

논문 2011-2-24

## 스마트카드를 이용한 모바일 교통결제 시스템 개발

### Development a Mobile Transportation Payment System of Used Smartcard

이상범, 정현호\*, 최이권

Sang-Bum Lee, Hyun-Ho Jung, Lee-Kwon Choi

요 약 오늘날 발달된 무선 네트워크 통신의 기술로 인해 많은 데이터를 빠르게 전송할 수가 있으며, 또한 더 좋은 단말기의 등장으로 멀티미디어 정보를 보여줄 수 있기 때문에 다양한 무선 인터넷 서비스가 가능하게 되었다. 따라서 SKT나 LGT와 같은 통신업체는 2.5세대 무선 기술과 스마트카드를 이용하여 온라인 쇼핑, 모바일뱅킹 등 많은 서비스를 제공하고 있다. 핸드폰을 대부분 사람들이 소유하고 있기 때문에 교통 시스템에 유용한 지불시스템으로 사용할 수 있는데, 모바일 지불 시스템이 무선 온라인 서비스로 이미 개발되어져 있기 때문이다.

본 논문에서는 모바일 교통 지불 시스템이 소개되고 있다. 운전자는 고속도로를 이용할 때 통행료를 지불해야한다. 최근에 하이패스 시스템이 개발되어 자동차는 요금을 지불하기 위해 정지하지 않고 톨게이트를 지나가면서 자동적으로 지불이 가능하다. 이러한 지불은 장착된 지불 시스템으로 인해 가능하다. 우리는 좀 더 발전된 형태로 스마트카드를 장착한 핸드폰을 지불 기기로 사용할 수 있는 시스템을 개발하였다. 개인과 금전 정보는 스마트카드에 저장되고 단말기와 OBU와의 통신은 블루투스 방식을 사용한다.

**Abstract** In these days, various wireless internet services are possible since the advanced wireless networks enable to transfer more data than before and smarter devices have developed to display the multimedia information more efficiently. Therefore, Korean MIPS such as SKT and LGT provide many kinds of advanced services such as wireless on-line shopping, mobile banking system and etc using 2.5 generation wireless network technology and smartcards. In a transportation system, a cellular phone can be used as a useful payment tool since most people always keep it and a mobile payment system has developed as infrastructure for wireless on-line services.

In this paper, a mobile transportation payment system is introduced. A driver needs to pay a toll fee when he/she uses the highways. Recently, a high pass system has introduced, in which a vehicle can pass a toll gate without stopping to pay the fee to a counter. The payment is done automatically by an attached payment device. We have developed an advanced mobile system in which a smartcard embedded cellular phone is used as a payment device. The personnel and financial information is stored at a smartcard and the communication between the phone and OBU(On Board Unit) is done by Bluetooth

**Key Words :** 무선 인터넷(MOBILE), 패스온 서비스, 결제, 스마트카드

#### I. 서 론

무선통신망이 발전하고 이동통신 단말기의 성능이 향

상되어 다양한 서비스가 제공되면서 이동통신 서비스 사업자 (MIPS)의 수익원으로 기존의 음성통신 서비스 이외에 모바일 전자상거래가 각광 받고 있다.<sup>[1][2]</sup> 특히 2.5세대 망과 스마트카드를 이용한 모바일 서비스가 활발해 지고 있으며 모바일 전자상거래의 기본 인프라로서 모바

\*준회원, 단국대학교 전자계산학과  
접수일자: 2010.12.21, 수정일자: 2011.2.15  
게재확정일자: 2011.4.15

일 결제서비스의 중요성이 커지고 있다. 또한 교통 서비스 분야에서 선불 충전을 통하여 요금을 결제하는 모바일 결제 시스템의 필요성이 증가하고 있다.<sup>[3][4]</sup>

이러한 모바일 교통 결제 서비스는 스마트카드에 개인 정보 및 관련 금융정보를 저장할 수 있는 이동통신 단말기 기술이 진일보함에 따라 결제에서부터 다양한 개인화된 맞춤형 서비스로 확대 되어지고 있다. 최근 도로공사 서비스인 모바일 교통결제 시스템은 그림 1과 같이 구성되어 있다.<sup>[5][6][7]</sup>



그림 1. 모바일 교통 결제 시스템  
Fig 1. Mobile Transportation payment System

본 논문에서는 기존 무선 인터넷 기반의 지급 결제 방식보다 빠르고 편리한 모바일 하이패스 시스템을 제안하였다. 이것은 차량이 고속도로 톨게이트를 통과할 때 교통카드 칩을 장착한 적외선 휴대폰으로 요금 정산단말기를 향해 휴대폰의 외장 키를 한번만 누르면 요금을 지불할 수 있는 시스템으로 기존의 RF 정산단말기에 적외선 모듈을 장착하며 RF통신방식과 병행하여 사용한다.

