



## 유무선 통합 모바일 응용서버에 관한 연구

한국전자통신연구원 김성훈 · 장철수 · 정승욱 · 서범수  
노명찬 · 박중기 · 이경호 · 김중배

### 1. 서 론

#### 1.1 연구의 필요성

컴퓨터 성능이 향상되고 네트워크 사용 환경이 개선됨에 따라 인터넷 사용이 보편화 되었다. 이에 따른 인터넷 서비스에 대한 요구도 증가하여 인트라넷 환경에서의 클라이언트/서버 환경이 범용 브라우저를 이용한 웹 응용서버 기반의 인터넷 서비스 환경으로 변화하였다. 이를 반영이라도 하듯, 국내 웹 응용서버 관련 시장의 경우 2001년 매출액이 500억 원에 달하고 있으며, 전 세계 응용서버 시장의 경우에는 2001년 70억 달러에서 2005년에는 252억 달러로 연평균 37%의 수준으로 성장할 것으로 전망되고 있다 [1].

한편, 이동통신 기술의 발전에 따라 비교적 넓은 대역폭을 갖는 셀룰러 폰, PDA, Post PC 등과 같은 모바일 단말기들이 광범위하게 보급되어 활용되고 있다. 이에 따라 2003년까지 인터넷 접근의 50%가 PC가 아닌 휴대용 단말기를 이용하여 접속할 것으로 예상되며[2], 2005년까지 전 세계적으로 10억의 사용자가 모바일 단말기를 이용하여 인터넷에 접속할 것으로 보인다[3].

즉, 웹 응용서버 시장의 비약적인 성장과 모바일 단말기의 보급과 더불어 이를 이용한 비즈니스 및 서비스 환경으로 급변하고 있다. 개인용 컴퓨터로 인터넷 서비스를 이용하는 환경에서 다양한 유형의 모바일 단말기로 인터넷 서비스를 이용하는 유무선 통합 인터넷 서비스 이용 환경으로 변화가 일어나고 있는 것이다. 바야흐로 모바일 단말기를 중심으로 한 인터넷 서비스 환경으로 변화하고 있다.

환경 변화에 따른 새로운 형태의 모바일 서비스의 등장은 필연적이다. 이를 반영이라도 하듯, 국모닝

증권에 따르면 국내 무선 인터넷 서비스 업체의 수는 2001년 2400 개에서 2003년 13만 개로 기하급수적으로 증가하는 추세에 있으며, 기존의 유선 인터넷 서비스를 제공하는 업체들은 무선 영역으로 서비스 영역을 확장하고 있다. 예를 들어, 기존의 응용서버에 모바일 단말기로 메시지를 보낼 수 있는 메시지 처리 시스템을 추가함으로써 제한적으로나마 모바일 서비스를 지원할 수 있도록 하고 있다.

이처럼 모바일 서비스에 대한 요구가 증가함에 따라 특정 기능 위주의 모바일 지원 기술이 개발되고 있다. 하지만 무선 인터넷 서비스를 효과적으로 제공하려면 기존의 유선 인터넷을 기반으로 구축되어 운영중인 서비스를 모바일 환경에서도 쉽게 사용할 수 있음은 물론 모바일과 관련된 지식이 없이도 새로운 서비스를 쉽게 개발할 수 있는 유무선 통합 모바일 응용서버에 관한 연구가 절대적으로 필요하다.

#### 1.2 모바일 응용서버 개념 및 응용 예

유무선 통합 모바일 응용서버란 기존의 유선 환경에서 뿐만 아니라 모바일 단말기를 이용한 무선 환경에서도 이용할 수 있는 비즈니스 서비스를 개발하고 운영할 수 있는 응용서버로서, 복잡한 하부 구조에 대한 지식이 없이도 분산 컴포넌트 로직을 쉽게 개발할 수 있음은 물론 개발된 로직을 효율적으로 관리할 수 있는 서버 측 미들웨어이다. 이 경우 비즈니스 서비스는 무선 단말기의 제약성과 이동 통신망의 간헐적 단절성을 극복할 수 있는 모바일 비즈니스 서비스로서, 기존의 인터넷 환경에서의 기업 내부 비즈니스나 기업간 비즈니스 등 유선 인터넷 서비스를 포함한다.

유무선 통합 인터넷 응용의 한 예는 기존의 전자상거래 서비스를 확장한 경우를 들 수 있다. 유선 인터넷 서비스의 경우 유선 랜 혹은 무선 랜으로 연결

된 PC 단말기를 이용하여 원하는 상품을 검색하고 주문할 수 있다.

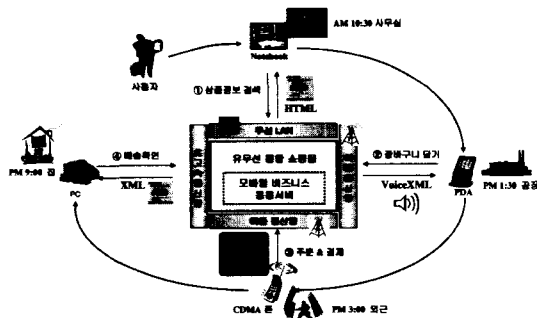


그림 1 유무선 통합 인터넷 서비스의 예

랜으로 연결된 PC 뿐만 아니라 다양한 모바일 단말기를 이용할 경우 보다 편리하게 전자상거래를 수행할 수 있다. 모바일 응용서버 시스템에서는 이처럼 PC 뿐만 아니라 다양한 모바일 단말기를 이용한 유무선 통합 환경에서의 전자상거래가 가능해진다. 그림 1은 유무선 통합 환경을 제공하는 모바일 응용서버 기반으로 유무선 통합 쇼핑몰에서 상품을 검색하고 주문하는 예이다. 예를 들어, 사용자는 회사에서 무선 랜으로 연결되어 있는 노트북을 이용하여 주문하고자 하는 상품을 검색할 수 있다. 이어 공장에서 CDMA가 장착된 PDA를 이용하여 검색한 상품을 장바구니에 담을 수 있으며, 퇴근 후 집으로 가는 도중에 CDMA 폰을 이용하여 주문하고 결제할 수 있다. 집에 도착한 후 초고속 통신망에 연결된 개인용 컴퓨터를 이용하여 배송을 확인할 수 있다. 이처럼 유무선 통합 모바일 응용서버를 이용할 경우 단말기 종류 및 네트워크에 무관하게 서비스를 제공받을 수 있다.

### 1.3 기술 현황

모바일 응용서버와 관련된 국외 기술은 크게 기업을 대상으로 독자적인 모바일 응용서버 솔루션을 개발하는 기업형 모델 분야와, 모바일 응용서버 호스트 서비스를 할 수 있는 컴포넌트와 플랫폼을 개발하기 위한 호스트형 모델로 나누어지며, 2001년부터 시제품이 출시되고 있다.

