

---

# XML 기반 스마트폰 미들웨어 Open API 구현

김영주\* · 김경주\*\* · 유영중\*\*\* · 박성호\*\*\*\*

Implementation of XML-based Open API for Smartphone Middleware

Young-ju Kim\* · Kyoung-ju Kim\*\* · Young-jung Yu\*\*\* · Seong-ho Park\*\*\*\*

## 요 약

최근, 스마트폰의 보급이 폭발적으로 성장하고 있다. 이러한 환경에서 많은 연구자들은 스마트폰 애플리케이션 개발의 중요성을 인식하고 효율적인 애플리케이션 개발을 위한 많은 연구를 수행하고 있다. 특히 서버-클라이언트 환경에서 효율적인 스마트폰 애플리케이션(앱) 개발을 위한 스마트폰 미들웨어에 관한 연구에 최근 관심이 집중되고 있다. 본 연구에서는 다양한 플랫폼을 가지는 스마트폰의 환경에서 애플리케이션 개발의 이식성, 확장성 그리고 편의성을 높이기 위하여 스마트폰 미들웨어 환경에 최적화된 XML 기반 Open API를 제안하였다. 그리고 이를 적용시켜 개발한 테스트 스마트폰 미들웨어 환경에서 성능을 평가하였다.

## ABSTRACT

Recently, use of Smartphones is growing at an extremely rapid pace. Many researchers have recognized the importance of developing Smartphones applications under these circumstances, and they have been conducted many studies to develop apps efficiently. The Smartphones middleware has become a focus of recent research to develop efficient Smartphones apps on a server-client environment in particular. In this study, for enhancing portability, expandability and convenience in the development of apps on the various platforms of Smartphones, We have proposed XML based Open API optimized in the Smartphones middleware. We also evaluated the performance, by applying it in Smartphone middleware environment.

## 키워드

스마트 폰, 미들웨어, XML, Open API

## Key word

Smart Phone, Middleware, XML, Open API

---

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

접수일자 : 2011. 02. 21  
심사완료일자 : 2011. 03. 21

### I. 서 론

최근 들어 전 세계적으로 개방형 운영체제를 지원하는 다양한 플랫폼의 스마트폰이 출시되었고, 스마트폰 시장의 크기가 폭발적으로 성장하고 있다. 특히 국내에서는 2009년 애플사의 아이폰의 출시되면서, 스마트폰 시장이 확대되는 계기가 되었다. 2010년도에의 국내 이동통신시장의 가장 큰 변화는 스마트폰 보급의 본격화와 아이패드와 갤럭시탭과 같은 태블릿 PC의 등장이다. 2011년도에는 이러한 스마트폰의 보급이 더욱 확대될 것이며, 태블릿 PC 또한 100만 대 이상이 보급될 것으로 예상되고 있다[1]. 이러한 스마트 단말 보급의 확대는 그림 1과 같은 다양한 스마트폰 환경을 형성하게 되었으며, 이를 활용하기 위한 다양한 스마트폰 응용프로그램 개발도 활성화되고 있다. 특히 스마트폰 응용프로그램은 어느 한 분야에 한정된 것이 아니라, 사회 다방면에서 활용되고 있다. 기업의 업무용, 개인의 여가용, Social Network Service 용으로 많이 이용되고 있으며, 나아가 더욱 더 많은 분야에서 스마트폰 응용이 개발 될 것이다. 스마트폰은 강력한 프로세서와 풍부한 메모리, 다양한 무선통신을 이용 할 수 있어서, 기존의 컴퓨터가 하는 작업뿐만 아니라 기존의 컴퓨터가 하지 못했던 다양한 분야에서 활용될 수 있다.



그림 1. 스마트폰 환경

Fig 1. An Environment of Smartphone

그림 1과 같이 다양한 응용과 무선통신, 플랫폼, 스마트폰이 이용되는 다양성이 존재하는 환경은 사용자가 자신의 취향에 맞는 스마트폰과 플랫폼 통신 방식, 응용을 선택 할 수 있는 큰 장점이 있다. 또한 스마트폰에 따

라서 다양한 센서와 카메라 등 다양한 센서를 탑재한 스마트폰도 보급되어 새로운 응용에 대한 요구를 증가시키고 있다. 그러므로 현재 스마트폰 환경은 그림 2와 같이 많은 응용서버에서 다양한 스마트폰을 지원하기 위하여 다양한 프로토콜을 지원하고 있다. 예를 들면, 자바로 구현되어 있는 헬스케어 서버의 경우 자바를 지원하는 안드로이드 환경에서는 서버와의 통신하는 응용 프로그램 개발이 용이하나 자바를 지원하지 않는 아이폰이나 블랙베리 환경을 지원하기 위하여 다른 통신 API를 구현하여 제공하여야 한다. 이러한 환경은 서버를 관리하고 개발하는 비용을 증가시킬 것이다.

