메모리 또는 하드디스크에 캐싱을 하므로 트랜잭션이 발생할 때마다 매번 데이터베이스에 접속할 필요가 없어서 속도가 우수하다.

3.2.5 단일 회사용에서 여러 회사까지 하나의 Application으로 사용 가능

WINIZ*Framework는 운영환경에 따라 1개의 Application Server로 여러 회사에 웹 서비스를 제공할 수 있다. Application 관리용 데이터베이스와 회사별 Business 데이터베이스가 별도로 존재하므로 ASP(Application Service Provider) 서비스가 용이하다(6).

3.2.6 다양한 형태로의 User Interface 적용 지원

WINIZ*Framework는 화면의 레이아웃 정보를 데이터베이스로 관리하므로 웹 페이지 구성이 편리하다. UI에 대한 통일성을 가져갈 수 있으며 필요한 만큼의 레이아웃을 구성할 수 있다. 한번 만들어진 레이아웃 정보는 재사용이 가능하므로 UI 설계에 대한 시간을 절약할 수 있다.

3.2.7 완벽한 동시 다국어 지원(Native Language Support)

WINIZ*Framework는 Application이 다양한 언어 환경에서 동작하도록 지원한다. CTL(Custom Tag Library)과 데이터베이스를 활용하여 End User가 설정한 언어 환경에 맞도록 웹 페이지를 생성해 준다. 또한 여러 나라 언어를 동시에 입출력할 수 있는 유니코드체계(UTF-8)를 완벽하게 지원함으로써 Global 환경 하의 기업에 적합하다.

3.2.8 안정적인 비즈니스 처리

Business Component를 통한 비즈니스를 처리할 때 싱크로나이징 기술과 트랜잭션 관리를 통해 안정적으로 비즈니스를 처리한다. 견고한 시스템을 위한 최적의 Framework 이다.

3.2.9 유지보수의 편리성

WINIZ*Framework는 각종 환경정보들을 데이터베이스와 XML로 관리하므로 유지보수가 쉽고 편리하다.

4. 기대효과

여러 번의 시행착오를 거쳐 2년만에 개발이 완료된 ERP Component인 WINIZ는 작년말에 시장에 출시되어 좋은 반응을 보이고 있으며 향후 고객사 마다의 제품에 대한 버전관리를 본사의 WINIZ Center에서 통합관리하여 WINIZ*Framework중 기술적컴포넌트(TC), 비즈니스 인프라스트럭처 컴포넌트(BIC)는 본사 WINIZ Center에서만 변경·수정하고 비즈니스 컴포넌트(BC), 애플리케이션 컴포넌트(AC)는 고객사의 프로젝트팀에서 변경·수정함으로써 ERP의 컴포넌트를 지속적으로 Upgrade해 나갈 예정이다. 이렇게 Component로 개발된 WINIZ ERP에 대한 기대효과는 다음과 같이 예상된다 [3,4,7,9,13].

첫째, WINIZ는 운영체제 및 하드웨어 플랫폼에 독립적으로 활용이 가능하므로 기존의 15종류에 해당하는 ERP 제품군을 3 종류의 제품군으로 통합 관리할 수 있음으로써 연구인력의 제품군 유지에 대한 고정투자를 감소할 수 있으므로 보다 효율적으로 연구개발에 집중할 수 있다.

둘째, ERP 비즈니스 프로세스의 50% 이상을 프로그램 소스에서 관리하지 않고 데이터베이스상의 Meta-Data에서 관리함으로써 비즈니스 프로

세스의 변경, 추가만으로도 프로그램 소스의 컴파일 없이 빠른 시간 내에 커스터마이징이 가능할 것으로 예상된다.

셋째, 요즘 ASP를 사용하고자 할 때에는 Application에서 제공하는 기능의 무조건 사용은 가능하지만 기업마다 다른 부분에 대해서는 지원이 매우 어려운 것이 현실이다. 그러나 WINIZ에서는 비즈니스 프로세스가 서로 달라도 하나의 Application을 사용하면서 서로 다른 비즈니스 프로세스에 대해서는 Component Configuration을 이용하여 기업마다의 요구조건에 맞는 컴포넌트를 실시간으로 지정하여 활용이 가능함으로써 ASP에 대한 본격적인 사업기회를 얻을 수 있을 것이다.

넷째, 요즘의 기업들은 이미 많은 다양한 IT자원, 즉, 하드웨어 및 운영체제를 보유하고 있다. 이러한 기업들이 IT 솔루션을 도입하려고 할 때 가장 고민하는 것 중의 하나가 기존에 보유하고 있는 IT자원을 재활용할 수 있는 솔루션을 찾고 있다. 이러한 필요성을 만족시킬 수 있는 지애텍의 WINIZ는 현재 상용화되고 있는 대부분의 하드웨어 및 운영체제에서 Application의 변경 없이 설치 및 운영이 가능하다.

따라서, 고객은 기존에 보유하고 있는 IT자원의 재활용으로 투자비용을 절약할 수 있으며, 지애텍은 이제 더 이상 ERP가 구동되는 플랫폼에 고민하지 않고 자유로워짐으로써 제품 자체에 핵심역량을 집중할 수 있을 것이다.

참고문헌

- [1] Blechar, M. Component-Based Development, Gartner Symposium/ITxpo 2000, Orlando, FL, October 2000.
- [2] Larsen, G. Component-Based Enterprise Frameworks, Communications of ACM, Vol. 43, No. 10, October 2000.
- [3] Marinescu, F. "2001: The Year of Web Services," Java Developer's Journal, April 2001.
- [4] Object Management Group (<http://www.omg.org/>)
- [5] 김기열, 컴포넌트 기반 개발(CBD), 경영과 컴퓨터, 2002년 9월,10월,11월호
- [6] 노상석, ASP 활용시 도입성과에 미치는 주요 성공요인 변수의 분석, 고려대학교 경영대학원 석사논문, 2001
- [7] 박준성, CBD에 대한 소고, 정보통신연구진흥지 제3권 제3호, 2001년
- [8] 박준성, CBD 시장 바로보기, 디지털 타임즈, 2001년 8월 3일
- [9] 박준성, CBD의 개요, 삼성SDS, 2001년 12월
- [10] 오영배, 박준성, CBD 적용사례 연구, 한국정보과학회 소프트웨어공학회지 1999년 9월호
- [11] 지애텍 홈페이지 (<http://www.gntech.co.kr/>)
- [12] 컴포넌트비전 홈페이지 (<http://www.componentvision.com/>)
- [13] 한국소프트웨어컴포넌트컨소시엄 (<http://www.component.or.kr/>)

저자약력



노상석

1982년 서울대학교 자연대학 지질학과 졸업 (이학사)
 2001년 고려대학교 경영대학원 MIS 전공 (MBA)
 1988년-1990년 대우엔지니어링 토목사업본부(CAD설계)
 1990년-1991년 큐닉스데이터시스템 소프트웨어사업부(MIS 개발)
 1991년- 현재 지애텍 정보기술연구소 연구소장(ERP제품개발)
 관심분야 : ERP, CBD, ASP, SCM, CRM, UML, WAS, J2EE, EJB..등
 이 메 일 : ssnoh@gntech.co.kr
 연 락 처 : 회사 02-562-6888(280),
 Mobile: 016-290-4992