국내의 경우 다양한 모바일 단말기에 적합한 콘텐츠로 변환하는 콘텐츠 변환기술, 모바일 단말기와 서버 측 데이터베이스와의 데이터 일치를 위한 데이터

동기화 기술, 유선 그룹웨어를 모바일 환경으로 확장하기 위한 모바일 그룹웨어 기술, 그리고 SMS (Short Message Service), MMS (Multimedia Message Service)와 같은 메시지 처리 시스템 등이 개발되고 있다. 이러한 국내 기술들은 특정 기능을 위주로 하고 있기 때문에 유무선 통합 서비스를 위한 기능이 미흡하고 모바일 비즈니스 서비스를 위한 미들웨어 기능이 통합적으로 제공되고 있지 않다.

국외의 경우 IBM, Oracle 등은 기존의 유선 응용서버에 콘텐츠 변환 기술과 같은 일부 모바일 기능을 추가하고 있으며, Brience 등은 무선 서비스 기능을 위주로 제공함으로써 상대적으로 유선 환경에서의 컴포넌트 미들웨어 기술은 취약하다. 전체적으로 유무선 통합 인터넷 환경에서 다양한 유무선 단말기, 네트워크, 콘텐츠 규약 등을 만족할 수 있는 통합 프레임워크 기술 개발이 필요한 실정이다.

한편 웹 응용서버와는 달리 모바일 응용서버의 경우 국제적으로 공인된 산업 표준이 없으며, 따라서 조사 기관 및 관련 연구들도 개별적인 시스템 구조를 제안하고 있다. 즉, 모바일 단말기의 소프트웨어 구동 환경, 액세스 게이트웨이(Access Gateway), SMS, MMS, J2EE, .NET, 워크플로우(Workflow) 등의 개별 표준은 존재하지만 모바일 응용서버 전체 프레임워크 및 관련 기능에 대한 표준은 존재하지 않는 실정이다.

### 1.4 모바일 환경에서의 고려사항

유무선 통합 모바일 응용서버 기술 개발 시 고려해야 할 사항에는 크게 세 가지가 있다. 첫 번째는 모바일 단말기 자체에 관한 제약으로서, 메모리 크기 및 단말기 계산 능력과 같은 소프트웨어 구동환경, 그리고 스크린 크기 및 키보드 등과 같은 입출력 장치에 관련된 사항이다. 또한 밴드에 따라 다양한 유형의 단말기가 있을 수 있다. 이러한 단말기 자체에 따른 제약 때문에 프로토콜 문제, 세션 유지 문제, 단말기 유형에 적합한 최적의 데이터 표현 문제 등을 고려해야 한다.

두 번째는 무선 네트워크와 관련된 사항이다. 무선 네트워크는 기존의 유선 네트워크와 비교해 볼 때 상대적으로 낮은 대역폭을 가지며, 이로 인해 데이터 전송 속도가 비교적 느리다. 또한 간헐적으로 연결이 단절되는 문제가 있으므로 모바일 응용서버 설계시

세션 유지 및 최적으로 데이터를 전송할 수 있는 방법에 대해 고려해야 한다.

세 번째는 기존 시스템과의 연동 문제이다. 유선 인터넷 환경에서 이미 구축되어 운영 중인 레거시 시스템 및 기존 서비스와 연동함으로써 보다 효율적인 모바일 응용서버를 구축할 수 있다. 즉, 기존의 유선 인터넷 서비스를 모바일 인터넷 서비스로의 손쉽게 확장할 수 있어야 하며, 한 번의 콘텐츠 저작으로 유무선 환경에서 동시에 사용할 수 있어야 한다.

본 논문에서는 이러한 문제들을 해결하기 위하여 유무선 통합 인터넷 환경에서의 네트워크와 프로토콜, 단말기에 독립적인 고가용성 모바일 비즈니스 응용서버 기술에 대해 설명한다. 이를 위해 모바일 단말기 특성 및 간헐적 연결 단절성을 고려한 모바일 환경 적응 기술을 제안하고, 기존 유선 응용을 무선 응용으로 확장하기 위한 유무선 연동 기술 방법을 제안한다.

다음 절에서는 본 연구에서 제안하는 유무선 통합 모바일 응용서버의 개요 및 특성에 대해 설명한다. 3절에서는 모바일 단말기 및 네트워크 환경에 적응함으로써 유무선 통합 서비스를 지원할 수 있는 환경 적응 기술에 대해 설명하고, 4절에서는 이미 구축되어 있는 레거시 시스템과 연동함으로써 시스템을 효과적으로 확장할 수 있는 유무선 연동처리 기술에 대해 설명한다. 대규모 서비스 요청을 효율적으로 처리하기 위한 클러스터링 기술에 대해 5절에서 설명하며, 마지막으로 본 논문에서 제안하는 방법에 대한 결론을 6절에서 내리고자 한다.

## 2. 유무선 통합 모바일 응용서버 개요

유무선 통합 모바일 응용서버는 모바일 환경에서 이용할 수 있는 분산 컴포넌트 즉, DB 로직, 응용 로직, 표현 로직 등을 쉽게 개발할 수 있으며, 이미 개발된 컴포넌트의 생명 주기를 효율적으로 관리할 수 있다. 이러한 분산 컴포넌트는 무선 환경에서 효과적인 서비스를 제공하기 위해 무선 단말기의 제약성과 이동 통신망의 간헐적 단절성을 극복할 수 있어야 한다. 이를 위해 본 논문에서는 모바일 단말기와 무선 네트워크, 그리고 특정 사용자에 적응적이고, 기존의 레거시 시스템과 쉽게 연동 확장할 수 있도록 그림 2와 같은 구조를 갖는 유무선 통합 모바일 응용서버를 제안한다.

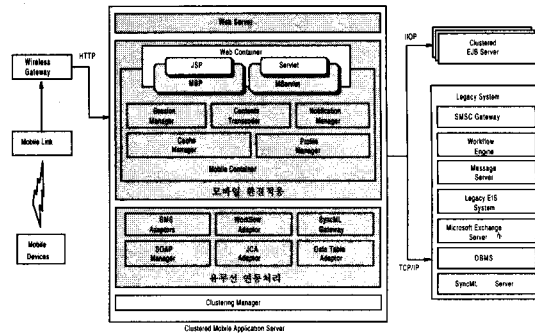


그림 2 유무선 통합 모바일 응용서버 구성도

유무선 통합 모바일 응용서버는 크게 두 부분으로 구성된다. 다양한 모바일 단말기 및 간헐적 연결 단절성, 그리고 사용자에 적응하기 위한 모바일 환경 적응 기술 부분과, 기존의 레거시 시스템과 연동할 수 있도록 함으로써 확장성을 갖도록 하기 위한 유무선 연동처리 기술이다.

모바일 환경 적응 모듈에서는 MServlet(Mobile Servlet)과 MSP(Mobile Server Page) 등과 같은 모바일 빈(Mobile Bean)을 통해 유무선 통합 서비스를 제공한다. MServlet이란 서블릿을 확장한 것으로서 애플리케이션 개발자로 하여금 모바일 환경에 대한 자세한 지식이 없이도 서비스를 개발할 수 있도록 하기 위한 일종의 API이다. 개발자는 단지 MServlet 클래스로부터 상속받아 기존의 서블릿을 작성하듯이 프로그램을 작성함으로써 모바일 단말기에 제공하는 서비스를 쉽게 개발할 수 있다. MSP란 기존의 JSP를 확장한 스크립트 언어로서, 모바일 서비스와 관련된 API를 지원하고 모바일 단말기에 콘텐츠를 표시할 수 있는 동적 웹 페이지 생성을 위한 서버 측 응용이다.

