

네트워크 컴퓨팅 시대의 필수 요소

21세기 기업의 경쟁력은 인터넷에 자신만의 비즈니스를 얼마나 효과적으로 접목시키는가에 달려있다. 이런 의미에서 최근 기존 시스템을 인터넷 환경으로 전환하려는 움직임이 일고 있다. 이 글에서는 인터넷 시스템 구축을 위한 애플리케이션의 추이와 파리크라상 점포지원 시스템 구축에 사용된 Oracle Designer, Oracle Developer & Server 6.0의 신기능 및 Feature에 대하여 간략하게 살펴본다. 더불어 인터넷 애플리케이션 구축 사례로 파리크라상의 점포지원시스템에 대하여 살펴 보도록 한다.

- 정팔교/한국 A.I SOFT 인트라넷 기술팀 팀장
(pkchung@aisoft.co.kr)

환경 및 기술의 급격한 변화

올해는 모든 산업 전반에 걸쳐 수많은 홈페이지, Data Warehouse, EC Solution, ERP, DB Marketing, CRM (Customer Relationship Management) 등 WWW 기술을 토대로 사내/외 업무를 효율적으로 적용하고자 하는 움직임이 유난히 컸던 해라고 생각 된다.

이전에는 많은 전산 담당자들에게 있어서 인트라넷은 사내 E-mail이나 BBS, 게시판 정도의 기능으로 국한시켜 인트라넷은 일종의 기업내 그룹웨어 기능이라는 생각을 가지고 있었으나 현재 많은 기업에서는 이 인트라넷을 보다 폭 넓게 활용하고 현재 대기업과 IT 업계 중심으로 여러 업무에 적용 중이거나 이미 적용을 하고 있다. 인트라넷의 근본적 개념은 기존의 클라이언트/서버 환경(Legacy System)에 투자된 모든 H/W, S/W를 그대로 유지하면서 기존에 운영하던 기업내의 모든 업무 프로그램을 Web으로 구축하는 것을 의미한다. 이를 위해 데이터베이스와 웹서버의 기술은 필수적인 것이 되었다.

그런데, 이를 위한 Web 기술의 뒷받침에 불신하는 많은 사람들은 인트라넷은 버블(Bubble)이 아니냐 라고 생각도 하였으나, 급속도로 발전한 기술로 인해 현재는 거의 대부분의 업무에 적용

되고 있다 해도 과언이 아니다.

또한 인트라넷의 핵심 요소 기술이 되는 Web과 DBMS와의 연동이 효과적으로 이미 선을 보인 상태이며, 기업 내에서는 위험 부담이 없는 분야에 조심스럽게 Pilot 시스템으로 적용 시켜 나가고 있는 실정이다.

희미하게나마 남아 있는 인트라넷의 부담감은 Web을 이용한 OLTP 분야에 대한 적용기술이다. 이를 위해 많은 벤더에서는 C/S환경에서의 개발도구의 Plug-in, Java를 이용한 방법, ActiveX 등의 방법으로 많은 방법론을 제시하고 있으며 이들 모두가 Web 상에서의 OLTP를 가능토록 하는 기술을 보유하고 있으므로 개발자들은 다음과 같은 부분에 좀더 많은 관심을 가지고 지켜 보아야 할 것이다.

첫번째로, 웹브라우저로 표시되는 작업 화면, 즉 프로그램의 속도이다. 이는 해당 프로그램의 크기에 좌우될 수 있다. 프로그램 자체가 브라우저내로 Download 되어야 하기 때문이다.

두번째로, OLTP의 가장 중요한 부분으로 데이터의 무결성(Data Integrity)이다. 동일한 화면으로 동시 여러명이 접속하여 동일한 데이터를 수정하고자 할 경우 첫번째 사용자가 해당 필드의 값을 먼저 수정하였을 경우 다른 동시 사용자가 해당 필드를

수정하려고 값을 입력시키려고 할 때 즉시 에러 메시지를 표시하여야만 한다.

