

U-learning 을 위한 OSGi 에 기반한 USB 단말기 시스템 개발

Development of the OSGi-based USB Terminal System for U-learning

김희선*, 김지홍, 이창구
(Hee-Sun Kim, Jee-Hong Kim, and Chang-Goo Lee)

Abstract : U-learning (ubiquitous learning) systems, which deliver learning materials anytime and anywhere, allow learners to watch live lectures on PDAs, tablet PCs and notebook computers via broadband and wireless Internet. These systems have various problems; first, terminal devices are expensive, and it is difficult to maintain their efficiencies. Secondly, Internet does not guarantee quality of service (QoS), and in general it does not provide real-time services. Finally, the security of these systems is weaker in a local network than in an external network. The USB-based terminal system based on the OSGi service platform was designed as a ubiquitous system, in order to solve those problems. The USB terminals, used in this system, are inexpensive, and it is easy to maintain their performances. Also, this system solves the problems of security in a local network and provides guaranteed QoS. To accomplish this, the number of USB terminals connected to the system has to be limited according to the formula proposed in our paper. This system uses the OSGi specification as a middleware. It supports the discovery mechanism of the USB terminals, maintenance and administration of the system. Finally, this paper shows a driver's license testing system as an example u-learning application.

Keywords : ubiquitous system, u-learning system, terminal system, USB, OSGi

1. 서론

최근 유비쿼터스 기술의 발달과 보급이 이루어짐에 따라 다가 올 유비쿼터스 사회에 대한 사회적 관심이 증대되고 있어 공공·민간기관에서 관련 연구가 진행되고 있으며 정치, 행정, 교육, 경제, 의료 부문 등에서 광범위하게 형성될 것이다. 그 중 교육 부문에서 그 움직임이 활발하게 이루어지고 있으며 2005년도에 교육인적자원부는 학교정착화 환경을 '유비쿼터스화' 추진을 선포하였다. 이를 U-러닝이라 명칭 지었으며 U-러닝은 유비쿼터스 러닝(ubiquitous learning)의 약자로 무선인터넷과 초고속인터넷을 이용해 PDA(Personal Digital Assistant)단말기나 태블릿 PC(Personal Computer), 노트북 컴퓨터상에서 교육을 받거나 실시간으로 자료를 검색, 다운로드 받을 수 있는 교육 서비스이다. 이렇게 개발된 학교 교육과 가정 학습의 'U-러닝' 모델은 학습자 개인별 맞춤형, 수준별 학습 서비스를 제공하고 개인별 학습 능력을 신장시킬 예정이다[1,2].

PDA, 태블릿 PC, 노트북과 같은 개인용 단말기를 유·무선 인터넷을 사용하여 네트워크를 형성했을 시 여러 가지 문제점이 있다. 먼저 적게는 수십 명에서 많게는 수백 명이 사용할 개인 단말기를 PDA, 태블릿 PC, 노트북을 사용하여 그 비용이 만만치 않으며 유지보수 또한 어렵다. 그리고 인터넷은 기본적으로 QoS(Quality of Service)을 보장 못할 뿐 아니라 상호간 실시간 동작을 요구하는 서비스 경우 적절하지 않다. 또한 보안에서도 매우 취약하며 외부 네트워크보다 내부 네트워크에서 단말기를 통해 서버에게 치명적인 유해를 가할 수 있다.

본 연구는 이런 문제점을 해결할 수 있는 OSGi(Open

Service Gateway Initiative)에 기반한 USB(Universal Serial Bus) 단말기 시스템을 개발하였다. USB는 적은 비용, 확장성, 자동 설정, 핫 플러깅(Hot-plugging; 자동 인식, 접속, 제거 동작 중에 가능하게 하는 기술)등 많은 장점을 가지고 있으나 네트워크 통신 기술로 주목 받지 못하고 있다. 그 이유로는 USB가 갖고 있는 단점 때문이다. USB는 IEEE1394보다 상대적으로 낮은 전송 속도를 가지며 통신 케이블의 길이도 5미터(meter) 정도로 짧다. 그리고 하나의 USB 호스트에 의해 다수의 USB 디바이스를 제어하는 중앙 집중 통신 토폴로지를 방식을 채택하고 있기 때문에 분산 네트워크 형성에 매우 불리하다. 그러나 USB 기술의 발전은 이런 문제점을 조금씩 해결하고 있다. USB 2.0은 480Mbps의 통신 속도를 충족시키고 있으며 최근에 소개되고 있는 무선USB는 최대 50M 거리에 있는 장치들과 통신이 가능하다. 더욱이 무선USB 기술은 무선 네트워크 기술로써 블루투스(bluetooth)나 지그비(zigbee)와 경쟁이 가능하리라 본다[3].

