

RFID와 USN을 사용한 지능형 주차관리 시스템 구축

서희석*

Intelligent Parking Management System using RFID and Sensor

Seo, Hee Suk

〈Abstract〉

There are several ways to solve parking problem in the downtown. One of them is to manage parking efficiently, that is, to raise use-efficiency of parking place and to help in the block efficiently, it can prevent disproportion that some parking space is over capacity and the other parking one is under. It causes not only the disproportion of economic efficiency but also the congestion of local road. The precondition to manage parking efficiently is to analyse the parking characteristics.

In this paper we build up total D/B by collecting every information about parking space. We developed the Parking Management System by using RFID and Sensor. The efficient Parking Management System is for convenience of drivers and efficient usage of parking space. And this system is also the basic study to be developed as the total Parking Management System by upgrading every information.

Key Words : Parking Management System, RFID, Sensor, ZigbeX

I. 서론

우리나라도 이제는 넘쳐나는 차량들로 도로가 몸살을 앓게 된지 오래고 주차장 문제 역시 심각하게 대두되기 시작하였다. 주차장의 땅 값과 주차장을 관리하는 인건비의 상승으로 주차관리 경비가 상승하고 이는 곧 차량 소유주들에게 적지 않은 부담을 안겨주고 있다.

비좁은 도심의 공간에는 빌딩의 지하와 옥상 등에 주차장이 들어서 있으며, 이는 주차관리에 새로운 문제점을 야기하게 되었다. 특히 지하주차장의 증가는 차량의

요금징수 및 주차관리, 범죄 등에 관한 여러 문제점들이 발생하게 되어 새로운 사회 문제까지 대두되고 있다.

특히 백화점 등 고객 확보를 위해 무료주차장 실시하는 기업들에게는 주차장 관리 문제는 기업 자체의 부담 증가가 되고 있는 실정이다. 그리고 차량을 주차시키는 소유주들의 입장에서는 차량 주차시 주차공간을 찾기 위해 주차장 안을 이리저리 돌아다니게 되어 지하 주차 공간의 공기 오염과 고객의 소중한 시간낭비 초래 및 주차장 내의 차량혼잡으로 이어지고 있다.

또 주차장에 차량을 주차시켜놓고 일을 마친 다음 자신의 차량이 주차된 장소를 찾지 못해 애를 먹는 주부들

* 한국기술교육대학교 인터넷미디어공학부 교수

과 일부이지만 주차권을 분실하는 사고와 차량의 도난 사고도 적지 아니 있어 이에 대한 종합적인 대책이 있어야 하는 실정이다[1-3].

지금까지 주차관리에 관한 여러 가지 해법들이 제시되었지만 아직까지 사람이 주차관리를 하는 주차장들이 많이 있는 실정이다. 이런 것은 주차관리 시스템이 전체적으로 자동화, 전산화가 이루어지지 않았기 때문이며, 이것은 주차공간의 특수성, 관리의 특수성, 전자동 시스템 설치 및 운영 경비 등의 여러 문제가 혼재하고 있다고 생각된다. 그러나 날로 증가해 가는 차량과 보다 많은 차량들을 주차시키기 위한 효율적인 주차관리 시스템은 계속 변화되고 또 발전되어야 한다고 본다.

이에 따라 최근 날로 그 용도가 증가하고 있는 RFID와 센서를 주차장 관리에 도입하여 보다 효율적인 주차관리 시스템을 구성하는 것에 대한 연구를 해보고자 한다. 기존 네트워크와 접속되어 무선으로 운영되는 센서 네트워크 시스템인 ZigbeX센서와 RF시스템을 결합하여 각 클라이언트 시스템에 연동함으로써 여러 곳의 주차장을 이용하는 고객들에게 각 주차장의 상황을 알려주는 맞춤형 주차 네비게이션 시스템을 제공하여 보다 신속하고 편리함을 제공할 것으로 기대된다.