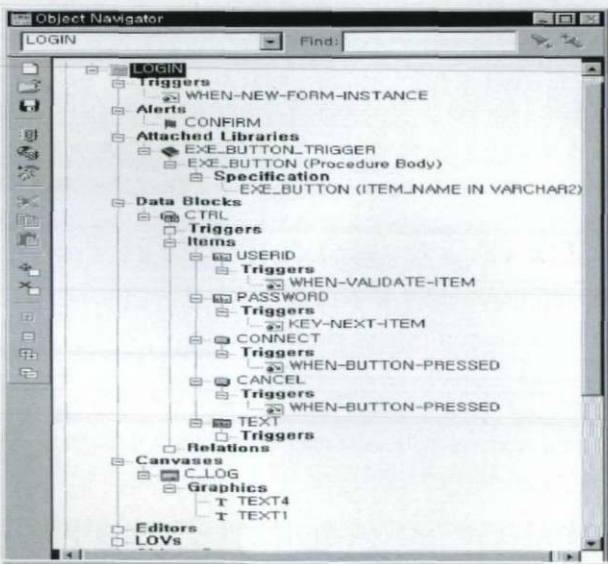
이러한 기능을 처리하기 위해서는 해당 프로그램이 지속적으로 서버의 데이터베이스와 접속을 유지하여야만 가능하다. 그러나 웹(HTTP Protocol)의 장점이자, 단점은 지속적으로 서버로 접속을 유지하고 있지 못하다는 점이다.

세번째로, 기존에 이미 투자하여 사용중인 Client/Server 개발 툴의 활용 및 Web OLTP 와의 병행처리 문제이다. 이제 막 완료하여 기업내에서 사용중인 프로그램을 버릴 수는 없는 일이다. 또한, 필요에 따라 동일한 프로그램을 웹의 장점을 이용하여 Web 환경에서 이용하고자 하는 End user 와 기존의 Client/Server 개발 툴에 의한 프로그램을 사용하고자 하는 요구가 공존 하리라 생각되며, 이러한 두 가지 요구를 충족시켜 줄 수 있는 기능이 제공되어야 한다. Web 상에서의 OLTP를 구현하고자 할 경우 위의 3가지 사항은 필수적으로 제공되어야 될 사항이다. 작년 한해는 Web OLTP의 원년도가 되었고 또한 웹을 이용한 전자 상거래(EC)가 활발하게 시작이 되었다고 생각한다.

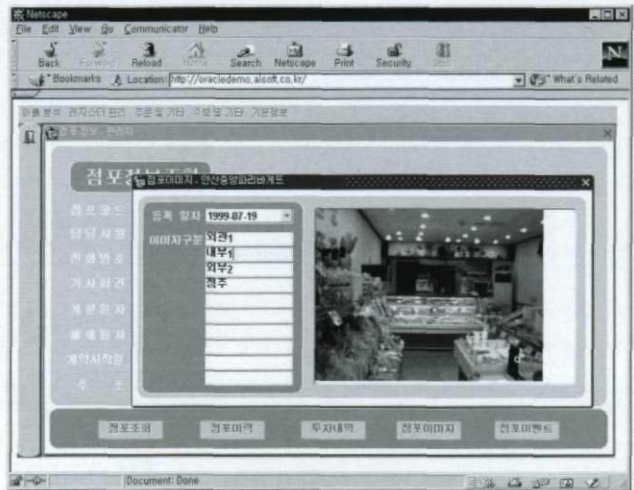
기존 PC통신을 이용한 상거래나 기업간의 EDI를 웹으로 구현함에 있어서 필요한 요소기술, 즉 전자 지불 시스템이나, 보안 시스템의 기술 소개로 소비자자와 기업, 기업과 기업, 은행과의 연계 등의 상거래 기술 소개로 인터넷이 더 이상 그룹웨어만의 적용 기술이 아니라, 기업 전반의 업무에 적용되고 있다.

