스마트폰 환경을 위한 미들웨어 설계 및 구현

김경주* · 문상호** · 유영중** · 박성호***

Design and Implementation of Middleware for Smartphone Environments

kyoung-ju Kim* · Sang-ho Moon** · Young-jung Yu** · Seong-ho Park***

요 약

현재 제공되고 있는 다양한 스마트폰 플랫폼은 효율적인 애플리케이션 개발을 어렵게 하고 있다. 이 문제를 해결하기 위하여 스마트폰 환경에 미들웨어를 도입하는 연구가 이루어지고 있다. 미들웨어의 도입은 서버 시스템과 스마트폰 플랫폼의 상호 운용성을 높여 효율적으로 스마트폰 애플리케이션을 개발하고 관리할 수 있도록 지원한다. 그러므로 급속하게 확장하는 스마트폰 환경에 능동적으로 대응하기 위하여 스마트폰 미들웨어 개발은 필수가 되고 있다. 본 연구에서는 다양한 스마트폰 플랫폼 환경에서 새로운 애플리케이션 서비스를 개발하고 유지하는데 소요되는 비용과 시간을 최적화할 수 있도록 스마트폰 미들웨어를 설계 및 구현하였다. 그리고 구현된 스마트폰 미들웨어의 성능 및 활용성을 검토하기 위하여 대학환경의 스마트폰 응용과 캠퍼스 트위터를 개발하였다.

ABSTRACT

The various Smartphone platforms that are used currently make difficult to build efficient applications(″apps″) for Smartphone. Introduction of middleware in the Smartphone environment is being studied to solve this problem. By enhancing interoperability between server systems and Smartphone platforms as introducing this middleware supports efficiently for Smartphone apps to be developed and managed. Thus, the development of this middleware for Smartphone has become essential for the purpose of responding actively to the rapidly expanding Smartphone market. In this research, we designed and implemented Smartphone middleware which optimizes the cost and time for developing new application service and maintaining it. In order to test this implemented middleware's performance and its capabilities, we also developed university Smartphone apps and activated campus twitter.

키워드

스마트폰, 미들웨어, 모바일 서비스, 캠퍼스 트위터

Key word

Smart Phone, Middleware, Mobile Service, Campus twitter

* 준회원 : 부산대학교 정보전산원 연구원
** 정회원 : 부산외국어대학교 컴퓨터공학과 교수
*** 정회원 : 부산대학교 정보전산원 교수 (교신저자, shpark@pusan.ac.kr)

접수일자 : 2011. 01. 05
심사완료일자 : 2011. 02. 08

Ⅰ. 서 론

최근 들어 애플사를 선두로 표준화된 개방형 운영체제를 채택하고 3G, 와이-파이(Wi-Fi), 와이브로(WIBRO)와 같이 다양한 방법으로 무선인터넷을 사용할 수 있는 환경과 멀티터치 등의 새로운 인터페이스 기술을 개발 탑재한 스마트폰의 보급이 급속하게 증가하고 있다. 스마트폰의 보급 확대는 현재 웹 방식을 주로 사용하는 정보의 유통 및 관리에 있어 많은 변화를 가져올 것으로 예상된다. 이러한 환경에 대비하기 위하여 각 기관들은 스마트폰 사용자를 위한 응용 서비스를 준비하고 있다.

그러나 스마트폰은 제조사별로 다른 운영체제를 채택하고 있으며, 동일한 제조사의 제품이라도 성능과 운영체제의 버전에 따라 다른 플랫폼을 가지고 있다. 그러므로 스마트폰 응용 개발자들은 특정 플랫폼의 스펙에 맞춰서 응용프로그램인 앱을 개발하여야 한다. 또한 대부분의 기관에서 사용되는 웹 프로그램은 Active X의 사용으로 마이크로소프트사의 인터넷 익스플로러(IE) 환경에 특화되어 스마트폰의 다양한 플랫폼을 지원하지 못하고 있다. 그러므로 개발자는 다양한 스마트폰 플랫폼과 인터페이스 특성에 부합하는 스마트폰 애플리케이션을 각각 개발하고 관리하여야함으로 현재 인터넷에서 제공되고 있는 서비스를 스마트폰 환경에서 제공하기 위해서는 많은 비용과 시간이 요구된다.