본 논문은 다음과 같이 구성된다. 2장에서는 주차관리 시스템에 대한 DEVS 방법론과 주차관리시스템의 전체 구조에 대해서 알아보고, 3장에서는 RFID와 센서를 이용한 주차관리시스템의 기능, 구성 및 구현에 대한 내용을 설명하고 4장에서는 결론과 향후 연구 과제에 대해 논의한다.

II. 주차관리시스템에 대한 DEVS 모델링

2.1 DEVS 방법론

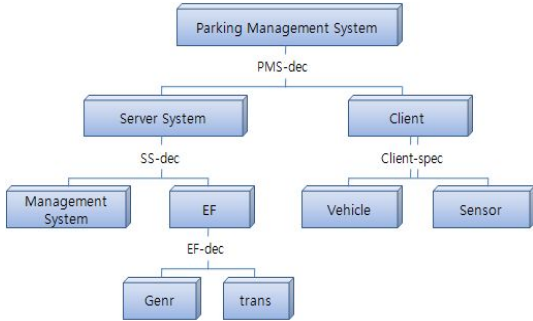
Zeigler에 의해 정립된 DEVS 방법론은 연속적인 시간 상에서 발생하는 이산 사건을 처리하는 시스템을 시뮬레

이션 하기 위해 이론적으로 정립된 모델링 방법론이다. 이는 모델의 구조와 행동을 시뮬레이션 수행으로부터 추상화시키기 위해 모델을 집합 이론적 방법으로 이용한 것으로, 시스템을 계층적(hierarchical)이고 모듈화(modular)된 형식으로 기술한다[4-6].

DEVS에서는 기본 (Basic) 모델과 결합 (Coupled) 모델을 정의한다. 기본 모델은 시스템의 동적인 특성을 표현하기 위한 모델이고, 결합 모델은 시스템의 구성 요소 간의 상호 작용을 표현하기 위한 모델이다.

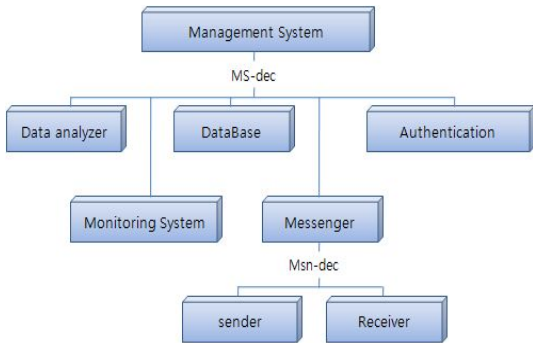
2.2 주차관리시스템의 전체 구조

<그림 1>은 주차관리시스템에 대한 전체 구조도이다. PMS (Parking Management System) 모델은 서버시스템 모듈과 클라이언트 모듈로 구성된다. 서버시스템 모듈은 주차장에 출입하는 모든 차량을 관리하고 처리하는 모델로서 Management System 모듈과 EF 모듈로 구성된다. Client 모듈은 실제 환경에 배치되는 모델로서 주차장 입구에서 RFID Reader에 의해 감지되는 RFID를 장착한 차량 모듈과 주차공간에 주차된 차량을 감지하는 Sensor 모듈로 구성된다. Management System 모듈은 MS 자체에 대한 모델이며 EF 모듈은 시뮬레이션의 입력을 생성하는 모듈로서 시뮬레이션의 입력을 생성하는 Genr 모듈과 시뮬레이션의 통계 수집을 위한 Trans 모듈로 구성된다. 본 구조에 따른 시뮬레이션의 수행은 다음과 같다. 시뮬레이션 수행을 위해서 Genr 모듈에서는 차량의 수, 주차공간의 수 등의 환경에 따른 입력을 발생하여 Management System모듈로 전달하게 된다. Management System 모듈은 이를 수신하여 자신의 각 영역에 입력값을 전달하여 시뮬레이션의 초기화를 수행한다.



