논문 14-39B-02-06

한국통신학회논문지(J-KICS) '14-02 Vol.39B No.02
http://dx.doi.org/10.7840/kics.2014.39B.2.102

모바일 비즈니스를 위한 모바일 미들웨어 시스템 설계 및 구현

이 일 주◆

Design and Implementation of a Mobile Middleware System for Mobile Business

Il-joo Lee◆

요 약

현재의 통신 및 네트워크 환경은 기존의 유선 인터넷에서 무선 및 모바일 기반으로 빠르게 움직이고 있으며 환경의 변화는 비즈니스 방법에도 많은 영향을 미치고 있다. 많은 기업들은 제품의 경쟁력확보를 위해 모바일 비즈니스 도입을 적극 추진하고 있으며, 효율적이고 안정적인 모바일 솔루션을 필요로 하고 있다. 그러나 최적의 모바일 컴퓨팅환경을 구축할 수 있는 방법과 기존의 업무프로세스 및 방대한 양의 데이터베이스를 어떻게 활용할 것인가에 대한 방안이 요구된다. 본 논문에서는 유선 상에서 운용되는 다양한 전산자원을 무선 LAN과 이동전화망, 모바일 디바이스로 연동할 수 있는 모바일 비즈니스 구축 지원 도구인 모바일 미들웨어 시스템을 구현하고자 한다. 이를 위해 일반기업에서 구축된 기간업무를 쉽고, 빠르게 모바일 환경으로 확장 시키는 강력한 무선 데이터 접근 도구를 제공한다. 제안 솔루션을 산업 현장에 적용 시 기존의 비즈니스 로직 및 자원의 변경 없이 경제적으로 레거시 업무 프로세스를 모바일 환경으로 변환 할 수 있다.

Key Words : mobile middleware, mobile server, mobile client, mobile business, wireless data, legacy system

ABSTRACT

Present communication and network environment have been moving rapidly toward wireless and mobile base from existing wired internet base. This change of trend influences greatly on business methods accordingly. Therefore many enterprises are trying hard to adopt mobile business in order to gain competitive edge of their products and they are in need of more effective and stable mobile solutions. However, the method of establishing optimal mobile computing environment and how to handle existing business process and use vast amount of database are still needed. Therefore this paper tries to realize a mobile middleware system as a mobile business establishment supporting tool that could link various computational resource on wired internet with wireless LAN, mobile phone network, and mobile devices. To accomplish that specific goal, this paper provides a powerful tool of mobile and wireless application data access that could expand the line of business already set up in general enterprises easily and rapidly into mobile environment. When this suggested solution is applied in the field of industry, it can economically change legacy business process into mobile environment without having to change existing logic and resources at all.

◆ First Author : 동원대학교 스마트IT콘텐츠과, ijlee@tw.ac.kr, 정회원
논문번호 : KICS2013-11-485, 접수일자 : 2013년 11월 6일, 심사일자 : 2013년 12월 16일, 최종논문접수일자 : 2014년 2월 10일

Ⅰ. 서 론

최근 언제, 어디서나 가능한 네트워크 액세스에 대한 수요 증가는 노트북, 태블릿, 스마트폰까지 포함하도록 확장되었다. 또한 원격 근무자나 직원이 사내에서 이동하면서 편리하게 기업의 네트워크를 사용할 수 있도록 기업의 무선랜 도입이 꾸준히 증가하고 있다[1]. 무선 데이터 통신 환경이 발달함에 따라 개인용 단말기를 이용하여 필요한 데이터를 실시간으로 접근하는 것이 필요한 다양한 어플리케이션 분야가 확산되고 있다[2]. 미들웨어(Middleware)는 분산 컴퓨팅 환경에서 서로 다른 기종의 하드웨어나 프로토콜, 통신 환경 등을 연결하여, 응용프로그램과 그 프로그램이 운영되는 환경 간에 원만한 통신이 이루어질 수 있게 하는 소프트웨어를 말한다[3]. 기존의 미들웨어 시스템은 유선 환경을 중심으로 개발되었기 때문에 미들웨어 및 관련 응용 프로그램들의 규모 또는 운용성이 크다. 따라서 이동 단말의 CPU 및 메모리 등 하드웨어 자원의 취약성과 네트워크의 낮은 대역폭을 가진 무선 이동성 환경에 적용하기 어렵다는 문제가 발생한다[4]. 모바일미들웨어 프로그램은 사용자의 단말기와 데이터베이스를 무선 LAN 또는 이동전화망으로 접속하여 필요한 데이터를 실시간으로 접근할 수 있는 어플리케이션을 개발하는데 필요한 도구이다[5]. 일반적인 레거시(legacy) 어플리케이션이 모놀리식(monolithic) 어플리케이션 서버와 ESB(Enterprise Service Bus)의 정보전달에 중점을 둔다면, 모바일 어플리케이션 구조는 웹서버와 어플리케이션 서버를 포함하는 하이브리드 모델로써 이루어진다[6]. 현재 SAP, IBM 등 국내외 여러 회사에서 모바일미들웨어 제품을 개발하여 보급하고 있는데 개발의 편의성이 확보된다는 점을 강조하고 있다. 그러나 실제 프로젝트에 들어가면 SI개발이 어쩔 수 없이 들어간다는 것이 현업 개발자들의 지적이며, 플랫폼의 도입 목적이 개발속도를 단축시키고 개발 편의성을 확보하는 데 있다는 점을 감안한다면 플랫폼 도입의 목적 자체가 희석되고 있는 것이다[7]. 본 논문에서는 일반 기업에서 구축된 기간업무를 쉽고, 빠르게 모바일 환경으로 확장시키는 강력한 무선 데이터 접근방법을 제안한다. 제안 도구를 이용하여 모바일 환경을 위한 업무 및 응용기술을 개발하고, 기업의 다양한 비즈니스를 모바일 컴퓨팅 환경으로의 확장·접목할 수 있도록 한다. 이를 위해 데이터베이스와 업무 프로세스를 모바일 환경에서 연동할 수 있는 미들웨어 프로그램을 개발하고, 기존의 ERP(Enterprise Resource Planning)환경을 모바일 ERP로 확장할 수 있는 시스템을 구현하고자 한다. ODBC(open data base connectivity)와 JDBC(Java data base connectivity)를 지원하는 모든 상용 RDB와의 모바일 연동 기술을 구현하며, Web 기반 미들웨어, TCP/IP 기반 미들웨어, SAP/R3등의 상용 어플리케이션과 원활한 호환이 가능하도록 한다. 본 연구에서는 첫째, 여러 가지 다양한 형태의 데이터베이스에서 수집하고 관리한 데이터를 모바일 업무영역으로 주고받기 위한 프로그램을 어떻게 설계하고 구현할 것인가 하는 모바일 미들웨어 시스템에 대해 설계한다. 둘째, 단말기 사용자가 요구하는 사양을 만족 시킬 수 있는 다양한 모바일 디바이스에 대한 호환 기술을 지원한다. 셋째, 기존의 복잡한 레거시업무 프로세스 및 자원을 모바일 환경으로 적용하기 위해, 최소한의 추가적인 경비를 투입하여 활용할 수 있는 도구 및 방법에 대한 연구한다. 그림 1에서 모바일 미들웨어는 클라이언트들에 대한 무선 랜 등의 접속을 처리하며, 서버에 대해 프로토콜 변환 요청 연결 및 작업결과의 정형화, 동시 접근자의 관리, 실시간 트랜잭션 등의 작업을 수행한다.

