

전자 지불 처리 시스템의 기능 및 구조*

송병열**, 함호상**, 박상봉**

The function and architecture of electronic payment system

Byoung-Youl Song, Ho-Sang Ham, Sang-Bong Park

Abstract

This paper shows the architecture of IPS(Intermarket Payment System), an implementation of the electronic payment function for internet-shopping merchant system. Because the original purpose of commerce transaction is the exchange of money for goods or services, it is very important to prepare an exchangeable economic value or method.

The electronic payment system is the hardware or software or both to process an electronic payment transaction. It has two type, the broker type and the electronic value type. The broker type means an intermediary between real bank network and internet commerce transaction. The electronic value type means a substitute for money in the real world.

This paper shows the architecture and the function to implement the broker type electronic payment system. The system has two parts. One is the mediator part to support multiple payment systems and to offer common access methods for merchant system database. The other is the executor part to implement the payment protocol and to process payment transaction.

Keyword: Electronic Commerce(EC), Electronic Payment System, Intermarket Payment System(IPS)

* 본 논문은 정보통신부가 지원한 "사이버커머스S/W기술개발" 과제의 연구 결과임.

** 한국전자통신연구원 전자상거래연구부

1. 서론

전자 상거래의 확산으로 인해 기존의 시장 구조에서의 유통 흐름 및 매체가 크게 변화하고 있다. 상품 흐름의 영역과 크기, 형태의 변화가 인터넷이라는 공용 매체를 통해 이루어지고 있으며 이러한 변화가 21세기의 실 경제에 점차 커다란 변화를 일으킬 것이라는 것은 분명하다.

상거래의 본질이 어떤 상품이나 서비스에 그에 상응하는 경제적 가치와의 교환이라고 볼 때, 인터넷상에서 교환 가능한 경제적 가치를 마련하는 것은 전자상거래의 활성화에 있어 아주 중요한 요소라고 할 수 있으며, 이러한 가치를 제공하는 것이 전자지불이다.

전자지불 시스템은 전자 지불을 처리하기 위한 하드웨어 및 소프트웨어를 의미하며 지불 매체에 따라 전자 현금, 신용카드, 전자 수표, 전자 자금 이체 방식 등으로 분류되고 다시 지불 방식에 따라 브로커형과 전자 현금형으로 나뉘어 간다. 브로커형은 신용카드, 전자 수표, 자금이체 등과 같이 실제 지불 처리는 은행 쪽에서 이루어지며 시스템은 단지 지불 처리를 위한 정보의 전달 매개체로 사용되는 경우를 의미하며 전자 현금형은 시스템을 통해 교환되는 정보 그 자체가 현금으로서의 기능을 지니는 경우를 의미한다.

현재까지 전자 지불 시스템은 주로 매체에 따라 분류되어 왔으며 시스템의 개발 또한 어떤 방식 즉, 매체를 무엇으로 할 것인가에 초점이 맞추어져 연구 개발되었고 사업적인 특성보다는 주로 기술적인 특성에 더욱 많은 비중을 두고 있다. 그러나 전자 상거래의 주요 수요자들이 기술자가 아닌 사업자임을 볼 때

개발되는 지불 시스템들은 비즈니스 영역에서의 활동 방식이 반영되어 실수요자가 되는 사업자들이 이해하기 쉽고 자신들의 목적에 맞게 시스템을 구성하는데 있어서 큰 어려움을 느끼지 않도록 할 필요가 있다. 물론 거래를 위해서 물리적인 현금이 아닌 디지털 데이터화된 가치의 교환이라는 특수한 환경과 인터넷이 가지는 개방성에 따른 거래의 안전성 문제를 해결하기 위한 보안 기술의 도입이 필수이다. 그러나, 이 역시 수요자 입장에서 보안 프로토콜의 구조까지 알아야 사용할 수 있는 기술적인 접근 방법이 아닌 비즈니스 관점에서의 접근 방법을 통해 전자 상거래를 시작할 수 있도록 하는 방법을 제공하는 것이 전자 상거래의 확산에 유리하다. 또 하나 주의할 점으로 현 시점에서조차 점차 변화하고 있는 전자 상거래 환경과 이에 따른 수요자들의 다양한 요구, 그리고 이를 해결하기 위한 지속적으로 제안되고 있는 새로운 형태의 지불 처리 방식 및 표준들에 대한 수용도 검토할 필요가 있다. 이러한 점은 전자 지불 시스템이 어떤 특정 프로토콜이나 구조에 국한되지 않고 다양한 방식들을 수용할 수 있도록 하기 위해 좀더 확장 가능한 구조로 설계되어야 함을 의미한다.

본 논문에서는 이러한 점들을 고려하여 먼저 2장에서 상거래에서 일어날 수 있는 지불 처리 유형을 분석하여 지불 처리 모델을 설계하기 위한 기본 자료를 제시하고 3장에서 기존 시스템들이 확장성있는 구조를 어떻게 구현하였는지 살펴본다. 4장에서는 현재 개발중인 지불 시스템의 구조와 기능에 대해 간략히 기술하고 5장의 결론에서 문제점과 향후 개발 방향을 제시한다.