본 논문의 구성은 2장에서 모바일 하이패스 충전시스템의 설계를 위해 UML을 사용하여 4가지 뷰를 나타내었다. 3장에서는 모바일 하이패스 시스템의 구현에 대하여 기술하였으며, 4장에서 결론을 맺는다.

## II. 시스템의 분석 및 설계

객체지향 개발 방법론을 준수하여 개발한다면 초기에 요구사항을 정확하게 분석하고 반영하여 개념적으로 명확한 시스템을 구축하고 구축 과정에서 있을 오류 작업을 줄일 수 있다. 이는 재사용성을 높일 수 있으며 개발 비용과 시간을 절감하고 보다 더 안정된 시스템을 구축할 수 있다.

모바일 하이패스 충전시스템은 객체지향 방법론을 이

용하여 개발하였다. 시스템의 개발 범위와 대상, 이해당사자(Stakeholder)에 대한 내용을 기반으로 시스템의 경계를 나타내는 컨텍스트 뷰를 작성하고 UML을 사용하여 4가지 뷰를 표현하였으며, UML을 사용하여 시스템을 개발하였다.<sup>[8]</sup>

### 1. 컨텍스트 뷰

모바일 하이패스 충전시스템은 그림 2와 같이 사용자, 휴대폰, OBU(OnBoard Unit), 충전시스템 등으로 구성된 이해당사자(Stakeholder)들이 모바일 기기를 통하여 가장 우선순위가 높은 기능인 OBU 인증을 거친 후에 모바일 하이패스 시스템의 주요 기능인 요금충전 및 충전알림, 예약충전, 잔액/내역, 교통정보, OBU 등록 정보관리 등을 이용할 수 있음을 보여주고 있다.

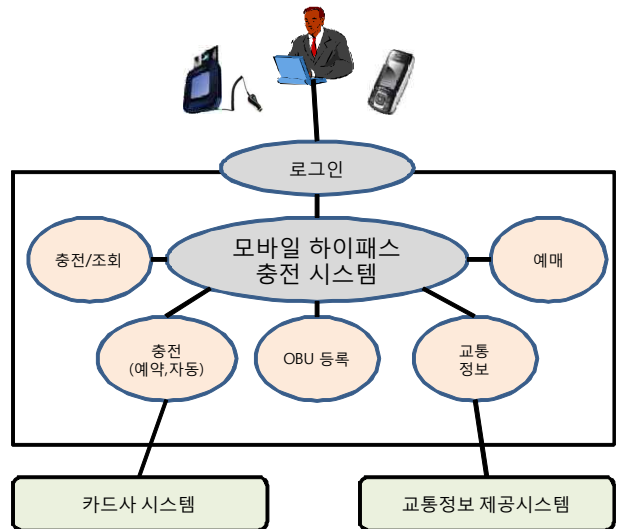


그림 2. 모바일 하이패스 충전시스템 컨텍스트뷰  
Fig 2. Context view of Mobile highpass charging System

### 2. 유스케이스 뷰

유스케이스란 요구 모델링 다이어그램으로 시스템을 사용하는 액터와 시스템과의 관계를 도식화 하여 사용자 시점에서 시스템을 모델링하는 UML 다이어그램이다. 유스케이스는 일반적으로 시스템의 사용 방법을 결정하는데 도움을 주는 역할을 하며 시스템의 요구사항을 보다 효율적으로 도출하기 위한 방식으로 쓰인다. 유스케이스를 작성하는 작업은 시스템 분석과 설계의 초기 단계에 해당하며 UML에서 논리적인 설계를 한 하나의 모델이다.

모바일 하이패스 충전시스템의 주요 기능들을 개념적인 모듈로 묶은 패키지 형태로 표현 하고 각각의 패키지들에 대한 유스케이스 뷰를 작성하는 단계로써 전체 시스템에 대한 패키지 형태의 유스케이스 뷰는 그림 3과 같다. 사용자가 시스템에 접근하는 것을 이벤트라고 표현하며, 이 메시지들은 각각의 접근하고자 하는 시스템을 이용하기 위한 명령어라고 할 수 있다.