이런 다양한 스마트폰 환경에 따른 개발 효율성 문제들을 해결하기 위해서 다양한 연구가 진행되고 있다. 진행하는 연구로는 가상머신(Virtual Machine)을 개발하여 플랫폼에 독립적인 개발환경을 제공하는 연구, 하나의 특정 언어로 개발한 다음 해당하는 플랫폼 언어로 자동 변환하는 연구, 다양한 플랫폼을 위하여 미들웨어를 도입하는 연구 등이 있다. 가상머신방식은 각각의 플랫폼에 해당되는 가상머신을 개발하여 탑재함으로써 플랫폼에 독립적으로 표준화된 언어로 개발된 애플리케이션을 지원할 수 있으나 성능 저하, 소비전력 과다 등의 문제가 발생할 수 있어, 현재 일부 스마트폰에서만 지원하고 있다. 하나의 특정 언어로 개발한 다음 해당하는 플랫폼 언어로 자동 변환해 주는 방식은 각 플랫폼에 대한 최적화된 응용 프로그램을 보급할 수 있으나 다른 플랫폼에 대하여 응용 프로그램을 이식하는 과정을 거쳐야함으로 추가적인 개발 비용이 요구되며, 응용 프로그램에 변경이 발생하였을 경우 모든 플랫폼에 대한 프로그램을 수정하여야 하는 등의 유지보수의 어려움이 발생한다. 그러므로 보다 효율적인 방법에 대한 연구가 요구되고 있으며, 최근에는 다양한 스마트폰 플랫폼을 위해 미들웨어를 도입하는 연구에 초점을 맞추고 있다. 미들웨어의 도입은 시스템의 상호 운용성을 높이고, 개발 성능을 높이는 역할을 한다. 그러므로 급속하게 확장하는 스마트폰 환경에 능동적으로 대응하기 위해 스마트폰 미들웨어 개발이 요구되고 있다. 즉, 각 기관의 스마트폰 애플리케이션의 요구를 지원하기 위해 새로운 시스템을 개발하는 것은 막대한 비용과 시간이 요구됨으로 보다 효율적인 스마트폰 애플리케이션 개발 환경을 구축하기 위해서는 스마트폰 미들웨어의 도입이 필요하다.

본 논문에서는 스마트폰 환경에서 다양한 서비스를 제공하기 위한 애플리케이션을 효율적으로 개발하고 관리할 수 있도록 지원하는 미들웨어를 설계 및 구현하여 대학 환경에 적용해봄으로써 사용의 효율성을 제시한다.

논문의 구조는 다음과 같다. 2장에서는 다양한 분야에서 연구된 미들웨어와 관련된 연구를 소개하고, 3장에서는 스마트폰 미들웨어의 전체 구조와 상세 설계에 대하여 설명한다. 4장에서는 설계에 따라 구현된 내용에 대해 언급하고 대학 환경에 적용하여 미들웨어의 효용성을 실험한다. 마지막 5장에서 결론과 향후 연구에 대하여 기술한다.

Ⅱ. 관련 연구

기존의 미들웨어의 연구는 모바일, 센서네트워크, 물류 등 다양한 산업 및 환경에 맞춰서 진행되었고 그것의 성능과 확장성, 상호 운용성 향상을 위한 연구도 동시에 진행되었다[1][2]. 미들웨어에 대한 연구는 하드웨어 장치와 소프트웨어를 연동시켜 주는 임베디드 미들웨어, 시스템과 시스템을 연동시켜 주는 비즈니스 미들웨어 등이 있다. 임베디드 미들웨어는 통신 프로토콜, 물리 메모리 등 소프트웨어가 직접 처리하기 어

