

모바일 디바이스 환경에서 어플리케이션에 따른 소프트 인터페이스 제작 및 제공 방안 연구

조창희*, 박상현**, 이상준***, 김진솔****

요약

본 논문에서는 사용자가 사용하는 어플리케이션 환경에 따른 인터페이스를 제공하고, 다양한 어플리케이션 환경에 적용할 수 있도록 웹을 통한 인터페이스 제작을 제공한다. 인터페이스 제작은 HTML5로 제작되어 마우스 드래그를 통해 다양하게 제작할 수 있으며, 기본적인 키 입력에 사용하는 아스키 코드와 안드로이드 OS에서 제공되는 키 이벤트를 사용하여, 문서뿐만 아니라, 멀티미디어, 게임등과 같은 어플리케이션에서 적용하여 사용할 수 있게 한다. 인터페이스 관련 데이터는 하둡 기반의 HDFS에 저장되어 관리되며, 간단한 로그인을 통해 자신이 제작 또는 선택한 인터페이스를 언제든지 제공받을 수 있다. 인터페이스를 빠르게 제공하기 위해 하둡 기반의 HIVE를 사용하여 검색하고 검색한 데이터를 Xml형식으로 제공하여, 제공받는 스마트 모바일에서도 빠르게 처리할 수 있도록 한다.

키워드 : 어플리케이션 키패드, 사용자 인터페이스, HTML5, 스마트 모바일, 웹 툴, 하둡 서버

Research of Soft-Interface Creation and Provision Methodology According to Applications Based on Mobile Device Environment

Changhee Cho*, Sanghyun Park**, Sang-Joon Lee***, Jinsul Kim****

Abstract

In this paper, we provide interfaces according to user application environments and provide tools through web-site that users can create interface to apply a wide range of application environment. HTML5 is used in the creation processing, so users can create various interfaces by dragging mouse and apply it to multimedia, game applications as well as documents by using the ASCII code and key events that are provided in the Android OS. Database of interfaces is stored in HDFS (Hadoop Distributed File System) based on Hadoop for management and users can have their own designed interface or select interfaces through simple login any time. In order to provide interface quickly, HIVE based on Hadoop is used for search and the data is provided in XML file which smart mobile can process quickly.

Keywords : Application keypad, Customized Interface, HTML5, Smart Mobile, Web Tools, Hadoop Server

1. 서론

※ 교신저자(Corresponding Author): Jinsul Kim
접수일:2013년 11월 29일, 수정일:2013년 12월 21일
완료일:2013년 12월 24일
* 전남대학교 전자상거래 박사과정
** 전남대학교 전자컴퓨터학부 석사과정
*** 전남대학교 경영학부 교수
**** 전남대학교 전자컴퓨터학부 교수
Tel: +82-62-530-1808, Fax: +82-2-880-7010
e-mail: jsworld@jnu.ac.kr
■ 본 연구는 BK21 Plus와 미래창조과학부 및 정보통신산업진흥원의 대학IT연구센터육성 지원사업의 연구결과로 수행되었음 (NIPA-2013-H0301-13-3005).

초창기 PC의 인터페이스는 Text만 나오는 CUI(Character User Interface)에서, 점차 화려한 GUI(Graphic User Interface)로 발전해 왔다[1]. PC가 화려한 GUI를 지원하게 되면서, PC시장은 급격하게 발전해갔고, 다양한 PC들이 출시되기 시작했다. 노트북, PDA, 태블릿 PC들이 등장하면서, 인터페이스들은 다양하게 변해갔다. 최근 스마트 디바이스가 출시되면서, 인터페이스의 자유도는 크게 확대가 되었다. 그 중에서도 가상키

보드는 많은 변화를 가져왔다. 기존에 가상키보드의 역할은 PC의 키보드 장치가 망가졌을 경우 마우스를 이용하여, 키보드의 역할을 대신해주는 도구로 사용되었다. 지금은 다양한 터치스크린 제품이 출시되면서, 보조역할이 아닌 물리적인 키보드를 대신하는 메인 키보드가 되었다. 이처럼 가상키보드[2][3]는 많은 부분에서 큰 역할을 하고 있다. OS의 배경화면, 메뉴 등 여러 인터페이스를 변경하듯이 가상키보드의 디자인도 다양하게 바뀌고 있다. 마이크로소프트사의 윈도우 OS는 다양한 개발 툴들이 있어, 가상키보드의 디자인뿐만 아니라, 기본적인 기능에 추가적인 기능을 더하여, 다양한 기능을 사용할 수 있는 가상키보드를 제작할 수 있다.