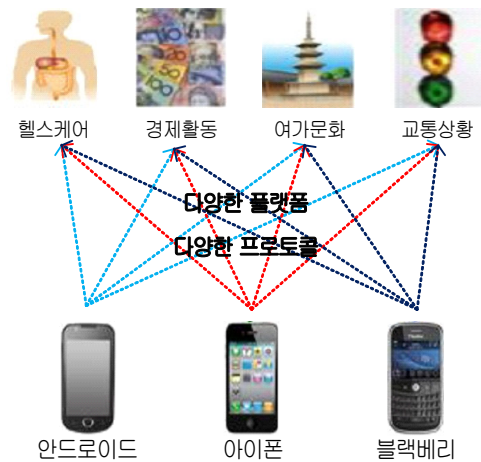


그림 2. 미들웨어를 사용하지 않는 스마트폰 환경  
Fig 2. An Environment of Smartphone that do not use middleware

이러한 문제를 해결하기 위하여 표준화된 웹 브라우저 응용 개발용 API인 웹킷[2]이나 모바일웹 표준화 연구[3]가 진행 중이나 속도 및 사용자 접근성 면에서 모바일 앱을 여전히 선호되고 있으므로 효율적인 스마트폰 애플리케이션(앱) 개발을 위한 스마트폰 미들웨어에 관한 연구에도 최근 관심이 집중되고 있다. 그러나 현재 상용으로 사용되는 미들웨어를 스마트폰 환경에 도입할 경우 스마트폰 환경에 최적화된 Open API가 제공되지 않아 이식성과 확장성 편의성이 떨어진다. 그러므로 본 논문에서는 이러한 문제를 해결하기 위하여 XML 기반 스마트폰 미들웨어 Open API를 설계 및 구현하고 성능을 평가하였다.

본 논문의 구성은 다음과 같다. 2장에서는 미들웨어와 Open API 관련된 연구를 소개하고, 3장에서는 스마트폰 미들웨어 Open API를 설계 및 구현에 대하여 설명하고 4장에서는 구현된 Open API의 성능을 평가 한다. 5장에서 결론과 향후 연구에 대하여 기술한다.

## II. 관련 연구

### 2.1. 웹킷

기존 스마트폰 응용프로그램 개발 시 단점으로 부각된 다양한 플랫폼 환경 설정 고려와 이종 기기의 호환성 등의 문제점이 일어나면서 자연스럽게 모바일 웹 응용프로그램 개발이 대안으로 부각되고 있다. 모바일 웹 응용프로그램 종류의 하나인 하이브리드용 웹 응용프로그램은 스마트폰에 내장된 웹킷 엔진을 이용하기 때문에 간단하게 개발될 수 있다는 장점이 있다. 즉, 스마트폰의 웹킷 탑재로 응용프로그램 개발자는 HTML과 자바스크립트, CSS만으로도 쉽고 빠르게 개발할 수 있을 뿐만 아니라 다양한 모바일 기기에 서비스를 제공할 수 있다[2,3].

그러나 모바일 웹은 스마트폰 응용프로그램(앱)에 비해 속도가 현저히 떨어지며, 다양한 응용프로그램에 대한 최적화된 사용자 인터페이스를 제공하기 어렵다. 또한 모바일 웹은 스마트폰의 다양한 하드웨어 접근에 한계가 있어 다양한 응용프로그램에 개발에 걸림돌이 되고 있다. 그러므로 많은 스마트폰 서비스 개발자들은 모바일 웹 환경보다는 스마트폰 앱 환경을 여전히 선호하고 있다.

### 2.2. 미들웨어

앞 절에서 언급한 스마트폰 응용프로그램 개발 시 단점으로 부각되고 있는 다양한 플랫폼 환경 설정 고려와 이종 기기의 호환성 등의 문제점을 이 해결하기 위한 방법으로 스마트폰 미들웨어에 관한 연구가 관심을 끌고 있다.