려운 작업을 수행하며, 수행된 결과를 소프트웨어가 사용할 수 있도록 인터페이스를 제공한다. 이와 같은 연구는 센서 네트워크, 물류 등의 분야에서 Impala[3], Mires[4], EPCglobal[5] 등의 연구가 수행되었다. 비즈니스 미들웨어는 이미 구축되어서 작동하고 있는 시스템과 다른 시스템이 동시에 사용되어야 할 때 두 시스템을 합쳐서 하나의 서비스를 제공할 수 있는 역할을 수행한다[6]. 비즈니스 미들웨어 연구는 Common Object Request Broker Architecture(CORBA)[7]와 같이 인터페이스를 제공하여, 언어와 상관없이 통신하여 데이터를 주고받을 수 있는 범용 미들웨어 기법과 실시간으로 발생하는 데이터를 효율적으로 응용 프로그램에 제공하는 기법이 연구되었다.

그러나 기존에 연구된 미들웨어는 그 환경에 최적화되어 있어 스마트폰 환경에 바로 적용하기 어려움으로 기존의 환경에 최적화 되어 있는 미들웨어의 구조를 변경 할 필요가 있다. 다양한 스마트 기기를 합쳐 놓은 스마트폰은 기존의 그 어떤 장치보다도 다양하게 응용이 개발되고, 이용되고 있다. 그러므로 그 응용들을 다 지원할 수 있는 API의 구조를 제공할 수 있는 미들웨어를 설계하고 구현하여야 한다.

Ⅲ. 스마트폰 미들웨어의 설계

본 연구에서는 스마트폰 응용의 다양한 환경을 지원하기 위하여 기존의 함수 호출 방식의 질의가 아닌, EPCglobal의 표준 프레임워크에서 정의해 놓은 정의(Spec) 방식의 질의를 사용한다. 이 방식은 질의를 XML로 변환하기 때문에, 다양한 응용에서 동시에 사용 할 수 있는 상호 운용성을 지원한다. 이를 기반으로 스마트폰 미들웨어 아키텍처를 정의하고 프로세서를 설계하고 구현하였다.

3.1 아키텍처 정의

스마트폰 미들웨어의 아키텍처는 그림 1과 같이 Service 계층, Process 계층, Capture 계층으로 구성된다. Service 계층은 클라이언트에 Open API를 제공하고, 그 API를 통하여 사용자로부터 질의를 받는다. 또한 사용자 질의를 분석하여 미들웨어에서 처리 할 수 있도록 Process 계층으로 전달하고, 사용자에게 질의의 응답을 전달하는 역할을 수행한다. Process 계층은 Service 계층으로부터 받은 질의의 조건에 따라, 즉각 처리해야 되는 질의는 Cache에서 먼저 데이터를 찾아보고 없으면 Capture 계층에 요청한다. Subscribe 질의와 Event 질의는 등록된 질의 수행 조건에 맞으면 질의를 수행한다. Capture 계층은 응용서버와 데이터베이스를 연동하는 Adapter 인터페이스를 제공하고, Adapter 인터페이스에 맞춰서 추가된 Adapter로 데이터를 요청하여, 데이터를 수집하는 역할을 한다.

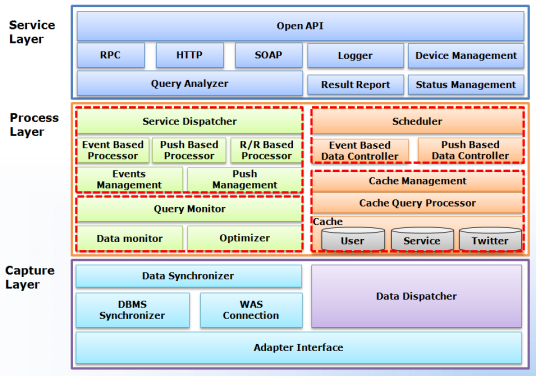

그림 1. 미들웨어 아키텍처
Fig 1. The Middleware Architecture

Service 계층에서 자바의 RMI와 HTTP, SOAP 방식의 다중 프로토콜을 지원하며 Query Analyzer를 통해 사용자로부터 질의를 받고, 질의를 분석한다. Query Analyzer는 사용자의 XML 질의를 미들웨어에서 처리 할 수 있는 Class 형태로 변환한다. Logger와 Result Report는 사용자가 응답 받는 시점과 장소에 미들웨어가 처리한 결과를 전달하는 역할을 수행한다. Device Management와 Status Management는 스마트폰의 통신 상태를 점검하는 역할을 수행한다. Status Management는 현재 스마트폰과 통신 상태가 어디서 문제가 발생하여 생기는지 점검 하는 일을 한다.