예를 들어, 모바일 단말기의 마이크로 브라우저를 통해 모바일 응용서버의 특정 서블릿(MServlet)이나 페이지(MSP)를 요청할 경우 WAP과 같은 무선 프로토콜을 이용하여 무선 연결을 통해 무선 게이트웨이로 전달된다. 무선 게이트웨이는 요청을 HTTP 프로토콜로 바꾼 후 웹 서버로 요청을 전송한다. 웹 서버에는 모바일 단말기의 요청을 처리하기 위해 특별히 작성된 서버 사이드 응용이 존재하며 이 응용이 모바일 환경 적응 모듈에 존재하는 응용, 즉 사용자가 요청한 서비스를 실행한다.

모바일 단말기에서 요청한 서비스는 필요에 따라

세션 관리자의 통제를 받으며 서비스가 수행된다. 서비스를 수행한 결과를 서비스를 요청한 단말기에 응답을 보내기 위하여 컨텐츠 변환기로 하여금 해당 단말기가 표시할 수 있는 형태로 컨텐츠를 변환하도록 요청하며, 변환된 컨텐츠를 제한된 네트워크 환경 하에서 메모리 오버런(overrun) 등을 방지하며 최적으로 데이터를 전송할 수 있도록 캐쉬 관리자에게 서비스를 요청한다. 세션 관리자, 컨텐츠 변환기, 캐쉬 관리자 등은 요청한 사용자 및 모바일 단말기에 관한 정보를 획득하기 위하여 프로파일 관리자를 이용한다.

이처럼 유무선 환경에서의 비즈니스 서비스에 대한 요구에 부응하기 위하여 컴포넌트 기반의 모바일 응용서버를 지원함으로써 컴포넌트를 신속하게 개발함은 물론 테스트와 배포, 그리고 유지 보수를 용이하게 한다. 또한 기존의 유선 인터넷을 기반으로 구축되어 운영 중인 서비스를 재개발하지 않고 모바일 환경에 맞도록 조정함으로써 기존 시스템의 변경을 최소화할 수 있다.

다양한 모바일에의 적용 기술 및 세션 유지 기술을 이용함으로써 모바일 단말기의 간헐적 단절성 및 컨텐츠 표현의 제약성 등을 극복할 수 있으며, 클러스터링 기술을 이용하여 유무선 환경을 통해 전달된 대규모의 서비스 요청을 효율적으로 분산시켜 시스템 성능을 향상시킬 수 있다. 마지막으로, 레거시 시스템과 연동할 수 있는 다양한 연동 기술을 제공함으로써 하나의 개발 플랫폼을 통해 기존 시스템과 손쉽게 연동할 수 있다.

다음 절에서는 모바일 환경적용 기술을 위한 각 모듈 별 세부 기능에 대해 설명한다.

### 3. 모바일 환경적용 기술

모바일 환경적용 기술은 세션 관리자, 컨텐츠 관리자, 통지 관리자, 캐쉬 관리자, 프로파일 관리자 등으로 구성되어 있으며, 이러한 모듈들을 이용하면서 클라이언트로부터의 요청을 받아 처리하는 모바일 요청 수용기가 있다(그림 3).

#### 3.1 모바일 요청 수용기

모바일 응용에 대한 관문 역할을 하는 모바일 요청 수용기(Mobile Request Acceptor)는 클라이언트의 요청에 관련된 전체적인 처리를 수행하는 모듈로서, 요청에 대한 처리를 위해 각 단계에서 필요로 하

는 모듈을 구동하여 모바일 요청을 처리하고 최종 처리 결과를 클라이언트에게 전송하는 역할을 수행한다.

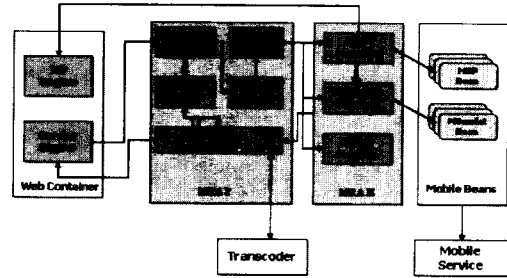


그림 3 모바일 요청 수용기 구성도

모바일 요청 수용기는 MServlet을 확장한 형태로 개발자에게 제공되는데, 개발자는 세션이나 프로파일 혹은 변환과 같은 모바일 환경에 관련된 자세한 내용에 대해 신경을 쓰지 않으면서 단지 제공된 모바일 요청 수용기를 확장하여 MServlet이나 MSP를 작성하기만 하면 모바일 요청 수용기가 실행 시간에 모바일 환경에 대한 처리를 자동으로 수행한다.

모바일 요청 수용기는 그림 3과 같이 클라이언트가 전달한 HTTP 요청 헤더로부터 각 통신사별로 특화되어 제공되는 사용자 정보 포맷을 인식하여 전화 번호 같은 사용자 식별번호, 화면 크기, 색상과 같은 브라우저 정보 및 위치 정보 등을 추출하여 일관된 형태로 가공하여 주는 요청 핸들러가 있으며, 로그 정보를 생성하여 로그를 남기는 로그 핸들러, 요청에 해당하는 페이지를 찾아서 해당 핸들러를 구동시키는 채널 핸들러, 해당 페이지에 대한 접근 권한 및 요금 등과 관련된 처리를 수행하는 보안 핸들러, 그리고 불안정한 모바일 통신망에서 안정적인 응답을 보낼 수 있도록 QoS를 보장하기 위한 응답 핸들러 등으로 구성된다.

#### 3.2 세션 관리

사용자 세션 관리를 위한 모바일 세션 관리자(Mobile Session Manager)의 기능은 다음과 같이 크게 세 가지로 분류할 수 있다. 첫 번째 기능은 모바일 네트워크의 간헐적 단절성을 극복하여 사용자의 네트워크 연결 상태와 무관하게 항상 세션을 유지하는 것이다. 두 번째는 쿠키를 지원하지 못하는 단말기를 위한 쿠키 프락시 기능이다. 세 번째는 사용자가 동

일 응용 내에서 여러 유형의 단말기를 사용하더라도 일관된 세션을 유지하도록 하는 것이다[4].