2. 상거래 지불 처리 유형

지불 시스템을 개발하는데 있어서 가장 먼저 해야 할 일은 기존 상거래에서의 지불 처리 유형을 분석하고 이를 토대로 적절한 모델을 세우는 것이다 하겠다.

일반적으로 상품을 구매할 때 대금의 결제는 현장에서 직접 이루어지는 경우와 물건의 인수 시점이 아닌 그 이후에 지불 처리가 수행되는 경우 두 가지로 나눌 수 있다. 이 장에서는 이 두 가지 유형에 따른 지불 매체와 방법을 분석하도록 하겠다.

2.1 현장에서 직접 처리하는 경우

보통 현금, 직불카드, 신용카드, 계좌이체, 전자 현금을 이용한 경우이다.

2.1.1 현금

(1) 처리 과정

비교적 소액의 경우에 해당되며 구매한 물품을 전달받는 과정에서 대금을 즉시 지불한다. 영수증은 금액 영수와 동시에 발행한다. 통신 판매와 같은 경우 배달원은 영수증 발행을 위해 고안된 HandTerminal을 통해 직접 대금 영수 내역을 회계 시스템에 입력하고 영수증을 발행할 수도 있다. HandTerminal과 같은 장치가 없는 경우 배달원은 직접 영수금액을 택배 본부에서 입금한다.

(2) 필요한 장비

통신 판매의 경우 배달원은 현금 매출 입력 및 영수증 발행이 가능한 휴대형 단말기를

휴대할 수 있다. 매장의 경우 전표의 입출력이 가능한 단말기를 필요로 한다.

2.1.2 직불카드, 신용카드

(1) 처리 과정

비교적 고액의 경우에 해당되며 구매한 물품을 전달받는 과정에서 고객은 현금 대신 직불카드나 신용카드를 이용하게 된다. 상점은 신용카드 단말기를 이용하여 은행에 대해 거래에 대한 승인을 요청하면 은행은 고객의 계좌 또는 신용을 확인하여 거래 승인 또는 거부로서 응답한다. 거래가 승인된 경우 이것은 현금의 지불과 동일한 효과를 얻게 된다.

(2) 필요한 장비

신용카드 또는 직불카드의 조회를 수행하고 영수증 발행이 가능한 단말기가 필요하다.

2.1.3 계좌이체

(1) 처리 과정

보통 통신 판매에서 사용하는 경우로 현재 테스트 시스템만이 나와 있는 IC 카드형의 전자 통장을 사용하는 경우에 해당한다. 처리과정은 먼저 고객이 계좌이체 기능을 가지고 있는 IC 카드를 사용하여 매장 또는 배달원에 계좌이체 거래를 요구하면 매장은 IC카드 단말기를 고객에게 제시한다. 고객은 매장의 IC 카드 단말기에 자신의 IC 카드를 삽입하고 거래를 위한 비밀 번호를 입력한다. 고객의 비밀번호와 계좌번호, 이체 금액등의 결제 정보가 은행으로 전송되고 처리된 후 그 결과는 IC 카드 단말기에서 영수증의 형태로 발행된다.

(2) 필요한 장비

계좌이체를 지원하는 IC 카드 리더기 또는 IC 카드를 입력받을 수 있는 단말기를 필요로 한다.

2.1.4 전자현금

(1) 처리 과정

현금 대신 IC카드에 저장된 전자 현금을 이용하는 경우로 비교적 소액의 경우에 해당된다. 처리 과정은 먼저 상품의 구매 시점에서 고객이 전자 현금 결제를 원하는 경우 상점이 IC 카드 단말기를 고객에게 제시하면 고객은 IC 카드 단말기에 자신의 IC 카드를 삽입하고 거래를 위한 비밀 번호를 입력한다. 이때 고객의 IC카드에 충분한 전자 현금이 남아 있다면 요청한 액수의 전자현금은 상점의 IC카드 단말기로 이전된다. 전자 현금의 이전이 완료되면 IC 카드 단말기는 해당 거래에 대한 영수증을 발행한다.

(2) 필요한 장비

현재 전자 현금을 지원하는 제품으로는 Mondex International의 Mondex 시스템이 가장 유력한 제품이며 최근 국내에서도 ICash라는 전자 현금을 발표하였다[MONDEX, 1998 : 동성정보, 1998]

2.2 사후 처리하는 경우(월말 정산 형태)

계좌이체, 지로이체, 자동이체 등의 방법이 있으며 보통 대형 매장을 통한 통신 판매나 신문이나 잡지 구독과 같이 매달 일정액의 금액이 결제되어야 하는 경우에 해당한다.

2.2.1 계좌이체, 지로 이체

(1) 처리 과정

판매자로부터 계약된 회계 기간내의 구매에 대한 요금 내역이 상품 구매자에게 배달된다. 구매자는 요금 내역에 해당되는 금액을 은행에서 직접 이체하거나 ATM을 통해 자신의 계좌로부터 판매자의 계좌로 이체한다.

2.2.2 자동 이체

(1) 처리 과정

계약자가 은행과 자동이체를 계약한 경우에 해당한다. 처리 과정은 먼저 판매자로부터 계약된 회계 기간내의 구매에 대한 요금 내역이 구매자에게 배달된다. 그리고, 판매자의 거래 은행은 구매자의 계약 은행에 해당 계좌에 대한 자동 이체 지급 청구를 하면 구매자의 계약 은행은 구매자의 계좌로부터 지급을 요구한 은행의 해당 계좌로 대금을 이체한다.

