

Adapter패턴을 이용한 EJB 컴포넌트 조립 기법 및 응용

An Approach to Composite Techniques and Application of EJB Component using Adapter Pattern

정 화 영*
Hwa-Young Jeong

요 약

디자인 패턴, 컴포넌트 기반 기법 등의 소프트웨어 개발기법 변화에 따라 웹 서버 시스템 개발 부분에서도 이를 도입하여 다양한 사용자 요구사항에 맞도록 능동적으로 대처하고 있다. 특히, EJB는 서버측 컴포넌트 모델로서 쉽고 다양한 개발지원으로 인하여 많은 연구 및 도입되고 있다. 또한, 컴포넌트 조립기법은 메시지 전달방법에 의한 아키텍처기반의 C2 스타일 아키텍처가 쉽고 효율적인 적용방안으로 많은 관심을 받고 있다. 그러나, 쓰레드를 호출할 수 없는 EJB컴포넌트의 경우, C2 스타일 아키텍처에 의한 컴포넌트 합성을 위해서는 C2 프레임워크의 수정이 불가피하다.

이를 위해 패턴 중 구조패턴의 하나인 Adapter패턴을 이용한 기법을 제안하고자 한다. 즉, Adapter패턴에 의하여 쓰레드를 호출할 수 있는 EJB로 변환함으로써 C2 스타일 아키텍처에 의한 컴포넌트 합성이 가능하도록 하였다.

Abstract

According to change of software development technique as design pattern, component based development etc., web server development part is applied activity to fit in various user requirement. Especially, EJB as server side component model much researches and introduce by easy and various development support. Also, component composing techniques is receiving many interests by easy and efficient application way that architecture based C2 style architecture with message handling. But, in case of EJB component that can not call thread, C2 framework modification is unavoidable for component composition by C2 style architecture.

For this, we propose techniques to use Adapter pattern to be one of structure pattern among pattern. That is, did so that component composition by C2 style architecture may consist changing to be able to call thread function in EJB by Adapter pattern.

키워드 : CBD, Server Component Application, Transaction Communicator, EJB, C2, Web Application

1. 서 론

소프트웨어 개발기법의 향상에 따라 웹 기반 시스템들도 많은 부분에서 이를 도입 및 응용하고 있다. 이러한 소프트웨어 개발 기법에서 디자인 패턴은 객체지향 소프트웨

어에서 경험을 활용하기 위해 제안된 기술이며[1], 시스템 구축시 이를 응용 및 도입하고있다. 또한, 컴포넌트 기반 개발기법(CBD)은 소프트웨어 프로그래밍에서 하드웨어 개발 환경처럼 소프트웨어를 Plug-and-Play 방식으로 시스템을 구축하는 '합성을 통한 시스템 구축'으로의 전환을 목적으로 한 방법이다[2,3].

* 정 화 원 : 예원대학교 정보경영학부 전자상거래학과 교수
jmichael@hanmir.com (제 1저자)

따라서, 현재의 웹 기반 시스템 구축은 각 기능단위의 비즈니스 로직을 포함한 웹 컴포넌트들을 조립하여 전체 시스템을 완성하고 있다. JAVA를 기반으로 한 서버측 컴포넌트 모델로는 EJB (Enterprise Java Beans)를 들 수 있으며, 체계적인 구조로 인한 쉬운 개발로 인하여 많은 관심을 받고 있다. 이러한 컴포넌트들이 서로 정확하게 결합하고 작동되려면 아키텍처를 기반으로 한 컴포넌트의 생성과 합성작업이 이루어져야 한다[4]. 컴포넌트 합성방법에서는 컴포넌트 간의 인터페이스 불일치를 해결할 수 있어야 하며, 독립적인 컴포넌트의 메소드 수정 없이 합성할 수 있어야 한다[5].

아키텍처 기반 컴포넌트 조립기술 중 C2 스타일 아키텍처[6]는 GUI방식의 멀티 쓰레드 기반 메시지 호출방식을 택하고 있으며, 비 동기적인 상호작용을 지원하는 대표적인 구조를 갖고 있어 쉽고 체계적인 조립 구축이 가능하다. 그러나 서버측 컴포넌트 모델인 EJB의 경우는 자체적으로 쓰레드를 호출 및 운용할 수 없으므로[7] C2 스타일 아키텍처에 의한 조립적용을 위해서는 수정이 필요하다.