OSGi는 가정 정보 기기 및 보안 시스템과 같은 인터넷 장비의 표준 연결 방법을 제안하고 있다[4]. OSGi에 기반한 USB 단말기 시스템은 OSGi 서비스 플랫폼을 탑재한 서버와 다수의 USB 단말기들이 고속의 USB 버스를 이용하여 네트워크를 구축하였다. 이 시스템은 개인 단말기를 저가의 USB 단말기를 사용했으며 아주 간단한 구조와 동작 원리를 갖고 있기 때문에 유지 보수가 쉽다. 네트워크는 USB 버스를 사용했기 때문에 USB 스펙에서 명시하고 있는 동기식 및 비동기식 전송방법을 사용하여 그에 맞는 다양한 서비스를 안정적으로 제공할 수 있다[3]. 그리고 단말기에서 이루어지는 모든 작업은 실제적으로는 OSGi 서버의 컨테츠 서비스 공간에서 제한적으로 이루어지기 때문에 단말기에서 서버에게 유해를 가할 방법은 전혀 없어 내부 네트워크에서 보안문제를 손쉽게 해결하였다.

본 논문의 구성은 다음과 같다. 먼저 OSGi에 기반한 USB 단말기 시스템 구성을 설명하고, 다음 3장과 4장에서는

* 책임저자(Corresponding Author)

논문접수 : 2007. 7. 2., 채택확정 : 2007. 10. 31.

김희선, 김지홍 : 전북대학교 제어계측공학과

(friendtd@empal.com/jeehong@chonbuk.ac.kr)

이창구 : 전북대학교 전자정보공학부(changgoo@chonbuk.ac.kr)

OSGi 서비스 플랫폼(service platform) 상에서 디바이스 액세스와 USB 디바이스/드라이버 서비스의 동작 원리에 대해서 기술하였다. 그리고 5장에서는 USB 단말기 드라이버 서비스를 설명하였으며 6장에서는 이를 이용한 실제 응용사례로서 운전면허 필기 시험을 구현하였다. 마지막 장에서는 본 연구에 대한 결론을 제시하였다.

II. USB 단말기 시스템 구성

USB 단말기 시스템은 그림 1과 같이 하나의 서버와 다수의 USB 단말기로 구성되어 있으며 USB 단말기들은 허브를 통해 확장되며 (USB 스펙상 USB 장치는 허브를 포함하여 총 127개의 USB 장치가 연결될 수 있다[3]) 허브를 제외하면 최대 100개 정도 연결될 수 있다. 그러나 시스템 용도에 따라 단말기당 필요한 전송속도를 계산하여 최대 USB 단말기 수를 결정해야 한다. 이론상 USB 2.0 스펙상 최대 전송속도가 480Mbps 이지만 실제 유효한 USB 호스트의 최고 전송속도는 320Mbps으로 알려져 있고 일반적으로 USB 호스트 그에 한참 못 미치는 속도를 지원하고 있다. 그 원인은 SOF(Start of Frame)간의 재 설정을 위해 소모되는 연산 시간 및 버스간 데이터 전송 시 발생하는 지연시간 등이 있다 [5]. 본 연구에서 사용하고 컴퓨터의 USB 호스트의 최고 속도를 측정된 결과 최고 116.9Mbps를 기록하였다. 따라서 단말기당 10Mbps의 전송속도를 확보해야 하는 시스템인 경우 USB 단말기 수를 10개 이내로 제한 해야 한다. 그렇지 않고 10개 이상의 단말기를 연결하기 위해서는 서버에 새로운 USB 호스트를 추가하여 전체 시스템의 대역폭을 높여 QOS를 보장해 주어야 한다. 또한 USB 버스의 길이를 5미터 이상 확장하기 위해서는 허브 또는 리피터(repeater)가 필요하며 이들 하나가 USB 버스의 전체 트래픽 요소 중 하나가 된다.

