

효율적인 전자지불 시스템 구성을 위한 Payment Adapter 설계

윤 범렬, 김 수동
송실 대학교 컴퓨터 학부

Payment Adapter Design for Developing Effective Electronic Payment Systems

Bum Ryeol Yoon, Soo Dong Kim
School of Computing, Soong-Sil University

Abstract

본 논문에서는 현재 구현되어 있는 전자 상거래 서버가 특정 지불 방식만을 Transaction Logic 안에 포함하고 있으므로 계속적으로 발전하고 있는 지불 방식의 추가를 위해서 서버 프로그램의 Transaction Logic 을 고쳐야 하는 문제점을 제시하고 전자상거래 Transaction Logic 으로부터 지불 방식을 분리하여 새로운 지불 방식 추가 시 효과적으로 새로운 지불 방식을 추가 적용할 수 있도록 서버 구현 시 각 지불 방식 사용에 있어 동일한 Interface 를 이용하여 구현을 쉽게 하는 Adapter 설계 방법을 제시하는 것을 그 목적으로 한다.

1. 서론

인터넷이라는 잠재 시장이 웹을 통하여 일반 사용자도 쉽게 인터넷을 이용할 수 있게 되면서, 인터넷을 상업적으로 사용하려는 기업들에게 네트워크를 기반으로 한 전자 상거래(Electronic Commerce)가 이루어지게 되었다. 인터넷을 상업적으로 이용하기 위해서 기업이나 개인은 상품을 구매하거나 유료정보를 얻는데 대하여 웹상에서 편리하게 그 금액을 지불할 수 있는 시스템이 필요하게 되었으며 금액을 지불하는 방법도 시행착오를 거쳐 기술적 관습적인 문제를 해결해 나가고 있다. 현재 인터넷상에서 구현되어 있는 전자 상거래 시스템은 Off-Line 지불 방식 즉 전자 지불의 형태가 CGI 를 통한 신용카드 번호 전달방식을 이용하여 상대방 회사(서버)의 단말기를 거쳐 은행의 결제를 이루는 방식이 주종을 이루고 있으며 은행 계좌를 이용한 계좌 이체 방식, 또 소규모의 전자상거래 회사에서는 현금결

제를 통한 지불이 이루어지고 있다. 현재 전자상거래 시스템에서 제공하고 있는 지불 방법은 HTTP-Protocol 을 이용하여 지불정보가 전달된다는 보안상의 허점이 있으며 서버마다 다른 지불방식을 제공하고 있어 사용자의 혼란을 발생시키는 문제점을 가지고 있다. 이러한 문제를 해결하기 위해서 SET(Secure Electronic Transaction) Protocol 이 제시되었으며 향후 전자화폐, 가상 지갑(Virtual Wallet), 스마트 카드(IC Card)를 이용하여 좀 더 안전한 전자 상거래 시스템을 구축할 수 있을 것이다.

전자 지불 시스템은 서버 자체 내에 독립적인 신용 구조를 가지고 있지 않고 신용 카드나 은행 계좌를 이용해 네트워크 상에서 지불을 하도록 연결 시켜주는 지불 브로커 형태와 지금까지는 금융 제도와 사회적 관습, 제도상의 미비로 실용화되어 있지는 않으나 선불 카드나 직불 카드를 응용하거나 순수한 디지털 현금을 응용하는 전자화폐 시스템으로 나누어 볼 수 있다. 이 논문에서는 현재 사용되고 있는 지불 브로커 전자 지불 시스템을 구축하는 것을 그 도메인으로 하여 지불 Adapter 를 설계하려 한다.[4][5]