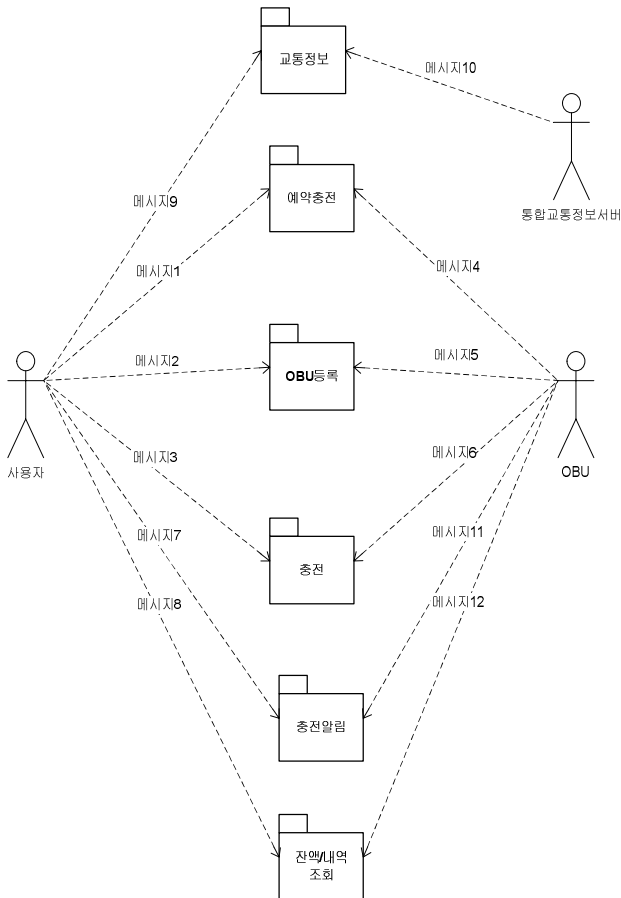


그림 3. 모바일 하이패스 충전시스템의 유스케이스뷰  
Fig 3. Usecase view of Mobile highpass charging System

### 3. 논리 뷰

논리 뷰에서는 모바일 하이패스 충전시스템의 패키지 구조에서 나타나는 추상집합과 그들의 관계를 보여주며, 3가지 정적 계층을 가진다. 시스템의 정적인 측면을 표현한 아키텍처로서 모바일 하이패스 충전 시스템의 논리 뷰는 그림 4와 같이 표현할 수 있다.

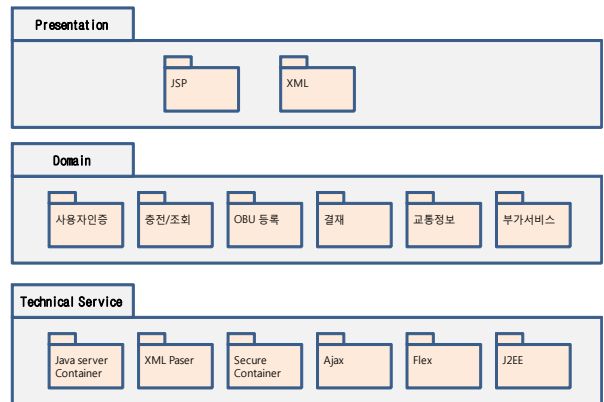


그림 4. 모바일 하이패스 충전 시스템 논리 뷰  
Fig 4. Logic view of Mobile highpass charging System

### 4. 컴포넌트 뷰

컴포넌트 뷰란 실제 구현된 모듈인 컴포넌트들 간의 관계를 표현해 주는 모델로 배치 뷰와 함께 구현 뷰를 표현하는 장치로 분류된다. 논리적 설계에 대한 구현은 이러한 컴포넌트 뷰와 배치 뷰를 이용하여 완성하게 된다.

모바일 하이패스 충전시스템의 컴포넌트 뷰는 그림 5와 같이 시스템에서 식별되는 컴포넌트들 간의 연관관계를 나타내는 아키텍처로서 각각의 컴포넌트들은 OBU 인증 컴포넌트를 통해 다른 컴포넌트들에 연결되어 있음을 알 수 있다.

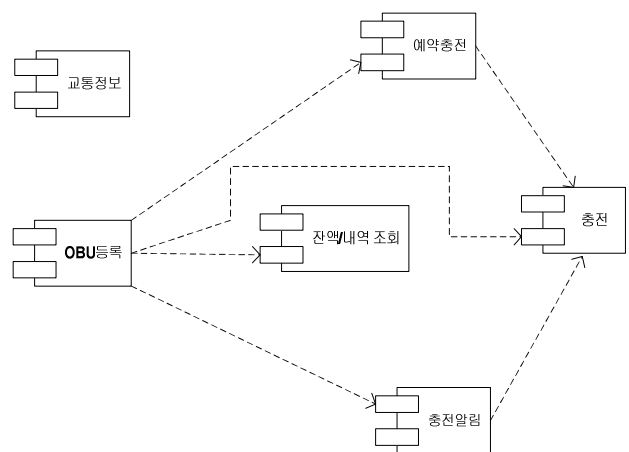


그림 5. 모바일 하이패스 충전시스템의 컴포넌트 뷰  
Fig 5. Component view of Mobile highpass charging System

### 5. 배치 뷰

배치 뷰란 노드와 노드에 존재하는 컴포넌트들의 물

리적 구성을 표현한다. 컴포넌트와 부가장치, 연결상태 등의 하드웨어적인 내용과 설치된 소프트웨어를 표현한다.