OSGi 서버에 연결된 다수의 USB 단말기들은 네트워크에 참여하는 개인에게 하나씩 지급되며 서로 다른 콘텐츠를 선택하여 자기가 원하는 서비스를 받을 수 있다. USB 단말기는 USB 컨트롤러, LCD 모듈, 터치 패널, 키패드 및 외부 확장 컨넥터로 구성되어 있다. USB 컨트롤러는 USB 버스를 통해 데이터를 주고 받는 일을 수행하며 LCD는 기본 출력 장치이며 키패드와 터치패널을 기본 입력 장치이다.

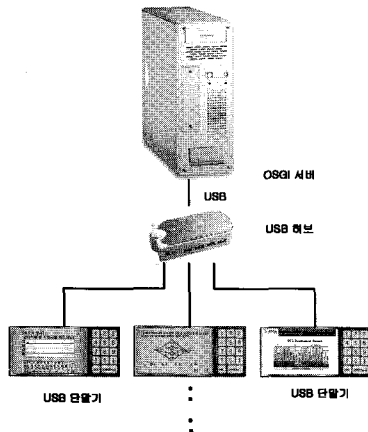


그림 1. OSGi에 기반한 USB 단말기 시스템.
Fig. 1. OSGi-based USB terminal system.

III. OSGi 표준과 디바이스 액세스

OSGi(Open Services Gateway Initiative) 표준은 가전 정보 기기 및 보안 시스템과 같은 인터넷 장비의 표준 연결 방법을 위해 OSGi 단체가 제안한 산업체 표준이다. OSGi 표준은 개방형 자바 내장형 서버 기반의 게이트웨이 소프트웨어를 만드는 것으로, 플랫폼, 응용 소프트웨어 등에 전혀 구애 받지 않고 보안 기능이 우수한 멀티 서비스를 장치나 설비에 제공하는 기능이 있다. 특히 블루투스, HAVi(Home Audio Video Interoperability), 홈 PNA(Phoneline Networking Alliance), 홈 RF(Radio Frequency), USB 등 다양한 유무선 네트워크 기술을 수용하는 개방형 네트워크 기술이다[6,7].

OSGi 서비스 플랫폼은 서로 다른 많은 개발업체들의 서비스와 디바이스들의 만남의 장소이다. 사용자가 서비스들을 추가할 때 새롭게 인스톨된 서비스는 자신의 목적에 맞게 입력 장치들을 찾고 그 장치들의 디바이스 드라이버를 로딩한다. 이런 일련 된 동작들은 OSGi 서비스 플랫폼상에서 프레임워크가 동작 중에 수시로 일어난다. USB와 같이 동적 플러그를 지원하는 기술들은 OSGi 서비스 플랫폼상에서 쉽게 이런 메커니즘을 구현할 수 있다. OSGi에서 정의한 디바이스 액세스(device access) 표준은 OSGi 서비스 플랫폼상에서 장치들을 자동으로 탐색하여 찾고 추가 하며 요구에 의해 디바이스 드라이버를 다운로드하고 설치하는 기술을 말한다.