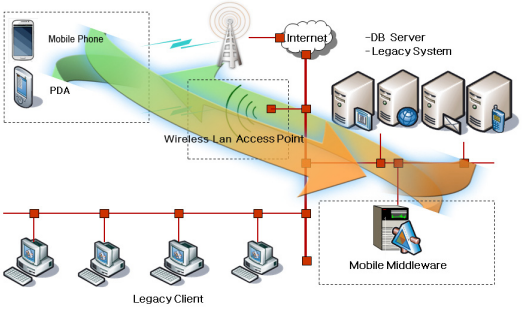

그림 1. 모바일 미들웨어 위치
Fig. 1. Position of Mobile Middleware

Ⅱ. 모바일 미들웨어 설계

2.1. 요구 기능 정의

2.1.1 실시간 모바일 컴퓨팅

기존의 업무를 모바일 환경에서 실시간으로 적용하기 위해서는 여러 가지 복잡한 프로그래밍 과정을 거쳐야 하므로 기존의 업무 환경과 모바일 환경간의 전이를 용이하게 해주는 강력한 하부 구조가 필요하다. 특히 모바일 환경에서는 실시간의 데이터 처리가 필수적인 경우가 많이 있으므로, 모바일 컴퓨팅 응용프로그램 개발 도구는 기존의 서버와 클라이언트간의

데이터 동기화(Synchronization)로 인한 실시간성의 제약의 한계를 뛰어넘는 기술을 제공해야한다. 그러나 지금까지의 대부분의 모바일 어플리케이션은 서버로부터 데이터를 단말기에 다운로드하여, 단말기에서 일정 시간동안 작업을 거친 후에, 서버의 데이터와 다시 동기화하는 방법이었다. 이와 같은 방법으로는 특정 시간에 단말기에 존재하는 자료가 수시로 변경, 추가되는 데이터를 반영하지 못한 부정확한 자료가 될 수 있다. 특히 의료 분야나, 금융 분야 등의 어플리케이션에서는 이와 같은 데이터의 일관성 결여는 심각한 문제를 초래할 수 있으므로, 모바일 컴퓨팅 환경에서는 실시간의 데이터를 지원함으로써 이런 문제를 해결할 수 있다.

2.1.2 기존 업무의 모바일 환경으로의 용이한 확장

각 기업들은 기존 업무 프로세서에서 서비스를 업그레이드 하거나 보안을 강화하는 등의 의도로 레거시 시스템을 수정하여야 하는 경우가 많은데, 그 때마다 광범위한 레거시 시스템을 이해하고 변경하는 일은 매우 비효율적이다[8]. 하지만 3-Tier 구조에 기반한 모바일 미들웨어 설계는 기존의 복잡한 업무를 모바일 환경으로 쉽게 확장 시켜 준다. 즉, 비즈니스 로직과 사용자 화면 UI를 분리하여 설계함으로서 기존 레거시 개발 시 활용했던 SQL구현 부분과 같은 기존 로직을 미들웨어가 그대로 활용함으로서 개발시간을 줄일 수 있게 되어 단시간 내에 모바일 환경으로 전환할 수 있게 된다. 따라서 기존의 기존 레가시 시스템이 3-Tier를 지원하면 모바일 Device에 맞추어 GUI를 설계하고 모바일 미들웨어는 기존 레가시 시스템의 로직을 활용하게 된다. 또한 3-Tier를 지원하지 않는 경우에는 레가시 프로그램의 소스코드를 분석하여 활용할 수 있는 SQL를 추출하여 비즈니스 로직을 구성한 후 모바일 환경에 맞게 전환 하면 되는데, 이 과정에서 모바일 미들웨어의 위저드기능을 이용하여 각종DB스키마를 활용한 비즈니스코드를 생성할 수 있다. 또한 무선 LAN 이나 이동통신망을 기존의 업무 환경에 추가하고, 자유롭게 클라이언트용 단말기를 선택할 수 있어, 단말기를 더욱 강력한 비즈니스 도구로 만들 수 있으며, 표준 인터넷 프로토콜을 사용하여 개발된 응용 소프트웨어는 인터넷을 통하여 원격 서버의 데이터에 접속할 수 있다. 따라서 개발자들은 회사의 업무를 진정한 의미의 모바일 시스템으로 확장 시킬 수 있다.

2.1.3 다양한 부가 기능의 지원

다양한 유틸리티의 지원을 강화하여 기본적인 모바일 컴퓨팅 환경의 구축을 쉽게 할 수 있도록 한다. 이를 위하여 완벽한 트랜잭션 처리를 위한 구조화된 자료를 모바일 컴퓨팅 환경에서 구현하며, 모바일 단말기에 파일단위로 자료를 복사 배포 할 수 있도록 한다. 또한 네트워크에 연결된 프린터를 활용하여 프린팅 기능도 기본적으로 지원하여 완벽한 업무 환경을 쉽게 수행하도록 도와주며, 공개·비공개된 채팅 서버를 통해 다양한 정보를 주고 받을 수 있는 서버와 클라이언트 기능도 제공한다.

2.1.4 모바일 시스템의 성능 향상

많은 전산자원을 필요로 하는 비즈니스 로직과 데이터 처리는 서버 상에서 처리하며, 상대적으로 컴퓨팅 파워가 적은 단말기에서는 비교적 간단한 데이터의 표현(Presentation) 부분만을 운영되도록 구성되어, 복잡한 모바일 어플리케이션에서도 성능의 저하 없이 시스템을 구축, 운영할 수 있도록 한다. 서버와 단말기간의 데이터의 동기화에 기반 한 시스템의 구성은 단말기와 통신 네트워크에 심각한 부하를 줄 수 있는데 3-Tier 구조를 통해 이 문제를 해결 하며, 또한 단말기 자료 저장 용량의 한계로 업무에 필요한 충분한 자료를 활용하지 못하는 문제와 상대적으로 보안성이 떨어지는 단말기에 중요한 데이터를 보관하지 않음으로써 보안 문제를 동시에 해결한다.