DB 마케팅, CRM의 솔루션

더불어 요즘 각광 받기 시작한 DB 마케팅, CRM(Customer



〈그림 1〉 Developer 6.0의 유저 인터페이스



〈그림 2〉 파리크라상의 영업정보시스템 메인 화면

Relationship Management) 같은 경우는 다음과 같은 시나리오도 가능하게 해주는 솔루션이 되어 버렸다.

- 1) 어느날 아침 우리(여기서 우리란 대한민국의 모든 상품공급회사)의 prospect가 E-mail, DM 등을 통하여 신상품에 대한 정보를 받게 된다.
 - 2) prospect는 우리의 web site를 search하여서 상품에 대한 정보를 획득하고, Internet을 통하여 상품을 주문한다. 그리고 해당 영업사원의 방문을 요청하게 된다.
 - 3) 영업사원의 방문 요청은 Call Center system과 연계가 되어서 Agent는 영업사원의 일정을 확인하여 다시 E-mail 또는 전화로 Prospect 에게 방문일정을 확인한다.
 - 4) 영업사원은 현장을 방문하여 고객과 상담을 한 뒤, Mobile Sales 를 이용하여 상품에 대한 주문을 최종 처리한다.
 - 5) 고객은 상품을 사용하다가 문제점 또는 up-selling의 요청사항이 발생되어 Web을 통하여 지원을 요청하게 된다.
 - 6) 이것은 다시 Call Center solution과 연계가 되어 Agent는 Field Service engineer의 일정을 확인하여 engineer 에게 job request를 보낸다. 이때, 사용되는 것이 Mobile-Service(GPS기능 포함)다.
 - 7) 엔지니어는 현장을 방문하여 문제점을 처리한 뒤, billing을 고객에게 하고, 이를 Mobile Service를 이용하여 시스템으로 전송한다.
- 그렇다면 우리는 위 시나리오를 이룰 수 있으려면 다음과 같은 것이 선결되어야 한다는 것을 알 수 있다.

첫째, Internet을 통한 재고확인, 주문처리, billing이 가능하여야 하고, 둘째, Internet과 Call Center와 연계된 business Flow 가 있어야 한다. 셋째, 현장에서 문제처리 정보, billing 정보 등을

remote 환경에서 upload하고, 엔지니어의 일정을 scheduling 할 수 있어야만 한다.

즉, 다시 말하면 크게 Internet/Call Center/Mobile의 세가지가 서로 통합(integration)되어 구동되어 진다는 것이다.

국내의 Call Center는 현재 Inbound Call 위주의 기능을 수용하고 있지만, 향후에는 Outbound Call 및 전사원의 Agent화(상담원화)가 되어 가리라고 본다(매우 큰 volume임).

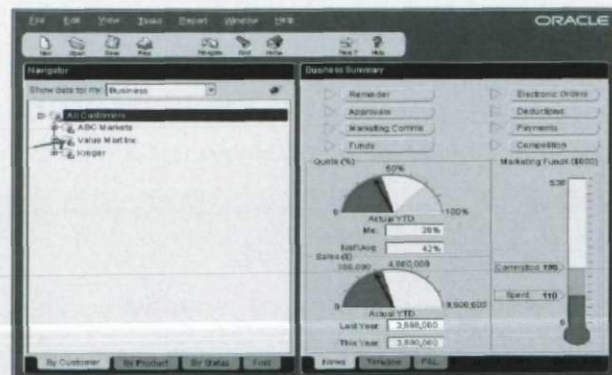
왜냐하면 이는 PBX/IVR 등과 연계가 되어 고객이 전화를 할 경우에 고객에 대한 모든 정보(profile, 구매정보, 문제처리정보 등)를 단말기를 통하여 볼 수가 있어 좀더 고객과 친밀하게 대화를 할 수 있고, 이를 통하여 고객은 친밀감을 느낄 수 있으며, 또한 기존구매 정보 등을 이용하여 cross selling/up selling이 가능하기 때문이다.