본 연구에서 USB 장치들을 자동으로 추가 및 삭제하는 USB 베이스 디바이스/드라이버 서비스(USB base device/driver service)와 USB 단말기들을 위한 USB 단말기 디바이스/드라이버 서비스(USB terminal device/driver service)를 구현하였다. USB 베이스 드라이버는 디스커버리 베이스 드라이버(discovery base driver)이다. USB와 같이 하드웨어가 장치를 자동으로 인식하고 장치의 명세서를 획득하는 메커니즘을 제공할 때 디바이스 서비스(device service)가 자동 인식되어 등록될 수 있

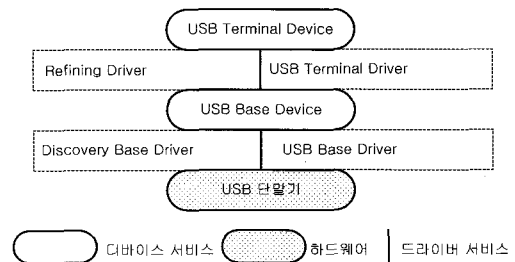


그림 2. USB 단말기 디바이스/드라이버 서비스.
Fig. 2. USB terminal device/driver services.

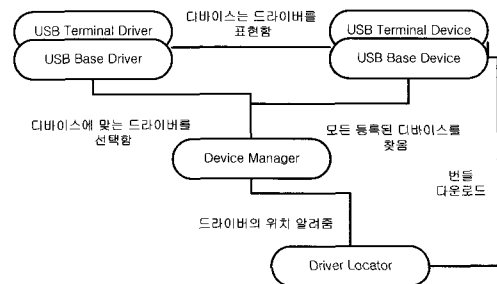


그림 3. 디바이스 액세스 클래스의 전체 구조.
Fig. 3. Device access classes for USB terminals.

도록 하는 드라이버를 디스커버리 베이스 드라이버라 한다. USB 베이스 드라이버는 USB 버스에 어떤 USB 장치가 연결되며 자동으로 USB 장치를 인식하고 USB 장치의 명세서를 획득하여 그에 해당하는 USB 디바이스 서비스를 찾고 자동으로 다운로드하고 설치한다. USB 단말기 드라이버는 리파이닝 드라이버(refining driver)이다. 디바이스 드라이버의 두 번째 범주를 리파이닝 드라이버라 부른다. 리파이닝 드라이버는 이미 프레임워크에 등록된 디바이스 서비스에 의해 표현된 물리적 디바이스의 새로운 관점을 제공한다. USB 단말기 드라이버는 USB 단말기를 대표하는 USB 단말기 디바이스 서비스에 의해 추가된다. 그림 2는 USB 디바이스 액세스 클래스(device access classes)의 추상적인 개념을 보여 주고 있다.

디바이스/드라이버 서비스는 OSGi 서비스 플랫폼 상에서 이들을 관리할 디바이스 매니저(device manager)와 드라이버 로케이터(driver locator)가 필요하다. 그림 3에서 보여 주듯 디바이스 매니저는 디바이스 서비스들의 등록과정을 도와주며 드라이버 서비스들간의 연계관계를 책임진다. 이런 과정은 드라이버 로케이터의 도움을 받는데 드라이버 로케이터는 디바이스 매니저가 드라이버 번들을 찾고 설치하는 것을 도와준다[8].

IV. USB 단말기의 네트워크 PnP (Plug and Play)

USB 단말기는 시스템에 자유롭게 연결 및 제거되며 그에 해당하는 드라이버 서비스 번들이 자동으로 설치 및 삭제되어야 한다. 그림 4는 OSGi 서비스 플랫폼 중 하나인 Oscar[9]에서 USB 단말기 드라이버가 자동으로 설치 및 제거되는 것을 보여 주고 있다. 이는 앞 장에서 설명한 것처럼 디바이스 액세스 클래스들(디바이스 매니저, 드라이버 로케이터, USB 베이스 디바이스와 USB 베이스 드라이버)에 의해서 이루어진다.

그림 5와 6은 USB 단말기가 연결될 때와 제거될 때 디바이스 액세스 클래스들간의 메커니즘을 보여 주고 있다.

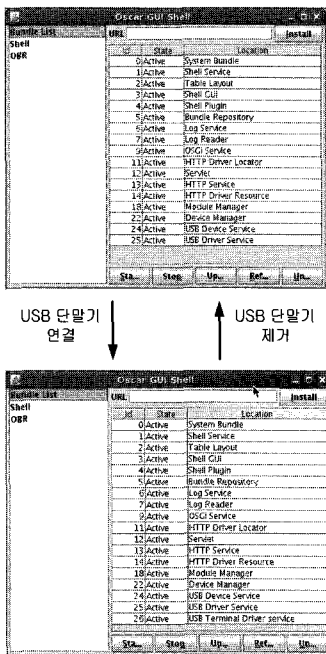


그림 4. USB 단말기 드라이버 설치 및 삭제 장면.
Fig. 4. The USB terminal driver added/removed.