미들웨어는 분산 컴퓨팅 환경에서 서로 다른 기종의 하드웨어나 프로토콜, 통신환경 등을 연결하여, 응용프로그램과 그 프로그램이 운영되는 환경 간에 원만한 통신이 이루어질 수 있게 하는 소프트웨어이다. 현재 미들

웨어의 연구는 모바일, 센서네트워크, 물류 등 다양한 산업 및 환경에서 성능과 확장성, 상호운용성 향상을 위한 연구가 진행 중이다.

특히 센서 네트워크, 물류 등의 분야에서 Impala[4], Mires[5], EPCglobal[6], RFID 미들웨어[7], USN 미들웨어[8] 등의 연구가 수행되었으며, 분산 애플리케이션 및 서버/클라이언트 분야에서는 Adaptive Communication Environment(ACE)[9], Common Object Request Broker Architecture(CORBA)[10], Internet Communications Engine(ICE)[10] 등의 연구가 수행되었다. 또한 미들웨어 표준 인터페이스(Open API)로 EPCglobal에서는 ALE(Application Level Event)를 제시하였으며, 이를 기반으로 유연성과 확장성을 가지도록 고안된 LGCNS RFID 미들웨어와 ETRI UbiCore 등이 발표되었다. 그러나 이들 미들웨어와 Open API는 그 환경에 최적화되어 있어 스마트폰 환경에 바로 적용하기 위해서는 기존의 미들웨어 및 Open API를 변경하여야 한다. 특히 다양한 응용을 지원하고 편리하게 사용할 수 있는 확장된 Open API가 요구된다.

## III. 스마트폰 미들웨어 Open API

본 연구에서는 스마트폰 응용의 다양한 환경을 지원하기 위하여 기존의 함수 호출 방식의 API가 아닌 정의(Spec) 방식의 API를 사용한다. 이 방식은 질의를 XML로 변환하기 때문에, 다양한 응용에서 동시에 사용할 수 있는 상호운용성을 지원한다.

### 3.1. 스마트폰 미들웨어의 기본 구조

본 절에서는 그림 3과 같은 스마트폰 환경에서 효율적인 스마트폰 미들웨어 API를 개발하기 위하여 스마트폰 미들웨어의 기본 구조를 제시한다.

스마트폰 미들웨어는 스마트폰에서 요청하는 많은 질의들을 각각의 응용 서버에 맞는 질의로 변환하고 스마트폰이 필요로 하는 데이터를 미리 가져옴으로써 스마트폰이 좀 더 빠르게 요청을 처리할 수 있도록 한다. 그리고 스마트폰이 원하는 형태로 응용서버의 데이터 형태를 변환해주는 역할을 수행한다.

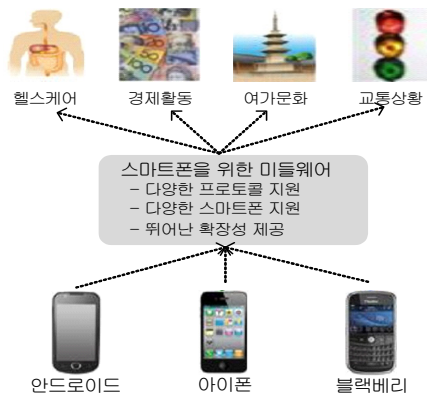


그림 3. 미들웨어를 사용하는 스마트폰 환경  
Fig 3. An Environment of Smartphone that used middleware

이러한 스마트폰 미들웨어의 도입은 각각의 스마트폰 응용프로그램이 각각의 응용 서버에 맞춰서 개발하지 않고 미들웨어에서 제공해 주는 Open API를 통해 미들웨어와의 연동만으로 스마트폰 응용 프로그램을 구현할 수 있도록 지원한다. 그러므로 효율적인 시스템 구축 및 개발이 가능하고 무엇보다도 서로 다른 기종의 스마트폰끼리의 상호운용성이 증가한다. 또한 서버 교체의 필요성이 발생하더라도 이식성 및 확장성이 뛰어나기 때문에 저비용 고효율의 유지보수가 가능하다. 스마트폰 미들웨어의 기본 구조는 그림 4와 같이 Service 계층, Process 계층, Capture 계층으로 구성된다.