2.1.5 표준 개발도구의 지원에 의한 생산성 향상

표준 개발 도구를 지원함으로써, 이미 구축된 어플리케이션을 용이하게 확장 시키는 것은 물론, 새로운 어플리케이션의 개발에 있어서도 프로그래머의 생산성을 향상시켜, 전체 모바일 어플리케이션의 구축에 필요한 비용과 시간을 감소시켜 준다. 개발도구는 단말기에 설치되어 복잡한 무선 통신 프로토콜을 지원하는 클라이언트와 기존의 데이터베이스와 연결을 해주고 비즈니스 로직을 지원하는 서버용 개발도구를 포함하는데, 클라이언트용 개발도구는 단말기에서 작성된 응용 소프트웨어가 서버에 있는 다양한 함수들을 손쉽게 사용하도록 지원 하며, 서버의 비즈니스 로직은 일반적으로 많이 활용되고 있는 개발 툴(Visual Basic, Visual C++)과 Delphi, Power Builder등의 다양한 개발 환경에서 작성이 가능하다. 모바일 단말기에서의 어플리케이션도 WinCE 상의 표준 개발도구인 Visual Basic 과 Visual C++을 활용하는 Microsoft WinCE Toolkit은 물론 다양한 단말기 제조사 개발툴과 SDK를 이용하여 손쉽게 개발할 수 있으며, 이동성이 보장되는 실시간 데이터베이스 접근 기술을 제

공하므로, 응용 소프트웨어 개발자들은 업무 기능의 처리절차에 더욱 집중할 수가 있다.

2.1.6 견고한 암호화 보안 솔루션의 지원

견고한 암호화 보안 솔루션을 사용하여 사용자에게 정보 보안에 대한 문제를 고려하지 않도록 지원한다. 즉, 사용자의 정보 보안을 위해 채택한 암호화 알고리즘은, 현존하는 암호화 알고리즘 중 가장 강력하면서도 적은 컴퓨팅 파워를 요구하는 ECC(Elliptical Curve Cryptography) 암호화 알고리즘으로써, 작은 키의 크기는 데이터를 암호화, 복호화 하는데 있어서 상대적으로 정보처리능력이 높지 않은 스마트카드 등의 기기에서 이용하기에 적합한 암호화 방식으로 알려져 있다[9]. 따라서 최소의 네트워 트래픽을 요구하면서 안정적이고 빠른 어플리케이션 레벨에 보안을 지원하기 때문에, 무선 데이터 통신에 최적화된 보안 솔루션이다.

2.2 엔진 설계

미들웨어의 엔진은 기존의 다양한 콘텐츠와 불특정 다수를 위한 서비스의 e-business를 m-business로 전환할 수 있는 전략을 수립 할 수 있도록, 각종DB 스키마와 SAP의 BAPI 함수 등으로 부터 얻은 정보를 이용한 서버용 비즈니스 로직으로 설계하며, 클라이언트 인터페이스 함수와 API를 자동으로 생성하는 프로그램 자동화 도구에 대한 설계를 포함한다. 모바일 비즈니스분야에 대한 응용프로그램 적용을 위해 Data와 Business Logic, Presentation을 분리하며, 3단계로 나누어 모바일 시스템을 구축 할 수 있도록 한다. 그림 2에서 Data는 모바일 비즈니스를 제공하기 위한 기업의 다양한 상용 DB와 그룹웨어와 같은 상용 프로그램, SAP등의 독립된 서버군 등이 포함되며, Business Logic의 모바일 미들웨어에서는 이러한 Data 서버들과 연동되는 모바일 서버에 접속하여 실시간 모바일 비즈니스를 제공받는 Presentation용 모바일 단말기와 모바일 서버의 연동을 담당한다.

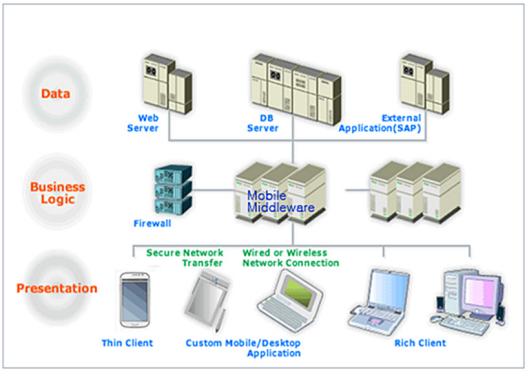

그림 2. 3-Tier 구조
Fig. 2. 3-Tier Architecture

- 1 단계: Server(Data ↔ Business Logic)

Delphi, Visual Tool등을 이용하여 COM기반의 서버 프로그램을 개발하며 개발된 서버는 COM을 이용하는 타 시스템에서도 재활용 가능하다.

- 2 단계: Interface (Business Logic ↔Application)

개발자는 서버의 요구사양에 맞게 오직 함수 또는 모듈을 호출 하는 것으로 시스템 인터페이스를 구현할 수 있다.

- 3 단계 Application (Client Device ↔ GUI)

최종사용자의 모바일 디바이스에 맞게 사용자 인터페이스 부분을 구현한다.

그림 3은 모바일 미들웨어 서버와 모바일 미들웨어 클라이언트로 구성되어있는 모바일 미들웨어의 블록 구성을 나타내고 있다. 미들웨어 서버는 Data서버의 다양한 자원을 모바일 자원으로 자동변환 및 생성 시키는 역할을 수행하는데, 각종 어댑터와 연동 수단을 제공한다. 이미지 어댑터는 모바일 장비를 이용한 협동작업과 실시간 커뮤니케이션을 위한 이미지를 실시간으로 전송하여 동일한 세션에 상주하는 사용자와 그림정보를 공유하는 역할을 하며, 텍스트 어댑터는 문자정보를 교환하는 역할을 한다. 파일전송 어댑터는

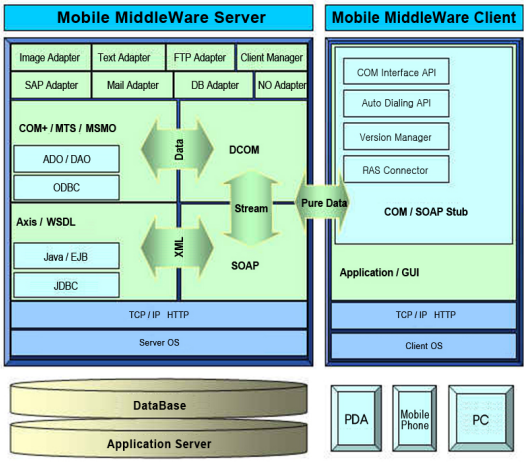

그림 3. 모바일 미들웨어 서버 및 클라이언트 구조
Fig. 3. Structure of Mobile Middleware Server and Client