이 액세스 클래스들간의 상호 데이터 흐름과 동작 메커니즘을 보여 주고 있다. 먼저 그림 5는 USB 단말기가 시스템에 추가되었을 때 USB 단말기 드라이버를 찾아 다운로드

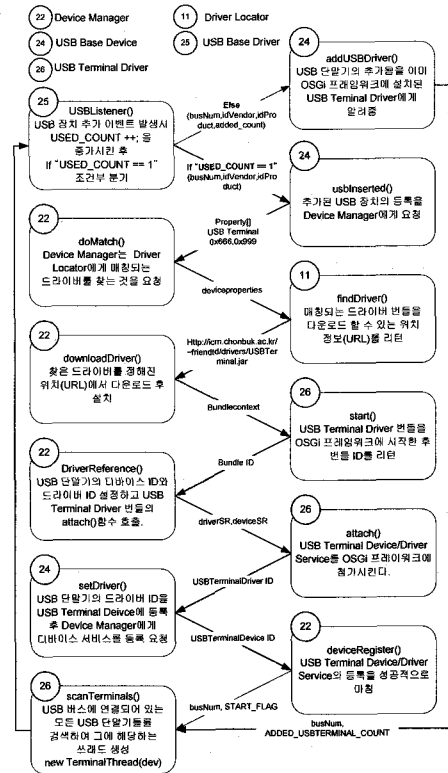


그림 5. USB 단말기가 연결될 때 디바이스 액세스 클래스들간의 메커니즘.
Fig. 5. Mechanism among device access classes when USB terminals are connected.

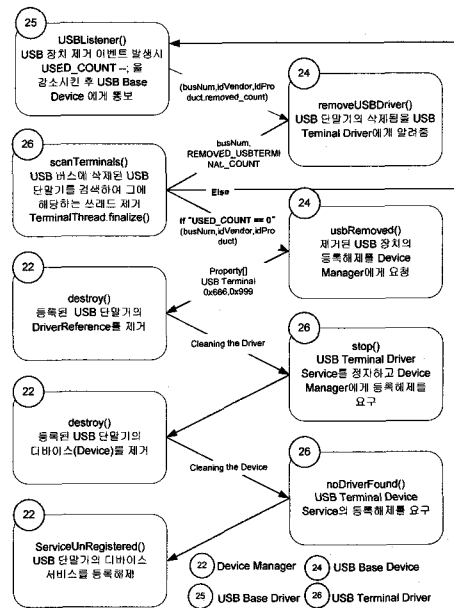


그림 6. USB 단말기가 제거될 때 디바이스 액세스 클래스들간의 메커니즘.
Fig. 6. Mechanism among device access classes when USB terminals are disconnected.

후 설치, 등록하는 과정을 보여주고 있다. 새로운 USB 장치가 연결되면 USB 베이스 드라이버는 새로운 USB 장치의 vendor ID와 product ID를 획득한다. 만약 USB 장치가 처음으로 연결된 거면(만약 USED_COUNT=1 이라면) 추가된 USB 장치의 등록을 디바이스 매니저에게 요청한다. 장치정보를 전달받은 디바이스 매니저는 연결된 USB 장치에 맞는 드라이버 서비스를 찾는다. 만약 새로운 USB 장치가 USB 단말기라면 드라이버 로케이터는 USB 단말기 드라이버를 지정된 서버로부터 다운로드한 후 설치한다. 설치된 USB 단말기 드라이버는 bundle ID와 USBTerminalDriver ID를 드라이버 매니저에게 보내 등록을 한 후에 USB 버스에 있는 모든 USB 단말기 장치를 검색하여 각 장치마다 쓰레드를 생성한다. USB 단말기 드라이버가 설치된 후에는 새로운 USB 단말기가 연결되면 그에 해당하는 쓰레드만을 생성한다.