<그림 2> 관리시스템의 구조도

2.3 관리시스템의 모델링

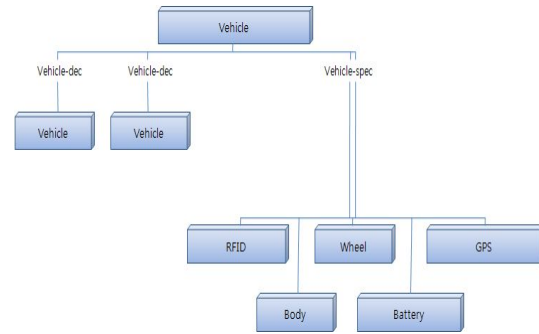


<그림 2> 관리시스템의 구조도

<그림 2>는 관리시스템에 대한 구조도를 상세하게 표현한 그림이다. Management System 부분은 Data analyzer, Monitoring system, Database, Messenger, Authentication 모듈로 구성된다. 각 Sensor들이 보내온 정보들을 Messenger 모듈의 Receiver 모듈을 통해 수신한다. 정보들은 인증 모듈을 통해 검사되고 Data analyzer 모듈을 통해 분석, 처리된 후 DataBase에 저장된다. 저장된 정보들은 실시간으로 Monitoring System에 업데이트된다.

2.4 차량의 모델링

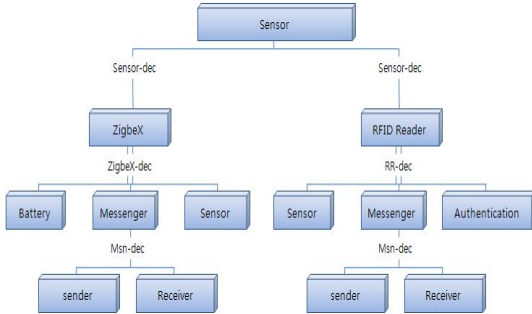
<그림 3>는 차량의 구조도이다. 각 차량은 RFID, Body, 바퀴, 배터리, GPS으로 구성된다. 차량에 장착되어 있는 RFID는 차량이 주차장에 들어왔다는 것을 알려준다.



<그림 3> 차량의 구조도

2.5 센서의 모델링

<그림 4>은 센서 모듈의 구조도이다. RFID Reader 모듈은 ZigbeX의 RFID Sensor 모듈을 통해 주차장을 출입하는 모든 차량을 감지하고 Messenger 모듈의 Sender 모듈을 통해 차량의 기본정보를 Server System로 읽어 드린다. RFID Reader 모듈은 인증 모듈을 통해 이를 검사한다. 주차 공간에 차량이 주차가 되면 ZigbeX 모듈은 조도 Sensor 모듈을 통해 이를 알아채고 Messenger 모듈의 Sender 모듈을 통해 주차공간의 영역을 Server System에 알린다. 차량이 출차가 되면 ZigbeX 모듈의 조도 Sensor 모듈은 차량이 출차된 정보를 Messenger 모듈의 Sender 모듈을 통해 Server System에 알린다. 차량이 주차장 입구를 벗어가면 RFID Reader 모듈은 ZigbeX의 RFID Sensor 모듈을 통해 출차된 정보를 확인한다.

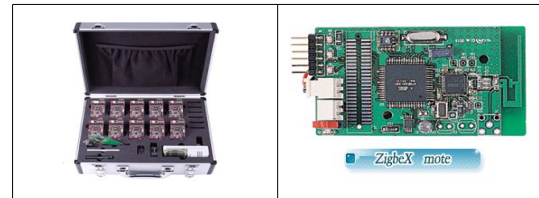


<그림 4> 센서의 구조도

- 각 Zigbee 센서를 통한 주차유무 확인
- 센서를 통한 주차 유무를 사용자에게 보여 줄 인터페이스 구현
- Zigbee 센서와 DB 연동을 통한 인터페이스 통합
- 각 센서들 모델에 적용

3.2 구성

3.2.1 하드웨어 구성



<그림 5> RFID 와 ZigbeX

III. 주차관리시스템의 구현

3.1 기능

1) 센싱 기능

- MFC로 제작한 프로그램으로 센서에서 읽어들이는 값들 중 조도 값을 읽어 들여 DB에 기록

2) 주차 공간 시뮬레이션

- DB에 저장된 값을 MySQL과 PHP를 연동하여 FLASH_MX를 이용한 주차 공간 시뮬레이션

3) ID센싱 기능 & 장소에 따른 주차 공간 시뮬레이션

- RFID리더기를 통하여 차량 내부에 부착되어 있는 RFID태그 값을 읽어 들여 ID를 읽어 들인 장소에 따라 다르게 제공되는 실시간 주차 공간 시뮬레이션을 제공