바이너리 형태의 각종 오브젝트를 전송하는 기능으로 통신비용을 줄이기 위한 압축과 해제 기능역할을 하고, SAP어댑터에서는 SAP시스템의 비즈니스로직을 호출하여 필요한 정보를 통신장비에 전달하며, 메일 어댑터는 각종 메일 서버와 다른 서버와 통신을 위한 역할을 수행한다. SOAP(Simple Object Access Protocol)연동 모듈은 자바를 이용한 비즈니스로직으로 웹어플리케이션 서버상의 함수를 호출하여 단말기에 처리된 결과를 전송하며, MTS(Microsoft Transaction Server) 연동 모듈에서는 DCOM(Distributed Component Object Model)과 호환 가능한 트랜잭션 보장함수를 호출하여 자료의 정합성을 보호하는 역할을 한다. 또한 미들웨어 클라이언트에서는 모바일 통신장비인 클라이언트의 환경에 적합한 다양한 비즈니스를 제공하는 역할을 수행한다. 버전관리모듈에서는 단말기 응용프로그램의 버전변경이 발생 시 모바일 미들웨어 서버로부터 압축된 신규 버전을 전송받아 압축을 해제한 다음 구 버전의 모바일 응용프로그램을 갱신하는 역할을 하고, RAS(remote access services)연동 모듈에서는 무선랜 환경이 아닌 CDMA와 같은 패킷통신 장비를 이용한 네트워크 연결장치에 대한 통신연결을 담당한다. 통신규약 관리모듈은 서로 다른 무선 통신 프로토콜과 다양한 모바일 장비의 운용체계에서 동일한 인터페이스를 유지하기 위한 정형화된 통신규약을 제공하며 서버와의 통신 및 자료전달을 위한 역할을 하고, COM(remote access services)연동 모듈은 모바일 미들웨어 서버와 DCOM과의 인터페이스 역할을 수행한다.

2.3 모바일 솔루션 확장

모바일 비즈니스 확장 구축을 위한 공정으로는 기 구축된 업무규칙을 이해하고 이를 모바일 환경에 맞게 수정하는 작업과 발생된 정보를 서버와 클라이언트에 안정적으로 전달하는 기술, 그리고 모바일 통신장비에 적절히 표현하는 기술이 필요하다. 이러한 단계 중에서 클라이언트와 서버에 안정적으로 정보를 전달하는 기술이 무선 모바일 비즈니스의 핵심으로 전체 구축 공정의 대부분을 소비하는 문제점이 있었다. 제안 솔루션을 통해 이러한 문제를 해결할 수 있도록 하며 또한 금융, 제조, 의료, 재난관리 등 산업 분야별 특성에 따른 시스템 구성을 용이하게 도와주며 소규모 모바일 비즈니스에서 대규모 모바일 비즈니스까지 손쉬운 확장이 가능하도록 한다.

- 뛰어난 확장성 및 안정성을 위한 각종 기능을 제공

동급 서버의 병렬 배치로 선형적인 성능 증가가 가능함으로써, 소규모 모바일 비즈니스를 위한 소규모 서버에서 전사적인 대규모 모바일 비즈니스로의 확대 전환이 매우 용이 하도록 설계하며, 모바일 비즈니스 서버의 자원 배분도 One 클릭으로 해주고, Load 밸런싱과 멀티쓰레딩(multi threading)을 지원하여 안정적인 서버 운용을 가능하게 한다.

- Backend 단의 기 투자된 자원과 호환성을 유지

기 투자되어 사용 중인 자원의 변경을 최소화하는 각종 Adapter를 지원하며, 유선상의 어플리케이션과 비즈니스 프로세스를 승계하여 전산자원과 인력이 빠른 시간 내에 모바일 비즈니스로의 전환이 가능하게 처리한다.

- 직관적 관리 및 운용 환경 제공

모바일 비즈니스 관리 및 운용환경을 보다 쉽게 관리하고 운용할 수 있도록 직관적인 GUI를 채용, 운용자 편의성과 효율성을 고려하여 설계하며, 서버에서 원격지의 모바일장비 사용자의 접근 현황 및 사용현황, 모바일 장비 리소스 상태 등을 실시간으로 감시 및 제어할 수 있도록 한다.

Ⅲ. 기능 구현

3.1 Real-time transaction

데이터의 충돌 발생 시 데이터 무결성을 보장하는 트랜잭션과 멀티쓰레드로 2 phase-commit을 지원하여 정보전달을 완벽하게 지원한다.

그림 4에서와 같이 오프라인 트랜잭션은 동기화 시점에서 데이터 불일치 가능성 존재하여 로컬 DB의 데이터가 최신인지 보장 못하지만, 온라인 트랜잭션은 기존 3-Tier와 같은 분산 처리 방식을 이용하여 데이터의 무결성을 보장하며 가장 최신의 데이터에 접근할 수 있다.

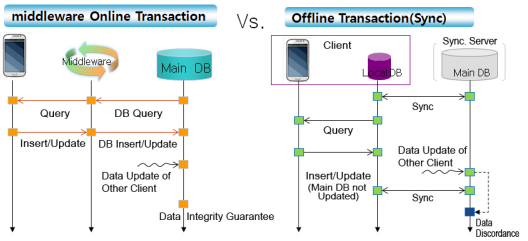

그림 4. 온라인 및 오프라인 트랜잭션 비교
Fig. 4. Comparison of On-line Transaction with that of Off-line

3.2 System Adaptation

개발 생산성 향상 및 용이한 유지보수 실현이 가능한 COM/SOAP based API를 구현함으로써 개발 코드양의 현저한 감소와 동적이며 유연한 프로그램개발이 가능하다. 또한 MS사의 표준 기술 수용이 용이하며 서버측의 DCOM에 대응하는 클라이언트 COM API를 제공하고 XML, HTTP 콘텐츠를 수용한다. 또한 ODBC와 JDBC를 지원하는 모든 상용 RDB를 지원하고, Web기반 미들웨어, TCP/IP 기반 미들웨어, SAP/R3 등의 상용 어플리케이션 등과의 원활한 호환이 가능하도록 지원한다. 따라서 비즈니스 로직의 재사용성이 증가하며 다양한 시스템에 대한 모바일화 요구를 충족하게 되어 복잡한 업무 환경에 용이하게 대응할 수 있다.

3.3 Code gen. Wizard

모바일 비즈니스를 손쉽게 구축할 수 있도록 제공되는 자동 개발도구는 각종 DB 스키마, SAP/R3의 BAPI 함수 등으로부터 얻은 정보를 이용하여 서버용 비즈니스 로직을 구현할 수 있는 코드를 컴파일러 별로 생성하고, 이를 이용한 클라이언트 인터페이스 함수와 API를 자동으로 생성하여 개발 생산성을 향상시켜 준다. 그림 5는 기 구축된 클라이언트/서버에서 사용된 비즈니스로직을 모바일 비즈니스에서 활용하기 위한 예로서 서버측의 DCOM에 대응하기 위해 생성된 클라이언트의 COM API를 표시하고 있으며, 그림 6은 데이터베이스로부터 SQL문을 추출하는 도구 화면이다.

