

J2ME기반 모바일 서버 연동 기법 연구

김 미* 이 양원**
*호남대학교 정보통신공학과

A Study On the Interface Techniques for the Mobile Server based on J2ME

Mi Kim, Yang-Weon Lee

E-mail : kimm@honam.ac.kr* ywlee@honam.ac.kr **

Dept. of Information & Communication Engineering, Honam University

요 약

인터넷의 접속환경이 유선에서 무선으로 급격하게 진행되고 있지만 무선에서 사용되는 인터넷 지원 환경은 Text 기반의 WAP 서비스가 주종을 이루고 있어서 멀티미디어 서비스와 같은 다양한 잇점을 제공하는 새로운 프로그램 개발방식과 구조가 필요한 실정이다.

본 논문에서는 Mobile 기기와 웹 서버간의 멀티미디어 및 다양한 컨텐츠 제공이 가능한 J2ME 기반 모바일 서버 연동 기법을 제안하였다.

본 연구에서 제안한 기법은 현재 많은 무선인터넷 서비스 업체에서 적용하고 있는 VM(Virtual Machine)에 적용시 예상되는 서버 연동 문제를 해결할 수 있음을 확인하였다.

1. 서 론

무선 분야는 지난 1, 2년 사이에 갑자기 활성화된 산업분야로서 가장 폭넓고 미래에 영향력 있는 변화를 주도할 분야로 인정받고 있다. 각종 통계와 지표에서 미래 무선 통신 분야의 발전이 어떠하리라는 예측을 보아왔을 텐데 하나같이 그 그림들이 지수적인 성장을 예고하는 것들이다. 그러나 실제로 무선분야에서 과거와 같이 폐쇄적으로 모든 Application을 하드웨어 회사 (단말기

제조사)들이 독점하는 것이나 무선인터넷을 주도하고 있는 Text 위주의 WAP서비스 만으로는 지속적인 성장을 이끌 수 없으며 멀티미디어 서비스등의 다양한 잇점을 제공하는 새로운 프로그램의 지원을 요구하는 실정이다.

본 연구에서 제안하는 기법은 멀티미디어 서비스 지원이 가능한 현재 많은 무선인터넷 서비스업체에서 적용하고 있는 VM(Virtual Machine)에 적용시 서버 연동 문제를 해결할 수 있음을 확인하였다. 본 연구에서 제안

하는 기법은 J2ME 기반의 모바일 서버환경 구축을 위해 웹서버와 서블릿 엔진으로 Tomcat, 그리고 데이터베이스 연동을 위해서 JSP를 사용하였다.

문제 등이 있다. [그림 2]는 J2ME Platform 의 CDC(Connected Device Configuration)와 CLDC(Connected Limited Device Configuration) 간의 차이를 나타낸 그림이다.

2. 모바일 자바용 J2ME

2.1 Java 표준 사양

자바의 환경은 J2SE, J2EE, J2ME 등 3가지 규격으로 구분되어져 있다. 이렇게 분류된 자바 플랫폼을 살펴보면 우리가 흔히 일반 PC 환경에서 접하는 J2SE(Java2 Standard Edition)와 서버환경을 위한 J2EE(Java2 Enterprise Edition), 그리고 본 논문에서 다루는 모바일환경용 J2ME(Java2 Micro Edition)로 분류된다. [그림 1]은 분류된 Java Platform 관련 기기들의 예를 보인 것이다.

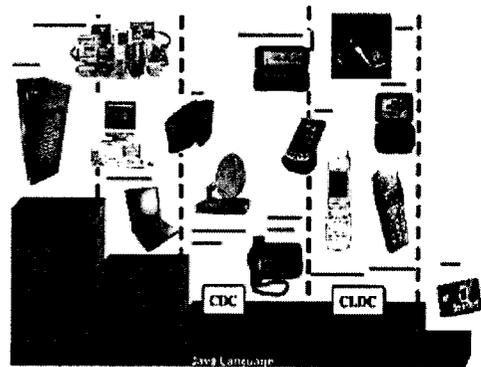


그림 3. J2ME 기반 CDC와 CLDC의 차이

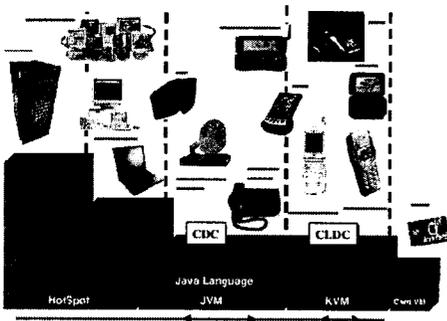


그림 2 Java Platform

2.2 J2ME(Java2 Micro Edition)

J2ME는 임베디드 시장을 위한 기술, 즉 비교적 작은 규모의 환경을 겨냥해서 나온 자바 플랫폼이다.