Call Center를 조그마한 기능이라고 생각하여, deal에서 간과하고 무시해서는 절대로 상황에 왔다고 보아도 될 것이다. 그리고 예상보다 빨리 CRM이란 Concept이 국내시장에서 수용되는 것 같다고 생각한다.

앞에서 기술한 모든 움직임, 즉 인터넷 애플리케이션 구축 목적의 가장 큰 이유는 한없이 커져만 가는 Client용 Desktop PC의 유지 비용이나 요구사항을 기업에서는 가장 효과적으로 대응하면서도 서버의 성능 향상, 단일화된 Presentation Layer(Web Browser)만으로 모든 업무를 처리하고자 하는 것으로 다가오는 Network Computing 시대를 효과적으로 준비하는 일이라 할 수 있으며, 향후 이런 방향으로 인터넷 애플리케이션이 구축되어질 것임을 필자는 조심스럽게 확신할 수 있다.

구축 틀과 파리크라상의 영업정보시스템

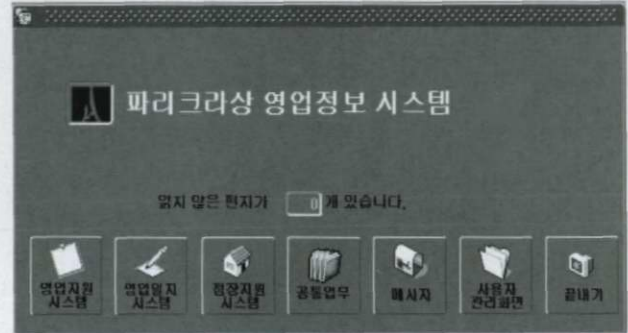
그러면 이런 3-Tier Solution을 가능하게 하여주는 Tool인 Oracle Designer와 Oracle Developer & Server 6.0에 대하여



〈그림 3〉 디벨로퍼 6.0의 Object Navigator 상에서 login 모듈을 만드는 화면

살펴보기로 한다.

일반적으로 정보공학 방법론에 의한 시스템 개발시 분석용 CASE 도구에 대한 활용 방안은 방법론의 각 공정에 따른 성과물의 작성을 CASE 도구로 자동화하며, 작성된 내용은 정보저장소(Repository) 내에 보관하여 개발자들과 모델을 공유하여 사용자 요구의 정확한 분석을 도모토록 하는데 있다.



〈그림 4〉 점포정보를 웹상에서 조회한 결과

또한 CASE Tool 도입은 사용자에게 정보공학론에 대한 마인드 형성을 가져오며 CASE Tool에 대한 이해 및 기술습득을 가져온다.

파리크라상 프로젝트를 통하여 발생된 기대 효과 정도는 다음과 같다.

- 1) IE(Information Engineering) 기법을 통한 Project 수행 경험 축적
- 2) CASE Tool을 이용한 작업품질 향상
- 3) 상위단계(Planning, Analysis, Design)에 많은 노력을 투자 Maintenance 노력 감소
- 4) 새로운 기술에 대한 직원들의 업무의욕 고취
- 5) 교육을 통한 CASE의 이해와 업무 패턴 변화의 기반 조성

Oracle Designer의 각 단계별 지원 과정을 상세히 살펴보면 〈표〉와 같다.

〈표〉와 같이 Oracle Designer의 기능을 이용하여 파리크라상의 점포지원시스템을 설계하였고 설계가 끝나자 ERD와 FHD만 보면 누구든지 Coding 할 수 있는 명확한 시스템 설계가 되었다. 개발자와 현업 담당자간에 계속된 의견 교환으로 시스템 설계를 확정짓고 나서 Developer 6.0을 가지고서 Coding을 하게 되었다.

여기서 Oracle Developer 6.0의 신기능에 대하여 살펴보면 다음과 같다.

