

## 모바일 응용서버의 구조에 관한 연구<sup>†</sup>

오동익\* · 이증섭\* · 이경호\*\* · 김중배\*\*

### 1. 서 론

오늘날 인터넷의 활용은 유선접속환경을 뛰어넘어 무선환경에서도 급속한 성장을 보이고 있다. 핸드폰과 같은 무선단말기를 이용한 새로운 데이터의 접근방법이 폭발적인 인기를 끌고 있으며, PDA나 소형 무선접속 단말기를 이용한 인터넷의 활용은 앞으로 상업적으로 무한한 구매를 도출하게 될 것임이 틀림없다. 그러나 m-commerce가 제대로 서비스되기 위해서는 무선통신, 인터넷, 콘텐츠 사업 등 지금까지 서로 독립적으로 움직여왔던 이질적인 분야의 다양한 기술, 애플리케이션과 서비스를 통합해야 하는 어려움과 복잡성이 있다. 최근에 들어서 이러한 다양한 기술의 접목이 필수적으로 요구되어지는 m-commerce의 서비스를 위하여 모바일 응용서버를 통한 프레임워크를 구축하고 이를 통해 효과적인 무선 인터넷 서비스를 제공하기 위한 연구가 진행 중이고, 해외에서는 일부 솔루션들을 제공하는 제품들이 속속 출시되고 있다[1,7-11].

그러나 아직은 관련분야의 기술이 정립되어 있지 못한 실정이며, 따라서 현재 진행중인 연구 및 개발된 제품에 대한 분석이 절실히 필요한 상황이

다. 이에, 본 연구에서는 새롭게 등장하는 m-commerce를 포함하는 무선 인터넷 응용을 위해 중심적 역할을 담당하는 모바일 인터넷 응용서버(또는 간단히 모바일 응용서버 - Mobile Application Server)의 기능 및 구성방법에 대해 살펴본 후, 이러한 서버들이 제공하는 유무선 통합 서비스제공의 문제점은 무엇인지 파악하고, 이들의 기능을 분석하여 기존의 다양한 무선 인터넷 서비스 및 무선 단말기 환경에서 효과적으로 활용될 수 있는 모바일 응용서버의 구성방안 및 향후 발전방향에 대해 살펴보고자 한다.

본 논문의 2장에서는 모바일 응용서버를 구성하기 위한 최신 기술에 대해 살펴본다. 3장에서는 효과적인 모바일 응용서버의 구성을 위한 방향을 미들웨어 솔루션에 입각해 살펴보고 모바일 응용서버에 접목하는 방법에 대해 논한다. 끝으로 4장에서 본 연구의 결론을 맺도록 한다.

### 2. 모바일 인터넷 응용서버 기술분석

#### 2.1 모바일 인터넷 응용서버의 소개

##### 2.1.1 개요

무선환경에서 구동되는 모바일 응용시스템의 개발을 위해 초창기 모바일 인터넷 응용은 특정한 단말기기, 네트워크, 콘텐츠에 종속된 솔루션을 제공하는 형태로 개발되어져 왔다. 하지만 모바일

<sup>†</sup> 본 연구과제는 2002년 순천향대학교 학술연구 조성비 일반연구 과제로 지원을 받아 수행하였음

\*순천향대학교 정보기술공학부

\*\*전자통신연구원

인터넷 응용이 대중화되면서 이렇게 장치, 네트워크, 콘텐츠에 종속적인 응용을 만드는 것에 한계가 있다는 것은 쉽게 파악될 수 있었다. 지난 수년간 몇몇의 회사들이 이러한 장치, 네트워크, 콘텐츠 종속의 문제를 해결하기 위해 모바일 인터넷 응용을 위한 소프트웨어적 프레임워크를 제공하게 되었고, 이를 구성하는 각종 소프트웨어 요소들을 모바일 응용서버라는 이름으로 부르고 있다. <그림 1>은 모바일 인터넷에서의 모바일 응용 서버의 역할을 나타내고 있다.

모바일 응용서버가 제공해야하는 기본적인 기능은 무선 단말기를 통한 인터넷 서비스의 활용이고, 이러한 서비스의 종류는 방대하다. 초기 모바일 응용서버에서는 단순히 기존의 정적인 웹 페이지를 모바일 기기에서 읽을 수 있는 한정된 기능을 제공하였지만, 최근의 모바일 응용 서버들은 단순한 웹 콘텐츠의 활용뿐만 아니라 Back-End 시스템의 자료활용 및 응용시스템과의 연동을 통해 비즈니스 로직을 담당하고, 또 모바일 응용에 특수하게 적용될 수 있는 기능을 구현하여 제공하고 있다.

2.1.2 모바일 응용서버의 정의

모바일 응용서버는 인터넷상의 서버장치에서 구동되는 프로그램으로서 아래와 같은 기능을 제공하는 소프트웨어 솔루션의 집합체로 정의될 수 있다 [1,2].