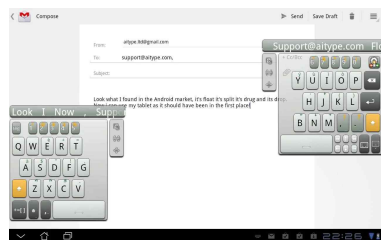
최근 안드로이드 OS기반의 모바일 디바이스가 증가하면서, 윈도우 OS처럼 테마와 레이아웃을 누구나 쉽게 변경할 수 있도록 다양한 어플리케이션들이 많이 출시되었다. 안드로이드 OS는 보통 스마트 모바일 디바이스에 최적화되어 윈도우 OS처럼 다양한 단축키를 쉽게 사용하지 못한다. 또한, 안드로이드 OS를 이용하는 사용자는 가상키보드의 스킨을 직접 제작하거나 쉽게 적용이 가능하지만, 레이아웃의 모양은 쉽게 제작하거나 자신이 원하는 레이아웃을 쉽게 가질 수가 없다. 다른 개발자가 개발한 가상키보드 레이아웃은 앱 형식으로 쉽게 다운받을 수 있지만, 이를 적용하기 위해서는 OS환경설정에서 사용할 키보드를 변경 해줘야하는 번거로움을 가지고 있다.

본 논문에서는 다양한 가상키보드를 사용자가 사용하는 어플리케이션에 따라, 실시간으로 제공하고, 단축키 기능을 사용하여 사용자가 어플리케이션을 사용하는데 보다 높은 효율을 제공하고자 한다. 가상키보드는 웹을 통해 누구나 쉽게 제작하여 사용할 수 있도록 한다. 웹을 통한 제작은 하둡 기반의 클라우드 서버를 이용하여 제공하며 생성된 인터페이스는 다른 사람들과 공유할 수 있고, 다른 사람이 제작한 인터페이스를 기반으로 자신에게 맞는 인터페이스로 변경하여 사용할 수 있다. 스마트 디바이스에서 인터페이스를 쉽게 적용할 수 있도록 Xml형식으로 다양한 인터페이스를 제공한다. 제작된 인터페이스는 하둡[4]의 map과 reduce[5]를 통해서 key와 Value로 분산되어 저장된다.

2. 관련 연구

스마트 모바일 기기와 관련하여, 다양한 모양의 가상키보드가 출시되고 있으며, 지금도 계속 개발 중에 있다. 태블릿 PC도 스마트 모바일 기기와 같이 안드로이드 OS를 많이 사용하고 있다. 출시된 키보드 어플리케이션 중에 인공지능 태블릿 키보드 어플리케이션은 안드로이드 OS에 호환되는 가상키보드로 태블릿 PC전용으로 출시되었다. 이 어플리케이션은 기존의 가상키보드와 다르게 분할이 되어있으며, 드래그 앤 드롭으로 가상 키보드의 사이즈를 줄일 수 있다는 장점을 가지고 있다. 사용자는 사용하는 어플리케이션에 따라, 가상키보드의 사이즈를 조절하면서 사용할 수 있으며, 가상키보드로 인해, 작업하는 화면이 줄어들어 불편함을 없애주는 역할도 한다. 또한, 문자를 입력하였을 때 맞춤법을 교정해주는 기능도 있어 많은 도움을 주고 있다. 이 어플리케이션은 문자입력, 문서입력과 같이 문자를 입력하는데 다양한 기능들을 포함하고 있으며, 한국어를 지원하는 장점을 가지고 있다. 하지만, 화면이 가로에서 세로로 전환되면 가상키보드 레이아웃의 틀이 망가지거나, 가상키보드가 사라지는 버그가 있다.