4) 데이터베이스 구축을 통한 고객 관리 기능

- 데이터베이스에 사용자의 인적사항을 기록함으로써 고객관리

5) Etc

- 각 Zigbee 센서들간의 통신 환경 구축
- Base Mote와 서버 컴퓨터간의 USB방식의 Serial 통신 구축
- 각 Zigbee 센서에 대한 주차 유무 확인 알고리즘
- 각 Zigbee 센서들에 대한 정보 저장 DB구축

<그림 5>의 좌측 그림은 한백전자의 HBE-ZigbeX kit를 보여주는 사진[9, 10]이고 구성품중에서 RFID reader, RFID tag card, ZigbeX mote를 사용하였다. 우측 그림은 주차장에 차를 인식하기 위해서 기본 센서인 ZigbeX mote의 모습을 나타내고 있다. 기본 센서는 온도, 조도, 습도 센서, TRC 기본 장착을 하고 있다. 본 논문에서는 RFID Reader는 주차장 입구, RFID tag card는 차량, ZigbeX mote는 주차장 영역 조도센서로 쓰였다.

1) Micro Controller

- ATmega128 (program 128Kbyte SDRAM 2KB EEPROM 4KB AD 10bit 8Channel)

2) RF part

- CC2420 2.4GHz Zigbee(IEEE 802.15.4)

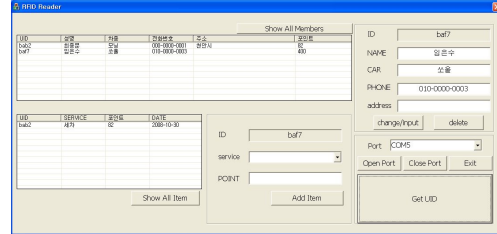
3) Security

- DSSS

4) Transfer BPS

- Maximum 250K BPS

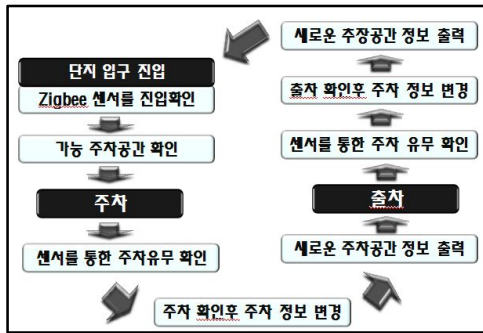
- 5) Base sensor
 - 온도, 습도, 조도 기본 장착, RTC기본 탑재
- 6) Power
 - 1.5V AA 2ea 또는 1.2V Rechargeable battery 2ea
- 7) RFID Reader DEVICE
 - USB단자와 연결되어 RFID태그 값을 읽어들이기 위한 장치
- 8) RFID
 - 고유 태그값을 가지고 있는 ID카드



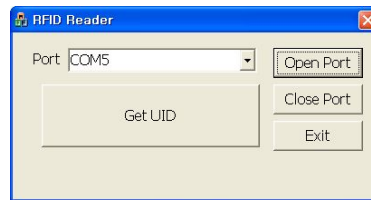
<그림 7> 주차장 메뉴 구성 화면

그림이다.

3.2.2 프로그램 구성



<그림 6> 주차 Navigation System 프로세스



<그림 8> 서버 프로그램 구동 화면

<그림 8>은 주차장에 차량이 진입하였을 경우 RFID ID TAG를 읽어들이기 위한 프로그램으로써 card TAG 정보가 들어왔을 때 기존에 저장되어 있는 사용자 ID라면 주차장 시뮬레이션 화면 <그림 9>를 로드하고 화면상에 보여주기 위한 프로그램이다.