Hierarchical Tree 구조가 가능하여 보다 효과적인 프로그래밍을 할 수 있으며 Java Virtual Machine을 자체적으로 내장하고 있어(Oracle Jinitiator), Builder 상에서(C/S 환경) Run form web button을 누름으로써, 웹 상에서 어떤 식으로 보여질지 previewer할 수 있는 기능이 추가 되었고 한 줄의 Parameter를 추가함으로써 web forms 구현시 swing library, background image, splash screen 등을 사용할 수 있다.

또한 Forms에서의 OCX 지원 이외에 WebForms를 위한 java Beans나 PJC(Pluggable Java Components)를 지원하며 Web Forms-java로 보았을 경우 화면의 구성 GUI가 투박하지 아니하고 미려한 UI를 가지게 되었고 무엇보다도 개발자들에게 보다 나은 개발 생산성을 제공하기 위해 Editor에서의 Reserved Word에 대한 Coloring, 보고서 Runtime시에 보고서 Layout 수정이 가능해졌다.

아울러 디벨로퍼 서버를 이용하여 C/S 환경에서 개발한 프

그램을 소스 코드 수정 없이 WEB 환경으로 전이가 가능하다는 것이 가장 큰 장점이다. <그림 1>은 Web에서 실행된 Developer 6.0의 미려한 UI이다. <그림 2>는 파리크라상의 영업정보시스템 메인 화면이다.

사람마다 권한이 있고 로그인 과정을 거쳐서 사용할 수 있는 시스템에 자동 연결이 된다. 실제로 <그림 2>는 브라우저 상에서 자바 애플릿 형태로 다운로드 되어진 화면이다.

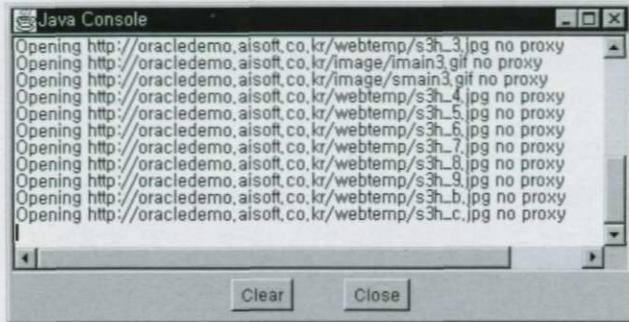
<그림 3>은 디벨로퍼 6.0의 Object Navigator 상에서 login 모듈을 만드는 화면이다. Base Template를 설정하고 Look and Feel을 지정하고 6.0의 새로운 기능중의 하나인 Run on the Web 버튼을 이용하여 C/S 프로그램을 웹에서 실행되었을 때의 결과 화면을 고려하여 Coding하였다.

<그림 4>는 점포정보를 웹상에서 조회해 본 결과이다. Graphic Server를 사용하여 Web 상에서 그래픽 정보를 GIF Format으로 보여준다.