대체로 Business-To-Consumer(BtoC)의 상거래 활동에서의 지불 처리는 이미 기술한 범주에 해당한다. 물론 Business-To-Business(BtoB)에서의 상거래인 경우 이와 여러 가지 방법 등이 있을 수 있으나 현재 본 논문에서 다루고자 하는 범위는 BtoC의 경우에만 국한하고자 한다.

3. 지불 시스템 개발 동향

이 장에서는 현재까지 발표된 또는 개발된 전자 지불 시스템들을 살펴보고 각각의 특징을 기술하도록 한다. 이 장의 주요 초점은 지

불 처리 프로토콜이나 방식 그 자체보다는 지불 처리 시스템을 구성하기 위한 구조 또는 프레임워크 위주로 기술하고자 한다. 경우에 따라서는 지불 처리 기능에 대한 언급이 부족한 경우도 있을 수 있으나 주요한 점은 전자 상거래에서의 지불 처리라는 비즈니스 활동을 수행하기 위한 구조의 고찰에 있다.

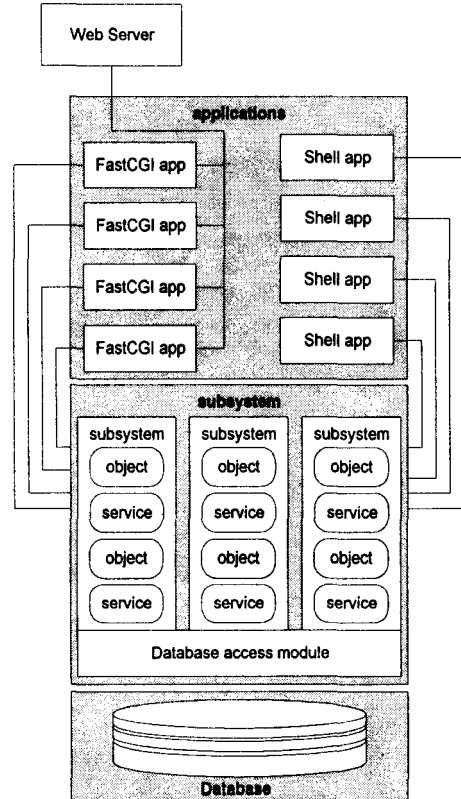
3.1 The Transact 4 Architecture

전자 상거래 솔루션 시장을 이끌어 나가는 주요 벤더의 하나인 OpenMarket은 초기에 SSL(Secure Socket Layer)와 각 상품의 URL 상에 MAC(Message Authentication Code)를 이용한 일종의 전자주문서(DO: Digital Offer)를 표시하여 거래를 처리하는 방식인 OM-Transact와 OM-SecureLink를 이용하여 지불 처리를 수행하였다.

1998년 초 OpenMarket에서는 분산화된 서버, 응용 프로그램, 서브시스템, 객체 및 서비스에 기반한 새로운 트랜잭션 구조로서 "The Transact 4 Architecture"를 발표하였다 [OpenMarket, 1998].

Transact 4 Architecture는 <그림 1>에 보인 것과 같이 application 부분과 subsystem 부분으로 구성된다. Application 부분은 시스템에 대한 일종의 인터페이스 부분으로 Fast CGI와 Shell 응용 프로그램으로 구현된다. 이 application들은 각기 특정한 기능을 제공하며 대부분 Transact 4가 subsystem으로 제공하는 공용 객체와 서비스를 이용하도록 되어 있다.

Transact 4에서의 subsystem은 객체와 서비스 그리고 데이터베이스 연동 부분으로 구성되며 전자 상거래 시스템의 기반 기능을 제공한



<그림 1> OM Transact 4 Architecture

다. Transact 4에서는 Order-invoice, Directory, Payment, Store, Microtransaction, Subscription의 6가지의 subsystem을 제공한다.

이들 서비스들은 COM interface를 이용하여 구현되었으며 Fulfillment API, Order Entry API, Buyer Profile API, Multi-Item API를 이용하여 application으로부터 호출될 수 있도록 되어 있다.

이 시스템에서 특정 지불 시스템은 Shared Object 또는 라이브러리로 구현하고 이에 대한 COM 인터페이스를 기술한다. 각각의 COM 인터페이스는 내장된 API에 의해 호출

되는 방식으로 고객 쪽에서의 지불 처리를 위해서는 API를 이용하여 지불 처리 COM 객체를 호출하는 CGI를 작성해야 한다.

3.2 IBM Multi-payment Framework

IBM은 자사의 전자 상거래 패키지인 Net.Commerce에서 지불 처리를 수행하기 위해 Consumer Wallet, Payment Gateway, Payment Registry, Payment Server로 구성된 Payment Suite를 발표하였다[IBM,1998a].

IBM Payment Suite를 구성하는 각각의 시스템은 SET의 거래처리 매커니즘을 구성하는 각 요소들을 바탕으로 구현되었으며 각기 Consumer Wallet은 고객 시스템, Payment Gateway는 금융기관, Payment Server는 상인, Payment Registry는 인증기관의 역할을 수행하게 된다.