Process 계층은 질의 종류에 적절하게 대응하기 위하여 Event Query를 처리하는 Event Based Processor, Subscribe Query를 처리하는 Push Based Processor, One-time Query를 처리하는 R/R Based Processor를 포함한다. Event Based Processor는 미들웨어 이벤트를 등록

하고, 등록 된 이벤트가 발생하면 질의를 수행하는 역할을 한다. Push Based Processor는 미들웨어에서 주기적으로 정보를 전달해야 하는 연속 질의 처리기로, 질의의 수행 조건이 맞을 때, 질의 처리를 한다. R/R Based Processor는 현재 조건에 맞는 질의를 바로 처리한다. Scheduler는 질의 수행 조건이 맞는지 계속적으로 체크하는 역할을 수행한다.

Capture 계층은 응용서버와 데이터베이스를 연결해서 데이터를 수집하는 역할을 한다. Adapter Interface는 Adapter에서 미들웨어가 필요로 하는 데이터를 전달하는 역할을 하고, Data Synchronizer는 데이터베이스와 미들웨어, 응용서버와 미들웨어 사이의 데이터를 동기화하는 역할을 한다. Data Dispatcher는 수집된 데이터를 Process 계층에 전달하는 역할을 한다.

3.2 패키지 정의

스마트폰 미들웨어는 그림 2와 같이 17개로 패키지로 구성하였다. 그 중 service패키지는 사용자에게 API를 제공하는 패키지로 사용자의 질의를 분석하는 query와 process 패키지에 질의를 전달하는 service. application 패키지가 있다. 그리고 service.util과 service.properties 패키지는 사용자의 질의결과를 전달하고, process 패키지는 미들웨어에서 실제적인 질의처리를 하며, process. subscribe 패키지는 사용자의 연속질의를 처리하며, process.event 패키지는 사용자의 이벤트 질의를 처리하는 프로그램을 포함한다. 또한 process.util 패키지는 process 패키지가 질의 처리 할 수 있도록 함수를 제공한다.

그리고 data.model과 data. cache 패키지는 미들웨어에서 사용하는 데이터와 캐시를 정의하고, 실제로 데이터를 메모리에 저장하는 프로그램을 포함한다. adapter 패키지는 미들웨어에서 필요로 하는 데이터를 응용서버나 데이터베이스에서 수집해 오는 역할을 수행하고 adapter.database는 데이터베이스 관련 데이터 수집을 처리하고 adapter.was는 응용서버 관련, adapter.filesystem은 로컬 파일시스템에서 데이터를 수집하는 프로그램을 포함한다. 그리고 adapter.util 패키지는 adapter가 데이터를 수집하는데, 필요한 함수들을 제공한다.

3.3 컴포넌트와 클래스 정의

본 논문에서는 컴포넌트와 클래스의 설계를 위한 툴로써 UML(Unified Modeling Language) 2.0을 지원하는 N3 Nabee를 사용하였다[8].

스마트폰 미들웨어는 그림 3의 컴포넌트 다이어그램과 같이 총 4개의 컴포넌트로 구성된다. Service 컴포넌트는 스마트폰과 통신을 하는 인터페이스를 제공하며, 사용자에게 결과를 전달하는 역할을 한다. Process 컴포넌트는 실제 미들웨어에서의 질의처리와 데이터 정제 및 수집을 하는 역할을 한다. Cache 컴포넌트는 미들웨어의 메모리에서 데이터를 미리 저장하여 스마트폰에게 좀 더 빠른 응답을 제공한다. 마지막으로 Adapter 컴포넌트는 미들웨어에서 필요로 하는 데이터를 수집하고, 타 시스템과 연동하는 역할을 수행한다.