<표> Oracle Designer의 각 단계별 지원 과정

BPR과 분석 단계		
프로세스 모델러 (Process Modeller)	프로세스를 모델링하고 비즈니스 활동을 재 디자인하기 위한 툴이다. 이곳서 정의되는 모델 정의 사항은 리퍼지토리에 저장되며, 이 내용은 이후의 툴에 대한 일반 정보 모델의 근간이 된다. 프로세스 모델러의 목적은 비즈니스의 사용자와 기술자간의 의사소통을 위한 인터페이스를 제공함에 있다.	
분석(Analysis)	정보와 기능 NEED의 시스템 요구사항을 모델화할 수 있는 툴들의 묶음. 지원 가능한 다이어그램은 Entity Relationship Diagram, Function Hierarchy Diagram, Dataflow Diagram 등이 있으며, 사용자는 시스템 모델러를 통해서 분석 작업을 수행	
	Entity Relationship Diagrammer	개념적 데이터 모델을 지원하는 도구로서 수퍼타입/서브타입, 아크관계, UID BAR관계, TRANSFERABLE 관계, RECURSIVE 관계 등을 지원
	Function Hierarchy Diagrammer	개념적 기능 모델링을 지원하는 도구로서 마스터/슬레이브, 엔티티/속성 사용정의 등 다양한 기능을 제공
	Dataflow Diagrammer	기능과 데이터간의 관계를 시각화하여 전체적인 업무 분석작업을 지원
디자인(Design)		
디자인 에디터 (Design Editor)	설계단계를 지원하는 도구로서 데이터와 프로그램 설계의 시각적 표현을 지원. Server Function은 논리적/물리적 데이터 설계단계를 지원하는 도구로서 각종 오브젝트는 물론 반정규화 작업도 지원. 이 도구를 통해서 분산 데이터베이스 설계지원 및 기존 데이터베이스로부터 역공학을 지원하며 Modules Function은 모듈에서 사용되는 테이블과 컬럼을 정의하는 다이어그램 도구로서 이 도구를 통해서 폼즈와 리포트, Visual Basic, Web Server등을 정의하여 자동코드생성기를 통해서 프로그램을 얻을 수 있게 된다. 또한 Server와 Module로의 역공학, Logic의 생성과 활용 등을 정의할 수 있다.	
제너레이터 (Generator)		
	서버 제너레이터	Oracle 7/8, Oracle Lite, Oracle RdB, Oracle Application Server 지원 타 데이터베이스 지원: DB2/2, Microsoft SQL Server, Sybase, ANSI SQL DDL, ODBC 데이터베이스 지원.
	클라이언트 제너레이터	Oracle Developer, Visual Basic, C++ Class
리퍼지토리		
리퍼지토리 서비스 (Repository Service)	매트릭스 다이어그램머	이 툴을 통해서 2차원적으로 모든 리퍼지토리의 내용을 서로 비교 검토할 수 있다.
	리퍼지토리 리포트	다양한 보고서 약 147개 이상의 기본 보고서를 제공
	리퍼지토리 오브젝트 내비게이터	정의된 모든 자료사전의 내용을 관리할 수 있는 도구
리퍼지토리 관리자 (Repository Manager)	리퍼지토리 설치, 사용자 등록, 사용자 정의 확장 등의 작업을 담당	

〈그림 5〉의 화면은 Oracle Jinitiator에 의하여 클라이언트 브라우저상에 다운로드 되어지는 이미지 정보를 Java Console에서 볼 수 있다.



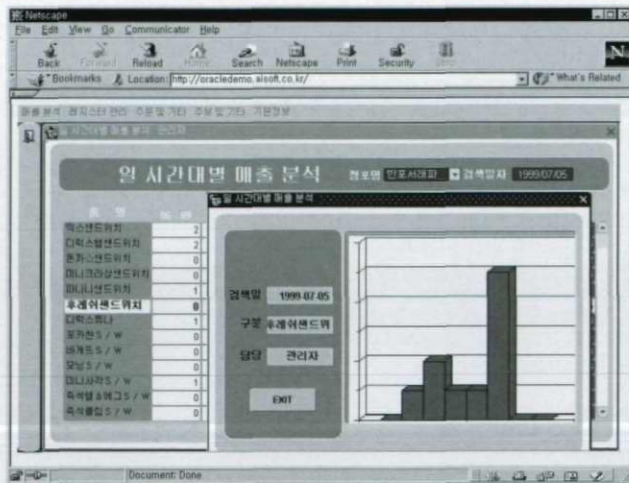
〈그림 5〉 Java Console 화면

〈그림 6〉의 화면을 보면 UI가 크게 강화된 모습을 볼 수 있다. 타원형의 스크롤바, 버티컬 메뉴틀바, 그리고 PJC(Pluggable Java Components)를 지원하기 때문에 자바 클래스를 사용할 수 있으므로 Beans Area를 설정하고서 프로퍼티에서 Implementation Class에다가 원하는 클래스를 연결하는 것으로 더 미려한 UI 환경을 유저가 직접 만들 수도 있다.