(그림 1) 인공지능 태블릿 키보드

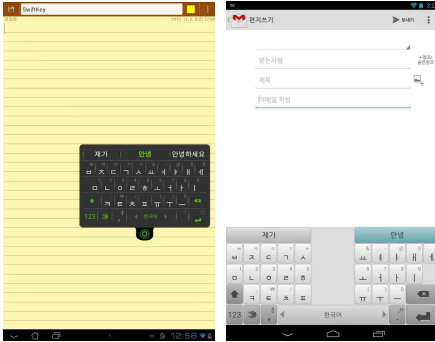


(Figure 1) A.I. Tablet keyboard

Swiftkey키보드는 이전에 인공지능 태블릿 키보드와 유사한 어플리케이션으로 이 어플리케이션은 태블릿 PC뿐만 아니라 스마트 모바일 디바이스에도 사용할 수 있다. 가로, 세로모드와 분할, 콤팩트와 같은 모드도 지원하여, 사용자의 어플리케이션에 맞게 사용할 수 있다. 입력방식은 기존 터치방식과 화면에 손가락을 떼지 않고 슬라이드 방식으로 입력 하는 방식이 있다. 이전 어플리케이션과 동일하게 가상키보드의 크기를

마음대로 조절할 수 있다. (그림 2)의 왼쪽그림과 같이 콤팩트 모양일 경우 제약 없이 위치와 크기를 조절할 수 있으며, 이전 어플리케이션과 같이 작업 화면이 줄어드는 문제를 해결해준다.

(그림 2) SwiftKey 키보드



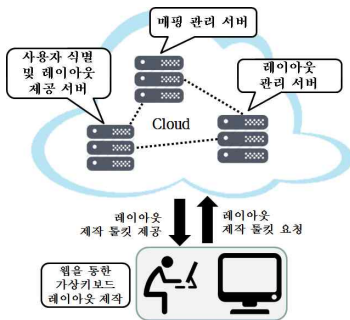
(Figure 2) SwiftKey Keyboard

이처럼 가상키보드 어플리케이션은 보편적으로 사람들이 쉽고, 빠르고 정확하게 사용할 수 있도록 제작되어있다. 레이아웃 모양은 어플리케이션에서 제공되는 것 외에는 사용할 수 없으며, 가상키보드 어플리케이션은 문서를 작성하거나 문자를 보내는 용도 외에는 사용할 수가 없다.

3. 시스템 환경

다양한 어플리케이션 환경에 따라 인터페이스를 제공하기 위해서는 누구나 가상키보드를 제작할 수 있는 환경이 제공되어야 한다.

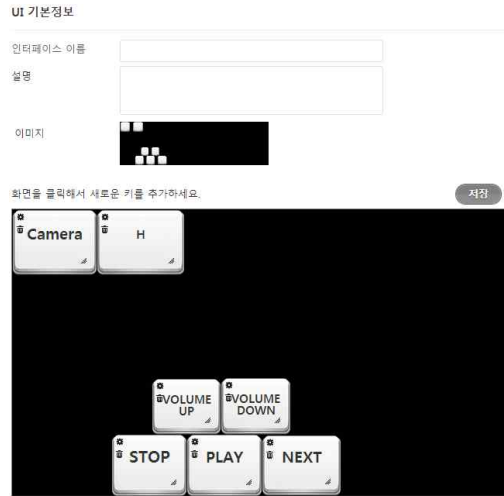
(그림 3) 가상키보드 제작환경 구조



(Figure 3) Virtual keyboard production environment Structure

(그림 3)은 어플리케이션 환경에 따른 가상키보드를 제공하기 위한 구조를 보여준다. 클라우드 서버[6][7]는 웹을 통하여, 가상키보드의 인터페이스를 제작할 수 있게 도와주며, 웹을 처음 이용하는 사용자는 간단한 로그인을 통해 가상키보드의 인터페이스를 제작할 수 있다. 가상키보드의 인터페이스는 한사람이 아닌 다수가 동시에 사용하기 때문에 사용자를 식별하기 위한 별도의 아이디가 필요하다. 아이디 생성은 회원가입을 통해 아이디와 패스워드를 작성하고 이메일을 입력하면 회원가입이 가능하다. 생성된 사용자의 정보는 (그림 3)과 같이 사용자 식별 및 레이아웃 제공서버에 저장된다.