모바일 하이패스 충전시스템을 구성하는 물리적인 구성요소들 간의 배치를 표현하는 배치 뷰는 그림 6과 같이 나타낼 수 있으며 사용자, 휴대폰(VM 어플리케이션 포함), OBU, 이통사 기간 시스템, 모바일 중계서버, 모바일 업무 서버, 교통사업자 업무서버, Legacy서버, 결제기관 서버군 들로 구성되어 있다.

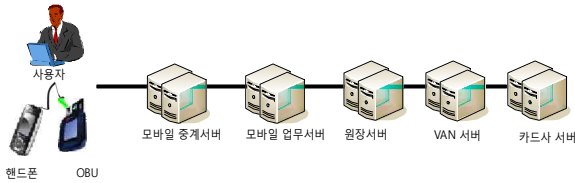


그림 6. 모바일 하이패스 충전 시스템 배치 뷰  
Fig 6. Deployment view of Mobile highpass charging System

### III. 시스템 구현

#### 1. 시스템 구현환경

모바일 하이패스 충전 시스템의 구현 환경은 그림 7과 같이 이동 통신사의 서버 군과 교통사업자 서버군 그리고 카드사 서버 군으로 나누어져 있다. 그중에서 모바일 하이패스 충전 시스템은 이동 통신사와 카드사의 서버를 연결하는 다리 역할과 비즈니스 로직을 담당하는 애플리케이션 역할을 하게 되는 교통사업자 서버군에 포함된다. 이러한 중계 역할을 하는 교통사업자의 서버군은 중계서버, 모바일업무처리서버, 원장서버와 결제를 위한 서버군으로 구성된다.

모바일 중계서버는 이동 통신사와의 중계를 위한 서버와 카드사와의 중계를 위한 두 가지 중계서버가 있으며 그 사이에 비즈니스 로직인 모바일 업무 서버가 존재하게 된다. 비즈니스 로직을 포함한 모바일 업무 서버에서는 고객의 요청에 따른 결제와 충전을 처리하는 시스템으로 구현된다.

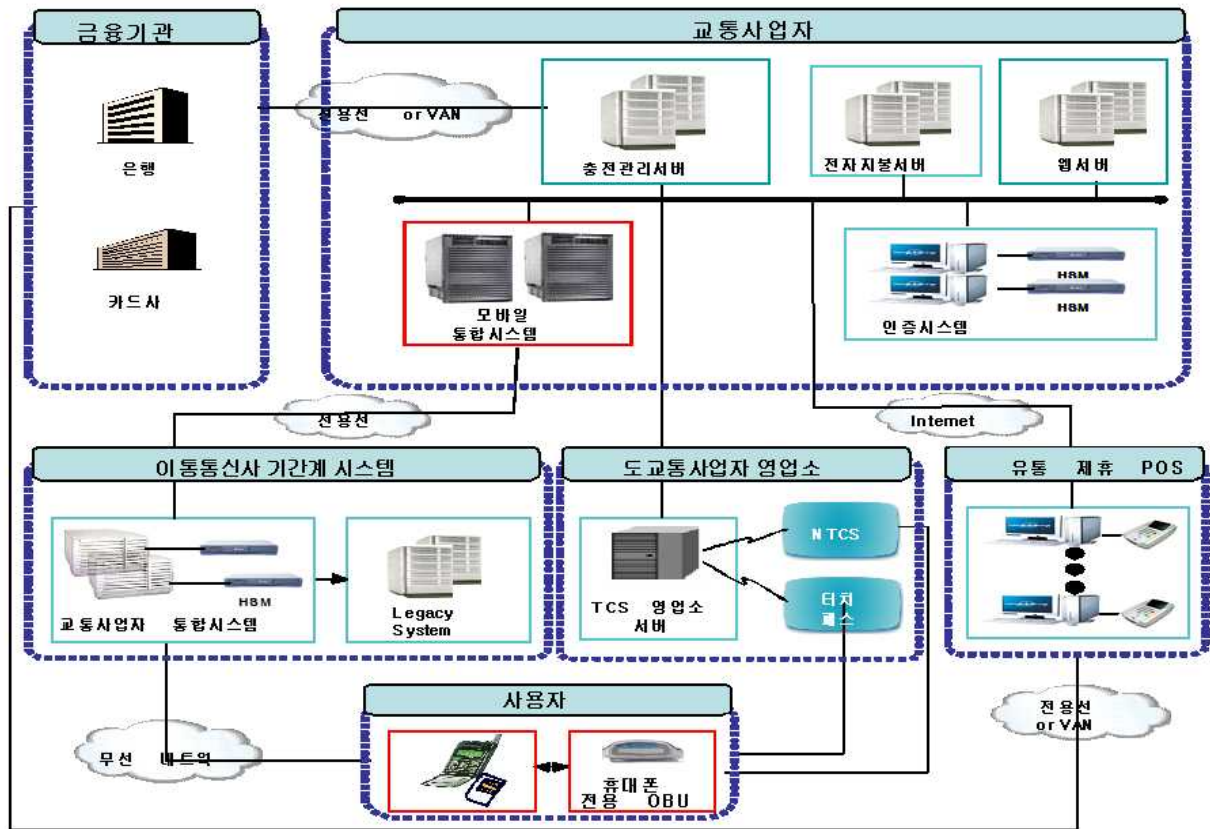


그림 7. 모바일 하이패스 충전시스템 도메인 아키텍처  
Fig 7. Domain Architecture of Mobile highpass charging System