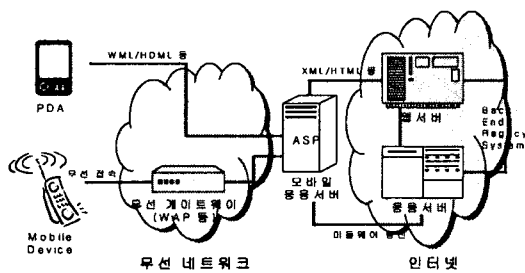


그림 1. 모바일 응용 서버의 역할

1) 클라이언트 이동 단말기기를 위한 프리젠테이션 서비스

모바일 응용서버는 Back-End 응용 시스템 (DBMS, 웹 페이지 등) 에서 추출된 데이터를 사용자의 무선 단말기에 디스플레이 할 수 있는 형태로 가공하는 역할을 담당한다. 이때, 단말기의 제한점(표시영역의 제한) 및 특성(인식할 수 있는 데이터의 표현방식)을 고려하여 콘텐츠 변환 (Transcoding)을 수행한다.

2) 응용 시스템 레벨 로직

이 로직은 비즈니스 로직을 처리하는 부분으로, Back-End의 DB 및 비즈니스 응용시스템(예로서, 기업 회계시스템, 재고관리, 자산 등)과의 연관을 통해, 또는 기존의 인터넷 응용서버와의 연동을 통해 로직을 수행한다.

3) 모바일 응용 특수성을 감안한 차별화된 서비스

모바일 응용서버는 모바일 인터넷 응용 시스템의 성격상 특수하게 발생하는 문제점에 대한 해결 방법을 제공해야 한다. 예로서, 많은 클라이언트의 서비스 접속을 감당하기 위한 멀티 스테딩, 잦은 Connection Drop을 극복하기 위한 연속성 제공 및 세션 복구의 기능 등이 제공되어야 한다. 또한, 모바일 응용 시스템에서 유용하게 활용될 수 있는 특징적 서비스들을 제공하여야 한다. 이러한 서비스의 대표적 예는 [표 1]과 같다

4) 모바일 응용 시스템 개발을 위한 툴 제공

쉽게 기존의 Back-End Business 로직을 연동하고, 응용시스템을 개발할 수 있는 Drag-And-Drop 방식 등의 응용 시스템 개발 툴이 제공될 수 있을 것이다. 이와 함께 새로운 응용을 작성할 수 있는 프로그래밍 레벨 API를 제공한다.

2.1.3 모바일 응용서버의 다양성

모든 모바일 응용서버가 동일한 기능을 제공하는 것은 아니다. 이 절에서 제공된 모바일 응용서

표 1. 모바일 응용 시스템에서 활용 가능한 서비스

서비스	설명
Notification (고지)	e-mail 과 Voice Mail의 Real-Time 배달, 정보 Push 기능
Offline/Online Data Synchronization	일정 데이터를 단말기에서 보유하고, 통신이 두절된 상태에서도 작업이 가능할 수 있도록 하는 기능
개인화 (Personalization) 기반 응용	사용자가 선호하는 선택 사항에 따라 자료의 표현방식 및 내용을 제공하는 기능
Location-Aware Context-Aware	위치정보에 입각한 정제된 관련 정보 제공을 유용하게 하는 기능
Multichannel 서비스	음성, 무선, 유선 등 여러 방식으로의 모바일 응용에 접근할 수 있도록 하는 기능
Session 유지 기능	Channel간의 이동을 통한 접속방법 스위칭이 세션중에 허용될 수 있는 기능
Scalability	많은 접속자를 동시에 서비스할 수 있는 확장성을 제공하는 기능
End-To-End 보안	단말기로부터 Back-End 시스템까지의 End-To-End 보안을 보장하는 기능

버에 대한 정의는 매우 광범위한 것으로서 결코 특정한 기능을 한정하지는 않는다. 이러한 모바일 특수 로직은 매우 다양한 형태의 것이 될 수 있으며, 이의 구현은 모바일 응용서버의 자체적인 서비스, 기존 응용시스템과의 연동, 기존 응용시스템의 수정을 통해 구현된다.

실제적인 모바일 응용서버의 기능은 이를 구현하는 벤더에 따라 기능이 천차만별이다. 어떤 모바일 응용서버는 단순히 기존의 웹서버에다가 모바일 기기와 연동될 수 있는 기능을 추가하고, 프로그램 개발자들에게 DB에서 자료를 추출해서 단말기에 있는 브라우저 기반의 소프트웨어에 보낼 수 있게 하는 기능이 있는 API를 제공한다. 또 어떤 모바일 응용서버는 비즈니스 응용 로직을 제공하여 기존의 Back-End DB의 자료를 추출하

여 이를 가공한 후 제공할 수 있게 한다. 어떤 모바일 응용서버는 데이터의 가공없이 WAP 프로토콜을 통해 단순히 정적 웹 페이지를 제공하기도 한다.

모바일 응용 시스템을 도입하고자 하는 기업은 모바일 응용서버 제공 업체들이 자신들이 원하는 충분한 기능을 모두 제공하는 것이 아니라는 것을 염두에 두어야 한다. 따라서 어떠한 벤더의 솔루션을 채택하기에 앞서 기존 시스템과의 연동성 및 기기 지원의 다양성 등, 기업의 상황에 따른 충분한 고려가 선행되어야 한다.