<그림 6>은 주차 Navigation System 프로세스로 차량이 주차장 입구에 진입하면 RFID로 차량의 진입을 확인하고 <그림 9>의 화면을 띄워준다. 차량의 주인은 주차 가능한 위치를 파악하고 주차공간에 주차를 한다. 조도 센서를 통해 주차된 공간의 영역이 DB로 보내어지고 실시간으로 주차 정보가 변경되어진다. 반대로 차량이 출차를 하게 되면 빈 공간의 영역이 나타나고 주차 정보가 변경되어진다.



<그림 9> 주차장 시뮬레이션 화면

<그림 9>는 플래쉬를 바탕으로 만들어진 프로그램은

3.2.3 화면 구성

<그림 7>은 주차장에 차량이 진입하였을 경우 RFID에 저장되어 있는 사용자 차량의 정보를 보여주는 프

로써 센서를 이용한 주차장 상태는 계속 DB에 저장된다. DB에 저장되어 있는 주차장 정보를 일정한 시간마다 지속적으로 읽어들이고 상태를 최신화 하도록 설정이 되어 있다. 또한, <그림 8>의 프로그램에서 다시 한번 ID를 읽어드려 출차가 되었다는 것을 확인하면 다시 화면이 상태를 최신화 하도록 되어 있다.

3.3 주요 함수

RFID ID TAG를 읽어 들이기 위해서는 Reader기와 프로그램 간의 상호 데이터 통신이 가장 중요하고, 또한 TAG 정보에는 고유 ID 값이 있지만 그 밖에도 다른 정보들이 내재되어 있기 때문에 필요한 부분인 고유 ID 값을 추출하기 위한 알고리즘이 주요 함수[7-10]이다.

ParkDlg 클래스 : 데이터 통신을 하기 위한 Reader기의 데이터를 불러오기 전에 설정을 담당하는 클래스로써 응용프로그램이 시작되는 동시에 초기화 작업을 해주고 기본적인 포트 열기와 닫기, ID 읽기 함수를 정의 해 놓았다.

CommThread : 통신 포트가 열렸을 때 통신 포트로 오는 정보들을 전부 읽어들이는 역할을 수행한다. 이 때, 기본적으로 Reader기의 모드세팅과 포트 설정이 정확히 안 되어 있을 경우 정보가 읽어 드러지지 않으며 정확한 자료의 크기를 모르면 필요없는 값들이 저장된다.

parseMSG(void) : 들어온 16진수의 정보들을 정수형 데이터로 저장하기 위한 함수로 통신 클래스에서 보내진 정보를 가공하는 역할을 한다. 여기서 저장된 값들에서 필요한 정보만을 추출해 내고 그 정보를 바탕으로 프로그램을 구동한다.

네비게이션 시스템은 통신 연결이 추가 되기 때문에 CommThread클래스의 사용은 동일하다. 다만 읽어오는 데이터의 ID카드는 ID에 대한 부분과 데이터에 대한 부분의 구분되어 전달되어지기 때문에 고유 값인 ID를 읽어와야만 하는 알고리즘이 필요하다

ParkDlg 클래스: 포트를 통해 카드 ID값을 읽어와서 해당 구역으로 신호를 주기 위한 클래스
void GetUID(): //포트에서 ID값을 읽어오기 위한 준비를 한다. 초기화 세팅
bool CheckReaderState(): //리더기를 통해 ID카드를 리더기에 체크시 데이터를 읽어온다
bool Start15693Cmds(): //리더기의 모드가 14592와 15693의 두가지 모드중 ID값만을 형식이 간단하고 ID값을 쉽게 처리할수 있는 15693모드를 선택한다. (초기화 세팅시)
void parseMSG(): //읽어온 데이터를 16진수 형태로 변환
afx_msg void OnBnClickedOpenPort(): //포트 열림
afx_msg void OnBnClickedClosePort(): //포트 닫힘

CommThread:

통신 클래스. 포트에서 데이터를 읽고 쓰는 작업을 수행

클래스 기능

포트에서 읽기 :

포트를 연 후에 포트에 자료가 도착하면 WM_COMM_READ 메시지가 메인 윈도우에 전달된다. ON_MESSAGE 매크로를 이용, 함수를 연결하고 m_ReadData String에 저장된 데이터를 이용 읽기

포트에 쓰기 :

WriteComm(buff, 30)과 같이 버퍼와 그 크기를 건네면 된다.