J2ME의 특징은 동적인 애플리케이션 다운로드, 플랫폼 호환성에 대한 강력함을 제공하고, 코드의 이식성, 익숙한 Java 프로그램 언어를 사용, 비연결성(통신비용절감), 보안

2.3 MIDP 개요

컨피그레이션 임베디드 프로그래밍을 위한 최소한의 API를 제공한다. 즉 임베디드 프로그램이 돌아가기 위한 기본 환경은 구축되었지만 세련된 프로그램을 위한 응용 개발적인 기능들은 제공하지 못한다. 이러한 점을 극복하기 위해 나온 것이 바로 프로파일이며 그중 하나가 MIDP(Mobile Information Device Profile)이다. [그림 3]은 Mobile Phone과 단말기 상에서 실제 MIDP와 CLDC가 어떤 위치에서 돌아가는지를 보여주는 그림이다.

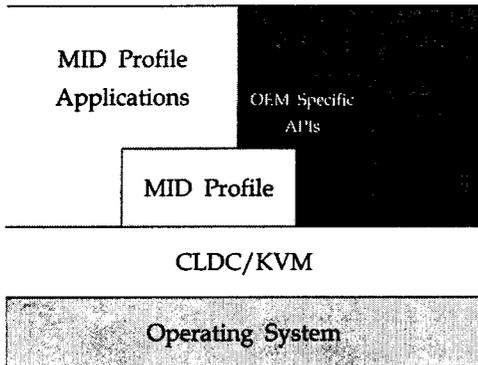


그림 3. CLDC 상의 MIDP

3. 서버 연동 기법

3.1 서버 환경 구축

서버 환경 구축을 위해 웹 서버와 서블릿 엔진 그리고 데이터베이스가 필요하며 웹 서버로는 Apache 웹 서버를 이용하였다. 웹 서버 환경 구축을 위해 필요한 서블릿 엔진으로는 Tomcat, 그리고 데이터베이스로는 MySQL을 이용하였다.

3.2 GET방식을 이용한 서버연동

서버 연동을 위한 예제로 GET방식을 이용하여 서블릿을 호출하고 JSP 연동을 통하여 id와 password를 입력하고 입력된 값들을 서버상으로 전송하여 서버의 데이터베이스내의 id와 password를 검색해서 로그인 처리를 하는 프로그램을 통해서 모바일 서버연동을 확인하였다. [그림 4]는 서버연동을 위한 시스템 구성도이다. 먼저 아파치 웹서버를 구동한 후 MySQL DB와 서블릿 엔진인 Jakarta-Tomcat을 구동한 후 예제파일을 실행 하였다.

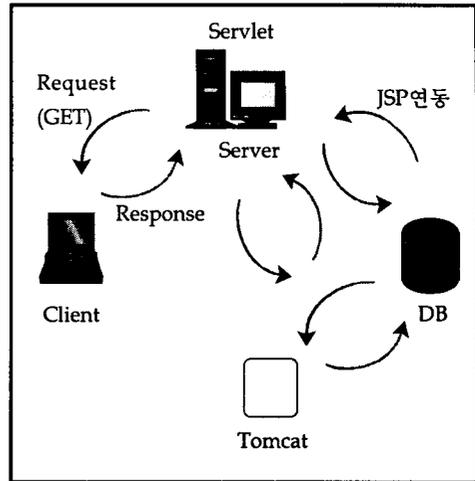


그림 4 서버 연동 시스템 구성도

3.3 JSP를 이용한 인증처리

[표 1]은 모바일 디바이스에서 id와 password를 입력한 후 로그인 처리를 위하여 JSP로 연동이 되는 파일구조이다.