## 2.2 모바일 응용서버의 구성 방법

### 2.2.1 Enterprise Model vs. Hosted Model[2]

모바일 응용서버를 구성하기 위한 방식은 대체적으로 두 가지 형태이다. 첫째는 기존 인터넷 응용 시스템을 운영하고 있는 기업들에게 독자적으로 모바일 응용서버를 구성할 수 있도록 관련기술 및 모듈을 제공하는 방법으로서, 이를 Enterprise Model 이라 부른다. 또 다른 방법은 벤더가 독립적인 모바일 응용을 위한 컴포넌트를 만들어 운영함으로써 기업들이 이를 활용하게 하는 방법으로 보통 Hosted Model 이라 불린다.

Enterprise Model은 기업이 소유한 내부 플랫폼에 설치되고, Hosted Model은 별도의 ASP 업체가 제공하는 포탈서버에 장착되어 제공되는 것이 일반적이다. 두 방식에는 다음과 같은 특징이 있다. 첫 번째 방식은 기업들에게 모바일 응용서버를 제어할 수 있는 유연성을 제공할 수 있는 반면, 두 번째 방식은 적은 비용으로 빠르게 모바일 응용 시스템을 구성할 수 있다는 장점이 있다.

Enterprise Model의 경우, 기업은 모바일 응용서버 관련 업체들이 제공하는 독립된 모듈들을 자신들의 기존 시스템 위의 새로운 계층으로 형성

하여 활용하는 방식을 택할 수도 있고(Connection), 또는 기존의 시스템을 수정함으로써 벤더에서 제공하는 기능을 통합하는 형태(Integration)를 취할 수도 있다.

2.2.2 모바일 응용서버의 구조 및 기능

모바일 응용서버는 서버 기반 소프트웨어로서 기존의 인터넷 응용 인프라에 접목되어 무선환경에서의 인터넷 응용이 가능케 하는 소프트웨어이다. 이 소프트웨어의 가장 핵심적인 역할은 데이터를 프리젠테이션 로직에서 분리시킴으로써 중복되는 불필요한 소프트웨어의 개발을 최소화시키는 것이다. 기술적인 관점에서 본다면 모바일 응용서버는 데이터에 대한 안전한 접근을 제공하고 이러한 데이터 콘텐츠를 어떠한 장치(PDA, 이동 전화기, 페이지, 데스크탑 PC) 에도 제공할 수 있어야 한다. 즉, 이 제품들은 데이터에 대한 접근을 제어하는 정책을 구현하고 콘텐츠와 서비스에 대한 제어 기능을 제공해야 하는 것이다.

구조적 관점에서 볼 때 모바일 응용서버의 구성은 [표 2]와 같다.

3. 효율적인 모바일 응용서버 구성을 위한 방향

인터넷과 응용 시스템의 접목은 여러 가지 미들웨어 플랫폼(예로서, 응용서버 및 Integration Broker) 및 컴퓨팅 모델(예로서, 브라우저 기반 클라이언트)이 새롭게 등장함으로써 가능하였다. 또한 이러한 컴퓨팅 모델을 위해서 기존의 기술(예로서, TP 모니터) 및 아키텍처(예로서, 3-Tier Client/Server)와 인터넷 기술 및 프로토콜(예로서, HTML/HTTP, Java) 이 접목되는 것이 필요하였다. 기존의 e-business 아키텍처에 무선단말기를 접목하고자 하는 m-commerce 응용을 위한

표 2. 모바일 응용 서버의 요소

계층	설 명
컨텐츠 계층	웹서버, 데이터베이스, 트랜잭션기반 응용, 음성 등의 데이터 기반의 소스에 대한 정형화된 인터페이스를 제공하는 계층
프리젠테이션 계층	단말장치에 따라 적합한 프리젠테이션을 위한 콘텐츠의 변환을 담당하는 계층
링크 계층	보안 및 암호화, 세션관리, 동기화(Synchronization), 고지(Notifica-tion), 음성인식 등의 기능을 제공하는 계층
프로파일 관리 계층	개인화(Personalization), 선호(Pre-ferences), 위치인식(Location Aware-ness), 인증(Authentication) 및 로그인을 관리하는 계층

시도도 현재 이와 유사한 방향으로 나아가고 있다. 새로운 미들웨어와 컴퓨팅 패러다임이 나타나고 있고, 이러한 개념들의 기존 시스템과의 연결을 통해서만 엔터프라이즈 급 m-business 응용 시스템을 구성할 수 있는 기반이 마련되고 있다 [12,13].

본 장에서는 이러한 기존의 미들웨어를 통한 모바일 응용시스템의 구성방법에 대해 살펴봄, 어떠한 형식의 미들웨어가 모바일 응용서버를 구성하기 위해 적합한지, 또 어떠한 기능 첨가에 의해 모바일 응용으로의 접목이 효과적으로 이루어질 수 있는지에 대해 살펴보기로 한다[14-17].

