

다중 장치 지원 모바일 응용서버 설계

정은경* · 변영철** · 이상준**

The Design of a Mobile Application Server for Various Mobile Devices

Eun-Gyoung Joung * · Yung-Cheol Byun ** · Sang-Joon Lee **

ABSTRACT

In this paper, we propose the design of the application server which provides internet services for the mobile devices such as a mobile phone and PDA, etc. The neutral form of mobile application which can be requested by various kinds of mobile devices is created and deployed to the proposed application server. The mobile application server supplies the content transformed according to the constraints and features of a mobile device. By using the server, a user can use many kinds of mobile devices at the same time, and only one session can be created and maintained at the server. The mobile application server is composed of the tree modules: MobileRequestGenerator, MobileEngine, and EnvironmentalAdoptation module. The Mobile RequestGenerator module receives the mobile request from a mobile device and creates the mobile request information supplied to the MobileEngine module. The MobileEngine module processes the mobile application requested and creates a content which will be sent to a mobile device. The EnvironmentalAdoptation module transforms the output of MobileEngine into the contents that a mobile device can display. In this way, the proposed mobile application server can support various kinds of mobile devices using a neutral form of internet service.

Key words : Mobile Application Server, Various Kinds of Mobile Devices, Internet Service

1. 서론

컴퓨터 성능 향상과 인터넷에 접속할 수 있는 네트

워크 환경이 개선됨에 따라 PC를 이용한 인터넷 서비스 사용이 크게 증가하고 있다. 특히 범용 브라우저를 이용한 웹 응용서버 기반의 인터넷 서비스 사용이 증가하는 추세이다. 한편 아울러, 이동통신 기술의 발전에 따라 셀룰러 폰, PDA, Post PC와 같은 다양한 대역폭을 갖는 모바일 단말기들이 광범위하게 보급되어 활용되고 있다. IT 비즈니스 연구 및 컨설팅 회사인 META 그룹에 의하면 2003년까지 인터넷 접근의 50%가 PC가 아닌 휴대용 단말기를 이용하여

* 제주대학교 대학원

Graduated School, Cheju Nat'l Univ.

** 제주대학교 통신·컴퓨터공학부

Faculty of Telecommunication & Computer Eng., Cheju Nat'l Univ.

접속할 것으로 예상되며, 2005년까지 전 세계적으로 10억의 사용자가 모바일 단말기를 이용하여 인터넷에 접속할 것으로 예상된다[1].

모바일 단말기의 보급과 이를 이용한 인터넷 서비스의 증가로 인하여 개인용 컴퓨터로 인터넷 서비스를 이용하는 환경에서 다양한 유형의 모바일 단말기로 인터넷 서비스를 이용하는 모바일 인터넷 서비스 이용 환경으로 바뀌고 있는 것이다[2]. 이처럼 모바일 단말기를 이용한 서비스에 대한 요구가 증가함에 따라 특정 단말기 위주 및 특정 기능 위주의 모바일 지원 기술이 개발되고 있다. 하지만 무선 인터넷 서비스를 효과적으로 제공하려면 기존의 다양한 모바일 단말기를 공통적으로 지원할 수 있음은 물론, 다양한 단말기를 이용하여 동일 세션을 수행할 수 있는 다중 장치 지원 모바일 응용서버에 관한 연구가 필요하다.

다음 장에서는 관련 연구 및 현황에 대해 설명하고 3장에서는 본 연구에서 제안하는 방법의 특징에 대해 설명한다. 그리고 4장에서는 다중 장치 지원 모바일 응용서버의 설계 및 구현 방법에 대해 설명하며, 마지막으로 5장에서는 본 연구에 대한 결론을 맺는다.

II. 관련 연구 및 현황

국내의 관련 연구의 경우, 여러 가지 유형의 모바일 단말기에 적합한 콘텐츠로 변환하는 콘텐츠 변환 기술, 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 개별적으로 개발되고 있다[3]. 이러한 기술들은 특정 기능의 구현을 위주로 하고 있기 때문에 독립된 서버 기능이 미흡할 뿐만 아니라 모바일 비즈니스 서비스를 위한 미들웨어 기능이 통합적으로 제공되고 있지 않다.