그림 6은 시스템으로부터 USB 단말기가 제거될 때 USB 디바이스 액세스 클래스간의 상호 데이터 흐름과 동작 메커니즘을 보여 주고 있다. 이때 설치된 USB 단말기 드라이버 서비스 번들은 모든 USB 단말기가 제거되며(만약 USED_COUNT = 0 이라면) 자동으로 삭제된다. 그렇지 않은 경우에는 제거된 단말기에 해당하는 쓰레드만을 제거한다.

V. USB 단말기 드라이버 서비스

앞장에서 기술한 것처럼 다수의 단말기들이 자유롭게 시스템에 참가와 탈퇴를 하며 원하는 서비스를 받을 수 있도록 OSGi 프레임워크(OSGi framework)상에서 서비스 번들(service bundle)을 설계하였으며 그 중 USB 단말기 드라이버는 USB 베이스 드라이버로부터 USB 단말기의 연결 및 제거 이벤트를 받아 쓰레드를 생성 또는 삭제한다. 하나의 USB 단말기가 연결될 때 마다 쓰레드가 하나씩 자동적으로 생성되며 각각의 쓰레드는 연결된 USB 단말기와 논리적인 USB 통신 라이프라인을 생성한다. USB 단말기들은 콘텐츠 데이터 베이스(contents data base)에 접근하여 자기가 원하는 콘텐츠를 선택하여 서비스를 받을 수 있다. 그림 7은 3개의 USB 단말기들이 각각 다른 콘텐츠 서비스를 받고 있는 시나리오를 보여 주고 있다.

USB 단말기 드라이버는 선택된 서비스를 콘텐츠 데이터 베이스(contents data base)를 통해 가지고 오며, 콘텐츠 서비스를 위한 LCD 프레임 데이터를 USB 단말기에 보내준다. USB 단말기는 LCD에 프레임 데이터 받아 LCD에 즉시 출력하며

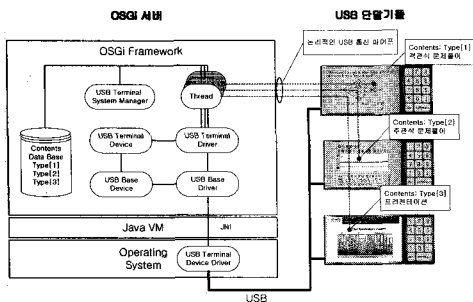


그림 7. OSGi에 기반한 USB 단말기 시스템의 소프트웨어 구성도.

Fig. 7. Middleware of the OSGi-based terminal system.

또한 USB 단말기의 키 값이나 LCD 터치 패널 값을 USB 단말기 드라이버에게 보내 준다. 즉, USB 단말기는 단순히 LCD 프레임 데이터 값을 받아 출력하고 키 값이나 LCD 터치 패널 값을 보내주는 단순한 일을 수행하고 콘텐츠의 데이터 및 사용자 데이터의 판단과 처리에서 저장까지 USB 단말기 드라이버가 하게 된다.

USB 단말기 드라이버는 JNI(Java Native Interface)을 통해 USB 단말기 디바이스 드라이버(USB terminal device driver)와 인터페이스를 하여 USB 단말기들과 통신한다. USB 단말기 디바이스 드라이버는 디바이스와 시스템 메모리 간에 데이터의 전달을 담당하는 커널 내부 기능이다. 드라이버는 위쪽으로는 파일시스템과 인터페이스를 가지며 아래쪽으로는 USB 단말기의 하드웨어와 인터페이스를 갖는다. 드라이버는 완전히 단일 쓰레드로 수행되며 다수에 USB 단말기는 각각 자신의 파일시스템 노드(/dev/usbterminal0, /dev/usbterminal1, /dev/usbterminal2)로 구별된다. 드라이버는 각각이 다른 프로세스에 속한 여러 시스템 콜에 대해서 순차적으로 반응 한다 [10,11].