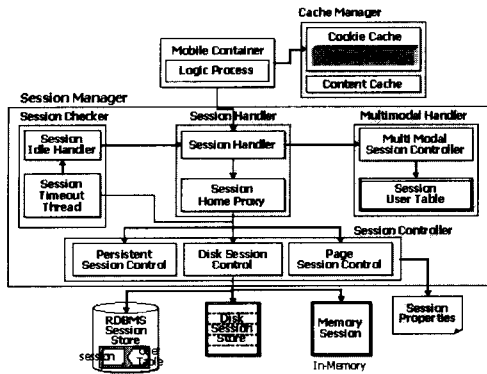


그림 4 모바일 세션 관리자 구성도

그림 4와 같이 모바일 세션 관리자는 세션 핸들러(Session Handler), 세션 체커(Session Checker), 멀티모달 핸들러(Multimodal Handler), 그리고 세션 컨트롤러 등으로 구성된다. 세션 핸들러는 모바일 컨테이너로부터의 세션 생성 및 소멸 요청을 처리하고 세션 컨트롤러의 접근 투명성 기능을 제공하며, 세션 체커는 단말기의 간헐적 단절성을 검출하고 세션의 생명 주기를 관리한다. 세션 컨트롤러는 세션 핸들러로부터 생성된 세션 데이터 객체를 보관하는 DBMS, 메모리, 파일 디스크 등의 디바이스를 컨트롤하며, 멀티모달 세션 컨트롤러는 멀티 디바이스 간 세션의 일관성을 위하여 디바이스 또는 사용자 고유번호를 키로 하여 세션 데이터를 관리한다.

모바일 세션 관리자는 모바일 컨테이너 상에서 시스템의 초기화와 더불어 유일하게 존재하며, 시스템의 기능상 멀티 모달 세션을 위하여 디바이스 프로파일 정보를 관리하는 프로파일 관리자와 연동한다. 또한 쿠키를 지원하지 않는 사용자 단말기를 위하여 가상 단말기를 생성하여 쿠키 정보를 관리하는 캐쉬 관리자(Cache Manager)와도 연동한다.

### 3.3 프로파일 관리

프로파일 관리자(Profile Manager)는 메모리, 화면 크기, 키보드 타입 등과 같은 콘텐츠 변환을 위한 사용자 단말기의 특성 정보, 콘텐츠 변환기로부터 생성된 변환 프로파일 정보, 사용자 정보 등을 관리한다(그림 5).

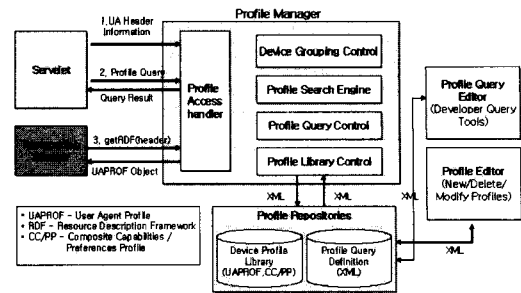


그림 5 프로파일 관리자 구성도

사용자 단말기 정보는 단말기에서 요청이 서버로 접수되는 시점에서 프로파일 관리자는 헤더 정보인 UAProf(User Agent Profile)를 분석하여 프로파일 레파지토리의 데이터를 갱신 또는 저장하며, 시스템 관리자가 프로파일 편집 도구를 이용하여 갱신할 수도 있다. 프로파일 레파지토리에 저장된 단말기 정보는 다음 절에서 설명할 콘텐츠 변환기에 의해 사용자 단말기와 가장 적절한 형태의 콘텐츠가 생성된다.

프로파일 관리자는 헤더 정보를 분석하여 해당되는 프로파일에 대한 레파지토리 정보를 생성/갱신하는 디바이스 프로파일 레파지토리 컨트롤러, 요청을 보낸 단말기의 프로파일 정보 또는 XSL을 반환하는 프로파일/XSL 검색 엔진, MSP 또는 MServlet 빈에 의해 전달된 프로파일 쿼리 정의를 분석하고 프로파일을 해석하여 쿼리 결과를 반환하는 쿼리 프로세서, 프로파일 레파지토리와 쿼리 레파지토리에 접근하여 데이터를 갱신하는 프로파일/쿼리 레파지토리 관리자 로 구성된다.

### 3.4 콘텐츠 변환 기술

콘텐츠 변환 모듈(Contents Transcoder)은 장치 독립적인 콘텐츠 출판을 위한 웹 콘텐츠 변환 기능을 수행한다. 무선 단말기를 이용한 인터넷 접속이 가능해지고 TV, PDA 등 PC 이외의 다양한 유무선 장치들이 인터넷 단말기 기능을 수행할 수 있게 됨에 따라 장치 독립성의 중요성은 날로 커지고 있는 실정이다.

W3C에서는 장치 독립성과 관련하여 다양한 권고안을 만들어내고 있다[5]. 그 중 대표적인 것이 CC/PP(Composite Capability/Preference Profile)이다. CC/PP는 장치의 기능 특성과 사용자의 콘텐츠 표시 선호 정보를 표현하기 위한 프로파일 기술 언어이다. 여기서 프로파일이란 장치, 네트워크, 사용자 등 웹

컨텐츠의 전송에 관련된 요소들에 대한 정보 모음을 가리킨다. 컨텐츠를 클라이언트 장치에 적합한 형태로 변환하기 위해서는 프로파일 정보가 필수적이다. WAP 진영에서는 이미 CC/PP를 채용하여 WAP 장치의 특성 정보를 기술하는 표준안을 UAPProf로 제정한 바 있으며, 앞으로 장치 독립성 분야에서 CC/PP의 채용은 더욱 보편화될 전망이다.

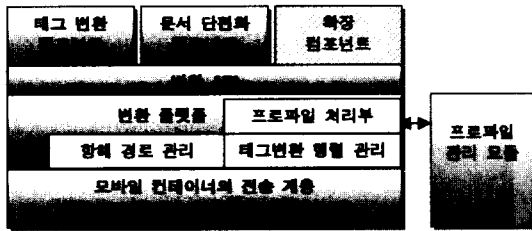


그림 6 컨텐츠 변환기 구성도

그림 6은 컨텐츠 변환 기술의 기본 구성도이다. 변환 플랫폼의 구조는 서블릿 컨테이너나 EJB 컨테이너와 같이 컨테이너 구조를 기반으로 한다. 변환 컨테이너는 구체적인 변환 기능을 수행하는 변환 컴포넌트들을 수용, 관리, 운영한다. 변환 플랫폼은 모바일 컨테이너의 프로파일 관리 모듈로부터 얻은 프로파일 정보를 기반으로 변환 작업을 수행한다.

태그 변환 컴포넌트에서 수행하는 태그 변환 기능은 입력 문서의 태그 집합을 제 3의 태그 집합으로 변환하는 기능으로서 태그 집합의 문법적 특성뿐만 아니라 의미적 특성까지 종합적으로 분석하며, 이를 위해 태그 변환 행렬(Tag Conversion Matrix)을 사용한다. 태그 변환 행렬은 확장성이 있어 새로운 태그 집합을 손쉽게 추가할 수 있기 때문에 다양한 태그 변환 기능을 쉽게 구축할 수 있다.

결국 컨텐츠 변환 기술은 모바일 응용서버의 웹 컨텐츠 제공 기능에 있어 장치 독립성을 강화함으로써 다양한 클라이언트 장치들을 수용할 수 있으며, 확장성 있는 구조와 프로파일 기반 기술을 바탕으로 지속적으로 새로운 장치와 전송 환경에 대응할 수 있다.