IBM, Oracle 등은 기존의 유선 응용서버에 콘텐츠 변환 기술과 같은 일부 모바일 기능을 추가하고 있으며, Brience[4]는 웹 응용 디자이너의 지식을 모형화하여 기존의 유선 응용을 다양한 단말기로 서비

스한다. M1[5]은 모바일 단말기용으로 작성된 응용을 다양한 단말기로 서비스하는데, 아파치 톰캣 서버에 의존함으로써 구현 방법이 용이하다는 장점이 있으나 독자적인 클러스터링이 불가능하고 톰캣 서버의 기타 정보를 효과적으로 사용하기 어렵다는 단점이 있다. 또한 이러한 기존의 시스템들은 다양한 모바일 단말기를 지원하지만 여러 단말기에 의해 동일한 세션을 지원하는 방법을 제공하지 않아 모바일 서버를 효과적으로 지원하기 어렵다.

III. 제안하는 방법의 특징

제안하는 방법은 기존의 JSP 혹은 서블릿(Servlet)에 단말기에 따른 콘텐츠 변환을 자동으로 제공함은 물론 다중 단말기에 의해 동일 세션을 수행할 수 있는 기능을 제공함으로써 보다 편리한 인터넷 서비스 개발 환경을 제공한다. 즉, 기존의 JSP 또는 서블릿을 작성하듯이 작성하기만 하면 손쉽게 다양한 모바일 단말기로 멀티 모달 세션 기능을 사용할 수 있으며, 모바일 환경과 관련된 복잡한 사항들에 대해 신경쓰지 않아도 된다. 본 방법의 특징은 크게 다음과 같이 3가지로 설명할 수 있다.

3.1 단말기에 독립적인 콘텐츠 뷰

본 연구에서 제안하는 방법의 특징 중 하나는 중간 형태의 언어로 콘텐츠 혹은 웹 응용을 저작함으로써 PC 뿐만 아니라 핸드폰, PDA, Post PC 등을 포함하는 다양한 모바일 단말기로 볼 수 있다는 것이다. 다시 말해, Fig. 1과 같이 응용 개발자가 단말기의 유형 및 기능에 독립적인 응용, 혹은 뷰(View)를 개발하면 모바일 응용서버는 요청을 보낸 단말기 형태에 맞도록 모바일 응용을 변환하여 보여준다. 이 경우 뷰는 비즈니스 로직을 이용하며, 레거시 어댑터를 통해 레거시 장치를 액세스 한다.

이러한 방법의 장점은 웹 응용의 개발 및 유지 보수 비용을 획기적으로 줄일 수 있다는 것이다. 즉, 특정 단말기에 의존적인 웹 응용을 단말기에 따라 다르게 작성할 경우 동일한 기능을 중복적으로 개발함으

로써 드는 비용과 특정 웹 페이지가 변경될 경우 다른 단말기용 웹 페이지를 변경함으로써 생기는 유지 보수 비용을 줄일 수 있다.

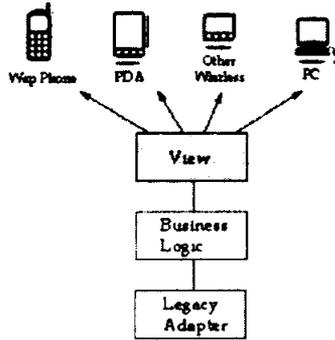


Fig.1. View of a web application and various kinds of mobile devices.

3.2 멀티 모달 세션 지원

모바일 단말기에 따라 웹 응용을 다르게 개발할 경우 특정 단말기로 세션을 수행하다가 다른 단말기로 앞서 수행하던 세션을 계속해서 수행하는 것은 원칙적으로 불가능하다. 하지만 PC, 핸드폰, PDA 등 동일한 여러 단말기를 사용하는 환경에서 보다 나은 인터넷 서비스를 지원하려면 PC를 포함하여 다양한 단말기로 동일한 세션을 수행할 수 있는 방법을 제공할 수 있어야 한다[6]. 따라서 다중 장치 지원 모바일 응용서버는 특정 단말기로 수행 중이던 세션을 다른 단말기로 계속해서 세션을 수행하도록 함으로써 보다 편리한 인터넷 서비스를 지원할 수 있다.

3.3 확장성을 갖는 구조

제안하는 응용서버는 5계층 구조에 기반한 독자적 요청 처리 구조를 갖는다. 독자적으로 모바일 빈의 생명 주기를 관리하면서 요청을 처리하기 때문에 대용량 처리를 위한 클러스터링 방법 구현시 시스템의 각종 정보를 효과적으로 이용할 수 있으며, 기존의 레거시 시스템과 효과적으로 구성할 수 있음은 물론 다양한 형태의 연결이 가능하다.



Fig. 2. 5-layered structure of a mobile application server.