이 논문에서 제시하는 지불 Adapter 는 여러 지불 방법이 가지고 있는 공통된 기능을 추상 클래스화하여 Transaction Logic 이 이 추상클래스에 정의된 동일 메소드 이름을 이용하여 여러 지불 방법에 접근할 수 있도록 하였다. 제시하고자 하는 Payment 추상클래스는 하부 Transfer Payment(은행 계좌 이용 지불 방법), Credit Card Payment, SET 기반 Payment 방식, Electronic Cash Payment 의 공통 기능들인 인증, 지불 요구, 지불 반환 요구, 총계 확인, 현재 상태(State) 확인, Abort 등을 RequestAuthorization(), RequestPayment(), RequestReversePayment(), RequestState(), RequestAmount(), Abort() 메소드들로 표현하고 있다. 여기서 Electronic Payment 지불 방법은 다른 3 가지 지불 방법과 달리 인증을 얻기 위해 재정 기관(은행)을 접근하지 않고 전자화폐의 서명(Signature) 인증 대행 기관의 인증을 받는다.

이 논문의 목적은 Payment 추상 클래스를 이용하여 새로운 지불 방법이 추가되더라도 Payment 클래스를 상속하여 쉽게 지불 방법을 구현, 적용할 수 있으며 구축 시 동일한 메소드 이름을 사용하여 여러 지불 방법에서 동일한 기능을 수행하게 함으로써 코드의 이해성을 높이고, 구축을 쉽게 할 수 있도록 하는 것이다.

이 논문에서는 관련연구로 전자 지불을 위한 전제 조건과 암호화, 전자 서명(Digital Signature), 전자화폐에 대해서 살펴 볼 것이며 본론에서는 현재 전자상거래 상의 전자 지불 시스템의 형태와 전자상거래 상에서 이루어지는 Payment 알고리즘과 이 논문에서 제시하는 Adapter 를 적용함으로써 발생하는 변화와 Adapter 의 각 메소드의 구체적인 기능을 살펴보고 결론에서는 Adapter 의 적용으로 인한 전자 상거래 구축의 장점을 설명하고자 한다.

2. 전자 지불을 위한 요구 사항

- 정보의 무결성

전자 거래로 이루어 지는 데이터는 손실되거나, 유실되거나 위조되어서는 안된다. 즉 전자 지불 시 보안이 철저하게 유지 되어 대금 지불자와 수금자 간에 안전한 거래가 이루어 져야 한다. 이를 위해서는 암호화를 통한 보안이 필수적이다.

- 지불 Transaction 비용의 최소화

전자 지불 시스템을 이용하는데 소비자 측과 상인 측 양쪽에 전자 지불 transaction 시 발생하는 경비가 최소화되어야 한다.

- 통화단위의 통일성

특정 국가에서만 사용되는 통화여서는 안되고 세계적으로 통용될 수 있는 화폐 단위를 이용해야 한다.

- 지불 비용 범위의 다양성

Internet 상에서 이루어지는 상거래의 물품이 다양해 짐에 따라 전자화폐의 단위도 다양하게 지원되어야 한다.

- On-line transaction

고객과 상인간의 상품 가격 지불이 인터넷을 통해 즉각적으로 이루어져야 한다.

- 현금과 같은 익명성 보장

전자 지불의 경우도 현실 세계의 현금이 갖는 익명성(Anonymity)를 보장하여 사용자의 개인정보 침해를 막을 수 있어야 한다.

- 경제 변동에 무관

사용자가 물건을 사기로 서명한 일자의 화폐가치가 적용되어야 한다.[2]

3. Payment Adapter 설계

이 논문에서 제시하고자 하는 Payment Adapter 는 여러 지불 방식이 가지는 공통된 기능을 추상 클래스화 하여 Transaction Logic 상에서 동일한 메소드를 가지고 별도의 변경 없이 새로운 Payment 방식을 적용할 수 있도록 한다.

3.1 Payment Adapter 의 업무 흐름

현재 사용되고 있는 지불 브로커 시스템을 이용하여 지불하는 방법을 살펴보면 다음과 같은 절차를 실행함을 알 수 있다.

1. 고객은 사고자 하는 물품을 파는 Site 로 접속을 한다.
2. 그 Site 에서 제공하는 Shopping Cart 페이지로 옮긴 후 고객은 물품을 선택한다.
3. 주문 양식을 작성한다.
4. 지불 방식을 선택한다.
5. 작성된 주문 양식을 서버상으로 발송한다.
6. 전자 상거래 서버는 재정기관(Payment Server)로 Payment Authorization 을 요구한다.
7. 인증이 확인된 경우 재정기관에 Payment 를 요구한다.
8. 지불이 확인된 경우 주문이 완료되었음을 고객에게 알린다.