3.1 모바일 응용시스템을 위한 미들웨어

미들웨어는 모바일 응용 아키텍처를 구축하고 이를 기존의 응용 시스템과 통합하는 데에 있어 핵심적인 역할을 할 수 있고, 앞으로 이를 위한 기능을 계속적으로 구현하게 될 것이다. 모바일 응용 아키텍처를 효과적으로 지원하기 위해서

MOM, RPC, ORB와 같은 통신 미들웨어, ODBC와 같은 데이터 접근 미들웨어가 모바일 기기와 서버에서 실행되고 무선네트워크를 기반으로 작동해야 한다. Application Server, Integration Server와 같은 다른 형태의 미들웨어는 모바일 기기에서 작동할 필요는 없으나 무선 네트워크 접속 기능을 제공하여야 한다. 문제는 기존의 미들웨어들은 모바일 기기나 무선 네트워크를 지원하는 기능을 가지도록 디자인되지 않았다는 데에 있다. MOM, RPC, ORB, ODBC 등은 모바일 기기에서 실행되기에는 너무 많은 용량을 요구하고, 무선 접속 상에서 발생하는 잦은 통신단절에 대해 잘 견디지 못한다. TP 모니터, Integration Broker, Application Server는 대신에 모바일 기기에 대한 접속기능을 가지고 있지 못하다. 따라서, 기존의 미들웨어들은 무선 인터넷의 이런 특징적 상황을 대처할 수 있는 방향으로 수정되어지고 있다. 어떤 식의 수정인지는 본 절에서 살펴보기로 하고, 이후의 절에서는 여러종류의 미들웨어가 어떻게 모바일 응용을 위해 활용될 수 있는지를 살펴보고자 한다.

### 3.1.1 적은 용량

모바일 응용은 일반적으로 자원이 한정된 기기에서 작동하게 된다. 메모리 및 저장공간이 적고 프로세싱 속도도 늦을 뿐만 아니라, 배터리의 용량을 적게 사용해야 하는 등의 문제점이 모바일 응용 시스템 개발자들에게는 큰 골칫거리이다. 따라서 모바일응용의 이러한 열악한 환경에 대처하기 위해 모바일 응용을 지원하는 미들웨어는 서버측의 프로그램과 상호작용 하는데 꼭 필요한 최소한의 기능만을 가지도록 기능이 축소되어야 할 것이다.

### 3.1.2 소규모 프로토콜

안정적이고, 상대적으로 빠른 네트워크에서 작

동하던 기존의 미들웨어는 아주 고차원적인 기능을 제공하기 위한 여러 가지 프로토콜을 지원하고 있다. 이러한 프로토콜을 구현하기 위해서는 용량이 커질뿐더러, 통신 속도가 낮은 무선 네트워크에서 복잡한 기능을 활용하기는 곤란하다. 따라서 프로토콜의 소형화 작업도 꼭 필요한 기능만을 중심으로 제한되는 작업이 있어야 한다.

### 3.1.3 대용량 Scalability

모바일 응용시스템은 많은 접속을 처리할 수 있어야 한다. 예를 들어, 무선전화기 사용자들의 숫자는 그 단위가 수천만으로 보통 얘기된다. 따라서 이들이 무선 포탈에 접속할 때 생기는 부하는 엄청난 것일 것임을 쉽게 알 수 있다. 이러한 상황으로 현재 미들웨어의 Scalability가 높다고 하더라도 충분한 동시접속 지원을 위해서는 역부족이다. 따라서 Scalability를 높이기 위해 미들웨어 벤더들은 고차원의 캐싱, 클러스터링, 로드 밸런싱 기법들을 제품에 제공하려고 노력하고 있으나, 이러한 기능이 미들웨어 기술개발의 비용을 높일 것이라는 것은 뻔한 사실이다. 따라서 이를 위한 저비용 솔루션의 개발이 필요하다.

### 3.1.4 높은 안정성

Message Queuing, TP 모니터, Application Server, Integration Broker와 같은 미들웨어를 제공하는 벤더들은 분산 처리를 위한 높은 안정성을 제공하는 제품들을 가지고 있다. 간헐적인 접속, 잦은 단절 등으로 불안정한 통신일 수밖에 없는 무선통신의 특성은 이러한 미들웨어를 사용할 경우에 감추어 질 수 있다. 그러나 일반적인 기존의 유선 네트워크 기반 미들웨어들에 있어 네트워크의 불안정성은 이제까지는 고려의 대상이 아니었던 바, 이러한 문제에 대한 대비가 전무한 실정이다. 그러므로 이러한 미들웨어를 네트워크의 접속이 불안정한 모바일 응용에서 활용하기 위해서

는 기존 제품에 수정이 가해져야한다.

### 3.1.5 양방향 통신 패러다임

모바일 응용에서는 단말기에서 서버측 응용 시스템에 대화의 요청을 하지만 그 반대의 통신도 제공되어야 한다. 그리고 이러한 양방향 통신이 많은 단위로 제공될 수 있어야 한다. 현재, 대량의 Many-To-Many 양방향 통신을 여러 형태의 기존 미들웨어를 접목시켜(예로서, Message Queuing, Publish-And-Subscribe) 구현하고 있는 제품도 출현하고 있다(무선 MOM 등).