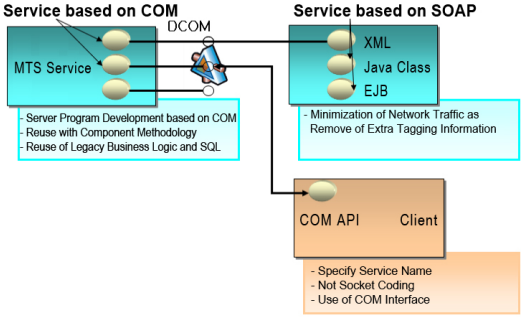

그림 5. COM/SOAP 기반 API
Fig. 5. API based on COM/SOAP

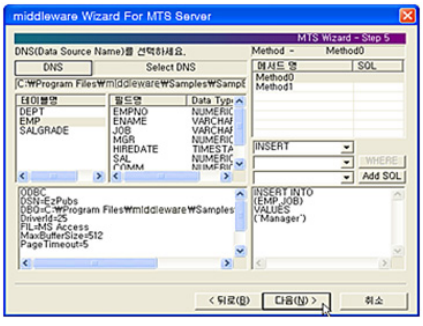

그림 6. 코드 생성 도구
Fig. 6. Code gen. Wizard

3.4 Security&Access control

종단간의 타원곡선(ECC) 암호화를 채택하여 사용자 정보를 보호하며, 외부 침투자를 사전에 통제하거나 분실된 모바일 장비를 통한 내부 접근을 사전에 차단한다. 그림 7은 모바일 미들웨어 서버와 클라이언트와의 정보흐름을 나타내고 있는데, 하드웨어별 접근통제와 불법사용자의 서비스 제한, 클라이언트자원 관리, 사용자 모니터링을 통한 접근제어와 사용자 아이디·패스워드 인증, 클라이언트 라이센스 인증, 시스템 자원 접근 보안 관리, 서버와 클라이언트간 트랜잭션 단위 세션 키 관리를 통한 어플리케이션 레벨의 보안 모델을 제공한다.

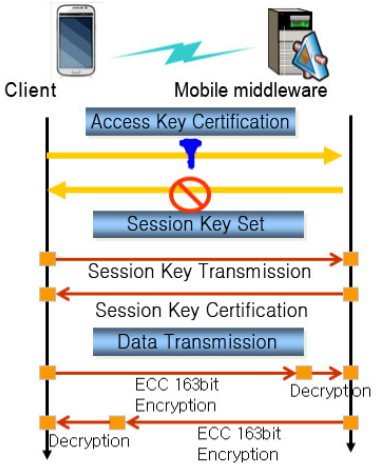

그림 7. 보안 및 접근 제어
Fig. 7. Security&Access control

3.5 Free Comm. Hang up

수시로 발생할 수 있는 통신 단절 시, 모바일 장비는 행업(Hang up) 상태가 되어 소프트 리셋(Reset)을 해야만 하는 경우가 자주 발생한다. 사용자 별 Time-Out기능 설정을 통해 설정된 유효시간 경과 후, 메시지를 알려주어 적절한 조치를 취할 수 있도록 리소스를 해제할 수 있도록 한다. 즉, 서버 접근 중 혹은 서버로부터 데이터 전송 중 네트워크가 단절되거나

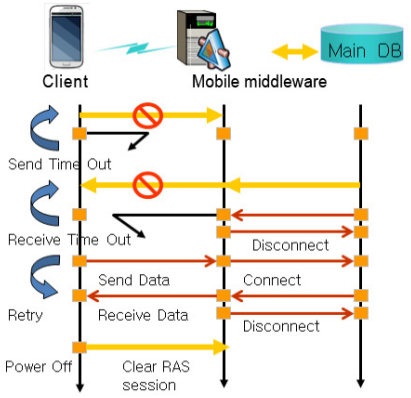

그림 8. 통신 단절
Fig. 8. Free Comm. Hang up

통신 세션 연결 중 네트워크 카드에 장애가 발생하는 경우, 기존의 클라이언트·서버 반응은, 단절되지 않은 상태에서 서버 커넥션 수가 증가하며 데이터베이스 Dead lock 가능성이 존재한다. 또한 클라이언트 행업 상태로 어플리케이션 Lock상태가 발생하여 클라이언트를 리셋하게 되면 시스템이 초기화되어 작업 중인 모든 정보가 소실될 가능성이 존재한다. 그림 8과 같이 모바일 미들웨어 서버는 비정상적인 세션은 종료시키며, 데이터베이스 접속 상태(Connection Pool)를 제거하고 클라이언트에 Time-Out 메시지를 발생한다. 또한 파워 오프(Power Off)시 네트워크 세션을 정리한다.

3.6 관리도구

모바일 비즈니스를 보다 쉽게 관리하고 편리한 운용환경을 이용할 수 있도록 그림 9와 같은 관리도구를 제공한다. 시스템 운용 시 손쉽게 시스템 현황의 모니터링 및 운용을 할 수 있는 GUI 기반 관리 툴 과 개발 및 관리를 위한 각종 로그 자료, 이벤트 정보를 로그로 기록하여 분석 가능한 트레이스(trace)기능을 제공한다. 또한 서버에 걸리는 부하 정도를 그래프로 분석과 모니터링이 가능한 트래픽 관리기능과 클라이언트접근 통제 및 자원을 관리할 수 있는 클라이언트 관리 기능을 제공한다.

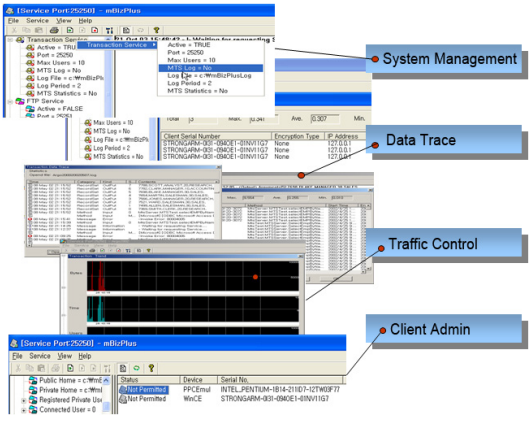

그림 9. 관리도구
Fig. 9. Admin Tool

Ⅳ. 사용자 응용

4.1. 서비스 만들기

비즈니스 로직을 구현하기 위한 서버는 MTS뿐만 아니라 SOAP 프로토콜을 사용하여 Web Service를 호출할 수 있다. WebService를 사용하는 것은 MTS보다 조금은 까다로운 구현 방법이지만 기존에 사용하던 WAS 등의 인프라 스트럭쳐를 재활용할 수 있다는 등의 장점이 있다. 또한 SOAP, HTTP 등 W3C(World Wide Web Consortium)의 표준 스펙을 따르기 때문에 다양한 언어와 다양한 플랫폼에서의 서버 작성이 가능하다.