Client	· idCheck 클라이언트.java	MIDP program
Server	idCheck서버.jsp	JSP bean 파일
	DB커넥션.java	JSP bean 파일
	ID_체크.java	JSP bean 파일

[그림 5]는 Client프로그램으로는 id값과 password를 받아서 서버로 전송하고 서버의 결과를 받는 프로그램으로 GET Method를 이용한 예이다. 이를 위해 데이터베이스 Table을 생성하며 [그림 6]은 데이터베이스 커넥션을 위한 beans 파일 예이다 또한 [그림 7]은 서버 측 서블릿으로 ID를 Check 부분이며 [그림 8]은 서버 연동후 실행 결과 화면 이다.

```

try {
    hc=(HttpConnection)Connector.open(target);
    hc.setRequestMethod(HttpConnection.GET);
    hc.setRequestProperty("User-Agent",
        "Profile/MIDP-1.0 Configuration/CLDC-1.0");
    hc.setRequestProperty("Content-Language",
        "eu-KR"); // Get Method 사용.
    is = hc.openInputStream();
        int ch;
        while((ch=is.read()) != -1) {
            sb.append((char)ch);
        }
    tb = new TextBox("Result Servlet ",
        sb.toString(), 1024, 0);
    tb.addCommand(CommandBack);
    tb.setCommandListener(this);
    } finally {
        if(is != null) is.close();
        if(hc != null) hc.close();
    }
    display.setCurrent(tb);
}
    
```

그림 5. Client측 메인 Method

```

public Connection getConnection()
{
    Connection con=null;
    try {
        con=DriverManager.getConnection
            (url,user,wooriwo);

    }catch(SQLException e) {}
    return con; //DB Connection
}
    
```

그림 6. DB connection

```

public void setDBInit(HttpSession session) {
    try {
        System.out.println("333");
        this.session = session;
        db.DBConnection
        db=(db.DBConnect)session.getAttribute("DB");
        con = db.getConnection();
        pstmt=con.prepareStatement("select password
            members where id =?"); // ID Check 부분.
    } catch(SQLException e) {
        System.out.println("e1 : " + e);
    }
}
: (중략)
    
```

그림 7. Server 측 ID check

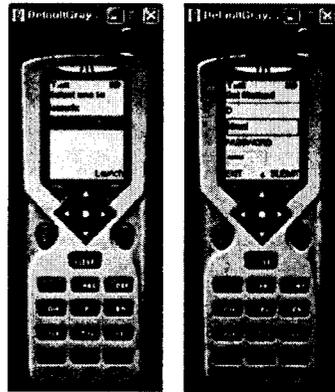


그림 8. 서버 연동 후 실행결과

4. 결 론

인터넷의 접속 환경은 유선 서비스 에서 무선 서비스로 한 층 더 가속화되어가고 있는 시점에서 Text위주의 WAP서비스의 한계로 다양한 콘텐츠와 멀티미디어 서비스가 필요한 실정이다. J2ME를 이용하여 모바일 서버와의 간단한 연동기법을 실험 확인 하였다. 본 연구에서는 서버 환경구축을 위해 웹 서버와 서블릿 엔진으로 Tomcat, 데이터베이스 연동을 위하여 JSP를 활용하여 서버와의 연동을 제시하였다. 현재 모바일 서비스 업체별로 다양한 VM(Virtual Machine) 이 등장하고 있으며 소형화 경량화 추세에서 제한된 메모리의 활용을 최적화하는 플랫폼 구조가 가속화 되 가고 있다. 이러한 실정에서 본 연구에서 제안한 서버 연동 기법을 적용시 서버 연동 문제를 해결할 수 있음을 확인하였다.

[참고문헌]

- [1] 2001 한국해양정보통신학회 vol5 No.2, "무선 인터넷 서비스 플랫폼 동향 분석", 김성한, 민재홍, 박기식, pp407- 410, 2001
- [2] "자바 모바일 프로그래밍", 강성윤, 이경범, 홍성인, 대림출판사
- [3] URL : <http://java.sun.com/docs/books/tutorial/servlets/index.html>, 2002
- [4] URL : <http://java.sun.com/j2me>, 2002