(그림 4) 웹을 이용한 인터페이스 제작



(Figure 4) Interface Making using the web

(그림 4)는 웹을 통한 인터페이스를 제작하는 모습을 보여준다. 제공하는 웹은 HTML5[8]를 기반으로 제작되어 그림과 같이 마우스를 이용하여 사용자가 원하는 인터페이스를 간단한 동작으로 버튼을 생성하고, 원하는 키 입력을 선택한다. 또한, 버튼의 크기, 위치를 마우스의 드래그 기능을 이용하여 쉽게 설정할 수 있다. 인터페이스 제작을 완료한 후 저장버튼을 눌러 제작을 완료한다. 인터페이스는 다른 사람과 공유하여 사용할 수 있고, 내가 다시 재수정하여 사용할 수 있다. 제작된 모든 인터페이스는 (그림 3)의 인터페이스 관리 서버에 저장된다. 또한 사용자가 어떠한 레이아웃을 저장하였는지의 대한

정보는 매핑 관리 서버가 관리한다. 매핑 관리 서버는 실제 데이터를 가지고 있지 않으며, 인덱스 정보만을 가지고 있기 때문에 사용자가 레이아웃을 요청하였을 때 빠르게 검색을 할 수 있도록 도와준다.

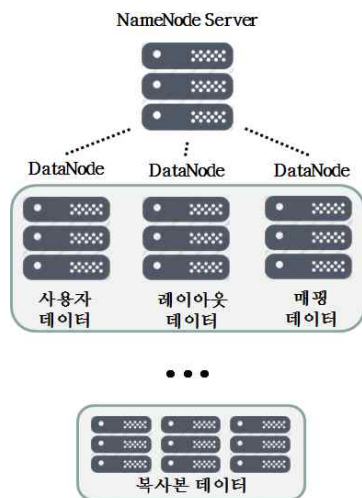
4. 인터페이스 저장 및 제공방법

스마트 모바일에 제공되는 인터페이스 데이터들은 하둡의 HDFS(Hadoop Distributed File System) 방식으로 구성된다. 보통은 Key, Value 방식으로 데이터가 저장이 된다.

4.1 인터페이스 데이터 저장방법

인터페이스 데이터를 저장하는 방법은 하둡의 HDFS방식을 기반으로 한다.

(그림 5) 전체 데이터 저장구조



(Figure 5) All data storage structure

데이터들은 (그림 5)와 같은 방식으로 데이터가 저장된다. NameNode Server[9]은 전체 데이터를 관리하는 서버로써, 데이터의 종류에 맞게 나눠주는 역할을 하게 된다. 또한, 데이터들의 인덱스 값인 메타데이터만을 가지게 됨으로 사용자가 자료를 요청하였을 때, 원하는 자료가 어디에 있는지를 바로바로 찾아준다. 실제 데이터가 저장되어 있는 저장소를 DataNode라고 하는데 (그림 6)과 같이 Key와 Value형식으로 구

성되어 있다. (그림 6)는 전체 데이터 저장구조를 간단하게 나타낸 것으로 데이터는 체계적으로 연결되어 있다. 사용자가 요청한 데이터를 Key값을 이용하여 빠르게 검색을 하며 검색한 레이아웃 데이터를 사용자에게 제공한다.

(그림 6) 전체 데이터 저장구조

Key	Value
L00012	List<...>
L00015	List<...>
...	...

Key	Value
A	X=0, y=1, W=10, H=10, Code = 0x45,
Camera	...
...	...