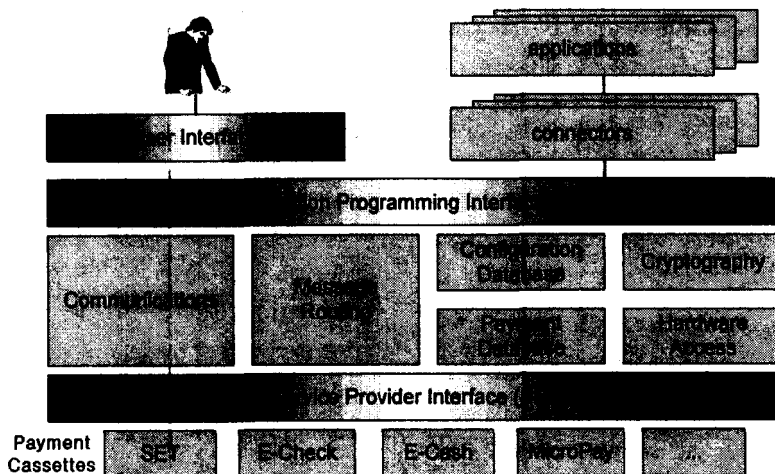
Payment Suite에서 특히 Payment Server

는 SET뿐 아니라 다양한 형태의 지불 처리 방식을 수용할 수 있도록 하기 위해 개방적이고 확장성을 지닌 구조인 Multi-Payment Framework(MPF)를 사용하고 있다[IBM, 1998b].

MPF는 <그림 2>에서 보인 바와 같이 응용 프로그램과 그림에서 cassette로 나타내어진 실제 지불 처리 모듈사이에서 메시지 라우팅, 통신, 데이터베이스 연동, 하드웨어 연동 등과 같은 기반 기능을 제공하여 다양한 형태의 지불 방식을 수용할 수 있는 구조이다.

이 시스템에서 실제 다양한 지불 처리 방식의 지원이 가능하도록 하는 부분은 Message Routing 부분과 API, SPI, cassette 부분이다.

Message Routing 부분은 MPF의 핵심으로 API 호출과 입력된 프로토콜 메시지들을 SPI를 통해 적절한 cassette로 연결 시켜주는 기능을 한다.



<그림 2 > IBM MPF Architecture

API는 응용 프로그램이 지불 요구를 받거나 처리하고 지불에 필요한 적절한 정보에 접근할 수 있도록 하기 위한 표준 인터페이스를 제공한다.

SPI는 cassette를 MPF에 연결할 수 있는 방법을 제공하는 것으로 cassette에서 구현되고 지불 관리자에 의해 호출되는 기능이나 MPF내에서 구현되고 cassette에 의해 호출되는 기능들을 정의한다.

Cassette는 각각의 지불 방식이나 프로토콜을 처리하는 소프트웨어로 SET, E-Check, E-Cash, MicroPay등과 같이 특정 프로토콜을 구현한 것을 의미하며 SPI에 의해 MPF와 결합된다.

3.3 MS Open Payment Architecture

Microsoft는 전자 상거래 관련 제품의 출시와 함께 Server Foundation, Payment, Solutions with Commerce Partners의 3 요소를 기반으로 하는 Internet Commerce Strategy를 발표하였다[Microsoft, 1998a] 이것의 주요 내용은 Site Server를 기반으로 하는 서버 제품과 지불 처리를 하기 위한 Microsoft Wallet, 그리고 시스템 지원을 위한 사업 파트너와의 공동 개발 전략 등이다. 이 중 여기에서 다루고자 하는 것은 Payment 요소로서 Microsoft Wallet과 Site Server를 기반으로 하는 Open Payment Architecture이다 [Microsoft, 1998b].

Open Payment Architecture는 Microsoft가 자사의 Client와 Server 시스템에 채용한 개방적이고 확장 가능한 지불 시스템 구조이다. Server인 Site Server에서는 이를 지원하기 위

해 Enterprise Edition APIs 또는 Order Processing Pipeline APIs를 제공하여 지불 시스템이 연동 할 수 있도록 하며 Client측인 Microsoft Wallet은 Wallet Payment Modules architecture로서 연동 방법을 제공한다.