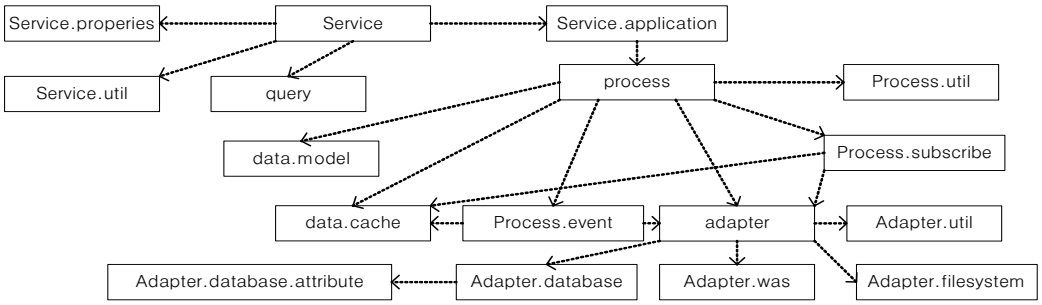

그림 2. 스마트폰 미들웨어 패키지 다이어그램
Fig 2. A Package Diagram of Smartphone Middleware

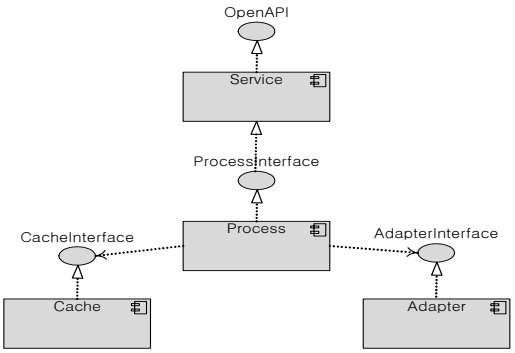

그림 3. 미들웨어 컴포넌트 다이어그램
Fig 3. Middleware Component Diagram

• Service 컴포넌트 클래스

Service 컴포넌트 클래스는 그림 4와 같이 1개의 인터페이스와 8개의 클래스로 구성된다.

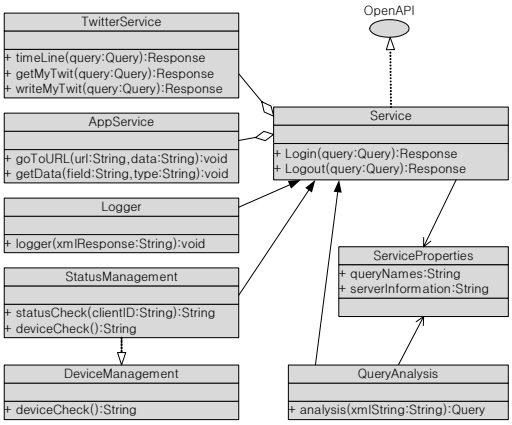

그림 4. Service 컴포넌트 클래스 다이어그램
Fig 4. Service Component Class Diagram

한 개의 인터페이스는 사용자가 제공되는 인터페이스로 OpenAPI이다. Service 컴포넌트가 하는 일은 사용자에게 인터페이스를 제공하고, 사용자로부터 받은 질의를 Process 컴포넌트에 전달하는 역할이다. 그러므로 새로운 응용을 추가하기 위해서는 Service 컴포넌트에 새로운 응용의 정의를 추가하여야 한다. 본 연구에서는 트위터와 앱 서비스를 위한 TwitterService와 AppService 클래스와 사용자에게 질의의 결과를 전달하는 Logger 클래스와 스마트폰의 상태를 체크하는 Device Management, StatusManagement 클래스가 있다.

• Process 컴포넌트 클래스

Process 컴포넌트 클래스는 그림 5와 같이 한 개의 인터페이스와 7개의 클래스로 구성되어 있다. Process 컴포넌트는 데이터 처리와 질의 처리를 사용자의 조건에 맞게 수행한다.