클래스 함수와 변수 설명

CommThread: CQueue

BYTE buff[BUFF_SIZE]: //큐의 버퍼 사이즈

int m_iHead, m_iTail: //큐 버퍼에 데이터를 넣고 빼 올 때 사용할 변수

void Clear(): //버퍼를 초기화.

int GetSize(): //현재 버퍼에 들어있는 데이터의 size를 리턴

BOOL PutByte(BYTE b): //큐버퍼에 1바이트를 넣음

BOOL GetByte(BYTE *pb): 큐버퍼에서 1바이트를 빼 옴

CCommThread: CCommThread

환경 변수

BOOL check;

HANDLE m_hComm: //통신 포트 파일 핸들

CString m_sPortName: //포트 이름 (COM1..)

BOOL m_bConnected: //포트가 열렸는지 여부를 나타냄.

OVERLAPPED m_osRead, m_osWrite: //포트 파일 Overlapped structure

HANDLE m_hThreadWatchComm: //Watch함수 Thread 핸들.

```
WORD m_wPortID: WM_COMM_READ와 함께 보내는 인
수.
CQueue m_QueueRead: //큐버퍼

//외부 사용 함수//
BOOL OpenPort(CString strPortName, DWORD dwBaud,
BYTE byData, BYTE byStop, BYTE byParity): //포트 열기
void ClosePort(): // 포트 닫기
DWORD WriteComm(BYTE *pBuff, DWORD nToWrite): //
포트에 데이터 쓰기

//내부 사용 함수//
DWORD ReadComm(BYTE *pBuff, DWORD nToRead): //
포트에서 데이터 읽어오기
```

```
MoteMonDlg
: 포트에서 읽어 들인 데이터를 인터페이스 화면으로 표시
하기 위한 다이얼로그 클래스

CCommThread m_ComuPort: //포트 연결을 위한 Thread
클래스
int baudrate: //포트 전송속도 설정 변수
CString portStr: //열기 위한 포트 번호
afx_msg void OnBnClickedButtonPortopen(): //포트를 열기
위한 함수
afx_msg void OnBnClickedButtonPortclose(): //포트를 닫기
위한 함수
OnCommunication(WPARAM wParam, LPARAM lParam):
//포트와의 데이터 통신 함수
parseMSG(void): //읽어온 16진수 형 데이터를 정수형 데이
터로 변환하기 위한 함수
```

3.4 관련 기술

- Thread(pthread) 관련 기술

MFC에서 제공하는 thread 함수를 이용하여 태그 입력, 센싱, 화면 출력 프로세스를 동시에 수행하여 유기적으로 동작할 수 있도록 개발하였다.

- 조도 센싱 기술

MFC에서 제공하는 Thread를 이용하여 포트를 연후에 포트에 ZigbeX 모뎀에서 전달한 자료가 도착하면 이를 ON_MESSAGE 매크로를 이용, 함수를 연결하고

m_ReadData String에 저장된 데이터를 이용 읽어들이는 다.

- ID 센싱 기술 & DB연동

Thread를 이용 연결된 RFID Reader 장치를 통하여 입력을 기다리고 있다가 ID값을 읽어 들이는 순간 함수를 연결하여 DB에 접속. ID와 일치하는 데이터를 가진 값을 DATABASE에서 검색하여 일치하는 테이블을 읽어와서 화면에 출력

- PHP & MySQL & FLASH를 동시에 연동한 주차 공간 시물레이션 기술

PHP & MySQL & FLASH를 동시에 연동하여 실시간으로 DATABASE에 기록된 조도와 ID값을 읽어들이어서 리더기가 읽어들이는 장소에 따라 개별적으로 사용자가 알아보기 쉽게 주차 공간 시물레이션 구성

IV. 결론 및 향후과제

본 논문에서는 유비쿼터스 네트워킹의 이용이 점점 활성화되고 있는 시점에서 RFID와 ZigbeX센서를 연동한 네트워크 프로그램을 구축하여 활용하기 위한 주차장 관리 시스템 시물레이션을 살펴보았다.