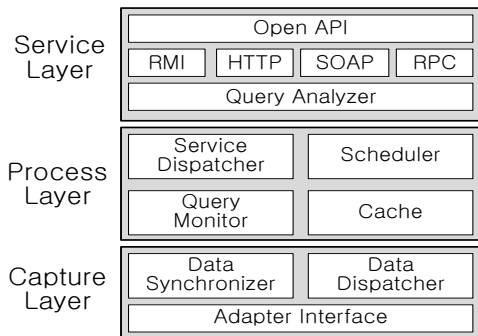


그림 4. 스마트폰 미들웨어의 구조  
Fig 4. The architecture of Smartphone middleware

Service 계층은 클라이언트에 Open API를 제공하고, 그 API를 통하여 사용자로부터 질의를 받고 사용자 질

의를 분석하여 미들웨어에서 처리 할 수 있도록 Process 계층으로 전달하고, 사용자에게 질의의 응답을 전달하는 역할을 수행한다. 또한 자바의 RMI와 HTTP, SOAP 방식의 다중 프로토콜을 지원하며 Query Analyzer를 통해 사용자로부터 받은 질의를 분석한다. Query Analyzer는 Class 형태로 전달된 사용자의 질의를 변환하여 처리한다. Process 계층은 질의 종류에 적절하게 대응하기 위하여 질의를 모니터링하며, Service 계층으로부터 받은 질의의 조건에 따라, 즉각 처리해야 되는 질의는 Cache에서 먼저 데이터를 찾아보고 없으면 Capture 계층에 요청한다. Capture 계층은 응용서버와 데이터베이스를 연동하는 Adapter 인터페이스를 제공하고, Adapter 인터페이스에 맞춰서 추가된 Adapter로 데이터를 요청하여, 데이터를 수집하는 역할을 한다.

### 3.2. Open API의 요구사항

기존 미들웨어의 API는 스마트폰의 다양한 응용을 지원하기 위한 확장성과 질의 편의성을 제공하지 못하고 있다. 이러한 문제점을 해결하기 위해서 서버와 스마트폰 사이의 효율적인 데이터 교환 포맷과 질의의 확장성을 고려한 질의 방식과 질의 정의가 필요하다.

스마트폰 미들웨어의 이식성을 높이기 위하여 다양한 플랫폼이나 언어 환경을 지원할 수 있으며, 다양한 종류의 데이터 타입을 지원할 수 있는 플랫폼 독립적인 언어로 API를 구성하여야 한다. 현재 이 두 가지를 모두 지원하는 언어는 XML과 JSON이 대표적이다. 이 연구에서는 이식성이 좀 더 높은 XML을 사용한다. 또한 미들웨어의 확장성과 질의 편의성을 위하여 질의 처리 방식은 XML-RPC를 이용한다. XML-RPC는 다양한 스마트폰의 플랫폼에서 XML 스트링 질의를 만들어서 HTTP 프로토콜을 이용하여 서버로 전달하는 방식으로 웹 서비스에서 많이 사용되는 방식이다. 그러므로 XML-RPC는 질의의 확장이 용이한 장점을 가진다.

### 3.3. Open API의 정의

본 논문의 Open API는 크게 One-Time Query와 Event Query, Query Response로 정의된다.

표 1의 One-Time Query는 사용자의 질의를 바로 처리해서 응답을 즉시 요청하는 Query로 데이터베이스나 응용서버의 데이터를 한번 질의를 통해 결과를 요청하는 역할을 한다.

표 1. One-Time Query의 정의  
Table 1. The definition of One-Time Query

속성 이름	타입	설명	
server Name	String	웹서버, DBMS 등의 이름	
function Name	String	웹서버의 함수 이름과, DBMS의 테이블 또는 Stored Procedure 이름	
support Store Procedure	Boolean	Stored Procedure를 사용 유무	
Attributes	List <Attribute>	질의에 조건들에 해당되는 조건들의 리스트	
Attribute	name	String	조건의 이름
	value	String	조건의 값

표 2의 Event Query는 사용자의 질의에 실행 조건을 추가하여, 실행 조건에 맞는 이벤트가 발생할 때마다 결과를 수집하는 역할을 한다.