Scheduler는 주기적으로 사용자의 질의 수행 조건을 체크하면서 QueryProcessor를 수행한다. 그리고 Process 컴포넌트는 사용자의 질의의 등록, 삭제, 갱신하는 역할을 QueryManager에서 수행한다. ProcessTwitter와 같은 경우에는 기능적인 것보다는 효율적으로 트위터 응용에 대해서 처리하기 위해서 추가한 클래스이다. 트위터와 관련된 일은 실제는 트위터 서버가 하지만, 사용자가 많아질 경우를 대비해서 트위터와 관련된 처리를 따로 분리하여 클래스로 구성하였다.

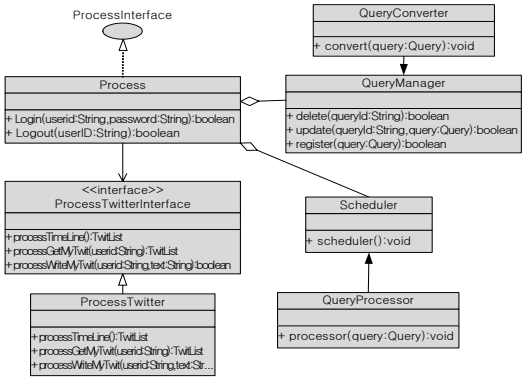

그림 5. Process 컴포넌트 클래스 다이어그램
Fig 5. Process Component Class Diagram

• Cache 컴포넌트 클래스

Cache 컴포넌트 클래스는 그림 6과 같이 3개의 인터페이스와 3개의 클래스로 구성되어 있다.

먼저 Cache 인터페이스는 Process 컴포넌트에 제공되는 인터페이스로 삽입, 삭제, 검색, 갱신을 하는 역할을 수행한다. 그리고 Cache 컴포넌트 내부에 있는 Twitter 인터페이스와 User 인터페이스는 트위터 캐시의 성능을 높이기 위해 나누었다. 실제 사용자의 정보를 저장하는 것보다 트위터의 저장이 많을 것으로 예상되기 때문에 트위터 처리에 최적화하였다.

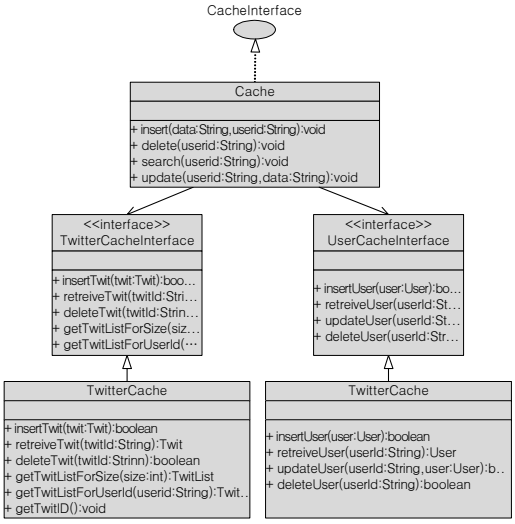

그림 6. Cache 컴포넌트 클래스 다이어그램
Fig 6. Cache Component Class Diagram

• Adapter 컴포넌트 클래스

Adapter 컴포넌트는 그림 7과 같이 4개의 인터페이스와 8개의 클래스로 구성되어 있다. Adapter 컴포넌트는 빈번한 변경에 대비하여 인터페이스를 확장이 가능하도록 하였다. 특히 데이터베이스 인터페이스는 현재 사용하는 대부분의 데이터베이스와 연동이 가능하며, 새로운 데이터베이스의 추가도 가능하도록 하였다. WAS 인터페이스는 현재 거의 모든 응용 서버들이 지원하는 HTTP 프로토콜 클래스를 가진다. LocalFileSystem 인터페이스는 미들웨어에서 로그를 남기거나, 로컬 데이터베이스를 구성에 필요한 클래스를 포함한다.