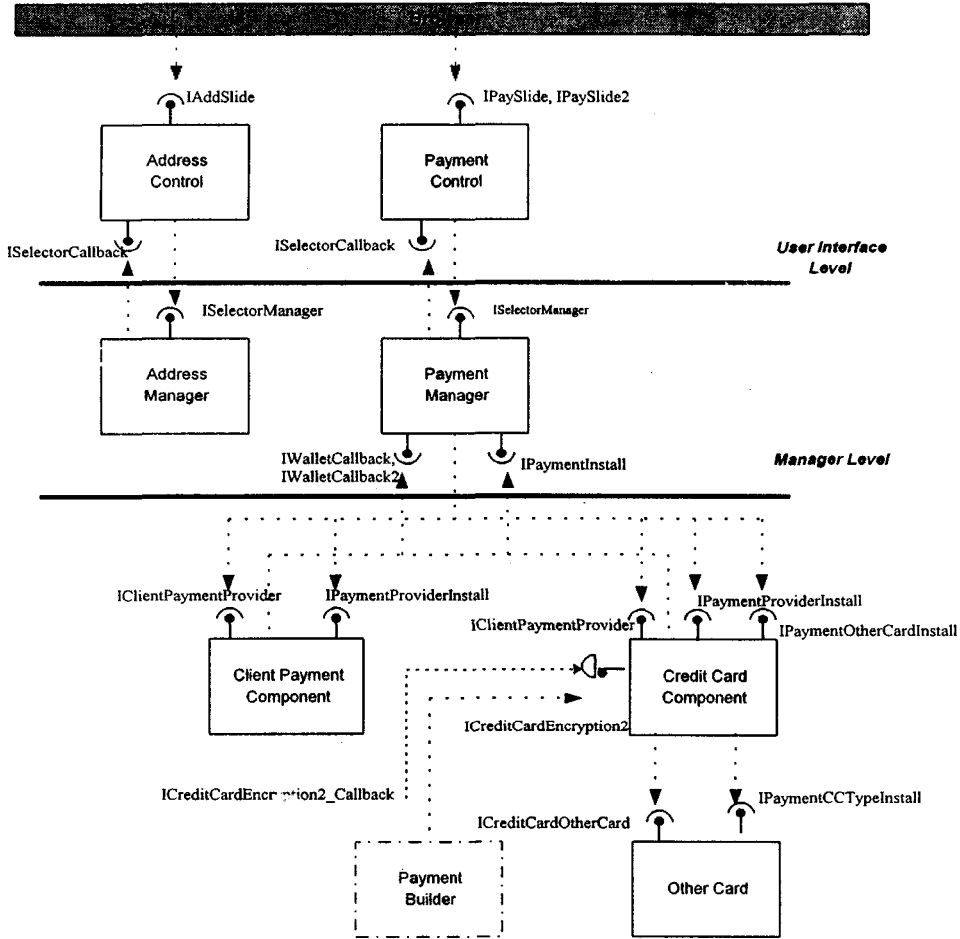
Order Processing Pipeline은 주문을 처리하기 위한 확장 가능한 데이터 구조와 API를 제공하여 개발자가 새로운 주문 처리 기능을 만들거나 기존의 시스템과 상거래 시스템을 통합할 수 있는 방법을 제공한다.

Wallet Payment Module architecture는 <그림3>에서 보인바와 같이 Payment Selector 기능과 Address Selector 기능으로 구성된다

Payment Selector는 다양한 형태의 지불 처리 방법을 구현한 모듈에 대한 일종의 관리 기능과 지불 정보를 담기 위한 보안성 있는 저장소를 제공하며 전자 현금이나 전자 수표, 또는 SET와 같은 지불 처리 모듈에 대한 연동 모듈을 제작할 수 있는 인터페이스를 제공한다.

Address Selector는 주문 처리시에 배송이나 대금 요청을 하기 위해 참조되는 주소등과 같은 정보에 대한 저장소와 API를 제공한다.

앞서 기술한 시스템들을 볼 때 현재 지불 시스템 또는 거래 처리 시스템들의 추세는 대체로 실제 주문을 처리하는 부분과 지불 처리 부분을 분리시키고 그 두 부분은 표준화된 API를 사용하여 연동되도록 한다. 그러나, 앞서의 시스템들은 대부분 가장 기반이 되는 기능들 예를 들어 암호화나 데이터베이스에 대한 접근, 또는 지불 처리에 대한 비즈니스 관점보다는 시스템 관점에서 객체성을 강조하여 지불 처리 시스템과 주문 처리 시스템을 연동



<그림 3 > MS Wallet Open Payment Architecture

하려는 방법을 사용하였다. 물론 이 방식들은 확장성을 보장하여 third-party 벤더에 의해 자신들의 비즈니스 모델이 내장된 다양한 형태의 시스템이 개발될 수 있도록 하는 면이 있으나 전자상거래 시스템에 대한 사전 지식이 없는 어제 막 전자 상거래를 시작하고자 하는 당사자들에게는 다소 시스템 규모가 크고 자신이 비즈니스 규칙을 세우고 그에 따라 시스템을 구성해야 한다는 어려움이 있다.

본 논문은 전자상거래를 시작하고자 하는 사람들이 좀더 간단하면서도 신속하게 편리한 상거래 시스템을 개발, 구축할 수 있는 시스템의 설계사양과 기술 구조를 다루고 있다.

4. 지불 처리 시스템: IPS

IPS(Intermarket Payment System)는 저자가 속한 연구팀에서 인터넷에서의 전자상거래

를 위한 쇼핑물 시스템으로 개발한 IMS (Intermarket Merchant System)에서 이용하기 위해 고안된 지불 처리 시스템이다. IPS는 현재 간단한 비즈니스 기능과 기본적인 지불 처리 모듈로 구성되어 있다.