Fig. 2는 본 연구에서 제안하는 5계층 모바일 응용서버의 구조이다. 뷰(Real View) 계층이란 실제로 단말기에 표시되는 콘텐츠 및 이를 생성하기 위한 모듈로 구성되는 계층이다. 가상 뷰(Virtual View) 계층은 모바일 응용서버에 배포되는 모바일 응용과 이를 구동시키는 뷰 핸들러(View Handler)로 구성된다. 컨트롤(Control) 계층은 모바일 요청을 받아 처리하는 모바일 엔진 및 기타 환경 적용 모듈들로 구성되어 있으며, 어댑터(Adapter) 계층은 레거시 시스템과의 연동을 위한 인터페이스로 구성된다. 모바일 응용서버의 계층 구조는 각 계층별 모듈성을 높인다.

IV. 설계 및 구현

4.1 전체 구조

다중 장치 지원 모바일 응용서버는 크게 네 부분으로 구성된다. 우선 (1)웹 서버로부터 HTTP 프로토콜로 요청(request)을 전달받아 모바일 요청으로 변환하는 모바일 요청 생성부(Mobile Request Generator)와 (2)모바일 요청을 해석하여 해당 모바일 빈(bean)을 찾아 실행한 후 클라이언트로 넘겨줌으로써 모바일 컨테이너의 전반적인 기능을 담당하는 모바일 엔진부가 있다. 그리고 (3)모바일 단말기의 제약 극복하고 모바일 단말기에 따른 특화된 서비스를 제공하기 위한 모바일 환경 적응부와, 마지막으로, (4)다양한 레거시 서비스를 지원하기 위한 유무선 연동 처리부가 있다. Fig. 3은 모바일 응용서버를 크게 네 부

분으로 나는 모습이다.

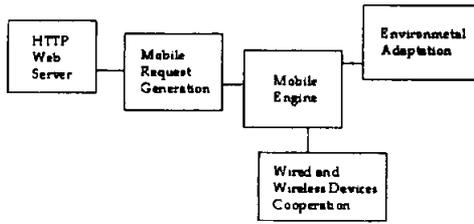


Fig. 3. The core modules of the proposed system.

4.2 모바일 요청 생성부

모바일 요청 생성부는 웹 서버로부터 AJP 프로토콜로 HTTP 요청을 받아 모바일 요청인 Mobile Request를 생성하는 모듈이다. AJP 프로토콜은 HTTP 아파치 웹 서버와 톰캣 웹 서버 연동에 사용되는 AJP Connector인 mod_jk.dll 아파치 plug-in 모듈에 의해 구현된다. 본 연구에서는 이를 HTTP 아파치 웹 서버와 모바일 응용서버를 연동하기 위한 모듈로서 사용하였다. 즉, HTTP 아파치 웹 서버로부터 온 JSP 혹은 Servlet 요청 중 모바일 관련 요청은 AJP 프로토콜로 8009 포트를 이용하여 모바일 응용서버로 전송된다.

모바일 요청 생성부에 의해 생성된 MobileRequest는 모바일 응용에게 모바일 단말기의 클라이언트에 대한 정보, 요청에 대한 파라미터, 요청을 통하여 전달된 헤더 등을 액세스할 수 있는 인터페이스를 제공한다. 모바일 요청 생성부는 Fig. 4와 같다. AJP 프로토콜에 의해 8009 포트로 요청이 들어 올 경우 RequestReceiver 모듈은 ServerSocket 모듈을 이용하여 소켓(Socket)을 요청 큐인 RequestQueue 모듈에 저장한다. 이때 스레드 풀에서 실행 중인 스레드 중 하나인 RequestProcessor 모듈은 RequestQueue에 저장된 소켓을 이용하여 AJP 프로토콜로 전송되는 클라이언트 요청으로부터 MobileRequest 요청을 생성한다. 스레드 풀에서 수행 중인 RequestProcessor 스레드의 초기 개수 설정과 이후 동적으로 필요에 따라 스레드를 생성하고 삭제하기 위한 모듈로서 Thread Monitor 모듈이 있다.