따라서, 본 연구에서는 서버측 컴포넌트 모델인 EJB를 C2 아키텍처에 의하여 조립 합성함으로써 전체 시스템을 구축하는 기법을 제안하고자 한다. 이를 위하여 C2 아키텍처에서 적용될 수 없었던 쓰레드 호출부분을 Gamma[8]의 디자인패턴 중 구조화패턴에 속하는 Adapter 패턴을 이용하여 EJB 컴포넌트를 위한 쓰레드의 생성 및 운영할 수 있도록 함으로써 C2에 의한 조립이 가능하도록 하였다. 본 연구의 구성은 2장에서 기존의 디자인 패턴 중 Adapter 패턴과 C2 아키텍처에 의한 컴포넌트 합성기법에

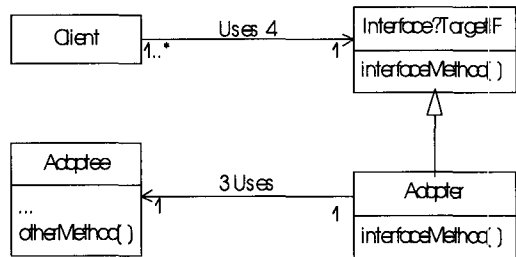
대한 연구들을 살펴본다. 3장에서는 C2 아키텍처에 따른 EJB 컴포넌트의 합성에 관한 기존의 연구된 제안기법과 본 연구에서 제시한 EJB컴포넌트 적용기법을 나타내며, 4장에서는 제안한 기법을 EJB 컴포넌트에 적용한 좌석예약 시스템을 구현함으로써 본 연구에서 제안한 기법이 실제 직접 적용될 수 있음을 보인다.

2. 관련연구

2.1 Adapter 패턴

디자인 패턴은 소프트웨어 설계에 관한 추상화 형태의 해결방법으로서, Gamma[8]는 생성, 구조, 행위패턴으로 분류하고있다.

Adapter 패턴은 이들 중 구조화패턴에 속하며, 그림 1과 같이 이미 제공되어 있는 것과 필요한 것 사이의 간격을 연결하는 패턴을 말한다[9].

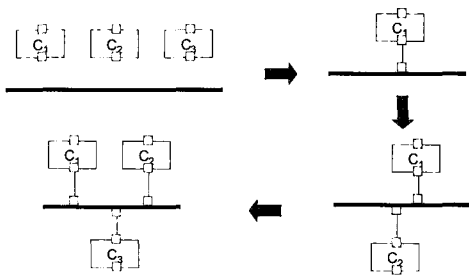


(그림 1) Adapter Pattern

TargetIF는 필요로 하는 기능을 제공하는 역할로서 Interface로 구성하되, Adaptee는 적합되는 측으로 이미 준비되어 있는 기능을 나타내고 Adapter는 적합하는 측으로 Adaptee의 기능을 사용해서 Target의 역할을 충족시키는 부분을 나타낸다.

2.2 C2 아키텍처에 의한 컴포넌트 합성 기법

C2 스타일 아키텍처에 의한 컴포넌트 조립의 기본원칙은 메시지기반의 컴포넌트간 통신, 멀티 쓰레드, 각 계층의 독립성, Message Routing Connector를 통한 컴포넌트들의 연결구성, GUI 소프트웨어 요구사항 지원 등이다. 또한, 분산 및 이 기종 환경, 분할된 주소공간을 갖지 않는 환경에서의 컴포넌트실행, 다중사용자 및 다중사용자 툴킷, 실행시간에서 변화될 수 있는 동적 구조 등에 적합하다[10]. C2 스타일의 컴포넌트 조립은 그림 2와 같이 각각의 커넥터를 중심으로 C2 컴포넌트들을 추가, 삭제, 재연결 할 수 있다[11].



(그림 2) C2 컴포넌트 조립

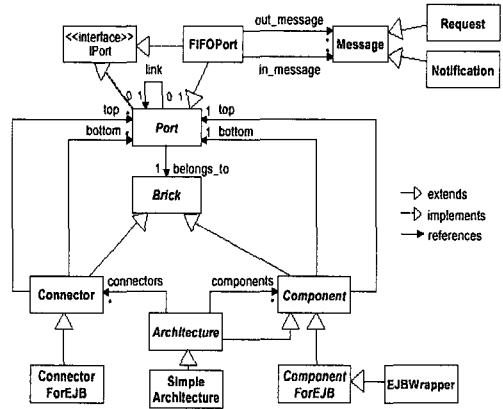
3. C2 아키텍처에 의한 EJB 컴포넌트 조립기법

3.1 기존의 연구기법

C2 아키텍처에서는 EJB 적용에 맞지 않는 컴포넌트와 커넥터를 위한 쓰레드를 사용하고 있다. 또한, EJB 컴포넌트의 수정 없이 합성될 수 있어야한다.