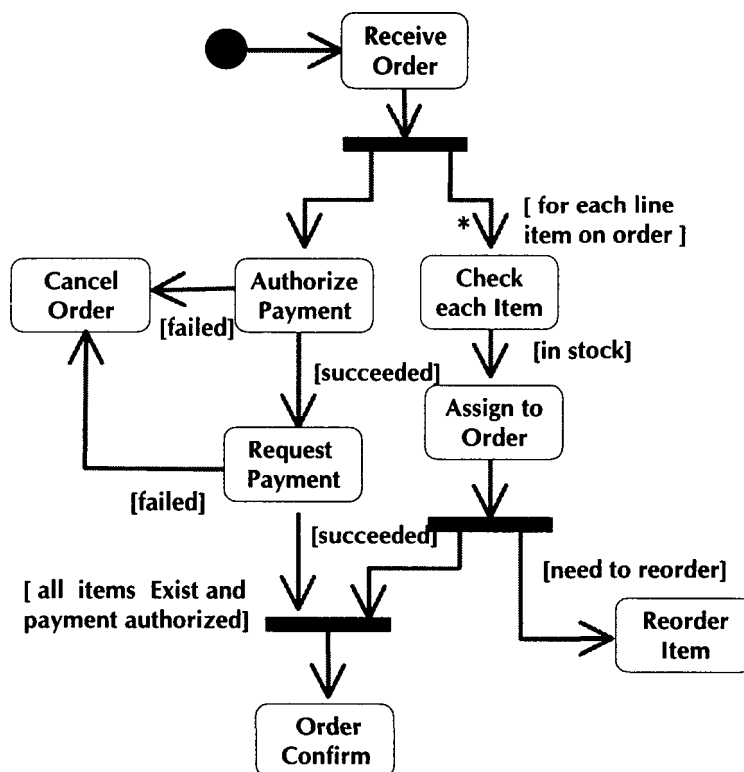


그림 1. 전자상거래 서버상에서의 프로세스

위에 제시된 절차는 지불 브로커 시스템을 이용한 지불방식의 전자상거래 실행 절차로 은행 계좌 이용 지불 방법, 신용카드 이용 지불 방법들에 공통으로 적용되는 절차이다. 위의 내용의 6번에서 8번 절차의 내용인 서버 상에서의 작업을 살펴보면 UML의 Activity Diagram으로 표현한 그림 1과 같다.[1] 그림 1에서 볼 수 있듯이 지불 브로커 시스템은 고객이 만든 주문을 받은 후 Payment Server로 계좌 조회를 신청한다. 계좌가 존재하는 경우 Payment Server로 지불을 요구하고 지불이 완료된 경우 상품 재고가 있을 경우 바로 고객에게 지불이 완료 되었음을 알린다.

3.2 Payment Adapter의 API 정의

그림 1의 내용을 바탕으로 Payment Adapter의 각 메소드들을 추출할 수 있을 것이다. Payment Server로부터 인증을 요구하는 RequestAuthorization(), Payment Server로부터 지불을 요구하는 RequestPayment() 메소드를 추출할 수 있다.

RequestAuthorization()의 경우 Parameter는 User의 신용카드 ID와 Password로 구성할 수 있으며, 또는 은행의 계좌번호 또는 신용카드 번호를 이용할 수 있다. 또 암호화를 통하여 지불 브러커 시스템상에서 사용자 정보를 알 수 없도록 사용자 익명성을 제공하도록 구성된 암호화된 코드를 넘길 수도 있다. 또 다른 파라메터로 여러 재정기관을 구분할 수 있는 내용도 있어야 한다. 그러면 RequestAuthorization()의 인터페이스를 다음과 같이 정의할 수 있다.