### 3.5 통지 관리

이동성을 지원하는 모바일 네트워크 상의 사용자 단말기는 SMS, Voice, Pager 등과 같은 다양한 형태의 통신 채널을 사용하고 있는데, 모바일 네트워크는 특성상 메시지의 전송에 있어서 유선 상에서 많이 사

용되는 동기식 방식의 전송과 함께 메시지의 저장 전달(Store and forward) 기능과 전달 보장 기능을 제공하는 비동기식 메시지 전송을 필요로 한다. 이를 위해 통지 관리자는 개발자가 작성한 응용에서 다양한 형태의 통신 채널을 통해 동기식 혹은 비동기식으로 메시지를 전송할 수 있도록 하고, 단말기가 전송한 SMS 메시지를 수신하여 처리한다.

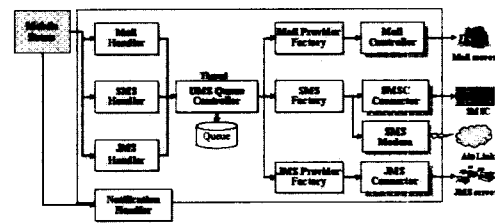


그림 7 통지 관리자 구성도

그림 7은 통지 관리자(Notification Manager)의 구성도이다. 통지 관리자는 MSP나 MServlet 등과 같은 모바일 빈에게 단일한 인터페이스로 다양한 형태의 메시지 전송 채널을 제공하는 채널 핸들러, 모바일 빈에서 전송된 메시지의 신뢰성을 보장하는 큐 컨트롤러, 채널 팩토리를 통해 디바이스 컨트롤러를 생성하고 실질적으로 메시지 전송 기능을 담당하는 디바이스 컨트롤러 등으로 구성된다. Mail 핸들러, SMS 핸들러, JMS 핸들러를 일컬어 채널 핸들러라고 하며, 각각의 채널 핸들러는 통지 관리자의 구성에 따라 다양한 형태로 추가되거나 제거될 수 있다.

### 3.6 캐쉬 관리

캐쉬 관리자(Cache Manager)는 대용량 서버 환경에서 사용자에게 보다 빠른 응답 시간을 보장하기 위해 한번 처리된 페이지에 대한 정보를 캐쉬에 저장하여 재요구시 캐쉬에 저장된 정보를 전송하여 응답 시간을 빠르게 함으로써 통신망 이용의 효율성을 높이는 기능을 수행한다. 또한 변환 규칙에 따라 변환기에 의해 이미 변환 완료된 이미지 데이터 등에 대해서는 불필요한 컨텐츠 변환 과정을 생략하여 전체 페이지의 생성 시간을 줄여 응답 속도를 개선할 수 있다.

웹 캐쉬는 크게 프락시 캐쉬(Proxy Cache), 투명 캐쉬(Transparent Cache), 리버스 캐쉬(Reverse Cache) 등으로 구별되는데, 프락시 캐쉬는 클라이언

트가 명시적으로 사용할 프락시 서버를 설정해 주면 페이지 요청 정보가 프락시 서버로 전달되도록 하는 방식이며, 투명 캐쉬는 네트워크 트래픽을 가로채서 캐쉬에 보내어 다른 네트워크 망과의 트래픽을 줄일 수 있는 방법으로 대부분의 ISP에서 적용하고 있는 방식이고, 리버스 캐쉬는 프락시가 하나의 캐쉬를 이용해 여러개의 불특정 서버를 캐쉬하는 것과는 반대로 하나의 서버를 위해 다수의 캐쉬를 이용하는 것이다.

유무선 통합 모바일 응용서버의 캐쉬 관리자는 위의 방법들 중 리버스 캐쉬의 성격을 갖고 있으며, 모바일 요청의 특징인 요청 폭주에 빠르게 대처함으로써 효과적으로 응답할 수 있다. 캐쉬 관리자의 중요 기능은 MSP와 같은 동적 문서와 이미지 및 프로파일 등과 같은 데이터에 대한 캐쉬 가능 여부를 결정하는 기능과, 캐쉬된 페이지의 일관성 관리 기능, 캐쉬간 데이터 동기화 기능, 자동 무효 처리 기능 등이 있다.

또한, 캐쉬 관리자는 각 모바일 단말기에 대한 메모리 용량 및 화면 사이즈 등의 제약 정보를 프로파일 관리자로부터 얻어 와서 콘텐츠 열람 편의성과 장치의 콘텐츠 수용 능력을 고려하여 요청된 원본 문서에 대해 지능적으로 문서를 분할한다. 이러한 단편화 기능을 통해서 클라이언트 단말기의 메모리 폭주를 막을 수 있을 뿐만 아니라, 데이터 전송 속도가 낮은 네트워크에서 최적의 속도로 데이터를 전송할 수 있다.

#### 4. 유무선 연동처리 기술

유무선 연동처리 기술은 유선 인터넷 환경에서 이미 구축되어 사용중인 서비스를 무선 환경에서도 사용할 수 있도록 시스템을 확장하기 위한 것으로, JCA 어댑터, JMS 어댑터, SyncML 게이트웨이, 워크플로우 엔진 어댑터, SOAP 엔진 인터페이스 등의 레거시 시스템 어댑터를 개발함으로써 유무선 연동 처리를 가능하게 한다.

##### 4.1 JCA 어댑터

최근 등장하고 있는 웹 서비스와 같이 상이한 시스템들을 연동하기 위한 다양한 방법들이 존재하며, J2EE 기반 응용서버에서는 JCA(Java Connector Architecture)를 이용하여 다양한 데이터베이스, ERP 등의 레거시 시스템과의 연동을 제공한다[6]. 본 연구에서 제안하는 모바일 응용서버에서도 JCA

를 이용함으로써 다양한 외부 시스템과의 연동을 통해 사용자의 요청을 처리한다. 그림 8은 리소스 어댑터를 이용함으로써 응용서버와 레거시 시스템과의 연동 방법을 보여준다.

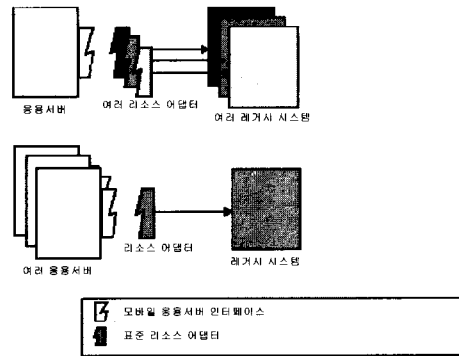


그림 8 응용서버와 레거시 시스템의 연동 방법

그림 8의 상단에는 응용서버와 서로 다른 외부 시스템들이 연동하기 위해서 각 외부 시스템들을 위한 리소스 어댑터들이 필요함을 보여준다. 리소스 어댑터는 각 리소스에 접근할 수 있는 방법과 이로부터 커넥션들을 생성하는 방법을 제공한다. 이 경우 특정 응용서버에는 n개의 리소스를 접근하기 위한 n개의 리소스 어댑터가 필요하며, 동일한 어댑터를 제 3의 애플리케이션에 적용하기 위해서는 애플리케이션에 종속된 구현 방법을 사용하여야 한다.