표 2. Event Query의 정의  
Table 2. The definition of Event Query

속성 이름	타입	설명	
QueryName	String	이벤트 질의의 이름	
Attributes	List <Attribute>	질의에 조건들에 해당되는 조건들의 리스트	
Attribute	name	String	조건의 이름
	value	String	조건의 값
execution Condition	Execution Condition	질의 실행 조건을 타나냄	
Execution Condition	Event Name	String	결과를 받을 이벤트 이름
	notification URI	String	결과를 받을 클라이언트 주소
	duration	int	이벤트 질의 처리 수행 시간
	report SetType	String	이벤트 질의 타입
	report Empty	Boolean	이벤트 질의 결과 유무

표 3의 Query Response는 One-Time Query와 Event Query에 의해 요청된 결과를 반환하는 역할을 수행한다.

표 3. Query Response의 정의  
Table 3. The definition of Query Response

속성 이름	타입	설명
QueryName	String	질의 이름
Text	String	결과에 대한 설명
Date	Date	결과 시간
numberofresults	int	결과 개수
List<Object>	any	결과 리스트 ( Event, String의 리스트 등)

### 3.4. Open API 동작 방식

표1, 표2, 표3에서 정의된 구조를 가지는 API의 동작 방식은 다음과 같다. 먼저 요구된 질의를 해당하는 API의 정의에 따라 클래스를 생성한다. 생성된 클래스는 각 API의 정의에서 기술된 구조를 가진다. 이 구조를 가지고 있는 클래스를 XML 스트링으로 변환한다. 그리고 HTTP의 Post 방식을 이용해서 미들웨어로 전송한다. 미들웨어는 전달된 XML 스트링을 다시 Query라는 클래스로 변환하고 해당되는 조건을 분석하여 질의 결과를 생성한 후 클라이언트에 전달하는 방식을 사용한다. 그림 5는 로그인 질의를 본 논문에서 제시한 Open API를 이용하여 처리하는 과정을 예로 들어 보여준다. 이때 로그인 질의의 변화된 XML 스트링은 그림 6과 같다.