MySQL과 PHP를 이용하여 DB를 구축하여 센서와 아이디에 대한 정보를 DB에 구축하고 시물레이션 하기 위해 플래시에서 PHP를 이용하여 DB에 등록된 정보를 읽어올 수 있도록 하였다. 아이디를 읽어들이는 장소, 즉 RFID의 리더기에 따라 플래시 주차 공간 시물레이션을 다르게 띄워야 했기 때문에 DB를 실시간으로 감시해야하므로 프로그램에서 CPU의 점유율이 꽤 높은 문제점과 센서에서 DB에 기록하는 시간과 이를 읽어오는 시간과의 차이가 미세하게 존재하여 실제 센서 값과 플래시에 표시 되는 것에 시간차가 존재하는 문제점이 있다. 이는 실시간으로 감시되어야 하는 주차장 관리 시스템 시물레이션

의 구현에 부족한 부분이라고 생각되며, 이를 극복하기 위해 좀 더 많은 기술을 연구하여 효율적인 프로그램을 개발해야 할 것이다. 또한, 이 시스템이 구축되어 사용된다면 RFID가 장착되지 않은 차량의 경우에는 임시 RFID 주차카드를 발급하거나 주차공간 내 RFID가 장착되지 않은 차량을 위한 공간을 따로 마련해야 할 것이다.

[8] 김선우, 신화선, 윈도우 프로그래밍; Visual C++ MFC Programing, 한빛미디어, 2005.
 [9] 한백전자 기술연구소, 유비쿼터스센서 네트워크시스템, ITC, 2006.
 [10] 한백전자 기술연구소, ZIGBEX를 이용한 유비쿼터스 센서 네트워크 시스템, ITC, 2007.

참고문헌

[1] 김성수, “주차관제시스템 개발에 관한 연구”, 고려대학교 산업대학원 석사학위논문, 1994.
 [2] 신영석 외 2인, “사전정보를 이용한 차량 관영역의 분리”, Vol. 13, No. 2, 1996. pp. 1-11.
 [3] 정연홍외 2인, “주차관제용 기계화 설비의 SMS(ShortMessage Service) 기반 원격관리시스템의 설계 및 구현”, 한국멀티미디어학회 추계 학술발표논문집, 2001. pp. 80-83.
 [4] Seo, Hee Suk and Cho, Tae Ho, “An application of blackboard architecture for the coordination among the security systems,” Simulation Modelling Practice and Theory, Elsevier Science B. V., Vol. 11, Issues 3-4, Jul. 2003, pp. 269-284.
 [5] Seo, Hee Suk and Cho, Tae Ho, “Simulation Model Design of Security System based on Policy-Based Framework,” Simulation Transactions of The Society for Modeling and Simulation International, vol. 79, no. 9, Sep. 2003, pp. 515-527.
 [6] Seo, Hee Suk, “Network security agent DEVS simulation modeling,” Simulation Modelling Practice and Theory, Elsevier Science B. V., Vol. 14, Issues, Oct. 2005, pp. 481-492.
 [7] 정일홍, MFC로 구현한 윈도우 프로그래밍, 생능출판사, 2006.

저자소개



서희석
Seo, Hee Suk

2005년 3월~ 현대 한국기술교육대학교
인터네트미디어공학과 정보보호전공
교수
 2005년 2월 성균관대학교 전기전자 및
컴퓨터공학과 (공학박사)
 2004년 3월~2005년 2월
(주)정보감리평가원 선임연구원
 2002년 2월 성균관대학교 전기전자 및
컴퓨터공학과(공학석사)
 2000년 2월 성균관대학교 산업공학과(공학사)
 관심분야 : 네트워크보안, 보안 시뮬레이션,
USN
 E-mail : histone@kut.ac.kr

논문접수일 : 2009년 4월 15일
 수정일 : 2009년 6월 1일
 게재확정일 : 2009년 6월 16일