(Figure 6) DataNode Data storage structure

4.2 인터페이스 데이터 제공방법

사용자가 로그인을 통해 인터페이스를 요청하게 되면 서버는 매핑관리 서버의 매핑데이터를 기반으로 레이아웃 관리 서버를 검색하기 시작한다. 검색을 위해서는 Hive[10]라는 쿼리를 사용한다. Hive는 하둡과 연계되어 사용자가 원하는 데이터를 쿼리문을 사용하여, 데이터를 빠르게 찾을 수 있게 도와준다. 쿼리문은 SQL쿼리문과 흡사하며, 기존 데이터 검색보다 빠르고 정확한 데이터를 검색할 수 있게 한다. 검색한 데이터는 스마트 디바이스가 분석할 수 있도록 XML파일 형태로 파일을 생성한다.

<표 1> Xml 인터페이스 데이터 형식

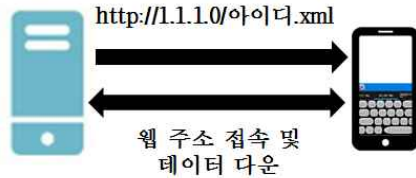
```
<UI = "U00012" name = "Media Keyboard">
<key="a" val = 0x61 x = "0" y= "1" width = "20" height = "21"/>
<key="Camera" val = 0xC2 x = "3" y= "5" width = "40" height = "30"/>
...
```

<Table 1> XML Interface Data Type

<표 1>은 스마트 디바이스에 제공되는 인터페이스 데이터다. 인터페이스 데이터는 고유번호, 인터페이스의 이름, 그 안에 세부적인 버튼의 키 이름, 스마트 디바이스에서 동작하는 코드 값 그리고 위치와 크기순으로 데이터가 전송이

된다. 1~2개정도의 인터페이스 데이터는 크지 않기 때문에 스트림을 이용해서 바로 전송이 가능하지만 그 이상의 데이터를 보낼 경우에는 인터페이스 데이터의 손상이 생겨 원하는 인터페이스를 적용하지 못할 수 있다.

(그림 7) 인터페이스 데이터 전송 방법



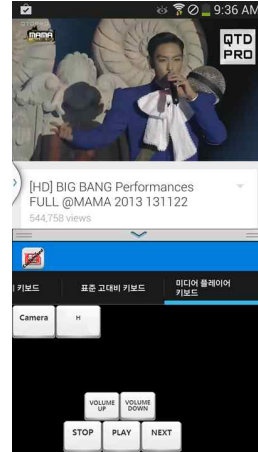
(Figure 7) Interface data transfer method

따라서 데이터를 전송하기 위해서는 (그림 7)과 같은 방법에 따라 데이터를 전송한다. 사용자가 요청한 인터페이스 데이터를 생성하게 되면 서버는 요청한 스마트 디바이스에 웹 사이트 주소를 보낸다. 웹사이트 주소를 받은 스마트 디바이스는 다시 웹사이트 주소로 접속을 하여, 요청한 인터페이스 데이터를 다운받는다. 이러한 형식의 데이터 전송 방식은 환경에 따라 속도가 변화는 무선 환경에서 데이터 손상 없이 전송하기 위한 방법이다. 중간에 네트워크가 끊어져도 스마트 디바이스는 다시 접속을 하여 받기 때문에, 데이터 손상이 발생하지 않는다. 레이아웃 데이터는 데이터 하나하나가 중요한 정보를 가지고 있기 때문에 위와 같은 방식이 아닌 데이터 자체를 스트림으로 보내지 않는다. 만약, 데이터를 스트림형식으로 보내게 될 경우 순간 접속이 끊어지면 데이터 손상이 발생하여 사용자가 요청한 인터페이스 정보를 제대로 받을 수가 없다. 따라서 웹사이트를 통한 데이터 다운방식은 보다 정확하게 데이터를 전송받을 수 있게 한다.