모바일 응용서버의 JCA는 공통 인터페이스를 사용함으로써 이러한 비효율성 문제를 해결한다. 즉, 그림 8의 하단에서 보는 바와 같이 다양한 유형의 리소스에 대해 하나의 인터페이스를 가지는 리소스 어댑터를 사용함으로써 다양한 리소스들이 유연하게 연동될 수 있다. 이를 위해서 각각의 리소스 벤더들은 JCA를 만족하는 리소스 어댑터를 제공해야 하며, 모바일 응용서버에서는 리소스 어댑터를 통해 생성된 커넥션의 생명 주기를 관리하고 이를 트랜잭션과 연관시키며 커넥션에 대한 접근 관리 등의 다양한 기능을 수행하는 모듈들을 제공한다.

이를 위해 JCA 아키텍처는 클라이언트가 각각의 리소스를 접근하기 위해서 사용하는 공통 클라이언트 인터페이스인 CCI(Common Client Interface)를 정의하고 있는 Application Contract, 리소스 어댑터와 애플리케이션 서버의 인터페이스를 정의하는

System Contract로 구성된다(그림 9).

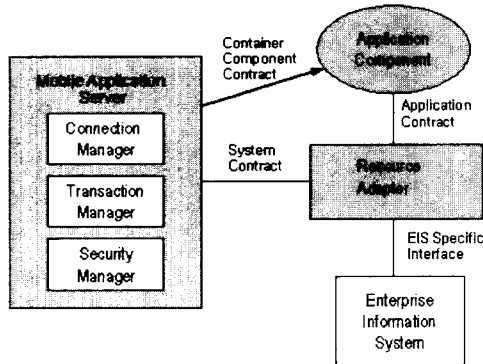


그림 9 JCA를 구성하는 인터페이스

리소스 어댑터와 레거시 시스템 사이의 인터페이스는 리소스 어댑터가 각각의 레거시 시스템에 맞도록 구현하여 제공한다. 연결 관리자는 JCA를 통해 기존의 레거시 시스템들과 연동함으로써 다양한 사용자의 유무선 요청을 처리할 수 있다. 이처럼 레거시 시스템들은 JCA 어댑터를 지원할 경우 모바일 응용서버와 쉽게 연동된다.

#### 4.2 워크플로우 어댑터

모바일 응용서버에서 제공하는 빈 컴포넌트는 재사용이 가능한 다양한 비즈니스 로직을 표현한 것이다. 빈 개발자의 의도에 따라 이러한 빈 컴포넌트는 복잡한 비즈니스 로직이 다양한 분기 조건을 가지고 하드 코딩되어 있는 경우가 많다. 하지만 이러한 업무의 경우, 일반적으로 사용자들은 자신이 가지고 있는 워크플로우를 통해 미리 정의된 업무 흐름과 분기 조건에 따라 보다 유연하게 처리하려고 한다. 모바일 응용서버에서는 이러한 외부 워크플로우 시스템을 사용하기 위한 방법을 제공한다.

모바일 응용서버에서의 워크플로우 사용 시나리오는 크게 두 가지가 있다. 첫째, 모바일 응용서버에서 특정 워크플로우 프로세스를 시작하여 빈이 그 결과를 이용하여 필요한 작업을 수행하는 것이다. 이 경우 워크플로우가 하나의 서비스로서 모바일 응용서버에 포함될 수도 있으며 단순히 어댑터를 통해 메시지만 전달하도록 구현할 수도 있다. 외부에 이미 정의되어 있는 복잡한 워크플로우 프로세스가 존재하고 모바일 컴포넌트에서는 이러한 워크플로우를

이용하면, 복잡한 워크플로우를 MSP나 MServlet 등의 컴포넌트로 재작성하지 않고 비즈니스 업무를 수행할 수 있다.

두 번째 사용 시나리오는 워크플로우에서 빈 컴포넌트를 사용하는 방법이다. 이것은 단순히 워크플로우 태스크에서 특정 빈 컴포넌트의 메소드를 호출하는 방법이다. 워크플로우 태스크가 빈 컴포넌트의 클라이언트로 동작하여 비즈니스 로직을 구현하는 빈을 사용함으로써 컴포넌트의 재사용성을 높일 수 있다.

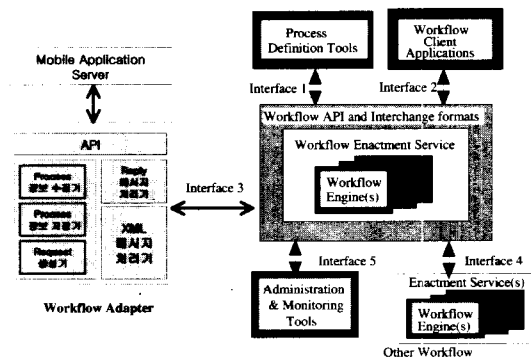


그림 10 WfMC 모델과 워크플로우 어댑터 구성도

그림 10은 WfMC[7]에서 제시하고 있는 워크플로우를 구성하는 모형과 이들 사이에 정의된 다섯 가지 유형의 인터페이스를 보여 준다. 워크플로우 어댑터는 인터페이스 3을 이용하여 워크플로우 엔진과 직접 통신하고 이를 통해 각각의 프로세스에 대한 정보를 수집하고 저장한다. 모바일 응용서버의 컴포넌트가 워크플로우 어댑터의 API를 통해 어떤 요청을 보내면 프로세스 정보 저장기에 보관되어 있는 프로세스 정보와 컴포넌트가 호출한 API를 분석하여 워크플로우에 전달하기 위한 요청을 생성하는데, 이러한 요청은 XML 메시지 처리기를 통해 워크플로우에 전달된다. 워크플로우로부터 받은 결과는 응답 메시지 처리기를 통해 요청한 컴포넌트에게 적절한 형태로 전달된다. 그림 10과는 다른 방법으로서 JCA 어댑터를 이용한 구현 방법이 있다. 이를 위해서는 각각의 워크플로우가 JCA 어댑터를 제공해야 한다는 제약 조건이 있다.

이와 같이 모바일 응용서버는 워크플로우와 효율적으로 연동함으로써 기존에 사용하고 있는 워크플로우를 변경 없이 재사용할 수 있는 구조를 제공할 수 있다.





SyncML 서버에 해당 요청을 전달한다. 또한, 데이터베이스가 변경되었을 경우 SyncML 서버는 SyncML 게이트웨이에 동기화 메시지를 전달하고, SyncML 게이트웨이는 해당 모바일 단말기에 모바일 단말기의 특성에 맞게 메시지를 변환하여 전송한다. 모바일 컴포넌트는 SyncML 게이트웨이에서 제공하는 API를 사용하며, SyncML 서버에 의존적인 부분은 SyncML 게이트웨이에 의해서 처리된다.

#### 4.5 SOAP 관리자

모바일 응용서버는 다양한 유무선 환경에서 효과적인 서비스를 제공하기 위해 HTTP, WAP, TCP/IP, JMS 메시지, SOAP(Simple Object Access Protocol) 등 다양한 프로토콜을 지원하기 위한 멀티 채널 구조를 제공해야 한다. 최근 들어 모바일 응용서버 선반 업체들은 모바일 응용서버 제품들에 다양한 프로토콜을 지원하기 위한 구조를 채택하고 있다.