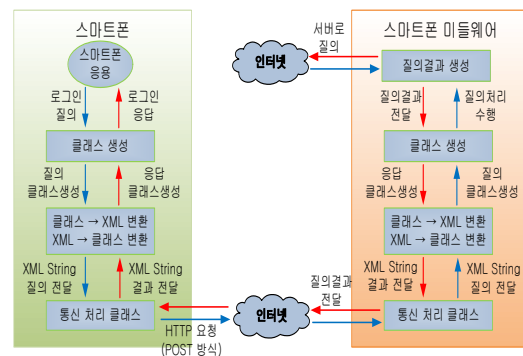


그림 5. 로그인 질의 처리의 예  
Fig 5. An Example of login query processing

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Query>
<serverName>LoginServer</serverName>
<targetName>Login</targetName>
<storedProcedure>FALSE</storedProcedure>
<attributes>
<attribute>
<name>user id</name>
<value>kimkk</value>
</attribute>
<attribute>
<name>password</name>
<value>1234!@#</value>
</attribute>
</attributes>
</Query>
```

그림 6. XML 스트링으로 변화된 로그인 질의의 예  
Fig 6. An Example of login query that is transformed into XML string

이 포함된 SQL문으로 처리된다. 그리고 데이터베이스 중 StoredProcedure를 사용할 경우에는 함수 이름은 결과를 생성하기 위한 구조체의 이름이 된다. 질의의 Attribute는 이름과 값을 가진 구조체이며 리스트 형태로 전달된다. 이러한 Attribute리스트는 속성 분석기를 통해 처리된다. 속성 분석기는 상위에서 분기되어서 두 가지 모듈로 수행된다. 응용 서버 처리 모듈일 경우 응용 서버에 관련된 Attribute 리스트들은 함수의 인수 값을 나타낸다. 데이터베이스 처리 모듈일 경우에는 데이터베이스 종류, 데이터베이스의 데이터 타입, 데이터베이스의 StoredProcedure 이름 등을 Attributes 리스트에 포함한다. 이러한 과정을 통해 전달되어진 질의는 미들웨어에서 사용할 수 있는 구조체로 변환된다.

### 3.5. 미들웨어의 Open API 처리 알고리즘

스마트폰에서 XML 스트링 형태로 미들웨어에 전송된 사용자의 질의는 미들웨어에서 사용하는 클래스로 변환된다. 클래스로 변환된 질의는 그림 7과 같이 서버이름 분석과정, 함수이름 분석과정, 속성 분석과정을 통해 수신된 질의를 해석하여 처리한다.

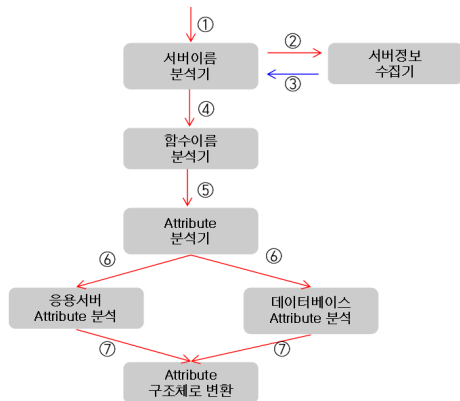


그림 7. 미들웨어의 API 처리 알고리즘  
Fig 7. A processing algorithm of middleware API

서버이름 분석기는 서버정보 수집기를 통해 질의 대상 서버가 응용서버인지, 데이터베이스 서버인지 판단을 하며, 함수이름 분석기는 서버이름 분석기의 결과를 바탕으로 응용 서버일 경우 처리 패스를 포함한 함수이름으로 처리하며, 데이터베이스 서버일 경우 테이블 명

## IV. 성능 평가

본 논문에서는 스마트폰의 다양한 플랫폼을 지원하는 애플리케이션을 효율적으로 개발하기 위한 스마트폰 미들웨어의 핵심적인 부분인 Open API를 제안하였다. 그러나 본 논문에서 제안한 API는 XML을 기반으로 하고 있어 확장성 및 이식성을 높여 개발의 효율성을 향상시킬 수 있으나 처리 속도가 증가하는 단점을 가진다. 그러므로 본 절에서는 일반적인 서버-클라이언트 통신 인터페이스(TCP/IP)를 적용한 방식과 본 논문에서 제안한 API를 적용한 방식의 성능을 질의의 처리 시간과 질의의 크기를 비교하는 실험을 수행하였다.

표 4. 실험환경  
Table 4. An Experimental environment

서버 환경	설명
미들웨어	자바로 구현된 자체 미들웨어
웹서버	jetty 6.1 webservice
프로토콜	HTTP(POST 방식)
클라이언트 환경	설명
네트워크	무선랜 와이파이
클라이언트 플랫폼	IOS 4.2 플랫폼(시뮬레이션)
프로토콜	HTTP(POST 방식)

본 논문에서는 제안한 API의 성능평가 실험을 위해 표 4와 같은 실험환경을 구축하였다.

현재 스마트폰은 일반 PC와 유사한 처리 능력을 가진 것으로 하나의 질의를 처리하는 데 있어 일반적인 통신 인터페이스(TCP/IP)와 본 논문에서 제안한 API 모두 수 ms(milli-second)이하로 의미 있는 차이를 보이지 않았다. 그러므로 본 절에서는 한 번의 질의에 많은 데이터가 발생할 수 있는 응답 처리에 대하여 성능 평가를 실시하였다.