따라서, 최유희[5]의 연구기법에서는 다음 그림 3과 같이 EJB에 적용하기 위하여

컴포넌트와 커넥터 쓰레드 부분을 각각 ComponentForEJB와 ConnectorForEJB로 변경하였다.



(그림 3) EJB 적용을 위한 기존의 C2 아키텍처 프레임워크 변경기법

또한, EJB 컴포넌트를 위한 중간 매개체 역할을 하는 EJBWrapper를 두어 EJB의 홈/원격 인터페이스를 가져와서 메소드를 호출하며, 그 결과를 메시지 형태로 변환하여 C2 아키텍처에 따라 합성할 수 있도록 하였다.

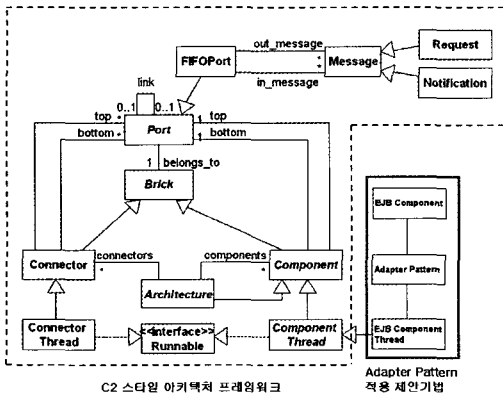
그러나, 이러한 기법은 EJB 적용을 위하여 컴포넌트 조립합성기법인 C2 아키텍처의 프레임워크 변경이 필요하였다. 또한, EJB 컴포넌트들을 위한 EJBWrapper가 별도로 필요하며, 이 부분에서 메소드 호출 및 결과를 메시지 형태로 변환하여야만 C2 아키텍처의 합성기법에 따라 운용이 가능하였다.

3.2 Adapter 패턴을 적용한 제안기법

쓰레드를 기반으로 한 메시지 호출방식의 컴포넌트 합성모델인 C2 스타일 아키텍처에서는, 쓰레드를 호출할 수 없으며 직접적

인 메소드 호출방식의 서버 컴포넌트 모델인 EJB에서는 적용할 수 없었다. 따라서, 직접적인 메소드 호출방식을 메시지 호출방식으로 전환하여야하며 자체적인 쓰레드 호출이 가능하도록 변환이 필요하다.

이를 위하여, 본 연구에서는 C2 아키텍처에서의 컴포넌트와 커넥터 쓰레드부분에서, EJB컴포넌트를 Adapter패턴을 이용하여 C2의 컴포넌트 요소에 바로 적용할 수 있도록 변경하였으며 커넥터 쓰레드는 수정 없이 사용하였다.

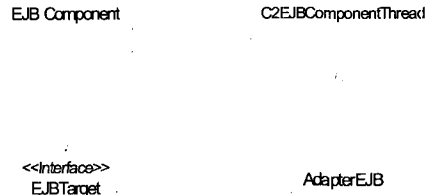


(그림 4) Adapter Pattern을 적용한 C2 아키텍처 프레임워크

즉, EJB컴포넌트의 C2 아키텍처 컴포넌트 적용을 위하여, 그림 4와 같이 Adapter 패턴을 이용함으로써 C2 프레임워크에서 EJB 컴포넌트 연결을 위한 Component Thread부분을 연결하도록 하였다. 따라서, C2 프레임워크에서는 EJB 컴포넌트에 관하여 컴포넌트 쓰레드를 호출할 수 있으며, EJB 컴포넌트 부분은 소스의 수정 없이 Adapter 패턴에 의하여 C2 프레임워크에 적용될 수 있다.

다음 그림 5는 이를 위한 EJB 컴포넌트와 C2 아키텍처 적용을 위한 컴포넌트 쓰

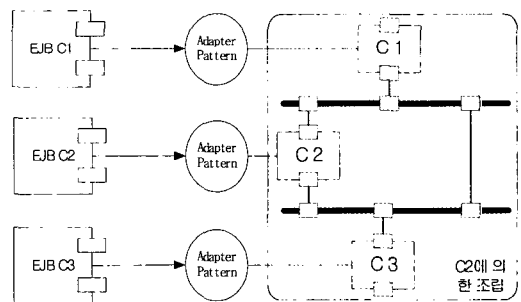
레드 사이의 구성을 나타낸다. 즉, Client는 EJB 컴포넌트가 되며, TargetIF부분은 필요한 구성요소로 변경하기 위한 인터페이스로서 EJBTarget부분이 된다.