## 3.2 Data Management Middleware

데이터관리 미들웨어는 프로그램이 (응용 프로그램 및 DBMS) 원격 데이터베이스 또는 파일에서 자료를 읽고 쓰는 것을 도와준다. 이러한 미들웨어는 대략 두 가지의 종류로 분류할 수 있다.

### 3.2.1 데이터 접근 미들웨어

지난 10년간 수많은 클라이언트/서버 형태 및 Web 응용 프로그램들이 ODBC, JDBC 및 이와 유사한 기능을 가진 API에 기반을 둔 데이터 관리 미들웨어를 이용하는 형식을 통해 개발되어져 왔다. 그러나 이 미들웨어는 빠르고 안정적인 네트워크의 접속을 요구한다. 즉, 이 미들웨어는 지역 네트워크(LAN) 상에서 가장 잘 작동하며, 빠른 속도가 지원되는 WAN이라면 잘 작동하나, 요구량이 많지 않은 응용프로그램이라도 광역 네트워크에서는 그 수행성능이 별로 좋지 않다. 데이터 접근 미들웨어는 통신단절에 잘 대응하지 못하며, Scalability 에 있어서도 클라이언트 응용과 서버의 특정 세션이 유지되어야 하는 관계로 그다지 좋은 성능을 보이지 못한다.

### 3.2.2 Database Synchronization 미들웨어

또 다른 형태의 데이터 관리 미들웨어로는 데

이터베이스 Replication(유사한 DBMS 간의) 또는 Propagation(서로 다른 제품간의) 을 통한 데이터 Synchronization기능을 제공하는 미들웨어를 들 수 있다. 이 기술을 사용하면 서로 다른 DB 간 데이터 동기화를 DB에 대한 수정 로그를 활용해 할 수 있게 된다.

모바일 및 무선 응용의 관점에서 볼 때 이 기능은 매우 유용하다. 즉, 모바일 사용자가 어느 정도의 데이터를 단말장치에 유지하고, 이 데이터가 중앙의 데이터 저장소에 때때로 업로드되던지 아니면 다운로드 되는 형식을 취하는 응용에서 이를 활용할 수 있다.

## 3.3 통신 미들웨어

통신 미들웨어는 모바일 기기를 위해 제공되는 개발 플랫폼을 활용해 개발된 클라이언트 측 응용 프로그램(J2ME, PocketPC, EPOC 등)과 서버에서 운영되는 플랫폼 미들웨어(TPM과 Application Server)를 접속하기 위한 핵심적 역할을 담당한다.

이러한 통신 미들웨어는 Request/Reply와 MOM 두 개의 형식으로 분류될 수 있다.

### 3.3.1 Request/Reply 미들웨어(RPC, ORB)

Request/Reply 미들웨어는 하나의 프로그램이 네트워크 상의 다른 프로그램을 마치 함수호출을 하는 것처럼 부를 수 있게 한다. 이때 두 프로그램은 클라이언트/서버 관계를 가진다. 클라이언트(호출자)는 서버(피호출자) 프로그램에게 파라미터를 전달함으로써 통신을 시작한다. 파라미터를 받은 서버 프로그램이 실행되게 되고, 결과를 만들어 이를 다시 파라미터를 통해 클라이언트 프로그램에 전달한다.

RPC는 이러한 Request/Reply 미들웨어의 가장 기본적인 형태이다. ORB 플랫폼(CORBA

ORB, Microsoft의 DCOM, Java RMI 등)은 보안, 트랜잭션 제어, Naming 등의 더 복잡한 기능을 제공하기도 하지만 이 범주의 기능을 바탕으로 작동한다. SOAP (Simple Object Access Protocol)도 이 범주에 속하는 미들웨어이다.

Request/Reply 미들웨어 클라이언트 소프트웨어는 용량이 크지 않아서 소용량의 모바일 단말기에서 운용하기에 적합하다. 하지만 이 미들웨어 프로토콜이 동기화에 바탕으로 구현되기 때문에, 비록 CORBA가 트랜잭션 환경에서 수천의 동시 접속을 처리할 수는 있다고는 하지만 Scalability에 문제가 있을 수밖에 없다. 가장 큰 문제는 이 제품들이 무선환경의 불안정한 통신접속에 잘 견디지 못한다는 것이다. Request/Reply 세션동안 통신이 끊어진다면, 그 세션이 후에 계속되는 것은 어려우며, 이 경우 트랜잭션이 완료되었는지에 대한 확신을 클라이언트는 가질 수 없게 된다.

### 3.3.2 MOM(Message Oriented Middleware)

구조상, MOM은 모바일 응용에 잘 어울리는 미들웨어이다. Messaging 기술은 RPC나 ORB보다 Scalability가 높는데 이는 메시지를 전달하는 프로그램 상호간에 긴 시간동안 세션을 유지할 필요가 없기 때문이다. Message Queuing은 메시지를 영구적 저장장치인 Queue에 저장함으로써, 불안정한 네트워크 상에서라도 안정적이고 확실하게 단 한번의 메시지 전달을 보장해 줄 수 있다(예를 들어 IBM의 MQSeries Everyplace[9]).