- **boolean RequestAuthorization(String ID, String Password, int BankNum);**
- **boolean RequestAuthorization(String AccountNum, int BankNum);**
- **boolean RequestAuthorization(Object SecureCode, int BankNum);**

RequestPayment의 경우 파라메타는 위에서 제시된 파라메타들을 포함하고 요청액과 지불 요청 기관, 지불 요청 날짜, 기관의 인증을 암호화한 코드를 가지고 있어야 하며 Return 값으로는 지불 완료와 잔액부족을 알 수 있도록 boolean 값을 Return 해야 한다. 그러면 RequestPayment()의 인터페이스를 다음과 같이 정의할 수 있다.

- **boolean RequestPayment (String ID, String Password, int BankNum, String RequestOrganization, Date RequestDate, Money RequestMoney, Object OrgSecureCertifiaction);**
- **boolean RequestPayment (String AccountNum, int BankNum, String RequestOrganization, Date RequestDate, Money RequestMoney, Object OrgSecureCertifiaction);**

- `boolean RequestPayment (Object SecureCode, int BankNum, String RequestOrganization, Date RequestDate, Money RequestMoney, Object OrgSecureCertifiacion);`

지불이 종료된 후 고객에게 주문이 완료되었음을 알릴 경우 고객에게 고객의 은행 잔고가 어느 정도 남아 있는지 잔고를 알려 주는 메소드인 `RequestAccount()`도 있을 수 있다. 이 메소드는 `RequestAuthoirization` 과 동일한 입력 파라미터를 가지며 출력 값으로는 은행잔고를 Return 한다.

- `Money RequestAccount(String ID, String Password, int BankNum);`
- `Money RequestAccount(String AccountNum, int BankNum);`
- `Money RequestAccount(Object SecureCode, int BankNum);`

3.3 주문 취소 기능의 모델링

고객이 주문을 취소하였을 경우 서버 상에서의 프로세스에 대해 그림 2에서 설명하고 있다.