모바일 중계서버와 업무서버는 J2EE Container로서 Java 프레임워크를 구동할 수 있으며, 실제로 휴대폰의 가상머신 어플리케이션으로 페이지를 전송한다. 그리고 결제 시 카드사를 연동하기 위하여 중계시스템과 VAN 서버 카드사 결제 시스템을 연동한다.

이러한 일련의 과정을 통해 이미 구축 되어진 이동 통신사의 서버와 카드사의 서버를 연동할 수 있으며, 비즈니스 로직을 포함하는 교통사업자 서버 군으로 모바일 하이패스 충전 시스템이 가능하게 구성된다.

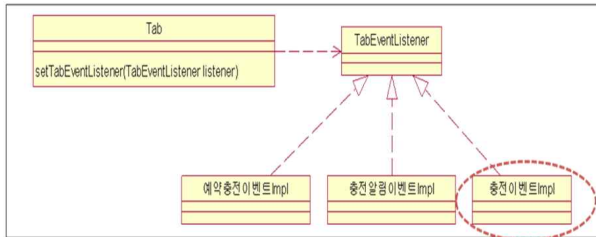


그림 8. 충전 프로세스의 이벤트 처리부분  
Fig 8. Class Diagram for charging event processing

모바일 하이패스 충전 시스템의 가장 중요한 부분이 라 할 수 있는 비즈니스 로직의 충전이벤트 발생에 대한 처리에 대한 부분은 그림 8의 다이어그램으로 표현하였다.

Tab 이라는 클래스는 UI 와 이벤트 처리 클래스들을 등록하여 이벤트가 발생하면 이벤트에 알맞은 기능을 처리 할 수 있는 부분을 호출하는 기능을 갖고 있으며, 이벤트 처리하는 클래스가 존재해야 한다.

TabEventListener 인터페이스는 특정 이벤트를 구현하기 위한 상위 인터페이스가 된다. 이러한 인터페이스 클래스를 상속하여 실제로 구현된 것으로는 예약 충전 이벤트 처리 클래스와 충전 알림 이벤트 클래스 그리고 충전 이벤트 클래스가 있다.

이것은 시스템에서 발생하는 이벤트에 대한 실제 처리 클래스들이며, 각각의 이벤트 역할에 맞는 클래스에서 이벤트를 처리하게 된다. 새로운 이벤트를 추가할 경우 이벤트 성격에 맞는 클래스는 TabEventListener 인터페이스를 상속하여 구현하게 되며, 새로운 이벤트에 대한 처리 루틴을 Tab 클래스에서 추가로 구현하게 되는 구조로 클래스를 구현하였다.