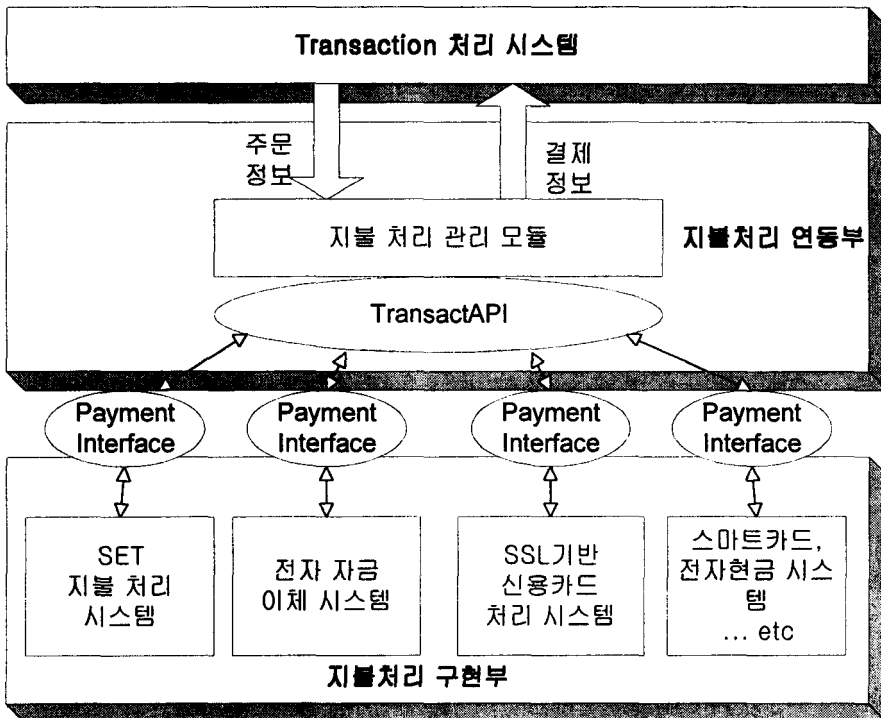
4.1 IPS의 구조

IPS는 먼저 구동 인터페이스 부분이 IMS 내부에 존재하며 사용자의 요구가 들어올 때 IMS 내부의 구동 인터페이스는 별도로 존재하는 IPS 모듈과 연동하여 지불 처리를 수행하게 된다. <그림 4>는 IPS의 구조를 보인 것이다. 그림에서 Transaction 처리 시스템은

IMS를 의미하며 고객의 주문이 들어오면 지불 요구가 지불 처리·관리 모듈로 전달된다. 지불 처리·관리 모듈은 Transact API와 연동된 Payment Interface를 통해 각 해당 지불 처리 모듈로 지불 요구를 전달하고 각 지불 처리 모듈에 의해 처리된 결과는 다시 Transact API를 통해 다시 Transaction 처리 시스템으로 전달된다.

4.2 IPS의 기능

IPS의 기능은 전자 상점 시스템에서 일어날 수 있는 지불과 관련된 행위로서 지불 요구와 환불 요구에 대한 처리를 기반으로 하며



<그림 4> IPS의 구조

<그림 5>에서 보인 바와 같은 작업 흐름에 따라 구성된다. 지불 처리의 경우 그림에서 보인바와 같이 동시처리 모델과 후처리 모델로 나뉘게 되는데 일반적으로 동시처리 모델은 지불에 대한 확인 또는 승인이 상품의 배송 이전 즉시 일어나는 모델, 예를 들어 SET, 전자 현금 등과 같이 지불 게이트웨이를 통한 연동 또는 기타 지불 승인을 얻을 수 있는 경우에 적용되며 후처리 모델은 구매 요구시 수집된 정보를 통해 구매요구 처리 이후의 임의의 시간에 지불 처리가 진행되는 것으로 구매요구시 지불에 대한 확인을 할 수 없는 경우, 예를 들어 금융망 또는 지불 게이트웨이가 직접 연결되지 않은 경우에 적용된다.

작업 흐름에 따른 IPS의 바즈니스 기능은 크게 지불처리 기능과 환불처리 기능, 조화 및

관리기능으로 구성된다.

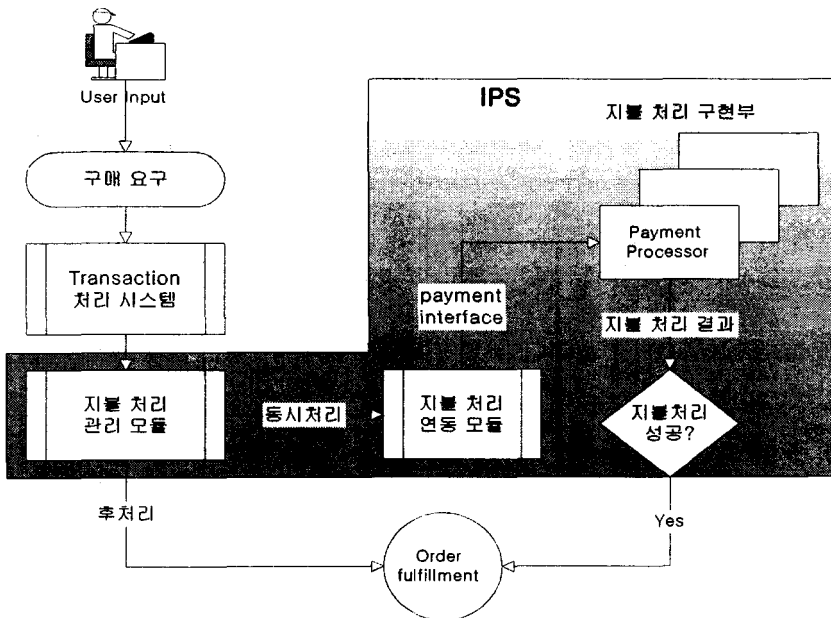
4.2.1 지불 처리 기능

(1) 거래 처리의 승인 및 알림(notify)

각 지불 방식을 처리하기 위한 모듈로부터의 결과에 따라 거래 처리의 승인 여부를 결정하고 이 결과를 데이터베이스에 기록하거나 또는 기타 메시지를 전달하는 방식으로 거래 처리 모듈에 알린다.