4.1.1 MTS Service 만들기

3-Tier구조를 가지고 있고, 비즈니스 로직을 구현하기 위해 Microsoft의 Transaction Server를 구현하는 방법으로 Visual C++를 주로 이용하지만 익숙한 프로그래머가 아닌 경우 다른 프로그래밍언어에 비해서 구현이 다소 까다로울 수 있다. 서버용 비즈니스 로직을 구현하기 위해서 Visual Basic, Delphi 등 프로그래머가 익숙한 언어를 선택할 수 있는 기능을 이용한다. 그림 10은 Visual Studio에서 Server Type을 Dynamic Link Library(DLL)로 선택하고, ATL Object Wizard의 Visual C++을 이용하여 MTS서버

그림 10. MTS Server 생성
Fig. 10. Generation MTS Server

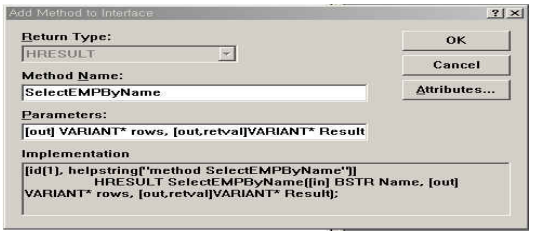

그림 11. 인터페이스에 메쏘드 추가
Fig. 11. Add Method to Interface

를 생성하는 화면이다.

기본적인 Class 작성은 ATL Wizard가 모두 해주었고, 다음 단계에서는 DB를 접근할 수 있는 DLL을 import하며 DB 조작을 위한 코딩을 직접 해야 한다. 그림 11은 클래스의 Method속성을 입력하는 화면인데, Method Name은 Client에서 접근을 하기 위한 이름이며, ODBC를 이용하여 DB를 접근할 수 있는 DLL을 import 한다. ADO의 객체를 사용하는 방법은 다양한데, 대표적으로 COM Interface 방법 중에서 Import를 이용하여 타입 라이브러리 안에 저장된 타입 정보들로부터 C++ 래퍼코드를 생성하는 방법이 있다. DB Open 과 RecordSet 복사, Connection 종료 기능을 코딩하면 기본적인 MTS Server가 만들어지는데 컴파일작업을 하면 해당 dll 파일이 생성된다. 작성된 서버를 MTS로 작동시키기 위해 Transaction Manager에 등록할 수도 있고, 그림 12와 같이 구성요소 서비스에 등록하여 MTS로 동작하도록 할 수도 있다.

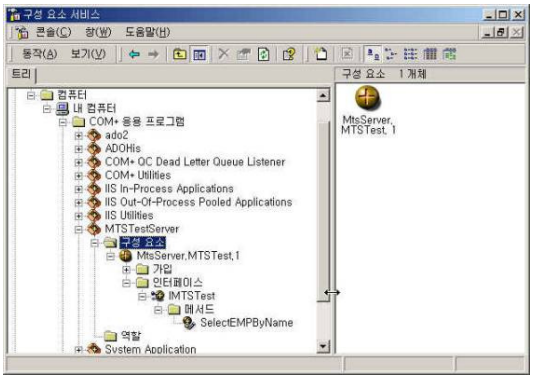

그림 12. 서버 등록
Fig. 12. Registration Server

4.1.2 Java로 Web Service 만들기

작성할 수 있다. 웹서비스 구현을 위해 Axis를 이용한 SOAP 서버 작성은 여러 가지 방법이 있으며 기본적으로 할 수 있는 방법은 다음과 같다.

a. WSDL 작성 → 자바 클래스 골격 및 WSDD작성 → 각 메쏘드 구현 → 서비스 등록
b. 자바 인터페이스 및 클래스 작성 → WSDL 작성 → WSDD 작성 → 서비스 등록
c. 자바 클래스 작성 → jws로 서비스 등록 → WSDL 생성

a방법의 경우 WSDL(Web Services Description Language)을 먼저 작성하고 org.apache.axis.wsdl.WSDL2Java 클래스를 사용하여 자바 인터페이스, 인터페이스 구현 클래스, 스텁 클래스등과 WSDD를 자동으로 생성시킨다. 그 후에 인터페이스 구현 클래스에 만들어진 메쏘드 골격에 코딩을 하여 서버를 작성한다. 코딩 후 컴파일을 완료 한 후에 WSDD 파일을 이용하여 서비스 등록을 한다. WSDL을 잘 작성하면 메쏘드 구현부분만 코딩하면 된다. b의 방법은 자바 인터페이스를 먼저 컴파일 한 뒤 그 클래스를가지고 org.apache.axis.wsdl.Ja va2WSDL클래스를 사용하여 WSDL을 만드는 방법이다. WSDD 파일은 만들어 주지 않기 때문에 서비스 등록을 위해서는 WSDD 파일을 직접 작성해야 한다. 또한 MS SOAP Toolkit과의 호환을 위해서는 자동으로 만들어진 WSDL을 수정해 주어야 한다. 그리고 자동으로 만들어주는 WSDL 파일에서는 변수명이 무조건 만들어지기 때문에 변수명 이름을 다르게 주기 위해서는 또 다른 수정 작업이 필요하며, 인터페이스를 구현하는 자바클래스를 작성 한다. c의 방법은 java 파일을 확장명만 jws로 바꾸어서 서비스폴더에 올리고 웹브라우저 상에서 호출을 하면 자동으로 서비스가 올라오게 된다. 또한 WSDL도 브라우저에서 호출하는 것으로 자동으로 생성할 수 있

```
<!-- Message -->
 <wsdl:message name="TestResponse">
   <wsdl:part name="rowset" type="xsd:anyType"/>
   <wsdl:part name="result" type="xsd:long"/>
 </wsdl:message>
 <wsdl:message name="GetNameByEmpnoRequest">
   <wsdl:part name="empno" type="xsd:string"/>
 </wsdl:message>

 <wsdl:message name=
    "UpdateNameByEmpnoRequest">
   <wsdl:part name="empno" type="xsd:string"/>
   <wsdl:part name="name" type="xsd:string"/>
 </wsdl:message>
```

그림 13. wsdl 생성 예
Fig. 13. Example of wsdl Generation

지만, 역시 MS SDK과의 호환을 위해서는 WSDL을 수정해야 한다. 또한 WSDL이 파일로 만들어지지 않고 웹브라우저로 접근해야 하기 때문에 WSDL을 수정하려면 약간의 작업을 거쳐야 한다. 그림 13은 wsdl의 생성 예이다

4.1.3 Wizard 사용하기

미들웨어 위저드를 사용하면 MTS 서버 및 웹서비스 작성을 보다 손쉽게 할 수 있다. 그림 14는 자동화 툴을 이용하여 해당 파일을 생성한 예제이다. 서비스이름.java 파일은 인터페이스파일이고 SoapBindingimpl.java는 인터페이스의 구현 파일이다. 이 두개의 파일을 Axis 설치 디렉토리에 복사한 후 컴파일을 하고 관련 wsdd파일을 이용해서 등록을 한다. 이후에 이 서비스를 호출할 때는 생성된 WSDL파일을 이용하면 된다.