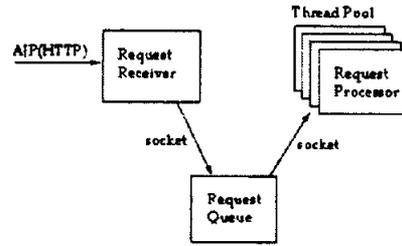


Fig. 4. The structure of mobile request creator module

4.3 모바일 엔진

모바일 엔진은 모바일 요청 생성부가 생성한 모바일 요청인 MobileRequest를 전달받아 Fig. 5에 나타난 환경 적응부의 관련 모듈들을 이용하여 요청을 보낸 모바일 단말기에 적합한 표현으로 변환하여 클라이언트로 전송한다. 우선 모바일 엔진은 프로파일 관리자인 ProfileManager 모듈을 이용하여 모바일 요청을 보낸 모바일 단말기의 MobileRequest 헤더 정보를 파싱하여 사용자 정보를 알아낸다. 예를 들어, 핸드폰을 이용하여 응용을 요청한 경우 ProfileManager 모듈로 MobileRequest 객체를 전송하여 관련 전화번호 및 가상 쿠키 사용 여부 등을 알아낸다.

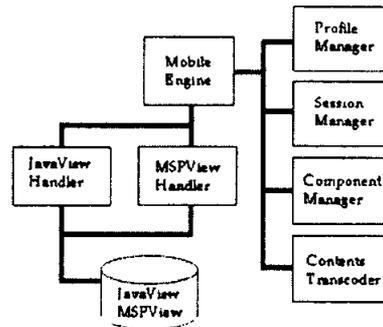


Fig. 5. The mobile engine and an environmental adaptation module

그리고 컴포넌트 관리자인 ComponentManager 모듈을 이용하여 클라이언트가 전송한 요청을 파싱하여 모바일 응용에 관한 정보를 얻어내고 모바일 페이지(빈)에 관한 정보(BeanContext)를 MobileRequest에

저장함으로써 이후 다른 모듈에서 사용할 수 있도록 한다. 또한 모바일 엔진은 세션 관리자 모듈인 SessionManager를 이용하여 요청에 있을 수 있는 세션 ID를 파악한 후 모바일 요청인 MobileRequest와 모바일 응답인 MobileResponse에 설정한다.

4.4 환경 적응부

세션 관리자(SessionManager)

세션 관리자는 크게 두 가지 경우에 의해 사용된다. 첫번째는 클라이언트가 요청을 전송할 경우이고, 두번째는 모바일 응용에서 세션을 요구할 경우이다. 전자의 경우 세션 관리자는 모바일 요청에 있을 수 있는 세션 ID를 찾아 이 후 사용할 수 있도록 요청과 응답에 세션 및 세션 ID를 설정하며, 후자의 경우에는 요청에 저장되어 있는 모바일 빈 컨텍스트로부터 모바일 응용 이름을 알아내어 해당 응용에 대해 세션을 관리하는 세션 핸들러인 SessionHandler 모듈을 이용하여 세션 ID 및 세션을 생성한다. SessionHandler 모듈은 특정 응용 단위의 세션 관리를 위한 모듈로서 StorageController라는 모듈을 이용하여 이미 메모리에 로드되어 있는 세션이 있을 경우 이를 재사용함으로써 성능을 향상시킬 수 있다.

클라이언트 요청에 대해 세션 관리자 모듈인 SessionManager는 세션 ID를 파악하고 설정하기 위하여 URL 파싱, 쿠키 파싱, 그리고 버추얼 쿠키 파싱 방법을 이용한다. 우선 세션 관리자는 URL에서 세션 ID를 파악하고 MobileRequest에 저장되어 있는 쿠키에서 세션 ID를 파악한다. 만일 어떠한 세션 ID도 찾지 못한다면 버추얼 쿠키에서 세션 ID를 파악한다. 만일 세션 ID가 있으면 요청에 세션 ID와 세션을 설정한다. 세션 ID를 쿠키에서 얻을 경우 모바일 응답인 MobileResponse에 세션 ID를 가지고 있는 쿠키를 설정한다.

컴포넌트 관리자(ComponentManager)

컴포넌트 관리자는 모바일 응용을 구성하는 각각의 모바일 컴포넌트, 혹은 모바일 빈의 생명 주기를 관리한다[7-10]. 즉, 시스템 구동 초기에 각각의 모바일 빈에 대해 설정되어 있는 수의 인스턴스를 생성하고

요청이 증가함에 따라 메모리에 로드되는 인스턴스의 수를 증가시킨다. 또한 요청이 감소함에 따라 인스턴스가 불필요하게 많이 생성되어 있을 경우 성능 향상을 위하여 인스턴스의 수를 감소시킨다.

컴포넌트 관리자는 응용 별로 응용 내의 각 빈의 생명 주기를 관리한다. 이렇듯 응용 위주로 관리함으로써 응용 단위의 컨텍스트의 관리가 용이하다. 가령 응용 단위로 멀티 모달을 지원할 것인지 여부를 설정할 수 있다. 또한 모바일 응용 개발자는 응용의 우선 순위(priority) 및 세션 타임 아웃 등과 같이 응용 단위로 쉽게 설정할 수 있다.