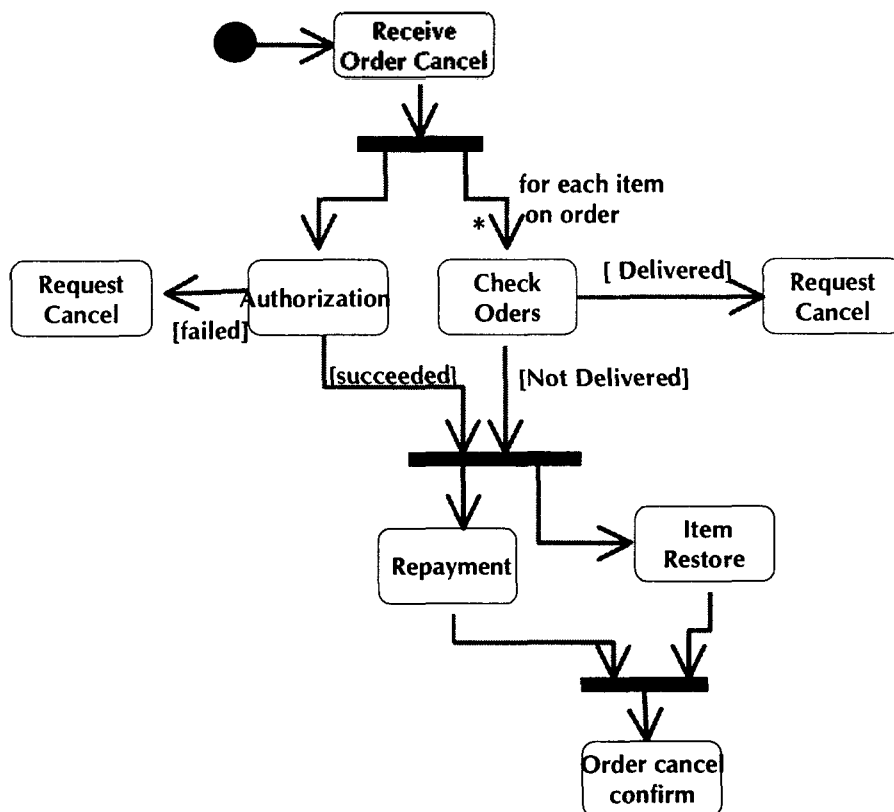


그림 2. 전자상거래 서버 내에서 주문 취소 프로세스

고객은 자신이 입력한 상품 주문에 대해서 배달이 이루어지기 전까지는 주문을 취소할 수 있어야 한다. 그림 2의 내용으로 알 수 있듯이 고객이 자신의 주문 목록 전체 또는 일부를 삭제할 수 있다. 먼저, 전자 상거래 서버는 주문 취소를 고객으로부터 받으면 지불 서버에서 그 고객의 계좌를 확인하고 주문 배달이 이루어졌는지를 확인한다. 계좌가 존재하지 않거나 주문 취소한 주문이 없는 경우, 주문 배달이 이루어진 경우에는 주문취소 불가를 알리게 되며 계좌가 존재하고 주문이 배달이 이루어진 경우에는 지불 서버상에 인출한 금액을 복구하고 주문 내용을 삭제하고 상품 수를 복구한다. 이 후 고객에게 주문 취소가 제대로 이루어졌음을 알리게 된다.

주문 취소 또한 지불 방식에 무관하게 적용할 수 있는 내용이며 전자 상거래 서버가 지불 서버에게 지불을 반환하는 메소드 RequestReversePayment()를 정의할 수 있을 것이다. RequestReversePayment()의 경우는 RequestPayment()와 동일하게 고객을 구분할 수 있는 내용, 반환액, 지불 반환 기관, 지불 반환 날짜, 기관의 인증을 암호화한 코드를 가지고 있어야 하며 Return 값으로는 반환 완료와 반환 실패를 알 수 있도록 boolean 값을 Return 해야 한다.

그러면 RequestReversePayment()의 인터페이스를 다음과 같이 정의할 수 있다.

- **boolean RequestReversePayment(String ID, String Password, int BankNum, String RepayOrganization, Date RepayDate, Money RepayMoney, Object OrgSecureCertifiaction);**
- **boolean RequestReversePayment(String AccountNum, int BankNum, String RepayOrganization, Date RepayDate, Money RepayMoney, Object OrgSecureCertifiaction);**
- **boolean RequestReversePayment(Object SecureCode, int BankNum, String RepayOrganization, Date RepayDate, Money RepayMoney, Object OrgSecureCertifiaction);**

전자 상거래 서버가 자신의 지불 Transaction의 상태를 파악하기 위해서 Payment Adapter 내에는 상태 정보를 Return 하는 메소드를 정의할 필요가 있다. 이를 위해서 멤버 변수로 State를 나타내는 변수들이 필요하며 그 내용은 다음과 같다.

StateType = { ordered, authorized, shipped, received, paid, cancelled, completed, error };

또 이 StateType 들을 Return 하여 Payment Transaction의 상태를 알아볼 수 있는 메소드 RequestState()를 정의하고 있다. RequestState()는 입력 파라미터가 없으며

Return 값으로 Payment Transaction 의 State 를 return 한다.

- State RequestState();

4. Payment Adapter 의 객체 모형도

위에서 정의한 각 메소드들과 멤버변수들로 구성된 Payment Adapter 는 그림 3과 같다.