(2) 지불 처리 내역 기록

지불 모듈에 따라 지불 처리된 내역은 각기 지불 모듈이 별도로 관리하는 데이터베이스에 기록되며 관리자에 의해 질의, 분석될 수 있다.



<그림 5> IPS 지불 처리 작업 흐름

(3) 지불 모듈 스위칭

거래 시 선택된 지불 방법에 따라 적절한 지불 처리 모듈을 결정하고 지불 처리에 필요한 정보와 제어권을 해당 모듈로 넘겨준다.

(4) 지불 데이터 수집

고객으로부터 요구된 지불 처리에 필요한 데이터들(고객번호, 거래액 등)을 트랜잭션 API를 이용해 수집하고 이를 각각의 모듈에서 필요로 하는 형식으로 변환한다.

4.2.2 환불 처리

(1) 환불 처리 내역 기록

고객의 요구에 의해 환불 처리된 경우 그 내용은 데이터베이스에 저장되어 이후에 조회될 수 있다.

(2) 환불 모듈 스위칭

지불 모듈 스위칭과 같이 환불 처리의 경우에도 각 지불 방법에 따라 별도의 처리 모듈이 존재하며 이것 역시 지불 방법에 따라 호출된다.

(3) 환불 처리 관리

특정 결제 방식의 경우 환불처리가 용이하지 못한 경우가 있으며 환불 건별로 유형에 따라 처리 방식이 달라지게 된다. 이때 서로 다른 처리 방식에 필요한 정보 및 허가, 금지 등의 환불처리 제어를 위한 정보를 관리한다.

4.2.3 결제 조회 및 관리

(1) 배치 처리 설정 및 해제

특정 방식 별로 일괄 처리를 위한 환경의 설정 및 해제. 신용카드와 같은 신용기반 지불 처리인 경우 이후에 실제 대금을 포착하기 위한 배치처리가 필요로 되며 이러한 배치를 위한 스케줄 설정 및 관리를 수행한다.

(2) 결제 내역 조회

상품의 구매에 대해 대금 결제가 완료된 거래를 조회, 분석할 수 있도록 한다. 손익 계산 및 세금 정산을 수행하는데 필요한 정보를 제공한다.

(3) 미결제 내역 조회

상품의 구매에 대해 대금 결제가 완료되지 않은 거래를 조회, 분석할 수 있도록 한다. 지로 방식의 결제나 기타 자금 이체 방식과 같이 온라인 결제가 아닌 경우 상점관리자가 미결제된 구매 내역으로부터 정보를 얻어서 지급요구서 발송 등과 같은 결제를 위한 행위를 취할 필요가 있다.

(4) 환불 내역 조회

고객이 구매한 상품에 대해 불만족하여 해당 거래에 대해 상품의 대금을 환불한 경우 그 내역은 환불 상태가 되며 환불 내역 조회는 이러한 거래들을 조회한다. 환불 내역 조회의 경우 특정 상품 및 상품군에 대한 고객의 불만족도를 알 수 있는 정보를 제공한다

(5) 환불 요청 내역 조회

고객으로부터의 환불 요청이 있는 거래의 경우 환불 요청 상태가 되며 관리자는 환불 요청이 있는 거래 내역에 대해서 적절한 행동을 취해야 한다.

4.2.4 지불 처리 모듈 관리

(1) 지불 처리 모듈 등록 및 설치

지불 처리 시스템 상에 특정 지불 방식을 처리 할 수 있는 모듈을 등록하고 설치한다. 이렇게 설치된 모듈은 거래가 수행될 때 연동되고 해당 거래에 대한 지불 처리를 수행한다.

(2) 지불 처리 모듈 활성화 및 비활성화

일단 등록된 지불 처리 모듈은 비활성화된 상태로 존재하며 이것을 사용하기 위해서는 활성화 시켜야 한다. 활성화된 모듈은 이후의 거래에 대한 지불 처리를 수행하기 위해 대기한다.

(3) 지불 처리 이용자 제한 및 등록

등록된 지불 처리 모듈은 초기에는 모든 사용자가 이용 가능하도록 되어 있으나 상점의 정책에 따라 특정 사용자에게만 유효하도록 설정할 수 있으며 이때 이용 가능한 사용자들을 개별적으로 또는 그룹별로 등록할 수 있다.