그림 8은 질의에 대한 결과의 개수에 따른 처리 시간을 보여준다. 질의에 대한 응답의 처리 시간은 응답의 개수가 많을수록 제안된 API의 처리 시간이 증가한다. 그러나 응답의 개수가 1000개 이하의 경우 일반적인 통신 인터페이스와 제안된 API 모두 처리 시간이 0.5초 이하의 성능을 보이고 있다. 이는 대부분의 스마트폰 응용프로그램이 한 번의 질의로 수신할 수 있는 결과를 수백개 이하로 제한하고 있으므로 응용 프로그램 측면에서는 심각한 전송 지연을 발생시키지 않는다.

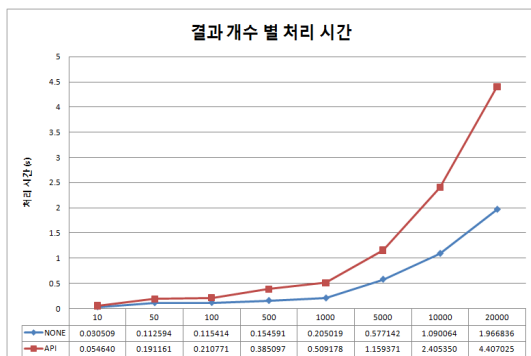


그림 8. 질의에 대한 응답 개수 별 처리 시간  
Fig 8. A processing time by number of responses to queries

그림 9는 질의에 대한 결과의 크기에 따른 처리 시간을 보여준다. 결과의 데이터가 클수록 제안된 API를 사용한 경우 처리 시간이 다소 증가하지만 심각한 성능 저하는 발생하지 않는다.

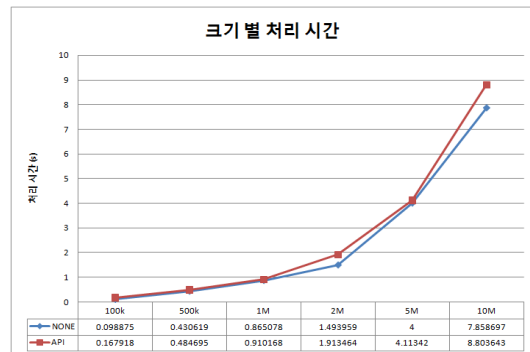


그림 9. 질의에 대한 응답 크기 별 처리 시간  
Fig 9. A processing time by size of responses to queries

위의 실험을 통해 본 논문에서 제안된 API는 일반적인 통신 인터페이스를 사용한 NONE API경우와 비교하여 심각한 처리 성능의 저하를 발생시키지 않는 반면에 이식성과 확장성을 제공하여 스마트폰 애플리케이션 개발의 효율성을 높일 수 있다.

## V. 결론

최근, 스마트폰의 보급이 폭발적으로 성장하고 있다. 이러한 환경에서 많은 연구자들은 스마트폰 애플리케이션 개발의 중요성을 인식하고 효율적인 애플리케이션 개발을 위한 많은 연구를 수행하고 있다. 특히 서버-클라이언트 환경에서 효율적인 스마트폰 애플리케이션 (앱) 개발을 위한 스마트폰 미들웨어에 관한 연구에도 최근 관심이 집중되고 있다. 본 연구에서는 다양한 플랫폼을 가지는 스마트폰의 환경에서 이식성, 확장성 그리고 편의성을 높이기 위하여 스마트폰 미들웨어 환경에 최적화된 XML 기반 Open API를 제안하였다. 그리고 이를 적용시켜 개발한 스마트폰 미들웨어 실험환경에서 성능을 평가한 결과 본 논문에서 제안된 API는 일반적인 통신 인터페이스를 사용한 NONE API경우와 비교하여 심각한 처리 성능의 저하를 발생시키지 않는 반면에 이식성과 확장성을 제공하여 스마트폰 애플리케이션 개발의 효율성을 높일 수 있음을 보였다.

### 참고논문

- [ 1 ] 김진영, “2011년 Mobile Device 대전망 보고서”, kt 경제경영연구소 Digieco 12월호, 2010
- [ 2 ] 이고은, 이종우, “스마트폰 상에서의 웹 응용프로그램 개발 환경 비교”, 한국콘텐츠학회논문지, pp. 155-163, 2010
- [ 3 ] 전중홍, 이승윤, “차세대 모바일 웹 애플리케이션 표준화 동향”, 전자통신동향 분석, 제25권, 2010
- [ 4 ] Ting Liu, Margaret Martonosi, “Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems,” San Diego : ACM p.107, 2003.
- [ 5 ] 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.
- [ 6 ] EPCglobal Architecture framework, URL : <http://www.epcglobalinc.org>
- [ 7 ] 박병섭, “대용량 데이터처리를 위한 XML기반의 RFID 미들웨어 시스템“, 한국콘텐츠학회논문지, 제7권, 제7호, 2007.
- [ 8 ] 김민수, 이용준, 박종현, “USN 미들웨어 기술개발 동향,” 전자통신동향분석, 제22권 제3호, pp.67-79, 2007.
- [ 9 ] <http://www.cs.wustl.edu/~schmidt/ACE.html>
- [10] Object Management Group, “The Common Object Request Broker: Architecture and Specification Revision 2.4, OMG Technical Document formal/00-11-07,” October 2000.

### 저자소개

#### 김영주(Yeong-ju Kim)

한국해양정보통신학회 논문지  
제14권 제11호 참조

#### 김경주(kyeong-ju Kim)

한국해양정보통신학회 논문지  
제15권 제3호 참조

#### 유영중(Young-Jung Yu)

한국해양정보통신학회 논문지  
제15권 제3호 참조

#### 박성호(Seung-Ho Park)

한국해양정보통신학회 논문지  
제15권 제3호 참조