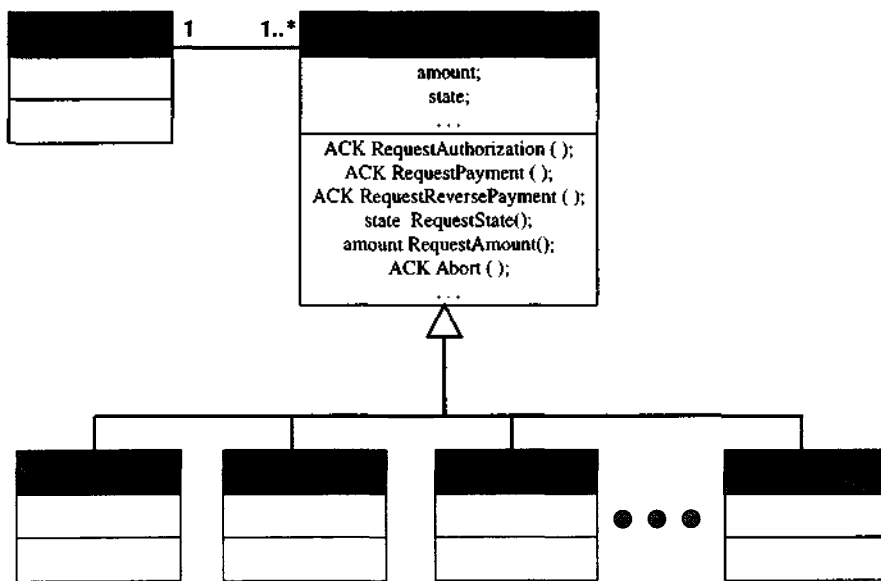


그림 3. Payment Adapter

그림 3에서 알 수 있듯이 Transaction logic 으로부터 지불 방식을 분리하여 여러 지불 방식을 지원할 수 있도록 하였으며 Payment Adapter 를 이용하여 각 지불 방식들의 공통되고 꼭 필요한 기능들의 Interface 를 정의하고 있다. Transaction 과 Payment Adapter 는 일대다 관계를 가지게 되며 Payment Adapter 를 각 Payment 방식이 상속하게 됨으로써 상속된 SubClass 의 함수는 Runtime 시 Dynamic binding 을 이룰 수 있으며 사용자가 지불방식을 결정할 때까지 Binding Time 을 늦출 수 있는 장점이 있다.

5. 결론

Payment Adapter 를 이용하여 전자상거래 시스템을 구현함으로써 시스템 관리자는 새로운 지불 방식을 시스템에 적용하는데 대하여 전자 상거래 시스템 전체적인 내용을 파악하지 않고, 다만 Payment Adapter Interface 만을 상속한 새로운 지불 방식을 적용함으로써 문제를 해결할 수 있게 되었다.

향후 Payment Adapter 를 적용한 전자 상거래 System 을 구축하여 이 논문에서 제시되지 못한 여러 문제를 해결하여 좀 더 쉽게 전자 상거래 지불 시스템을 구축할 수 있도록 할 것이다.

참고 문헌

- [1] Martin Fowler , Kendall scott , UML Distilled, Addison-wesley 1997
- [2] Davis, R.D. and C.Lagoze. *A protocol and server for a distributed digital technical report library*. Tech. Rep. 94-1418, Dept. of Computer Science, Cornell University, 1944
- [3] Handley, M. and Crowcroft, J. *The World-Wide Web*. UCL Press, 1995.
- [4] Baran, N. *Inside the Information Superhighway Revolution*, Coriolis Group Books, Scottsdale, AZ. 1995
- [5] Bennahum, D. "The trouble with e-cash", *Marketing Computer*, Vol.15 No.4, pp.25-7, 1995

저자소개

김 수동

1984, 전산학 학사, Northeast Missouri State University

1988, 전산학 석사, The University of Iowa

1991, 전산학 박사, The University of Iowa

1991 - 1993 한국통신 연구개발단 연구실장/선임연구원

1993 - 1994 The University of Iowa, 교환교수

1994 - 1995 현대전자 S/W연구소 책임연구원

1995.9 - 현재 송실대학교 컴퓨터학부 조교수

주요 관심 분야

객체지향 개발방법론 (UML, Java)

웹 기반 분산 객체 컴퓨팅 (CORBA, Intranet, Client-Server Web, Web Agents)

인터넷 상거래 시스템 기술

윤 범렬

1997, 전산학 학사, 경원 대학교

현재, 송실대학교 컴퓨터학부 대학원생 (석사과정)

주요 관심 분야

객체지향 개발방법론 (UML, Java)

분산 객체 컴퓨팅 (Mobile Agent, CORBA, Client-Server Web)