그러나 MOM 제품들은 소용량의 모바일 단말기에서 운영되기에는 용량 면에서나 프로세싱 속도 면에서의 요구사항이 높다. 따라서 벤더들은 제품의 기능을 줄여 소용량 단말기에서 운용될 수 있는 MOM 버전을 속속 출시하고 있다.

이러한 종류의 제품을 사용함으로써 모바일 응용 시스템 개발자들은 무선네트워크를 통해 효율

적이고 안정적으로 작동되는 복잡한 응용 프로그램을 개발할 수 있다. MOM을 통해 응용 프로그램들은 기존의 서버측 응용시스템과 메시지를 주고받을 수 있을 뿐만 아니라, 서버측 응용프로그램에서 보내는 Notification에 대해 반응할 수도 있게 된다.

MOM은 이러한 여러 가지 장점으로 인해서 (대중성, Scalability, 유연성, 모바일 아키텍처와의 좋은 연관성) 모바일 응용과 기존의 응용 시스템과의 연결을 제공하는 통신 미들웨어로서 가장 각광받게 될 것으로 보인다.

## 3.4 Platform Middleware

PDA나 이동전화기에 보기 좋은 디스플레이를 제공하는 것이 모바일 응용의 전부는 아니다. m-business를 포함한 모바일 응용은 서버 및 데이터 소스에 대한 접근 로직을 가지고 있어서 다양한 정보를 DB를 통해 읽고 갱신하며 이에 대한 논리를 처리하는 일들도 담당해야 한다. 따라서 모바일 응용 시스템 구축을 위해 시스템 개발자에게 중요하게 대두되는 이슈중의 하나가 어떤 적합한 플랫폼 미들웨어를 선택하느냐의 문제이다. 가장 일반화된 플랫폼 미들웨어는 TPM과 Application Server 이다.

### 3.4.1 Application Server

1990년 후반에 나타난 Application Server의 주된 용도는 인터넷 응용을 담당하는 것이고, Thin/Ultrathin 클라이언트 장치를 위해서 사용되는 기술로 웹 브라우저를 이용한 HTML 프로세싱에 사용하는 방식이 일반적이다. 최근에 벤더들은 e-business가 SMS, WAP 장착 모바일 단말기 등을 지원하는 방식으로 이루어져야 한다는 것을 인식하고 이를 위해 Multichannel을 지원하도록 Application Server 제품을 확장시키는 노력

을 기울여 왔다. 근래에 출시되는 많은 Application Server들은 다양한 형태의 클라이언트 아키텍처를 지원하는데, 이에선 일반 PC, 웹서비스 프로토콜, 무선 단말기, JMS(Java Messaging Service) 등이 있다. 많은 Application Server 벤더들은 WAG(Wireless Application Gateway)와 다른 모바일 환경에서 작동되는 기존 제품의 기능 접목을 통해(모바일 MOM, Synchronization, Notification 등) Application Server의 기능을 확장하는 형식을 취하고 있다. WAG는 모바일 단말기를 지원하기 위한 제반의 자료변환 및 응용로직을 담당하는 모바일 응용 아키텍처의 일부로서, 대체로 Front-End의 모바일 기기를 위한 지원기능을 담당하는 경향이 강하다. 따라서 WAG는 Front-End 서비스기능에 중점을 둔 모바일 응용 서버라 할 수 있다.

Application Server는 빠른 속도로 변화/발전하고 있다. 비록 아직까지는 Unix/Windows 기반의 TPM의 가용성(Availability), 성능 및 Scalability를 쫓아가지는 못하지만 대용량 Multichannel을 지원하기 위한 개발이 진행중이다. 모바일 미들웨어 기술이 성숙되고 사용자의 요구사항이 명백해짐에 따라, 다양한 클라이언트용 아키텍처(Always-On, 마이크로브라우저 기반, 간헐적으로 접속되는 Thin Client와 Offline 기능)가 Application Server에 반드시 지원하는 기능에 포함 될 것으로 예상된다.

### 3.4.2 TPMs

TPM은 아직까지도 대형의 OLTP 비즈니스 응용을 위해 가장 많이 사용되는 플랫폼 미들웨어이다. CICS, IMS 및 Tuxedo는 일반적인 Application Server나 ORB와 견주어 볼 때 성능, Scalability, 보안, 가용성 면에서 월등히 뛰어나다. TPM은 또한 여러 종류의 클라이언트 아키텍처,

효율적인 클라이언트/서버 프로토콜 및 메시징을 지원할 수 있다. 그러므로 이론상으로는 새로운 대형의 서버 측 모바일 응용을 지원하기 위해서는 TPM이 가장 적합한 플랫폼일 수 있다.