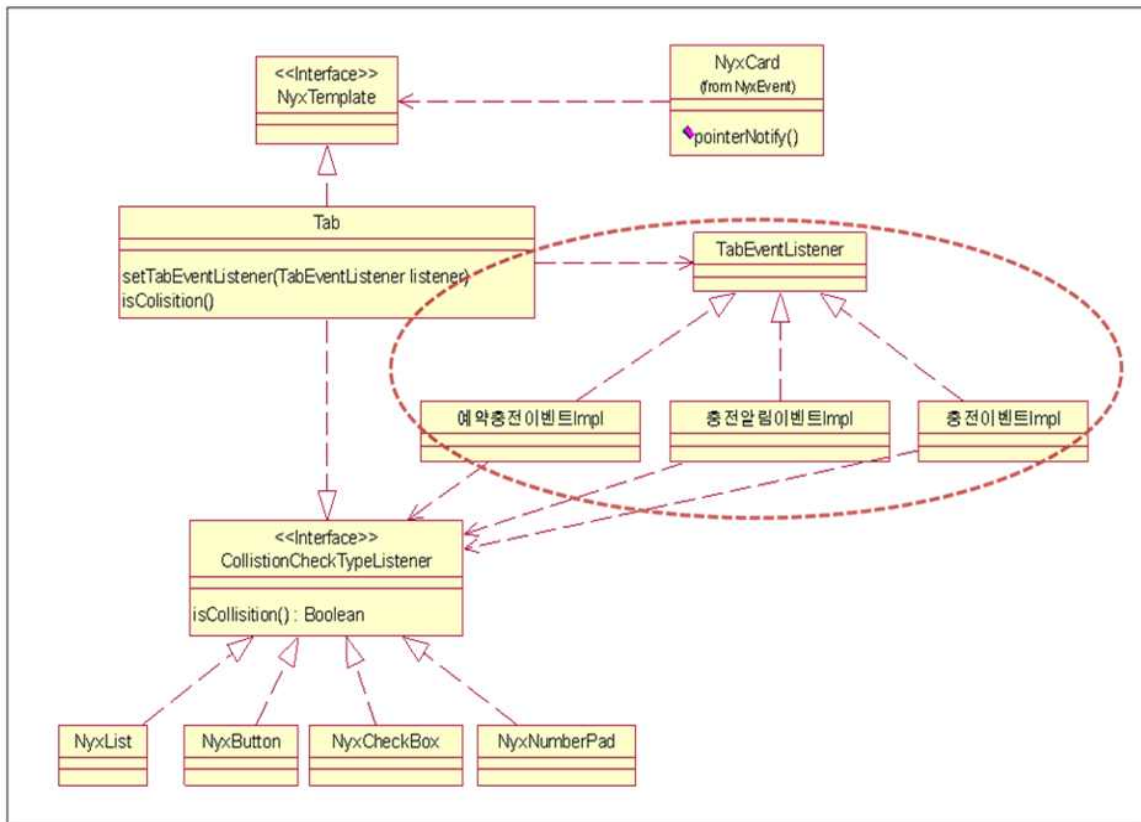


그림 9. 모바일 하이패스 시스템의 이벤트 구조  
Fig 9. Class Diagram for event structure

UI 컴포넌트로부터 받은 이벤트, 카드사로부터 받은 이벤트 그리고 사용자로부터 받는 충전 및 결제에 대한 이벤트 등 모바일 하이패스 시스템 전체의 각 프로세스 이벤트 구조는 그림 9와 같이 다이어그램화 된다.

CollisionCheckTypeListener의 인터페이스를 구현한 각각의 UI 컴포넌트들이 존재 하며, 이들은 이벤트 처리 프로세스를 담당하는 클래스 역할을 수행하게 된다. 한편, 이러한 컴포넌트(List, Button, CheckBox, NumberPad...)를 추가하여 UI 전체를 구성한 클래스가 Tab 클래스가 되며 UI 컴포넌트를 받아들여 각 이벤트에 맞는 루틴을 실행하는 기능도 가지고 있다. 또한 카드사 서버로부터 받은 기능을 수행하는 NyxCard 클래스가 NyxTemplate 인터페이스를 구현하여 정의 된다. 이 또한 Tab 클래스에서 이벤트 클래스로 참조(등록)하게 되며 이벤트에 대한 처리 루틴을 호출하게 된다.

표 1. 충전 비즈니스 처리 플로우  
Table 1. Code for charging business processing

```
public class ChargeMaker{ // 충전 프로세스 UI 구성
    public ChargeMaker() {
    }
    public Tab makeTabInPane(int pageNumber, TabEventListener listener){
        Tab tempTab = null;
        switch(pageNumber) {
            case NyxState.CHARGE_0 :
                tempTab = chargeTab0(); // 첫 페이지 UI 구성
                break;
            case NyxState.CHARGE_1 :
                tempTab = chargeTab1();
                break;
            case NyxState.CHARGE_2 :
                tempTab = chargeTab2();
                break;
            case NyxState.CHARGE_3 :
                tempTab = chargeTab3();
                break;
            case NyxState.CHARGE_4 :
                tempTab = chargeTab4();
                break;
            case NyxState.CHARGE_5 :
                tempTab = chargeTab5();
                break;
        }
        if(tempTab == null)
            return null;
        tempTab.setTabEventListener(listener); // Listener 등록
        return tempTab;
    }
}
```

ChargeMaker클래스는 충전 프로세스의 UI를 구성하는 클래스로 표 1과 같다. 각 페이지의 인덱스값을 넘겨 받아 해당 페이지의 UI구성을 하게 되며, 이때 각 페이지의 UI를 표현하는 클래스가 Tab 클래스가 된다. 또한 UI

구성 후 EventListener를 참조하여 이벤트 발생시 이벤트 담당 클래스에서 처리하도록 하였다.

chargeTab0 ~ 5의 클래스는 각각의 페이지에 따른 인덱스 값과 매핑되는 UI를 가진 클래스이며 UI 로 표현된 각각의 charTab#N 클래스를 이용하여 사용자는 원하는 충전 프로세스 과정을 진행할 수 있게 된다.

TabChargeEventImpl 클래스는 TabEventListener 클래스를 상속 받아 충전 프로세스 중 해당 Component의 이벤트를 처리 하는 클래스로 표 2와 같이 구성된다. 이 이벤트를 처리하는 TabChargeEventImpl 클래스의 gotoCommonPad 메소드는 표 1에서 선택되어진 각각의 Component의 이벤트가 발생 할 경우 페이지의 인덱스 값을 구분하여 해당 로직을 실행 하게 된다.