모바일 응용서버에서는 다양한 프로토콜 중 SOAP 메시지에 대한 처리를 위해 그림 13과 같은 SOAP 관리자 구조를 제공한다. SOAP 관리자는 SOAP 메시지에 대한 처리 방법을 크게 두 가지 시나리오로 구분한다. 첫 번째는 웹을 통해 요청되는 SOAP 메시지로써 Tomcat에서 제공하고 있는 SOAP 메시지 처리기가 처리하며, 두 번째는 JMS 게이트웨이 및 소켓 게이트웨이를 통해 요청되는 SOAP 메시지로써 SOAP 엔진을 통해 워크플로우 제공자(Provider)와 JMS 제공자에게 전달되어 처리되는 경우이다. SOAP 관리자는 SOAP 메시지를 처리함에 있어 Tomcat을 이용하는 구조를 채택하고 있으며, 워크플로우 및 JMS 메시지 서버와 연동하기 위한 워크플로우 제공자와 JMS 제공자 인터페이스를 제공한다.

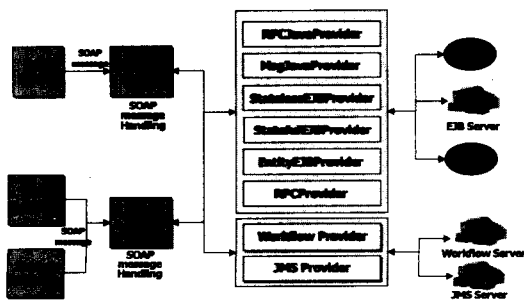


그림 13 SAOP 관리자 구성도

이러한 SOAP 메시지 제공자들은 두 가지 시나리오에 의해 클라이언트로부터 요청된 SOAP 메시지를 분석하여 워크플로우 및 JMS 메시지 서버로 전달하고 각각의 서버로부터 나온 결과를 클라이언트에게 전달하기 위하여 SOAP 메시지로 부호화하는 역할을 담당한다.

이와 같이 모바일 단말기 특성을 고려한 메시지 전송 수단으로서 플랫폼에 독립적이고 HTTP 프로토콜을 기반으로 하는 텍스트 프로토콜인 SOAP을 지원함으로써 모바일 환경에서 다양한 콘텐츠 서비스가 가능하다.

#### 5. 모바일 응용서버 클러스터링

이 절에서는 유무선 환경을 통해 전달된 대규모의 요청을 효율적으로 분산시켜 시스템 성능을 높일 수 있도록 하기 위하여 모바일 응용서버를 클러스터링 하기 위한 방법에 대해 설명한다. 그림 14는 클러스터링을 위한 구조도이다. 앞서 살펴보았던 모바일 응용서버와 함께 클러스터 관리자, 클러스터 콘솔, 그리고 배치 도구 등으로 구성된다.

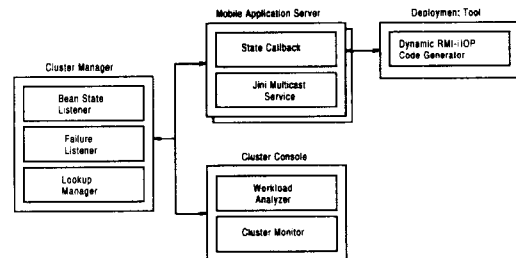


그림 14 모바일 응용서버 클러스터링 구성도

클러스터 관리자 : 클러스터 관리자는 모바일 응용서버를 클러스터링 하는 핵심 구성요소로서, 빈 상태 감시기, 장애 감시기, 룩업(lookup) 관리자로 구성된다. 빈 상태 감시기는 모바일 응용서버에 배포되어 실행되는 빈의 실행 상태 정보를 주기적으로 수집하여 연속성을 갖도록 저장한다. 장애 감시기는 클러스터를 구성하는 모바일 응용서버에서 모바일 컨테이너에 대한 장애 발생 여부를 감시한다. 장애 감시기는 모바일 응용서버의 구성 요소인 지니(Jini) 멀티캐스트 서비스를 주기적으로 작동시켜 클러스터 구성 노드인 모바일 응용 서버로 지니 멀티 캐스트 메시지를 전달한다[9]. 룩업 관리자는 서비스를 중단하

지 않고 동적으로 클러스터의 형상을 관리한다. 이를 위해 모바일 클러스터드 응용서버에서는 지니/JMX 기반의 동적 클러스터 관리 기능을 제공한다[10]

**클러스터 콘솔 :** 클러스터 콘솔은 모바일 응용서버 클러스터 시스템의 성능 감시 도구로서, 부하(Workload) 분석기와 클러스터 모니터로 구성된다. 워크로드 분석기는 각각의 모바일 응용서버의 부하 상태를 분석 가공하는 모듈로서, 배포된 빈의 수, 빈의 실행 상태, 동작중인 쓰레드 수와 같은 주로 EJB에 대한 실행 정보를 분석한다. 클러스터 모니터는 부하 분석기와 상호 연동하여 모바일 응용서버의 부하 정보를 다양한 형태의 그래프로 가시화 한다.

**배포 도구 :** 이 도구는 모바일 응용서버 애플리케이션 코드를 생성하는 도구로서, 클러스터 구조에 적합한 응용 코드를 생성하기 위한 동적 RMI-IIOP 코드 생성기를 포함한다. 동적 RMI-IIOP 생성기는 모바일 응용서버와 클라이언트 간의 통신 메커니즘인 RMI-IIOP 규약을 준수하며, 특히 클러스터 구조의 모바일 응용서버 시스템에 대한 정보를 담은 지능형 코드를 생성한다[13]. 예를 들어, 클라이언트가 지능형 코드를 통해 임의의 모바일 응용서버에 배포된 빈을 호출할 때 호출을 받은 빈에 장애가 발생하면, 메소드 호출 수준에서 이 빈과 동일한 다른 응용 서버에 배포된 빈에게 서비스를 전달하는 장애 탐지 및 복구가 이루어진다.

**모바일 응용서버 :** 모바일 응용서버는 클러스터를 구성하는 단위 노드로서 클러스터 기능과 관련하여 빈 상태 콜백과 지니(Jini) 멀티 캐스트 서비스를 포함한다. 빈 상태 콜백은 클러스터 관리자의 빈 상태 수집기와 연동하여 모바일 응용서버에서 실행 중인 빈의 실행 시간 상태 정보를 수집하여 빈 상태 수집기에 전달한다. 지니 멀티 캐스트 서비스는 클러스터 관리자의 장애 탐지기의 제어에 따라 모바일 응용서버 간에 장애 발생 여부를 확인하기 위해 지니 멀티 캐스트 메시지를 발생한다.

## 6. 결론

본 논문에서는 모바일 환경에서 인터넷 서비스를 제공할 경우의 고려 사항을 살펴보고, 이를 해결하기 위한 응용서버로서 유무선 통합 모바일 응용서버를 제안하였다. 또한 기존의 시스템과 연동함으로써 시스템 확장 방법에 대해서도 살펴보았다.