프로파일 관리자(ProfileManager)

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

사용자 단말기의 특성 정보는 단말기에서 요청이 서버로 접수되는 시점에서 헤더 정보인 UAProf(User Agent Profile)를 분석하여 저장되며, 시스템 관리자가 프로파일 편집 도구를 이용하여 갱신할 수 있다. 프로파일 레퍼지토리에 저장된 단말기 정보는 콘텐츠 변환기에 의해 사용자 단말기에 가장 적합한 형태의 콘텐츠가 생성된다.

문서 변환기(ContentsTranscoder)

특정 모바일 단말기가 모바일 빈을 요청할 경우 모바일 엔진은 가장 마지막 단계에서 MobileRequest 객체에 설정되어 있는 단말기 정보와 Mobile Response 객체에 저장되어 있는 XHTML을 입력받아 변환한 후 그 결과를 MobileResponse 객체에 저장한다.

문서 변환기의 주요 모듈로서 특정 콘텐츠만을 골라 문서를 만드는 AnnotationProcessor가 있는데, 이 모듈은 편집기 도구를 이용하여 미리 정의한 클리핑 정보와 단말기 유형에 따른 중요도 등을 표시한 어노테이션 정보, 그리고 장치 프로파일에 있는 장치 정보를 이용한다. 또 다른 모듈인 ProfileCollector는 장치 프로파일 이외에 AnnotationProcessor가 생성한 문서로부터 문서 프로파일 정보를 추출한다. 마지막

으로, ProfileAggregator 모듈은 장치 프로파일과 문서 프로파일에 있는 속성값을 이용하여 어떤 변환을 수행할 지를 결정한다. 이때 적용되는 변환에는 태그 필터링(tag filtering), 태그 변환(tag conversion), 그리고 모바일 단말기의 특성을 고려하여 문서를 여러 조각으로 자르는 문서 분할(fragmentation)이 있다.

V. 결론

본 논문에서는 다양한 모바일 단말기의 유형에 독립적인 중간 형태의 인터넷 서비스를 지원할 수 있는 다중 장치 지원 모바일 응용서버와, 다양한 모바일 단말기를 이용하여 동일한 세션을 수행할 수 있는 방법을 제안하였다. 인터넷 서비스 요청을 보낸 모바일 단말기에 따라 서로 다른 콘텐츠를 생성하기 위한 프로파일 관리자와 문서 변환기, 단말기가 바뀔지라도 동일 세션을 계속해서 수행하기 위한 세션 관리자 등을 포함한 환경 적용 모듈들에 대해 설명하였다. 이러한 방법을 이용함으로써 응용 개발 및 유지 보수에 있어서 비용을 줄일 수 있을 뿐만 아니라 멀티 모달 세션에 의한 다양한 응용을 지원함으로써 모바일 환경에서 보다 편리하고 효과적인 방법으로 인터넷 서비스를 제공할 수 있다.

참고 문헌

- 1) Peter Lowber, 2000, Wireless application gateways: A technology perspective, Gartner Technology, Overview.
- 2) Peter Lowber, 2002., Wireless application gateways, Gartner Technology Overview.
- 3) 오동익, 이종섭, 이경호, 김중배, 2002., 모바일 응용 서버의 구조에 관한 연구, 한국멀티미디어학회지, 제6권 1호.
- 4) <http://www.brience.com>
- 5) <http://www.aligo.com>
- 6) 김성훈, 잠철수, 정승욱, 서범수, 노명찬, 박중기, 이경호, 김중배, 2002, 유무선 통합 모바일 응용서

버에 관한 연구, 정보과학회지, 제20권 1호.

- 7) <http://www.ovum.co/go/product/flyer/MECtm>
- 8) <http://www.metagroup.com/cgi-bin/inetcgi/jsp/home.do>
- 9) Geoff Johnson, 2001.m-Commerce Scenario, Gartner Group.
- 10) Jeffrey M Capone, 2001, Extending J2EE for mobile application development. <http://www.oreillynet.com/pub/a/onjava/2001/10/17/mobilej2ee.html>.
- 11) www.w3c.org
- 12) Rahul Sharma, 2000, J2EE Connector architecture specification, SUN Microsystems.
- 13) <http://www.wfmc.org>
- 14) SyncML Initiative, 2000, Building an Industry-wide mobile data synchronization protocol, SyncML White Paper.
- 15) <http://java.sun.com/Jini>.
- 16) <http://java.sun.com/jmx>.