Ⅳ. 스마트폰 미들웨어 구현 및 적용

본 논문에서 구현된 스마트폰 미들웨어의 성능을 평가하기위하여 대학환경에서의 스마트폰 응용과 트위터에 적용하였다. 대학의 스마트폰 응용을 위해서는 그림 8과 같이 인증시스템, 학사시스템, 응용 서버나 데이터베이스 등의 다양한 시스템에 접근이 요구된다. 인증시스템과 같은 경우에는 외부 응용에서 로그인이 가능하도록 API를 제공하고 있으나 학사시스템과 데이터베이스는 내부 네트워크에서만 접근이 가능하여 외부 응용에서 바로 접근 할 수 없다. 그러므로 미들웨어는 교내 내부시스템과의 연동과 외부시스템과의 연동이 요구된다.

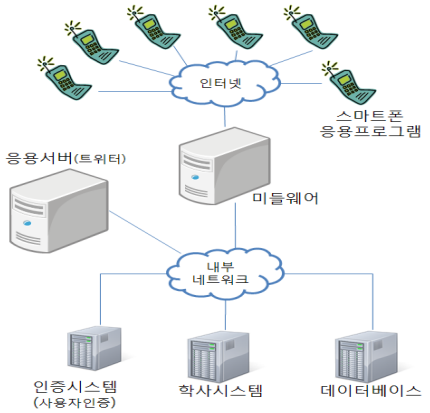

그림 8. 스마트폰 미들웨어 적용 환경
Fig 8. An Environment that applied to Smartphone Middleware

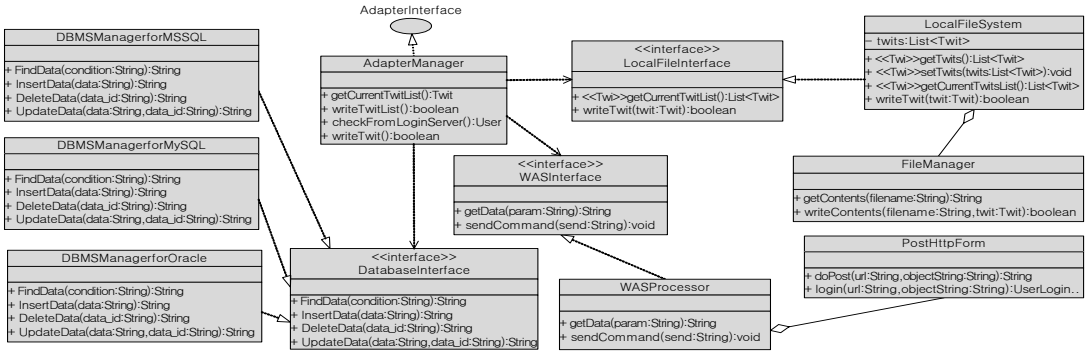

그림 7. Adapter 컴포넌트 클래스 다이어그램
Fig 7. Adapter Component Class Diagram

4.1 개발 환경 및 구동 환경

본 논문에서는 구현된 스마트폰 미들웨어의 개발환경과 구동환경은 표 1, 표 2와 같다

표 1. 스마트폰 미들웨어 개발 환경
Table 1. Development Environment for Smartphone Middleware

요소	사양
CPU	Inteldual core 2.66GHz
Memory	3GB
Language	Java SE Development Kit 6u21
IDE	Eclipse classic
OS	Windows XP SP3

표 2. 스마트폰 미들웨어 구동 환경
Table 2. Operation Environment for Smartphone middleware

요소	사양
OS	Linux CentOS 5.3
CPU	Intel dual core 2.4GHz
Memory	4GB

4.2 스마트폰 응용 개발

본 연구에서 구현한 미들웨어를 테스트하기 위하여 아이폰 기반 응용 애플리케이션인 PNU 트위터를 우선적으로 구현하였다. 스마프폰 응용 개발환경은 표 3과 같다.

표 3. 스마트폰 응용 개발 환경
Table 3. Development Environment for Smartphone Application

이름	항목
플랫폼	IOS 4.0 (아이폰 OS)
IDE	Xcode
SDK	IPhone 4.0 OS
OS	Leopard 10.5.6

• 캠퍼스 트위터의 활용

그림 9는 본 연구에서 구현된 캠퍼스 트위터 화면의 예를 보여준다. 캠퍼스 트위터는 인증시스템을 통해 로그인하며, 학사 시스템의 수업 정보를 이용하여 같은 강의를 수강하는 팔로우(follower) 그룹을 생성할 뿐 아니라 개인별 팔로우 그룹을 생성한다. 또한 각 개인이 생성한 트윗은 트위터 응용서버를 통해 관리된다. 이러한 과정을 스마트폰 미들웨어가 처리함으로써 다양한 플랫폼에 대한 개발의 효율성을 높일 수 있다.