하지만 TPM의 선호는 급격히 떨어지고 있고, 이러한 제품의 가격은 매우 고가이다. 점점더 많은 서버측 응용이 J2EE나 Microsoft의 .NET을 지원하는 Application Server 상에서 개발되고 있으며, 이러한 Application Server가 많은 응용시스템 개발을 위해 사용될 것으로 보인다.

### 3.5 Integration Middleware

모바일 기기에서 실행되는 무선 응용프로그램은 유선 상에서 작동하는 응용 프로그램과 근본적으로 다를 바가 없다. 이 응용 프로그램들은 비즈니스 프로세스를 수행하고 데이터를 관리하며, 이벤트에 대한 Notification을 받는 기능들을 수행하기 위해서 다른 응용 시스템과 연함(Integration)된다. 지난 수년간 많은 제품들이 이러한 응용 프로그램의 연함을 좀더 구조화된 형태로 제공하고 출시되었다. 이러한 제품들의 집합을 Integration 미들웨어라 부르며 이에선 Integration Broker, BPM(Business Process Manager) 및 Integration Server가 있다.

Integration Broker는 소프트웨어 플랫폼으로서 MOM 이나 다른 형식의 Communication 미들웨어 위에서 메시지 및 데이터의 변경, 라우팅 등의 작업을 담당한다. 이들은 또한 패키지가 된 응용, 기존 시스템 환경, DBMS를 위한 어댑터 및 이들을 개발 관리 할 수 있는 도구들을 제공한다.

보통 Integration Broker와 함께 제공되는 BPM은 여러 응용 및 사람과의 연함, 기업 외 작용들을 포함한 복잡한 여러 단계의 응용을 모델링하고 실행하는 일을 담당한다.



Integration Server는 웹 서버 기능의 일부 기능(예로서, HTTP 기능 및 웹 페이지 생성 기능), Integration Broker 기능(예로서, 라우팅 및 변환), 경량 응용프로그램 지원 기능, 어댑터 및 개발 틀들을 제공한다. 이들은 보통 B2B 응용 개발을 위해 활용되거나 아주 복잡한 웹 Integration 시나리오를 구현하기 위해 사용된다.

### 3.5.1 Integration Brokers와 BPM

Integration을 위한 다양한 기능을 제공하는 Integration Broker 및 BPM은 모바일 응용 시스템을 구축하는데 매우 중요한 역할을 할 수 있다. 모바일 응용이 다른 유선 상에서의 응용과 마찬가지로 Integration 시나리오의 일부분이 될 수 있는 것이다. 이들은 메시지를 주고받을 수 있고 이 과정에서 메시지의 변형이나 라우팅을 Integration Broker가 담당해 줄 수 있다.

모바일 응용은 또한 BPM을 이용해 자동화된 Multistep 프로세스에 참여 할 수 있다. 비록 몇몇의 BPM 벤더(예로서 Staffware 등)가 자사의 프로세스 플로우에 모바일 사용자를 포함시켜야 한다는 사실에 대해 잘 인식하고 있으나, 많은 Integration Broker 및 Integration Broker의 코드에 기반한 BPM은 모바일 아키텍처 및 무선 프로토콜을 위한 지원을 제공하고 있지 못하다.

### 3.5.2 Integration Server

Integration Server는 Web Integration 시나리오 상에서 보통 Application Server 대신에 사용되는 경우가 많다. 따라서 이들은 Application Server가 모바일 및 무선 기능을 접목하기 위하여 취하고 있는 방식과 유사한 방법으로 이에 대한 솔루션을 찾아나가고 있다.

비록 Integration Server와 WAG가 처음 보기에는 비슷해 보이지만, 전자는 Back-End Integration을 위한 다양한 기능을 제공하며, 후자는

개인화(Personalization) 및 모바일 기기, 무선 네트워크를 위한 지원에 있어서 강점을 가진다.

하지만 WAG가 모바일 단말기를 지원하는 쪽의 좁은 영역의 일을 담당하는 반면 Integration Server는 더 넓은 도메인인 Web Integration 마켓을 대상으로 하고 있다는 데에 Integration Server 벤더들의 어려움이 있다. 그러나 Integration Server를 사용하는 것이 Integration Broker와 WAG를 접목시켜 도입하는 것보다 더 위험부담이 클 것이라고 이야기 할 수는 없다. 왜냐하면 WAG가 이러한 솔루션을 제공하기 위한 가장 불안정한 요소 일 수 있기 때문이다. Integration Server를 사용한 해결 방법은 Integration/WAG 콤비네이션을 사용하는 솔루션에 비해서 비용부담이 덜하다. 따라서 아직 안정되지 않은 기술을 도입하는 입장에서 이러한 방향으로 나가는 것이 위험부담이 적을 수 있다.

## 4. 결론

기존의 무선 네트워크 환경에서 m-commerce 응용을 대중화시킨다는 것은 모바일 응용시스템이 가지는 특수성을 감안할 때 쉽지 않은 작업이 될 것이다, 최근에 들어서 이러한 서비스를 위한 근간을 제공해주는 솔루션이 속속 등장하고 있다. 이러한 솔루션에 중심이 되는 기술은 모바일 응용 서버 기술이며, 이 서버는 클라이언트 단말기와 서버측의 데이터 소스 및 로직을 연결시켜주는 역할을 담당한다. 본 연구에서는 최근에 출시되는 모바일 응용서버 제품들의 기능에 바탕을 두고, 전형적인 모바일 응용서버가 제공해야하는 기능에 대해 정의하였다.