(그림 8)은 서버가 보내준 웹 주소 값을 가지고 데이터를 다운받아 적용한 모습이다. 레이아웃 상단에는 사용자가 선택한 다양한 레이아웃의 목록이 나와 있으며, 선택을 하면 바로 레이아웃이 적용되어 어플리케이션 상황에 맞게 사용할 수 있다. (그림 8)처럼 멀티미디어 어플리케이션을 이용할 때는 미디어 플레이어 키보드를 사용하고 문서를 작성하거나, 문자를 보낼 때

는 바로 옆에 있는 가상키보드를 선택하여 바로 적용하여 사용할 수 있다.

(그림 8) 인터페이스 적용 화면



(Figure 8) Apply the Screen Interface

4.3 다양한 키 이벤트 처리 방법

우리가 주로 사용하는 스마트 모바일 기기는 다양한 키 이벤트를 가지고 있다. 하지만, 기본적으로 제공되는 가상키보드는 많은 키 이벤트를 사용하지 않는다. 본 시스템에서는 다양한 어플리케이션 환경에 따른 인터페이스를 제공하기 위해 안드로이드에서 제공하는 키 이벤트를 최대한 사용하며, 그 외 추가적인 사용자 키 이벤트를 지원한다. 기본적인 알파벳, 숫자, 기호와 같은 문자는 아스키코드[12]를 사용한다. 아스키코드는 표준규격인 0x00~0x7F까지의 범위를 사용하며, 문서나, 문자와 같은 어플리케이션에서 주로 사용하게 된다. 영상시청, 음악듣기, 게임 등 문서를 작성하는 어플리케이션 외에 다양한 어플리케이션에서는 사용자 정의 및 다른 코드 값이 필요하다. 기본적으로 사용하지 않는 키값의 범위는 0x96이후의 코드범위로 지정된 아스키코드가 아닌 사용자 정의 코드 값으로 키 이벤트를 사용하여, 다양한 어플리케이션에서 적용하여 사용할 수 있게 한다. 키 이벤트관련 자료는 안드로이드 API문서[13]를 참조하였으며, API에 나와 있지 않는 기능들은 별도로 정의하여 사용할 수 있도록 하였다. (그림 8)의 경우 소리를 키우거나 줄였을 때 내부적으로 KEY CODE _ VOLUME _ UP 와

KEYCODE_VOL_UP, ME_DOWN과 같은 키 이벤트를 사용하여 처리하게 된다. 사용자는 자신이 주로 사용하는 어플리케이션 환경에 맞추어 여러 가지 키를 정의하고 이를 상황에 맞게 언제든지 제공받아 사용한다.

5. 결론

본 논문에서 모바일 디바이스에 있는 가상 키보드의 인터페이스를 다양한 어플리케이션 환경에 적용하기 위한 웹 제작 도구 및 저장 방법을 제안하였다. 사용자는 웹을 통하여 자신이 원하는 인터페이스를 제작할 수 있고, 다양한 어플리케이션 환경에 맞도록 사용할 수 있다. 또한, 간단한 로그인 방식으로 인터페이스를 제공받을 수 있기 때문에, 다른 스마트 디바이스를 사용하더라도 이전에 사용하였던 인터페이스를 그대로 사용할 수 있다. 하둠 방식의 데이터를 저장하고 관리함으로써, 단일 서버보다 효율적으로 데이터를 관리 할 수 있다. 또한 Hive쿼리를 사용하여, 사용자가 레이아웃 데이터를 요청하였을 때 빠르고 정확하게 데이터를 검색할 수 있도록 한다. 추후 과제로는 현재키 입력뿐만 아니라 기능키 즉, 카메라를 동작시키거나, 전화를 걸거나, 소리를 조절하는 등과 같은 기능키를 지원하지만 더욱 다양한 기능키를 지원할 수 있도록 할 것이며, 다양한 스마트 디바이스에서 사용할 수 있도록 지속적인 연구를 할 예정이다.