표 2. 충전 프로세스에 대한 이벤트를 처리하는 클래스  
Table 2. Code for charging event processing

```
// 충전 프로세스의 이벤트처리를 담당하는 클래스
public class TabChargeEventImpl implements TabEventListener{
    .
    .
    // 버튼의 이벤트 처리를 담당하는 메소드로 CHARGE_4의 값은 충전 버튼을
    // 클릭 할 경우 실행되게 된다.
    private int gotoCommonPad(int index, NyxCommonButtonPad pad {
        Tab tempTab = (Tab)oObject;
        switch(tempTab.getState()) {
            case CHARGE_4: {
                .
                .
                //////////////////////////////////////////////////// 충전 실행 ////////////////////////////////////////
                OBUMessageMaker.allMessageMake(Protocol.I100);
                OBUMessageMaker.allMessageMake(Protocol.I110);
                OBUMessageMaker.allMessageMake(Protocol.I120);
                ////////////////////////////////////////
                .
                .
            }
        }
    }
}
```

## 2. 모바일 하이패스 충전 시스템 클라이언트 실행화면

그림 10은 모바일 홈페이지의 접속에 대한 실행화면이다. 사용자가 휴대폰의 인터넷 접속 버튼을 누르고 'URL 입력'을 선택하여 원하는 URL을 입력하 후 확인 버튼을 누르면 모바일 홈페이지에 접속된다.

충전, 충전알림, 예약충전, 잔액/내역, 미납요금, 교통정보, OBU 등록에 대한 메뉴를 선택할 수 있으며, 사용자는 원하는 서비스를 선택하여 다음 과정을 진행할 수 있다. 이때 시스템은 가입 여부를 체크하여 사용자에게

공지사항을 전달한다.



그림 10. 모바일 어플리케이션 실행화면 1  
Fig 10. Captured screen 1 for Mobile Application

그림 11은 OBU를 시스템에 등록하는 과정을 보여주는 화면을 나타낸다. 모바일 홈페이지에 접속한 후 "OBU등록"이라는 메뉴를 선택하면 OBU의 기능을 사용자에게 전달한다. 이 때 "등록" 버튼을 누르게 되면 OBU를 검색하기 시작한다. 검색에 성공하면 해당 칩에 맞는 OBU를 선택하고 비밀번호를 등록하게 된다. 이러한 과정을 거치면 휴대폰에 OBU가 성공적으로 연결된다.



그림 11. 모바일 어플리케이션 실행화면 2  
Fig 11. Captured screen 2 for Mobile Application

다음의 그림 12는 충전 과정을 나타낸다. [그림 10]에 서처럼 모바일 홈페이지 첫 화면에서 충전이라는 화면을 선택하게 되면 충전에 대한 과정이 시작된다. 결제 종류에 따른 신용카드, 계좌이체, 폰빌 등을 선택할 수 있으며, 각 선택된 상태에 따라 맞는 충전 과정이 이루어지게 된다. 이는 충전 과정의 결제 시스템에 따라 카드사나 은행으로 연결되어 결제를 진행하게 된다.



그림 12. 모바일 어플리케이션 실행화면 3  
Fig 12. Captured screen 3 for Mobile Application

### 3. 모바일 하이패스 시스템의 진행 절차

모바일 하이패스 시스템의 진행 절차는 다음과 같다.

- 1) 차량이 영업소에 진입하기 전, 운전자는 전자카드가 내장된 휴대폰을 전용 OBU에 부착한다.
- 2) 차량이 Wake-up 영역을 통과하면 휴대폰전용 OBU는 차로제어기로부터 BST를 수신하기 위한 수신대기상태가 된다.
- 3) 차량이 통신영역에 진입하면 차종분류가 시작되고, 무선통신으로 징수명령이 이루어진다.
- 4) 휴대폰전용 OBU는 징수명령을 받으면 휴대폰과의 통신을 거쳐 휴대폰내 내 장착된 전자카드에 징수명령을 전달하고 전자카드의 처리결과를 기다린다.
- 5) 휴대폰 내 전자카드 처리결과가 휴대폰전용 OBU로 전달되면 휴대폰전용 OBU는 처리결과를 안테나제어기를 통하여 차로 제어기로 전달하고, 통행료 징수 결과를 운전자에게 통지한다.
- 6) 차로제어기는 차종분류결과, B/L (Black List) 검색결과, 전자카드처리결과를 확인하여 위반차량으로 판별될 경우 차량의 번호판을 촬영한다.