(그림 5) C2 프레임워크 적용하기 위한 Adapter Pattern

EJBTarget은 EJB 컴포넌트를 C2 아키텍처의 컴포넌트에 매칭 시키기 위한 인터페이스를 구현한다. 또한, Adapter부분은 AdapterEJB가 담당하며 인터페이스를 실제로 구현하여 EJB 컴포넌트가 C2아키텍처의 컴포넌트로서 작동하도록 변환한다. Adapter는 C2EJBComponentThread 부분이며 적용하고자하는 C2 아키텍처의 Component Thread를 상속받은 EJB 쓰레드이다.

Adapter 패턴에 따라 재구성된 각 컴포넌트는 다음 그림 6과 같이 합성하여 운용되어진다.

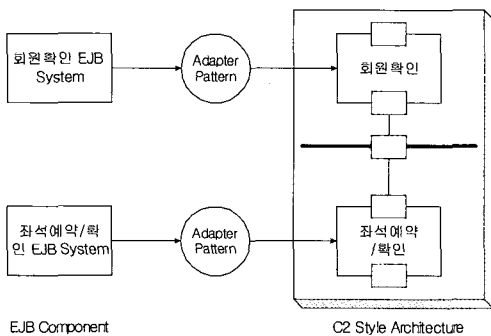


(그림 6) C2 스타일 아키텍처에서의 EJB 컴포넌트 조립을 위한 Adapter 패턴 적용

각 EJB 컴포넌트 EJB C1, EJB C2, EJB C3에 대하여 Adapter Pattern을 적용해서 C2 스타일 아키텍처의 컴포넌트 C1, C2, C3로 변환됨으로서, 변환된 컴포넌트는 커넥터 쓰레드와 컴포넌트 쓰레드에 의하여 해당 포트를 통한 EJB 컴포넌트 메소드 호출 메시지를 전달받고 처리하며 이를 다음 컴포넌트에 전달할 수 있다.

4. EJB 기반 좌석예약 시스템

본 제안기법인 Adapter 패턴을 이용한 EJB 컴포넌트의 조립 및 운용이 실제 적용됨을 보이기 위하여 전자상거래 시스템에서 사용될 수 있는 간단한 좌석예약 시스템에 적용 및 구현하였다. 이에, 본 예제 시스템은 인터넷을 기반으로 한 극장의 좌석예매, 일반 대중교통의 좌석예매, 야구장 등의 스포츠 관람 좌석예매 등에 간단히 사용될 수 있는 것으로서 J2SDK1.3과 EJB를 위한 J2SDKEE1.2.1, 데이터베이스를 위한 Cloudscape 를 이용하였다. Adapter 패턴을 적용하여 좌석예약/확인 EJB에 의한 C2 스타일 아키텍처 구성은 그림 7과 같이 구성하였다.

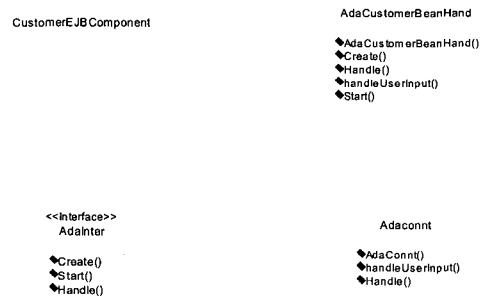


(그림 7) 예제 시스템 구성도

회원확인 EJB System은 Adapter Pattern

에 의해 회원확인 C2 스타일 아키텍처를 위한 회원확인 컴포넌트로, 좌석예약/확인 EJB System은 C2 스타일 아키텍처를 위한 좌석예약/확인 컴포넌트로 각각 변환된다. 이들 사이에 커넥터를 두어 합성함으로써 C2 스타일 아키텍처 모델을 구성하였다.

회원확인 컴포넌트를 위한 Adapter Pattern 부분은 다음 그림 8과 같이 구성하였다. 즉, 기존의 EJB 컴포넌트인 CustomerEJBComponent 부분을 C2 스타일 아키텍처에서 사용 가능한 AdaCustomer BeanHandler 부분으로 변환하였다.