콘텐츠 데이터 베이스는 USB 단말기가 서비스 받을 수 있는 콘텐츠와 그로 인해 발생한 사용자의 저장 데이터가 모두 저장되고 USB 단말기 시스템 관리자(USB terminal system manager)는 콘텐츠 데이터 베이스를 접근하여 각 단말기들의 사용 정보 및 결과 등을 출력 하며 각 USB 단말기들의 쓰레드를 관리하여 시스템을 설정하고 실시간 모니터링 한다.

VI. 실험 및 결과

본 연구에서는 USB 단말기 시스템을 이용하여 운전 면허 필기 실험을 실시하였다. 그림 8과 같이 두 개의 단말기를 사용하여 각기 다른 문제 유형으로 시험을 실시하였으며 USB 단말기 시스템 매니저는 그림 9와 같이 웹을 통해 단말기들을 관리한다. 설정 페이지는 각 단말기들의 시험 유형 선택 및 단말기의 가동 유무를 설정할 수 있고 감시 페이지는 각 단말기의 상태를 모니터링 할 수 있으며 결과 페이지는 각 단말기의 실험 결과를 알려준다.

이와 같이 USB 단말기 시스템 활용을 통하여 시험을 봤을 때 다음과 같은 유익한 역할과 기술을 제공받을 수 있다. 첫째, 시험 문항에 대한 반응 결과와 시험 결과를 신속하고 효

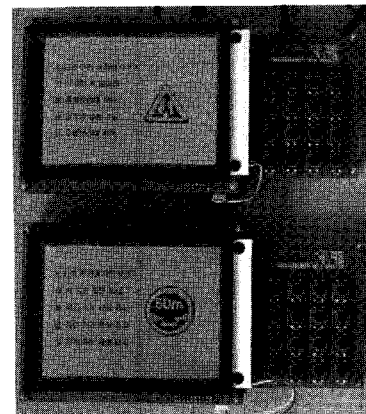


그림 8. USB 단말기들의 실행 사진.

Fig. 8. The picture of tested USB terminals.

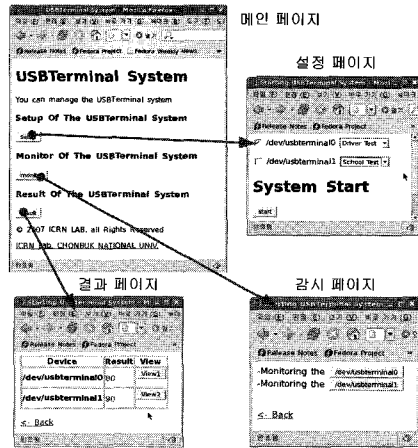


그림 9. USB 단말기 시스템 관리자의 실행 화면.
Fig. 9. The picture of executed USB terminal system manager.

울적으로 처리할 수 있다. 즉 시험을 실시한 직후에 시험 문항의 반응 결과를 즉시 알아볼 수 있다. 둘째, 각종 정보의 저장 및 기억이 용이하며 분석에 필요한 주변 기기의 통제가 가능하다. 즉 다양하면서도 대량의 검사 자료를 저장할 수 있는데 이와 같이 광범위한 정보에 대한 데이터베이스를 구축함으로써 검사에 대한 새로운 접근과 연구가 가능하게 될 것이다. 셋째, 저렴한 비용과 신뢰도를 확보할 수 있다. 이는 실시 요령과 문항 제시의 방법이 일정하게 통제 되어질 수 있고 피험자의 신체적 특성이나 상황에 영향을 받지 않고 시험을 실시할 수 있다[12].

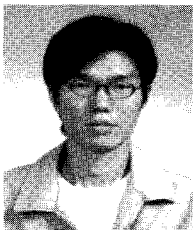
VII. 결론

OSGi에 기반한 USB 단말기 시스템은 다수의 USB 단말기들과 OSGi를 탑재한 서버가 고속의 USB 버스를 이용하여 네트워크를 구축한 유비쿼터스 시스템이다. 이 시스템은 6장에서 소개한 응용 사례에서 볼 수 있듯 기존의 PDA나 PC를 사용하는 네트워크 시스템을 대신할 수 있으며 이들 시스템이 갖는 여러 문제들도 해결할 수 있다. 즉, 이 시스템의 USB 단말기를 저가이며 간단한 구조와 동작 원리를 갖고 있기 때문에 유지 보수가 쉽고 단말기에서 이루어지는 모든 작