## IV. 결론

본 논문에서 제시하는 모바일 하이패스 충전 시스템은 시나리오에 따른 각각의 설계를 UML로 도식화 하였고 이에 관련된 시스템을 구현 하였다. 클라이언트는 일반 무선인터넷을 사용할 수 있는 휴대폰(VM 어플리케이션)

선 WIPI)과 교통사업자의 지불시스템과 연동할 수 있는 OBU(On Board Unit)와 서버(모바일서버, 중계처리, 인증서버, 과금서버, 결제서버, 충전지불 서버 등)을 구현하였다. 모든 서버의 처리 로직은 컴포넌트 기반으로 구성하였으며 TCP/IP 프로토콜을 이용한 통신처리 모듈(각 인터페이스별 별도 데몬 처리)로 이용하였다. 모바일 서버 통신부와 어댑터, 컨트롤러, 입출력 데이터 처리부(비즈니스 로직)은 각각 컴포넌트화 하여 향후 추가 개발에도 모듈 재사용이 가능하도록 설계 구현 하였고, 전문 로직은 무선 환경에서 최적화할 수 있게 패키지형태로 설계 및 구현하였다. 클라이언트는 이통사에서 제공하는 WIPI 플랫폼을 사용하여서 개발하였으며, 휴대폰과 OBU 간의 통신은 블루투스 기반으로 개발 하였다. OBU와 교통사업자간은 통신은 RF 통신으로 개발 하였다.

현재 유·무선 인터넷상에서 전자상거래가 갈수록 확대 되는 추세에 있음에도 불구하고 편리하고 안전한 지불결제 수단과 모바일 서비스가 별로 없다고 할 수 있다. 앞으로 휴대폰과 무선인터넷의 발전으로 인한 유비쿼터스 시대가 도래 할 것이고, 스마트카드나(USIM 카드)나 IPV6 환경에서의 IP를 통한 지불 결제가 발전하게 될 것이다. 향후 이러한 새로운 요구와 기술력에 변화에 맞춰 이러한 환경에서의 다양한 모바일 결제 시스템에 대해서 연구 할 계획이다.

## 참고문헌

- [1] 이동일, 박천교, 임영이, “M-Commerce 서비스 개요 및 시장 전망”, 정보통신산업진흥원, 정보통신 연구진흥원 학술정보 주간기술동향 1073호, 2002.
- [2] 정지범, 김한주, “국내 모바일 전자상거래 이용실태 조사 분석”, 정보통신산업진흥원, 정보통신 연구진흥원 학술정보 주간기술동향 1139호, 2004.
- [3] 조진만, 김수형, 문기영, “모바일 지급결제의 시장 현황 및 표준화 동향” 한국전자통신연구원, 전자통신동향분석 제20권 제1호, 2005.
- [4] 이충열, 김재필, “전자정부 구현을 위한 스마트카드 도입의 경제적 효과 연구”, 정보통신산업진흥원, 2005.
- [5] 한진희, “스마트카드 플랫폼 기술”, 한국정보통신 기술 협회, TTA Journal 제 90호, 2003.
- [6] 이기한, 김윤정, “전자지불시스템에서 이용되는 스마트카드의 시험 모듈 구성에 대한 연구”, 한국전자거래학회, 한국전자거래학회지, 제9권 제1호, 2004.
- [7] 최이권 외1명, “PDA를 활용한 모바일 banking 시스템의 설계 및 구현”, 한국산학기술학회 논문지 제8권 제6회 pp 1530-1536, 2007년 12월
- [8] Colin Atkinson, “Component-based Product Line Engineering with UML, Addison-Wesley, 2002.



저자 소개

이 상 범



• 학위(학, 석, 박사)  
학사 1983. 한양대학교 기계공학과  
석사 1989. Louisiana State University  
전산학  
박사 1992. Louisiana State University  
전산학

• 경력

단국대학교 전자컴퓨터학부 부교수 (1993 - 현재)  
아람미디어텍(주) 대표이사 (2000.2 - 현재)  
한국전자통신연구원 초빙연구원 (1994 - 1995)  
한국전자통신연구원 선임연구원 (1992 - 1993)  
제일모직 전산실 프로그래머 (1983 - 1986)  
<주관심분야 : RFID/USN 기술, 이동무선통신, 안테나 및 전  
파전파, 통방융합기술>

최 이 권



• 학위  
학사 1999. 단국대학교 전자계산학과  
석사 2003. 단국대학교 전자계산학과  
박사 2008. 단국대학교 전자계산학과

• 경력

백석대학교 겸임교수(2005 - 2007)  
모전스랩(주) 대표이사(2005 - 현재)  
<주관심분야 : 모바일 컴퓨팅, 콘텐츠 개발, 소프트웨어 프로  
세스>

정 현 호(준회원)



• 학위  
학사 2009. 단국대학교 컴퓨터학과

• 경력

단국대학교 전계산학과 석사과정(2010  
- 현재)

<주관심분야 : 소프트웨어 프로세스, 소프트웨어 방법론, 소프  
트웨어 아키텍처>