(그림 8) 회원확인을 위한 Adapter Pattern

EJBComponent에서 Adapter에 의해 변환 적용할 인터페이스인 AdaInter부분은 다음과 같이 구현하였다.

```
public interface AdaInter {
    public abstract void create(String _name);
    public abstract void start();
    public abstract void handle(Request r);
    public abstract void handle(Notification n);
}
```

이를 실제 구현한 Adapter부분인 Adacontnt에서는 C2 아키텍처의 Component-Thread부분에서 처리하는 Ada Customer

-BeanHand의 메소드들을 호출하도록 하였다. 따라서, 다음과 같이 각 메소드들이 직접 처리하지 않고 부모 클래스가 메소드에 관한 처리를 하도록 하였다.

```
public class AdaConnt extends
    AdaCustomerBeanHand implements
    AdaInter {
public AdaConnt (String _name)
{super(_name);}
public void create(String _name)
    { super.create(_name); }
public void start() { super.start(); }
public void handle(Request r) {
super.handle(r); }
public void handleUserInput (Request r)
    { super.handleUserInput(r); }
public void handle(Notification n)
    { super.handle(n); }
}
```

C2 아키텍처에서 ComponentThread부분을 상속받은 AdaCustomerBeanHand는 Adapter에 의해 다음과 같이 구현하였다.

```
public class AdaCustomerBeanHand extends
ComponentThread {
public void start () {
super.start ();
Request 메시지 등록
}
public synchronized void handle(Request r)
{
Request 메시지 전송
}
public synchronized void
handle(Notification n) {
Notification 생성 및 전송
}
}
```

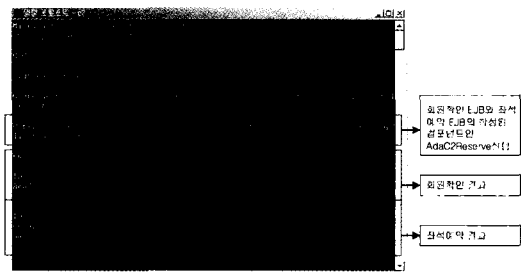
이와 같은 방법으로 좌석예약 및 확인 EJB 컴포넌트도 구현하였다. 따라서, 회원 확인 EJB Adapter와 좌석예약/확인 EJB Adapter를 BindingBus 커넥터에 연결하여 그림 7과 같이 합성하면 다음과 같다.

```
AdaConnt CustHnd = new
AdaConnt("Customer");
AdaReserv ReservHnd =
new AdaReserv("Reserve");
ConnectorThread binding_bus =
new
ConnectorThread("BindingBus");
addComponent (CustHnd);
addComponent (ReservHnd);
addConnector (binding_bus);
weld (CustHnd, binding_bus);
weld (binding_bus, ReservHnd);
start();
```

즉, 회원확인 EJB Adapter 객체인 CustHnd와 좌석예약/확인 EJB Adapter인 ReservHnd를 각각 생성하고, 생성된 컴포넌트를 연결하는 커넥터 binding_bus도 생성한다.

이를, C2의 addComponent와 addConnector 메소드를 통하여 C2아키텍처에 컴포넌트와 커넥터를 등록하고 Weld 메소드를 통하여 생성된 컴포넌트와 커넥터를 연결하며 합성시킨다. 이를, start() 메소드에 따라 스레드를 동작시키면 회원확인 컴포넌트는 회원확인 요구 메소드 호출 메시지를 받아서 처리한 후 그 결과를 커넥터에 전달하고, 커넥터는 해당 회원의 예약/확인 요구 메소드 호출 메시지를 받아서 좌석예약컴포넌트에 전달하여 그 결과를 얻음으로서 C2 스타일 아키텍처에 의한 컴포넌트 합성 결과를 얻을 수 있었다. 이에 따라, EJB서버의 가동을 위한 J2EE 서버를 가동한 후, 데이터베

이스를 위하여 J2EE에 내재된 Cloudscape 엔진을 기동한 다음 이를 실제 적용하여 구현된 결과는 다음 그림 9와 같다. 이는, 회원관리 EJB 컴포넌트인 MyCustomer AppClient.jar과 좌석예약 EJB 컴포넌트인 MyReservationAppclient.jar을 합성한 Ada-C2Reserve를 실행함으로써 회원확인 결과와 좌석예약결과를 나타낸다.