그림 9. 캠퍼스 트위터 화면의 예
Fig 9. Examples of campus twitter's screen

• 캠퍼스 스마트폰 응용

그림 10은 본 연구에서 구현된 캠퍼스 스마트폰 응용 화면의 예를 보여준다.

그림 10. 캠퍼스 앱 화면의 예
Fig 10. Examples of Campus App's screen

이 스마트폰 응용은 미들웨어를 통해 로그인과 데이터베이스 서버에 접근하여 학사 정보 및 좌석잔여현황, 캠퍼스맵, 오시는길 등 대학의 다양한 정보를 제공한다.

V. 결론 및 향후 연구

본 연구에서는 급속하게 확장하는 스마트폰 환경에 능동적으로 대응하기 위해 스마트폰 미들웨어를 설계 및 구현하였다. 스마트폰 미들웨어는 다양한 스마트폰 플랫폼 환경에서 새로운 애플리케이션 서비스 개발과 유지에 소요되는 비용과 시간을 효율적으로 활용할 수

있는 기회를 제공한다. 또한 본 연구에서는 대학환경의 스마트폰 응용과 캠퍼스 트위터를 개발하여 구현된 스마트폰 미들웨어의 성능 및 활용성을 점검하였다. 현재는 시간적인 한계로 본 연구에서는 교내 메신저 역할을 수행하여 학내 구성원 및 졸업생 간의 Social Network Service를 제공하는 캠퍼스 트위터 및 간단한 응용 프로그램을 아이폰 환경에서만 구현하였으나 향후 다양한 플랫폼에서 응용프로그램 및 표준화된 API를 추가로 연구 개발할 예정이다.

참고문헌

[1] 김민수, 김광수, 이용준, “USN 미들웨어의 특징 및 기술 개발 동향,” IITA, 주간기술동향, 통권 1284호, 2007. 2.

[2] 박병섭, “대용량 데이터처리를 위한 XML기반의 RFID 미들웨어 시스템“, 한국콘텐츠학회논문지, 제7권, 제7호, 2007.

[3] Ting Liu, Margaret Martonosi, ″Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems,″ San Diego : ACM p.107, 2003.

[4] Eduardo Souto, Germano Guimaraes, Glauco Vasconcelos, Mardoqueu Vieira, Nelson Rosa, Carlos Ferraz. ″A Message-Oriented Middleware for Sensor Network,″ New York:ACM p.127, 2004.

[5] EPCglobal Architecture framework, URL : http://www.epcglobalinc.org

[6] R. E. Schantz and D. C. Schmidt, ″Middleware for distributed systems: Evolving the common structure for network-centric applications,″ in Encyclopedia of Software Engineering, J. Marciniak and G. Telecki, Eds. New York: Wiley & Sons, 2002.

[7] Object Management Group, ″The Common Object Request Broker: Architecture and Specification Revision 2.4, OMG Technical Document formal/00-11-07,″ October 2000.

[8] N3 Nabee Document. URL : http://www.n3soft.co.kr

김경주(kyeong-ju Kim)

2008: 부산대학교 컴퓨터공학과(공학사)
2010: 부산대학교 컴퓨터공학과(공학석사)
2010 ~ 현재: 부산대학교 정보전산원 연구원
※관심분야 : RFID 미들웨어, RTLS 미들웨어, 질의 처리

문상호(Sang-ho Moon)

한국해양정보통신학회 논문지 제14권 제4호 참조

유영중(Young-jung Yu)

한국해양정보통신학회 논문지 제14권 제11호 참조

박성호(Seong-ho Park)

한국해양정보통신학회 논문지 제14권 제11호 참조