5. 결론

본 논문에서 기술한 IPS 시스템은 아직 제한된 비즈니스 요소만을 가지고 있다. 따라서 만일 사용자가 미리 정의된 기능 외의 기능을 추가할 경우 다른 부분들과의 연관성이 없으므로 거래 처리에 관한 부분들에서부터 최종적인 지불 처리 이후의 결과에 대한 처리 기능까지 개발하여야 한다. 물론, 기존에 존재하

는 부분을 사용할 경우에는 해당 API를 사용하면 되지만 그 외의 기능인 경우에는 직접 시스템 내부 구조에 접근해야 한다. 그것은 초기에 이 시스템이 소규모 사업자를 대상으로 간단한 구조와 시스템 계층을 단순화 시켜 이러한 개발이 쉽게 이루어 질 수 있도록 한 의도도 있었지만, 반대로 시스템을 확장하고자 하는 경우에는 전체 구조에 대해 충분히 숙지하고 직접적인 방법, 예를 들어 시스템의 개발에 사용하였던 데이터베이스에 대한 직접적인 접근과 CGI 루틴의 작성들을 필요로 한다. 이것은 앞서 2장에서 다루었던 다른 지불 시스템 구조에서 취한 방법과 동일하나 다소의 다른 점이라면 지불 처리에 가능한 비즈니스 형식인 지불 처리와 환불처리에 대한 모델을 기초로 시스템이 구성되었다는 점이다.

그러나, 이것만으로는 도입부에서 기술하였던 비즈니스 표현을 이용자가 쉽게 하도록 하자는 목적에는 미치지 못한다. 실제 비즈니스 표현을 쉽게 할 수 있는 시스템이라면 시스템에서 사용되는 각각의 비즈니스 요소들을 기능별로 객체화하고 이들간의 동적인 통신이 가능하게 하는 공통 프레임워크를 필요로 하며 비즈니스 요소를 기술하여 하나의 업무 흐름을 만들 수 있는 어떤 도구를 필요로 한다.

이에 대한 몇 가지 해결책으로서 Open Financial eXchange나 Open Trading Protocol, Open Buying on the Internet 등과 같이 특정 지불 매체에 제한되지 않고 상거래 활동을 비즈니스 관점에서 모델링한 전자 상거래 기술들이 제시되고 있으며 이들 기술들을 하나의 프레임워크 내에 통합시키기 위한 노력의 일환으로 전자 상거래 개방형 프레임워크인 eCo Framework Project 등이 국제 전자 상거래 컨

소기업인 CommerceNet에 의해 진행되고 있다
[OFX, 1998 : OTP Consortium, 1998: OBI
Consortium, 1998: CommerceNet, 1998].

IPS의 차기 버전은 이들 기술들에 대한 접
목이 그 한가지 방향으로 제시될 것이며 현재
고정된 업무 흐름을 유연하게 변화시킬 수 있
는 구조를 가지도록 할 것이다.

참고문헌

- [동성정보, 1998] 동성정보통신, 전자상거래 대금결제용 전자화폐시스템-ICash, <http://www.dsi.co.kr/ICash.htm>
- [CommerceNet, 1998] CommerceNet, eCo Framework Project, <http://www.commerce.net/project/current/projects/eco/>
- [IBM, 1998a] IBM, IBM Payment Suite: Solutions for more secure e-commerce OverviewBrochure, <http://www.software.ibm.com/commerce/payment/nsb3828f.pdf>, 1998
- [IBM, 1998b] IBM, Multipayment Framework White Paper, <http://www.software.ibm.com/commerce/payment/mpf.pdf>, 1998
- [Microsoft, 1998a] Microsoft, Microsoft Internet Commerce Strategy White Paper, <http://www.microsoft.com/BACKOFFICE/commerce/whitepaper.htm>, 1998
- [Microsoft, 1998a] Microsoft, Microsoft Wallet, <http://www.microsoft.com/wallet/default.asp>, 1998
- [Mondex, 1998] Mondex International, <http://www.mondex.com>
- [OBI Consortium, 1998] The OBI Consortium, The Open Buying on the Internet(OBI), <http://www.openbuy.org/>, 1998
- [OFX, 1998] OFX, Open Financial Exchange, <http://www.ofx.net/ofx/default.asp>, 1998
- [OpenMarket, 1998] OpenMarket, The Transact 4 Architecture White Paper, http://www3.openmarket.com/lpbin/lpext.dll/vl/whitepapers/1125/tra_arch.pdf, 1998
- [OTP Consortium, 1998] OTP Consortium, Open Trading Protocol, <http://www.otp.org/>, 1998

저자 소개

송병열 (e-mail : sby@etri.re.kr)

1995 전북대학교 전자공학과 학사

1997 전북대학교 전자공학과 석사

1997~1998 한국전자통신연구소 부설 시스템공학연구소

1998~현재 한국전자통신연구원 전자상거래연구부 연구원

함호상 (e-mail : hsham@etri.re.kr)

1977 고려대학교 산업공학과 학사

1983 고려대학교 산업공학과 석사

1995 고려대학교 산업공학과 박사

1979~1981 새한자동차(주)

1982~1998 한국전자통신연구원 부설 시스템공학연구소 전자거래연구팀 팀장

1998~현재 한국전자통신연구원 전자상거래연구부 전자거래연구팀 팀장

박상봉 (e-mail : hspark@etri.re.kr)

1974년 고려대 산업공학과 학사

1976년 고려대 경영학 석사

1975~1981 한국과학기술연구소 전자계산부

1982~1993 한국과학기술원 시스템공학센터

1993~1996 한국과학기술원 시스템공학연구소

1996~1998 한국전자통신연구소 부설 시스템공학연구소

1998~현재 한국전자통신연구원 전자상거래연구부 부장