References

- [1] Campbell-Kelly, Martin, and William Aspray, "Computer: a history of the information machine," Vol.2, Boulder: Westview Press, 2004.
- [2] Al Faraj, Khaldoun, Mustapha Mojahid, and Nadine Vigouroux, "Bigkey: A virtual keyboard for mobile devices," Human-Computer Interaction, Ambient, Ubiquitous and Intelligent Interaction, Springer Berlin Heidelberg, pp.3-10, 2009.
- [3] Jain, Siddharth, and Samit Bhattacharya, "Virtual keyboard layout optimization," Students' Technology Symposium (TechSym), 2010 IEEE, pp.312-317, 2010.
- [4] K. Shvachko, H. Huang, S. Radia, R. Chansler, "The hadoop distributed file system," in: 26th IEEE (MSS T2010) Symposium on Massive Storage Systems and Technologies, pp.1-10, May 2010.
- [5] Dean, Jeffrey, and Sanjay Ghemawat, "MapReduce: a flexible data processing tool," Communications of the ACM, Vol.53, No.1, pp.72-77, 2011.
- [6] Brian Hayes, "Cloud computing," Communications of the ACM, Vol.7, pp.9-11, 2008.
- [7] Jong-Hei Ra, "Qualitative Study on Service Features for Cloud Computing," Journal of Digital Contents Society, Vol.12, No.3, pp.319-327, Sep 2011.
- [8] Lubbers, Peter, Frank Salim, and Brian Albers, "Pro HTML5 programming," Apress, 2011.
- [9] Carstoiu, D., A. Cernian, and A. Olteanu, "Hadoop hbase-0.20.2 performance evaluation," New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on. IEEE, pp.84-87, 2010.
- [10] Thusoo, Ashish, et al, "Hive-a petabyte scale data warehouse using hadoop," Data Engineering (ICDE), 2010 IEEE 26th International Conference on. IEEE, pp.996-1005, 2010.
- [11] Ling, Yun, and Guangjian Xu, "A Distributed Keyword Search Algorithm in XML Databases Using MapReduce," Computer, Informatics, Cybernetics and Applications, Springer Netherlands, pp.1307-1316, 2011.
- [12] Hieronymus, James L, "ASCII phonetic symbols for the world's languages: Worldbet," Journal of the International Phonetic Association, 1993.
- [13] Meier, Reto, "Professional Android 4 application development," John Wiley & Sons, 2012.

조창희



2010년 : 전남대학교 전자상거래 석사

2010년~현재 : 전남대학교 전자상거래 박사과정
2013년~현재 : ㈜자이숨 CEO
관심분야 : 전자상거래, 자동화 시스템, 모바일 인터넷
페이스, 클라우드 컴퓨팅

김진술



2001년 : Computer Science (BSC S), University of Utah, USA

2005년 : KAIST 정보통신공학과 디지털미디어공학전공 (공학석사)

2008년 : KAIST 정보통신공학과 디지털미디어공학전공 (공학박사)

2005년~2008년 : ETRI 한국 전자통신연구원 연구원

2009년~2011년 : 나사렛대학교 멀티미디어학과 교수

2012년~현재 : 전남대학교 전자컴퓨터공학부 교수
관심분야 : 방송·통신 융합미디어 처리, 디지털미디어 처리, 휴먼-컴퓨터 인터랙션, 클라우드컴퓨팅, 등

박상현



2010년 : 나사렛대학교 멀티미디어학과 (학사)

2010년~2012년: ㈜미디어플로우 시스템개발 개발원
2012년~현재 : 전남대학교 전자컴퓨터공학부 석사과정
관심분야 : 인터랙티브 미디어, 시스템 관리, 임베디드 시스템, 디지털미디어, 클라우드 컴퓨팅

이상준



1991년 : 전남대학교 전산통계학과 (이학사)

1993년 : 전남대학교 전산통계학과 (이학석사)

1999년 : 전남대학교 전산통계학과 (이학박사)

1995년~2005년: 서남대학교 경영전산정보학과 조교수

2005년~2007년: 신경대학교 인터넷정보통신학과 조교수

2007년~현재 : 전남대학교 경영학부 부교수

관심분야 : 경영정보시스템, 스마트컴퓨팅, 소프트웨어공학 등