(그림 9) Adapter 패턴에 의해 합성된 EJB 컴포넌트 실행 결과

5. 결론 및 향후 연구 방향

GUI 기반의 컴포넌트 합성기법인 C2 스타일 아키텍처에서는 쓰레드를 기반으로 컴포넌트와 커넥터를 통한 메시지 교환방식을 택하고 있다. 따라서, 쉽고 효율적이며 시각적인 컴포넌트의 합성기법으로 알려져 있다. 그러나, 쓰레드를 호출할 수 없는 직접적인 메소드 호출방식을 택하고 있는 EJB 컴포넌트에서는 이를 활용할 수 없었다.

따라서, 본 논문에서는 C2 스타일 아키텍처에서 쓰레드를 호출할 수 없는 EJB 컴포넌트의 적용기법을 제안하였으며, 실제 예제 시스템을 구현하여 적용된 결과를 확인하였다. 이를 위하여, 디자인 패턴중 구조화 패턴의 하나인 Adapter 패턴을 적용하여 EJB컴포넌트에서 쓰레드를 호출할 수 있도록 전환하였다. 따라서, C2 스타일 아키텍

처에서 EJB 컴포넌트 쓰레드를 호출할 수 있었으며 EJB 컴포넌트를 합성하여 실제 구현됨을 확인할 수 있었다.

향후 연구과제로는 각 컴포넌트를 C2 EJB 쓰레드 적용을 위한 개별 Adapter 패턴이 아닌, 보다 일반화되고 구조화된 단일 Adapter 패턴에 의한 핸들링이 필요하다.

참고 문헌

- [1] C. Marcos, M. Compos, and A. Pirotte, "Reifying Design Patterns as Metalevel Constructs", *Electronic Journal of Sadio*, 2(1) pp.17-19, 1999.
- [2] F. Brosard, D. Bryan, W. Kozaczynski, E. S. Liongorari, J. Q. Ning, A. Olafsson, and J. W. Wetterstrand, "Toward Software Plug-and-Play", in *Proc. of the 1997 Symposium on Software Reusability*, 1997.
- [3] P. C. Clements, "From Subroutines to Subsystem : Component-Based Software Development", *Component Based Software Engineering*, IEEE CSPress, 1996.
- [4] 신동익 외6인, "C2 스타일의 아키텍처 기술을 지원하는 ADL 지원도구의 개발", *한국정보처리학회 논문지 Vol. 8-D, No 6*. 2001.
- [5] 최유희, 권오천, 신규상, "C2 스타일을 이용한 EJB 컴포넌트 합성방법", *한국정보처리학회 논문지, Vol. 8-D, No 6*. 2001.
- [6] The C2 Style, <http://www.ics.uci.edu/pub/arch/c2.html>, Information and Computer Science, University of California, Irvine.
- [7] Sun Microsystems Inc, "Enterprise Java Beans Specifications", at URL: <http://java.sun.com>
- [8] E. Gamma, R. Helm, R. Johnson, and J.

- Vlissides, "Design Patterns-Elements of Reusable Object-Oriented Software", Addison-Wesley, 1995.
- [9] Adapter Diagram : Overview of Design Patterns, Patterns in Java Volume 1: A Catalog of Reusable Design Patterns Illustrated with UML, http://www.mindspring.com/~mgrand/pattern_synopses.htm, 2002
- [10] R. N. Taylor, N. Medvidovic, K. M. Anderson, E. J. Whitehead, Jr. Robbins, J. E. Nies, K. A. Oreizy, P. and D. L. Dubrow, "A Component-and Message-Based Architectural Style for GUI Software", IEEE Transactions on Software Engineering, Vol.22. No.6., June, 1996.
- [11] Nenad Medvidovic and Richard N. Taylor, "A Classification and Comparison Framework for Software Architecture Description Languages", IEEE Transactions on Software Engineering, Vol. 26, No. 1, January 2000.

○ 저 자 소 개 ○



정 화 영

1991년 목원대학교 수학교육과 (학사)

1994년 경희대학교 대학원 전자계산공학과 (석사)

2001년 경희대학교 대학원 전자계산공학과 수료(박사)

2000년 ~ 현재 : 예원대학교 정보경영학부 전자상거래학과 교수

관심분야 : 소프트웨어 공학, 컴포넌트 기반 개발기법, 컴포넌트 조립/합성, 웹 엔지니어링.

E-mail : jmichael@hanmir.com, hy0917@orgio.net