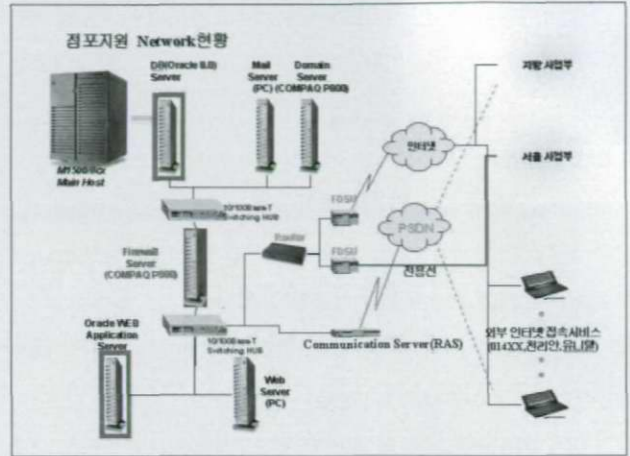
파리크라상의 Network 구성도를 살펴보면 〈그림 7〉과 같다.

본사 내부에서는 그대로 C/S 환경이나 Web을 사용하고 리모트 지역인 점포 같은 곳에서는 웹브라우저만 컴퓨터에 있으면 3-Tier인 Web으로 본 시스템을 사용할 수가 있었다.

파리크라상의 개발 동기는 기존 HOST에서 운영해오던 영업지원 시스템은 TCP/IP 프로토콜을 이용한 터미널 형태의 네트워크로 구성되어 정보를 제공해 오고 있었다. 이러한 시점에서 현재 환경에서의 보다 더 나은 서비스를 제공하기에는 한계가 있다고



〈그림 6〉 일시간 대 점포별 매출 그래프 화면



〈그림 7〉 파리크라상의 Network 구성도

생각하게 되었고 사용자의 편의와 IMAGE 정보 제공을 위하여 인트라넷 환경에서 신규로 점장지원 시스템을, 그리고 기존 HOST에서 운영되어 오던 영업지원 시스템과 임원정보 시스템도 이러한 환경적인 변화가 필요하여 동시에 개발하는 것으로 내부적으로 결정 추진하게 되었다.

늘어나는 유저의 유지보수, HOST 자원의 한계, GUI 요구, 현 네트워크 추세 등 이러한 모든 요소를 충족하기 위해 결국 인트라넷 환경의 솔루션으로 눈을 돌리게 될 수 밖에 없었다. 인터넷은 고객의 관리 능력을 향상 시키기 위한 좋은 도구라고 생각하고 특히 필드에서 영업하는 영업 사원들과 직접 고객과 대면하는 점장들, 변화하는 고객의 요구와 기대를 파악하고 차별화 된 가치와 만족을 계속적으로 제공 해야 하는 이러한 환경에서 움직이는 서비스를 생각할 때, 고객과의 관계를 보다 잘 관리하기 위하여 다년간 축적된 노하우와 진보된 네트워크 환경의 결합이 된 것이었다.

사람들이 각 기능을 수행하는데 동일한 정보를 사용하고 동일한 목표를 추구하면서 마케팅, 판매, 주문 그리고 서비스간에 동일한 자료를 제공하는 WEB 기반의 시스템, 몇 년간 소개되어 온 새로운 기술 활용에 초점을 두고 인터넷, 모바일 컴퓨팅 그리고 통신에서 진보된 기술 마케팅, 효과적 고객관리 등 기본 사항을 충족 할 만한 것들을 선택 하게 된 것이다.

결국 파리크라상 시스템 구축으로 인하여 직원들은 사용자 별도 교육 없이도 인터넷만 사용할 수 있으면 누구나 쉽게 사용할 수 있었고 웹 브라우저상에서 구동이 되기 때문에 클라이언트에 별도의 유지나 보수를 할 필요도 없으며 인터넷 환경으로 전환이 되었기 때문에 통화 대기란 용어도 사라지게 됐으며 향후 인터넷 컴퓨팅 환경으로 가는데 일조를 하게 되었다.