또한 향후 모바일 응용서버의 효과적인 구현을 위한 방안을 살펴보았는데, 특히 기존 유선 응용시스템의 인프라를 그대로 활용할 수 있는 방법이

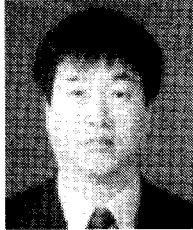
빠르게 성장하는 모바일 응용시스템을 위해 적합한 접근 방법으로 판단되었다.

본 논문에서 제시된 기존 시스템의 활용을 통한 모바일 응용시스템의 개발을 위한 방향에 대해 요약하면 다음과 같다. 첫째, 무선 응용시스템을 위해 개발된 기존의 프로그램을 수정 없이 활용할 수 있도록 한다. 둘째, 이렇게 변환 개발된 프로그램들이 모바일 응용 시스템 상에서 구동될 수 있도록 하는 인프라를 제공하는데, 이는 기존에 존재하는 미들웨어들을 수정하여 활용함이 바람직하다. 단말기의 용량 제한, 모바일 응용에 있어서의 특수성을 감안해 기존의 미들웨어의 기능이 축소되거나 변형되는 작업이 필요하나, 기존의 시스템에 입각한 모바일 응용시스템의 생성 및 구동으로 쉽고 빠르게 모바일 응용시스템을 구축할 수 있게 될 것이다.

**참 고 문 헌**

[ 1 ] <http://www.mobileinfo.com>, Mobile Computing Application Server.  
 [ 2 ] P. Lowber and K. Dulaney. Wireless Application Gateways: An Emerging Technology(an Update). *Gartner Technical Paper, October 2001*.  
 [ 3 ] S. Hayward. WAGs-Beyond Wireless. *Gartner Technical Paper, September 2001*.  
 [ 4 ] K. Dulaney. SyncML: A Standard for Offline

Computing. *Gartner Technical Paper, September 2001*.  
 [ 5 ] M. Basso. Service Platforms: A Good Choice for Multichannel Portals? *Gartner Technical Paper, September 2001*.  
 [ 6 ] M. Heijden and M. Taylor. *Understanding WAP*. Artech House Publishers, Norwood, MA, 2000.  
 [ 7 ] <http://www.oracle.com>  
 [ 8 ] [www.aligo.com](http://www.aligo.com)  
 [ 9 ] <http://www.ibm.com>  
 [ 10 ] <http://www.brience.com>  
 [ 11 ] <http://www.2roam.com>  
 [ 12 ] M. Pezzini. Middleware Glues Mobile Devices and the Enterprise. *Gartner Technical Paper, September 2001*.  
 [ 13 ] S. Hayward and M. Pezzini. Marrying Middleware and Mobile Computing. *Gartner Technical Paper, September 2001*.  
 [ 14 ] M. Pezzini. Do MOM, ORBs and Data Access Middleware Suit Mobile? *Gartner Technical Paper, September 2001*.  
 [ 15 ] M. Pezzini. Mobile Applications Look for Hooks in the Enterprise. *Gartner Technical Paper, September 2001*.  
 [ 16 ] M. Pezzini. Are Application Servers Ready for M-Business? *Gartner Technical Paper, September 2001*.  
 [ 17 ] J. Sinur. Display Limits in Mobile Apps Aren't Everything; Enter BPM. *Gartner Technical Paper, September 2001*.  
 [ 18 ] J. Capone. Java 2ME Bridging Wireless Gap?, The O'Reilly Network, August 2001.



오 동 익

- 1985년 뉴욕시립대학교 전산학 학사
- 1989년 플로리다 주립대학교 전산학 석사
- 1997년 플로리다 주립대학교 전산학 박사
- 1997년~현재 순천향대학교 정보기술공학부 조교수
- 관심분야 : m-commerce, e-commerce, 운영체제, 실시간 시스템, Ada 프로그래밍언어 등



이 경 호

- 1987년 고려대학교 산업공학과 학사
- 1990년 고려대학교 산업공학과 석사
- 1990년~현재 ETRI 책임연구원
- 관심분야 : e-Business, 모바일 어플리케이션 서버



이 중 섭

- 2001년 순천향대학교 컴퓨터학부 학사
- 2001년~현재 순천향대학교 전산과 석사과정
- 관심분야 : m-commerce, e-commerce, 모바일 어플리케이션 서버



김 중 배

- 1986년 고려대학교 산업공학과 학사
- 1988년 KAIST 산업공학과 석사
- 1998년~현재 KAIST 산업공학과 박사과정
- 1988년~1991년 대한항공(주) 시스템부 대리
- 1991년~현재 ETRI 모바일응용서버연구팀장
- 관심분야 : e-Business, 모바일 어플리케이션 서버