모바일 환경에서의 제약 사항을 해결하는 방향은 크게 다음과 같이 두 가지로 설명할 수 있다. 첫째는 모바일 단말기를 이용한 통신과 관련된 제약 사항을 해결하는 것이고, 둘째는 다양한 모바일 단말기를 이용할 경우 나타날 수 있는 단말기의 데이터 표현 및 관리와 관련된 문제를 해결하는 것이다. 제안한 모바일 응용서버는 네트워크 통신 환경 및 모바일 단말기의 다양성에 적용할 수 있다. 통신과 관련된 제약 사항으로서 모바일 환경의 특성인 간헐적 단절성을 극복하고 통신 환경 및 단말기 자체의 제약을 극복함으로써 안정적으로 세션을 유지할 수 있다. 또한 단말기 콘텐츠 표현 및 관리 기술을 이용함으로써 다양한 유형의 단말기를 수용할 수 있고 사용자에게 따른 특성을 반영할 수 있다.

제안한 모바일 응용서버는 적응성을 가짐과 동시에 고 가용성과 확장성을 갖는다. 즉, 한번의 저작으로 유무선 동시 서비스를 가능하게 하며, 클러스터 구조의 모바일 컨테이너를 기반으로 다양한 연동 기술을 이용하여 기존의 레거시 시스템과 연동함으로써 자원을 효율적으로 이용할 수 있다. 이는 SMS와 JMS, SyncML, 워크플로우 등을 공통적으로 이용할 수 있는 공통 인터페이스를 제공함으로써 가능하다.

다양한 서비스를 지원하는 공통 프레임워크로서 유무선 통합 모바일 응용서버를 이용할 수 있다. 그럼으로써 대규모 유무선 인터넷 비즈니스 업무를 빠르고 안정적으로 처리하고자 하는 애플리케이션 소프트웨어 분야의 미들웨어 개발을 위한 기반 기술로 활용할 수 있다.

## 참고문헌

- [1] <http://www.ovum.com/go/product/flyer/MEC.htm>,
- [2] <http://www.metagroup.com/cgi-bin/inetcgi/jsp/home.do>
- [3] Geoff Johnson, "m-Commerce Scenario," Gartner Group, 2001.
- [4] Jeffrey M Capone, "Extending J2EE for Mobile Application Development," <http://www.oreillynet.com/pub/a/onjava/2001/10/17/mobilej2ee.html>, 2001.
- [5] [www.w3c.org](http://www.w3c.org)
- [6] Rahul Sharma, "J2EE Connector Architecture

Specification," SUN Microsystems, 2000.

[7] <http://www.wfmc.org>

[8] SyncML Initiative, "Building an Industry-Wide Mobile Data Synchronization Protocol,"

SyncML White Paper, 2000.

[9] <http://java.sun.com/Jini>

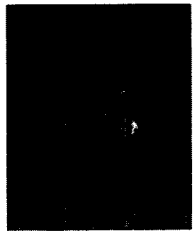
[10] <http://java.sun.com/jm>

### 김 성 훈



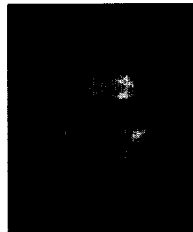
1995 광운대학교 전자공학과 졸업(학사)  
 1997 광운대학교 전자공학과 졸업(석사)  
 1996~1998 시스템공학연구소 연구원  
 1998~현재 한국전자통신연구원 컴퓨터 소프트웨어연구소 모바일응용서버 연구팀 선임연구원  
 관심분야: 모바일응용서버, 시스템소프트웨어, 미들웨어 등  
 E-mail: saint@etri.re.kr

### 장 철 수



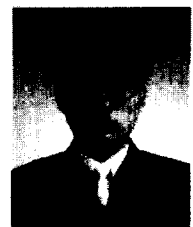
1995 인하대학교 전자계산공학과 졸업(학사)  
 1997 광주과학기술원 정보통신공학과 졸업(석사)  
 1997~1998 시스템공학연구소 연구원  
 1998~현재 한국전자통신연구원 연구원  
 관심분야: 모바일응용서버, 웹기술, 데이터베이스 등  
 E-mail: jangcs@etri.re.kr

### 정 승 욱



1996 전남대학교 전산학과 졸업(학사)  
 1998 광주과학기술원 정보통신공학과 졸업(석사)  
 1998~현재 한국전자통신연구원 컴퓨터 소프트웨어연구소 모바일응용서버 연구팀 연구원  
 관심분야: 웹 응용 서버, 모바일 응용 서버, 미들웨어 등  
 E-mail: swjung@etri.re.kr

### 서 범 수



1996 서울시립대 전산통계학과 졸업(학사)  
 1998 서울시립대 전산통계학과 졸업(석사)  
 1998~현재 한국전자통신연구원 컴퓨터 소프트웨어연구소  
 관심분야: 전자상거래 미들웨어, 유무선 응용서버, 메시지소모바일응용서버 연구팀 연구원 미들웨어, 워크플로우시스템

E-mail: bsseo@etri.re.kr

### 노 명 찬



1996 방송통신대학교 전자계산학과 졸업(학사)  
 1991~현재 한국전자통신연구원 컴퓨터 소프트웨어연구소 모바일응용서버 연구팀 선임기술원  
 관심분야: J2B서버, 모바일환경적응기술 등  
 E-mail: mcroh@etri.re.kr

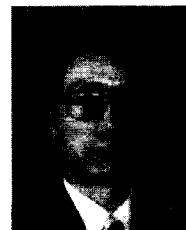
### 박 중 기



1993 한남대학교 공과대학 컴퓨터공학과 졸업(공학사)  
 1995 충남대학교 자연과학대학 컴퓨터학과 졸업(이학석사)  
 2001~현재 충남대학교 자연과학대학 컴퓨터학과 박사과정  
 1995~현재 한국전자통신연구원 컴퓨터 소프트웨어연구소 모바일응용서버 연구팀 선임연구원  
 관심분야: SyncML, 모바일 데이터베이스, 클러스터 컴퓨팅

E-mail: jkp@etri.re.kr

### 이 경 호



1997 고려대학교 산업공학과 졸업(공학사)  
 1990 고려대학교 산업공학과 졸업(공학석사)  
 1990~현재 한국전자통신연구원 컴퓨터 소프트웨어연구소 모바일응용서버 연구팀 책임연구원  
 관심분야: e-Business, 모바일 응용서버

### 김 중 배



1996 고려대학교 공과대학 산업공학과 졸업(공학사)  
 1998 한국과학기술원 산업공학과 졸업(공학석사)  
 1998~현재 한국과학기술원 산업공학과 박사과정  
 1988~1991년 대한항공(주) 시스템부  
 1991~현재 한국전자통신연구원 컴퓨터 소프트웨어연구소 모바일응용서버연구팀장

관심분야: 미들웨어, 시스템 소프트웨어, 인터넷 정보검색 등  
 E-mail: jikim@etri.re.kr