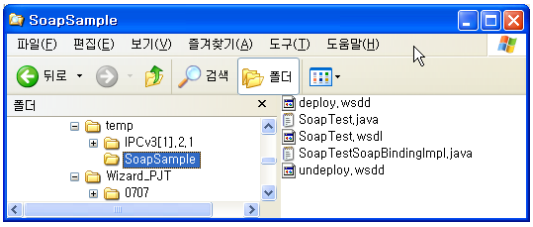

그림 14. 서비스 파일 예
Fig. 14. Example of Service File

4.2. 클라이언트 만들기

클라이언트 프로그램을 구현하기 위해 Visual Basic과 Visual C++을 이용하는 두 가지 방법을 사용한다. 먼저 손쉽게 작성할 수 있는 Visual Basic인 경우 개발은 쉽지만 C++와 비교했을 때 응용프로그램의 실행속도가 느리고 실행파일의 크기도 차이가 많다. 따라서 embedded Visual C++로 구현하는 방법은 Visual Basic보다는 구현 과정이 다소 복잡하지만 파일의 크기도 작고, 속도가 빠르기 때문에, 제한된 자원을 염두에 두어야 하는 스마트디바이스 이용환경에서 많은 이점이 있다. 다음은 Visual C++로 Windows CE 클라이언트 프로그램을 만드는 간략한 코딩절차이다.

① MicroSoft Embbeded Visual Tools에서 Visual C++을 이용하여 프로젝트 파일을 작성한다.
② 미들웨어가 설치되어있는 폴더에서 해당 파일을 프로젝트로 파일 복사를 한다.
③ COM 초기화를 위한 코딩 작업을 한다. COM과 관련된 함수를 참조할 수 있도록 하며, 미들웨어 서버와 통신 하기위한 인터페이스가 정의된 Header File과 서버와 클라이언트 간에 데이터 암호화 모드에 관한 값을 정의한 Header File을 include하여, 서버와 통신하기 위한 함수를 참조할 수 있도록 한다.
④ AccessKey 초기화 작업을 수행한다. AccessKey는 Client의 ID 값을 가지게 되는데, 이는 서버에서 Client를 식별하기 위해 반드시 필요한 값이다. 따라서 RequestMTS 함수를 호출하기 이전에, 반드시 초기화 함수를 호출해야한다.
⑤ Arguments로 넘길 변수 처리와 서비스를 호출 하기위한코딩 작업을 한다.

```
hr=
pRequestService->RequestMTS(AccessKey,EncryptionMode,serName,&Args,&Error,&Rowset,&Result);
```

함수 구성은 다음과 같다.

```
HRESULT RequestService(BSTR AccessKey, int EncryptionMode, BSTR serName, VARIANT * Args, VARIANT* Error, VARIANT* RowSet, VARIANT* Result)
```

- AccessKey : [in] Initialize된 AccessKey
- EncryptionMode : [in] Data Encryption Mode 값을 Int Type로 넘긴다. 데이터 암호화 모드 값은 다음의 4가지 값 중 하나의 값을 갖는다.

i) Encryption_Both: 서버에서클라이언트로, 클라이언트에서 서버로 보내는 데이터를 모두 암호화
ii) Encryption_Client: 클라이언트에서 서버로 보내는 데이터만 암호화
iii) Encryption_Server: 서버에서 클라이언트로 보내는 데이터만 암호화
iv) Encryption_None: 모든데이터를 암호화 시키지 않음

- serName : [in] MTS Service Name
- Args : [in] 서비스에서 사용할 Arguments
- Error : [out] runtime Error발생 정보
- RowSet : [out] MTS에서 얻은 Record Set
- Result : [out] MTS에서 얻은 Record Set의 Rows 정보

⑥ RequestService가 실패한 경우 Error 정보를 해석한다.
⑦ RecordSet 접근을 위한 첨자의 증가작업을 수행한다.
⑧ 프로그램이 종료되기 이전, 즉, 더 이상 RequestService을 하지 않는 시점에서 AccessKey Uninitialize함수를 호출한다.

4.3 응용사례 : 모바일 원격서버 관제시스템

모바일 서버와 모바일 단말기와의 실시간 데이터 통신을 위한 단계적 동작 흐름을 사례를 통해 설명한다. 그림 15는 제안된 모바일 미들웨어 시스템을 이용한 응용프로그램 개발 지원 방법의 연관관계를 나타내고 있다. 미들웨어 서버의 개발도구를 이용하여 서버코드를 생성하기 위해 우선, 데이터베이스 혹은 다른 상용 프로그램으로부터 절차 처리를 위한 스키마를 추출하여 모바일 통신장비와 정보전달을 위한 정보를 획득한 후 응용 시스템에 적합한 컴파일러에 해당하는 서버코드를 생성하게 된다. 기본적으로 생성된 서버 코드의 로직부분을 수정 및 보완하고, 완성된 서버코드를 선택된 컴파일러를 이용하여 실행 가능한 오브젝트 코드로 완성한다. 클라이언트용 소스를 생성하는 방법은, 클라이언트 응용프로그램 중에서 호출된 함수를 선택하고, 서버의 논리적 위치정보를 알려주면, 응용프로그램에서 사용된 인터페이스 속성을 선택하여 정의된 입출력 정보 속성에 맞는 기본 클라이언트 코드를 생성하게 된다. 정의된 속성 정보를 이용하여 서버와의 인터페이스를 위한 모바일 미들웨어 관련 API를 생성하고, 생성된 클라이언트 코드를 선택된 모바일 단말기에 맞는 컴파일 과정을 거쳐 실행 오브젝트 생성이 완성된다.

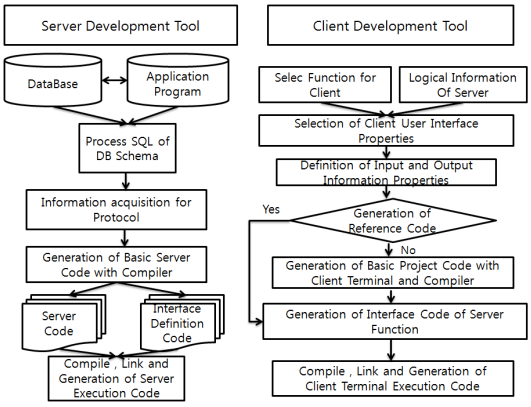

그림 15. 모바일 미들웨어를 이용한 응용프로그램 개발
Fig. 15. Application Program Development using Mobile Middleware

그림 16은 장애발생시 이동 중에도 원격에서 해당 서버·네트워장비·기계설비의 장애상황을 확인, 조치, 결과보고를 즉시 입력할 수 있도록 기존의 시스템 모니터링툴과 연동한 정보시스템의 모바일 단말기를 구현한 시스템이다. 우선 모바일 단말기에서 물리적으로 멀리 떨어진 위치에 존재하는 서버에 접속하여 자료를 요청하면, 모바일 단말기가 무선랜에 노출된 상황인지 단말기 모듈의 데이터 네트워크를 이용하는 상황인지를 판단한 후, 네트워크 연결을 시도하게 된다. 자원을 요청 받을 서버와의 네트워크 연결이 가능한 상태이면 요청정보를 모바일 미들웨어 통신규약에 맞게 자료를 정형화 시키고, 서버 세션을 요청하게 된다. 서버와의 세션 연결 요청이 수락되면 다중 연결자 처리를 위한 멀티스레드로 다중 세션을 생성하게 되며, 각각의 세션은 독립적으로 요청자료를 수신하고, 요청자료의 특성을 분류하여 데이터베이스를 연결할 것인지 또 다른 서비스 연결자를 호출할 것인지를 결정하게 된다. 데이터베이스에 연결하는 경우, 비즈니스로직에 맞게 처리하여 처리된 자료를 정형화 하는데, 모바일 통신규약에 맞는 정형화된 자료는 암호화 모드,