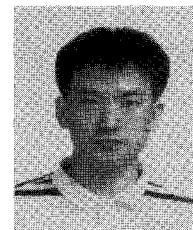
업은 실제적으로는 OSGi 서버의 컨텐츠 서비스 공간에서 제한적으로 이루어지므로 보안에 유리하며 컨텐츠 기반 서비스를 제공하고 있기 때문에 시스템 활용 목적에 맞추어 컨텐츠만 개발하며 다양한 분야에 적용이 가능하다. 예를 들어 학교의 일반 시험문제 풀이를 이 시스템을 이용하여 부정행위를 줄일 수 있을 뿐만 아니라 한 반에서 학생들이 원하는 타입의 문제 유형이나 과목을 선택하여 시험을 볼 수 있다. 그리고 수준별 학습용 시스템, 맞춤형 설문 조사 프로그램, 다수가 같이 하는 게임 시스템으로도 사용이 가능하다.

참고문헌

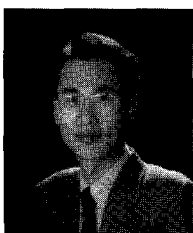
- [1] B. C. Doherty, P. T. O'Hare, M. J. O'Grady, and G. M. P. O'Hare, "Entre-pass: Personalising u-learning with Intelligent Agents," *WMUTE '06*, pp. 58-62, Nov. 2006.
- [2] G-J. Hwang "Criteria and strategies of ubiquitous learning," *SUTC '06*, vol. 2, pp. 72-77, 2006.
- [3] The OSGi appliance, *OSGi Service Platform Core Specification*, Release 4, OSGi Alliance, August 2005.
- [4] J. Axelson, *USB Complete: Everything You Need to Develop custom USB peripherals*, 3rd Ed, Lakeview Research, Dec. 2005.
- [5] I. Chiu, William DeVercelly, Everything USB, <http://www.everythingusb.com/usb2/faq.htm>
- [6] 김희선, 이창구, "Jini Surrogate에 기반한 광대역 PLC 홈 제어기 개발," 제어 · 자동화 · 시스템공학논문지, 제 12 권 제 1 호, pp. 1-8, 2006.
- [7] 김희선, 이창구, "USB에 기반한 홈 제어 시스템 개발," 제어 · 자동화 · 시스템공학논문지, 제 12 권 제 4 호, pp. 405-410, 2006.
- [8] The OSGi Alliance, *OSGi Service Platform Service Compendium*, Release 4, OSGi Alliance, August 2005.
- [9] Richard S. Hall, Oscar forum, <http://oscar.objectweb.org/>.
- [10] S. Liang, *The Java Native Interface Programmer's Guide and Specification*, Addison-Wesley, June 1999.
- [11] J. Corbet, Alessandro Rubini, and Greg Kroah-Hartman, *Linux Device Drivers*, 3rd Ed, O'REILLY, June 2001.
- [12] 서정희 외, "미래교육을 위한 u-러닝 교수-학습 모델 개발," 한국교육학술정보원 연구 보고, 12월, 2005.



김희선
1997년 전북대학교 제어계측공학과 학사. 1999년 전북대학교 전기공학과 석사. 2006년 전북대학교 제어계측공학과 박사. 관심분야는 홈 네트워크, 내장형 시스템.



김지홍
2000년 전북대학교 제어계측공학과 학사. 2003년 전북대학교 제어계측공학 석사. 2005년~현재 전북대학교 제어계측공학과 박사과정 수료. 관심분야는 내장형 시스템, 로보틱스, 미들웨어.



이창구
1981년 전북대학교 전기공학과 학사. 1983년~1991년 한국전자통신연구원 선임연구원. 1991년 전북대학교 전기공학과 박사. 1992년~현재 전북대학교 전자정보공학부 교수. 관심분야는 지능제어, 홈 제어 시스템.