그림16. 원격 서버 제어
Fig. 16. Remote Server Control

수행될 서버 함수, 요청자료, 장애정보, 결과자료, 결과정보로 구성된다. 서버세션이 연결되어 요청한 자료의 불필요한 정보를 제거하고 정보를 압축한 후, 미들웨어 통신규약에 맞는 정형화된 자료를 전송한다. 자료전송이 완료된 후 서버의 반응 이벤트가 있는지 대기하게 되는데, 서버의 반응 이벤트가 있고 서버 반응 자료를 수신하면, 결과 값이 정상으로 수신되었는지를 검토한 후 제한된 시간을 경과하면 타임아웃메시지를 사용자에게 전달하여 다음 행동을 할 수 있도록 하며, 정상적으로 수신된 자료를 사용자 인터페이스로 구현할 수 있도록 결과값을 넘긴다. 요청사항이 정상적으로 처리가 되었으면 연결 단말기를 정리하고 세션의 자원을 정리하여 서버세션을 종료하게 된다. 모바일 원격서버 관제시스템은 모니터링툴과 모바일 단말기의 유기적인 결합을 통하여 장소에 관계없이 각종 장애를 확인, 조치, 결과보고가 가능한 모바일시스템을 제공하고, 장애상황에 효과적으로 대응하는 지원체계를 실시간으로 지원 할 수 있는 시스템이다. 인증된 단말기에서 원격서버에 직접 접근하지 않고 미들웨어 시스템 서버에 제어모듈을 개발 등록하여 간접 제어함을 목표로 하는 경우이다. 장애발생시 SMS(Short Message Service)를 이용하여 문자를 수신한 후 모바일 디바이스의 통신망을 통하여 네트워크에 접속한다. 모바일 미들웨어 서버를 경유한 장애에 대한 발생 내역과 장애 서버의 시스템 상태를 확인하며, 모바일 서버를 경유한 조치내역을 보고한 후 클라이언트의 네트워크 접속을 해제한다.

V. 결 론

국내의 무선 통신 환경은 네트워크 음영지역이 거의 없는 구축 환경을 자랑하며, 본 연구에서 제안한 모바일미들웨어는 이런 국내 통신 환경과 실시간의 트랜잭션을 보장해야 하는 비즈니스를 위한 무선 프로세스 구축에 이용될 수 있다. 구현된 솔루션은 오프라인 비즈니스뿐만 아니라, 온라인 비즈니스에서도 완벽한 트랜잭션 보장을 위한 시스템 설계에 역점을 두었다. 데이터베이스 독립, 비즈니스로직 레이어와 프리젠테이션 레이어를 분리하여 견고한 시스템 구축 및 유선 어플리케이션과의 통합을 위한 아키텍처를 채택하였다. 따라서 개발자는 기존 유선 상 어플리케이션에서 충분히 고민하고 경험한 비즈니스 프로세스를 무선으로 확장만 시키면 된다. 또한 모바일 미들웨어 플랫폼 뿐만 아니라, 비즈니스로직, 프리젠테이션 부분을 위한 코드 생성 위자드 등의 무선 개발 환경 및 클라이언트 버전관리, 타임아웃 메시지 핸들러 등 무선 환경에서 노출될 수 있는 문제들을 쉽게 해결하기 위한 다양한 API를 제공하였다. 개발자들은 API사용법만 숙지하면 유선상의 어플리케이션을 위한 데이터베이스 혹은 비즈니스 로직은 변경 없이 기존 전산자원을 그대로 활용할 수 있다는 장점이 있다. 특히, 자바·SAP R/3·그룹웨어·또 다른 엔터프라이즈용 미들웨어와의 통합을 위한 다양한 어댑터를 제공함으로써 유무선 통합 어플리케이션 구축이 수월하다. 또한 개발생산성 향상을 위한 자동화 도구를 이용하여, 비즈니스 로직 생성에서 클라이언트 생성까지의 작업을 직관적인 인터페이스를 통해서 안정적이고 일관된 개발이 가능하다. 향후 새롭게 등장하는 다양한 형태의 모바일 디바이스에서 활용이 가능 하도록 지속적인 업그레이드가 필요하며, 제안 시스템을 활용한 모바일 비즈니스 구축을 통해 국내 모바일 산업의 활성화를 기대한다.

References

[1] J. S. Park, M. H Park, and S. H. Jung, "A whitelist-based scheme for detecting and preventing unauthorized AP access using mobile device," *J. KICS*, vol. 38B, no. 8, pp. 632-640, 2013.

[2] Travis Broughton, Mobile Middleware for the Enterprise, Retrieved Oct 2013, from http://blogs.intel.com/application-security/2013/02/15/mobile-middleware-for-enterprise/

[3] Doosan Corporation, Middleware, Retrieved Oct. 2013, from http://www.doopedia.co.kr

[4] S. I. Yang, T. G. Lee, and S. H. Park, "Design of lightweight mobile middleware naive system," *J. KCA*, vol. 9, no. 9, pp. 41-50, 2009.

[5] P. Bellavista and A. Corradi, *The Handbook of Mobile Middleware*, Auerbach Publications, 2007.

[6] Intel White Paper, Mobile Middleware Buyer's Guide, Retrieved Oct. 2013, from https://cloudsecurity.intel.com/

[7] Digital daily, MEAP(2011), Retrieved Aug. 2013, from http://www.ddaily.co.kr/news/news_view.php?uid=82366

[8] S. M. Lee, K. I. Seo, and E. M. Choi,

"Translation and adaptation technique for mobile software using AOP," *J. Korea Institute of Information Scientists and Engineers Association*, vol. 37, no. 1(B), pp. 59-63, 2010.

[9] G. B. Lee, and B. S. Lee, "A study on RFID information protection with the modified ECC algorithm," *J. Korea Institute of Information Scientists and Engineers Association*, vol. 8, no. 3, pp. 77-85, 2010.

이 일 주 (Il-joo Lee)

1988년 2월 : 아주대학교 전자계산학과 공학사

1994년 8월 : 한양대학교 전자계산학과 공학석사

2006년 8월 : 아주대학교 컴퓨터공학과 공학박사

1989년 9월~1998년 1월 현대미디어시스템/현대정보기술 책임

1998년 3월~현재 동원대학교 스마트IT콘텐츠과 부교수

<관심분야> 정보검색, 모바일 프로그램