DBpia

이러닝 웹 서비스를 위한 ADL 기반의 서비스 브로커 인터페이스

정회원 정 화 영*

The ADL based Service Broker Interface for E-Learning Web Service

Hwa-Young Jeong* *Regular Member*

요 약

최근, 웹을 기반으로 한 이러닝 분야에서도 동적인 프로세스 운용이 가능한 웹 서비스 적용에 관한 연구가 진행되고 있으며, 웹 서비스 브로커는 웹 서비스들 사이의 연결을 담당한다. 그러나 많은 웹 서비스들이 이러닝 분야에 효율적으로 적용되지 못하고 있으며, 정형기반의 서비스 브로커를 제시하지 못하였다. 본 연구에서는 정형기반의 ADL을 이용하여 웹 서비스 브로커를 설계하였으며, 이러닝 웹 서비스에 합성 및 적용하고자 한다. 적용된 ADL은 CSP기반의 Wright를 이용하였으며, 이러닝 시스템은 성적조회와 결과 값에 대한 프로세스에 적용하였다. 제시된 방법은 ADL기반의 정형화 기법을 통하여 이러닝 웹 서비스 처리의 연결구조를 증명하였으며, 이에 따라 예제시스템에서 이러닝 시스템 개발 및 운용의 효율성을 나타내었다.

Key Words : E-Learning, 웹 서비스, 서비스 브로커, 브로커 인터페이스, ADL

ABSTRACT

Recently, in the web based E-learning part, it is researching to apply web service that is able to do dynamical process. Web service broker is deal with connection between web services. But many web services couldn't apply the E-learning part efficiently and suggest formal based service broker. This research designed the web service broker using formal based ADL and propose to apply E-learning web service composition and application. Applied ADL is used the CSP based Wright and it is processed the pre-score checking and learning result value in E-learning system. The proposal method has given proof of connect structure of the E-learning web service using ADL based formal method. According to this method, example system has showed the efficiency of development and operation of E-learning system.

Ⅰ. 서 론

웹 서비스는 클라이언트-서버 형식의 정적 웹 구조를 보다 적극적이고 동적인 환경으로 바꿀 수 있는 환경을 제공함으로써 인터넷을 통하여 정보의 흐름뿐만이 아니라 활동의 흐름까지 가능하도록 만들고 있다[1]. 이러한 장점으로 인하여 웹을 기반으로 한 이러닝 분야에서도 학습 프로세스의 효율적인 운용을 위해 적용을 위한 많은 연구[2,3,4]가 진행되고 있다. 그러나 웹서비스 이용하기 위해서는 먼저 사용자가 원하는 프로세스가 있는지 검색하여야 하며 이를 사용가능 하도록 이용하는 방법을 고려

* 경희대학교 교양학부(hyjeong@khu.ac.kr)
논문번호 : 08035-0527, 접수일자 : 2008년 5월 27일

www.dbpia.co.kr

하여야 한다. 이를 위해 웹 서비스 합성[5]에 관한 연구가 이루어지고 있으며, 그 방법으로는 웹 서비스 브로커가 있다. 웹 서비스 브로커의 주된 기능은 사용자가 요청한 웹 서비스의 위치와 정보를 제공하여 주는 일을 수행한다. 그러나 이는 웹 서비스 정보의 매개체 역할 뿐만 아니라 원격의 서비스를 연결하여 주는 일을 하기에는 어려움이 많으며, 이러닝의 특성을 반영하여 효율적으로 운용하는 하고, 각 이러닝 웹 서비스의 합성 및 연결을 담당하는 서비스 브로커 내의 처리 및 인터페이스에 대한 정형화된 로직 구성이 이루어 지지 않았다.

따라서 본 연구는 이러닝 웹 서비스의 정보를 파악하고 이를 합성하는 웹 서비스 브로커를 설계하였다. 또한 각 웹 서비스 프로세스의 정형적인 연결처리를 위하여 ADL(Architecture Description Language)기반의 합성 인터페이스를 제시하고자 한다. 정형적 프로세스 명세로 사용되는 ADL은 CSP(Communicating sequential processes)기반의 Wright[4]를 이용하였다.

Ⅱ. 관련 연구

2.1 SOA(Service-Oriented Architecture)

웹 서비스는 클라이언트 응용과 서버 응용간의 상호작용 형태를 갖으며, 클라이언트는 사용자의 요구사항을 받아서 요구사항을 처리하기 위해 필요한 서비스들을 다시 찾아 이를 처리한 후 결과를 사용자에게 전달할 수 있다[6]. SOA는 이러한 서비스들을 기반으로 하는 소프트웨어 아키텍처로 애플리케이션의 기능들을 사용자(Consumer)에게 적합한 크기로 공개한 서비스들의 집합을 제공하고 정책 또는 프레임워크를 통해 바인딩 가능하도록 구현한다.

이때 서비스는 단일한 표준 기반의 인터페이스 형태를 통하여 구현하며 독립적으로 추상화되고 호출(Invoke), 공개(Publish), 발견(Discovery)과 같은 오퍼레이션을 수행 한다. 그림 1은 SOA의 구성요소와 상호관계를 나타낸다[7].

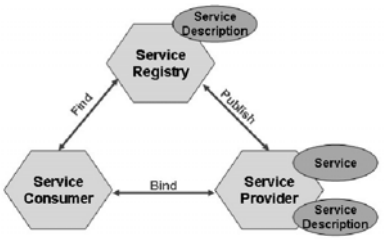

그림 1. SOA의 구성요소와 상호관계

2.2 ADL

ADL은 소프트웨어 시스템 모델링 컨셉 구조를 위한 요소들을 제공하는 언어이며[8], 정형방법으로 소프트웨어 구조를 기술하고 특정 소스 모듈의 세부사항을 구현하는 것 보다 더 높은 단계의 소프트웨어 설계를 표현한다[9]. 그림 2는 ADL의 하나인 Wright[10]의 간단한 예를 나타낸다.

```
Style Client-Server
    Component Client
        Port p = request → reply → p⊓§
        Computation = internalCompute →
            p.request → p.reply → Computation⊓§
    Component Server
        Port p = request → reply → p□§
        Computation = p.request →
        internalCompute → p.reply →
            Computation □ §
    Connector Link
        Role c = request → reply → c⊓§
        Role s = request → reply → s□§
        Glue = c.request → s.request → Glue
            □ s.reply → c.reply → Glue
            □ §
        Constraints
            ∃!s ∈ Component, ∀c ∈ Component
            : TypeServer(s) ∧ TypeClient(c) ⇒
            connected(c,s)
EndStyle
Configuration Simple
    Style Client-Server
    Instances C : Client ; L : Link ; S : Server
    Attachments C.p as L.c ; S.p as L.s
EndConfiguration
```

그림 2. Wright의 클라이언트-서버 예

이때 request는 이벤트의 발생을 나타내며, ⊓은 프로세스의 내부선택, □은 프로세스의 외부선택을 나타내고, 프로세스의 §은 종료를 나타낸다.

2.3 선행연구

웹 서비스를 합성을 위한 연구는 합성 및 운용을 통한 애플리케이션의 효율성 때문에 많은 연구가 진행되어왔다. 웹 서비스 브로커를 통해 서비스를

www.dbpia.co.kr

합성 및 운용하는 연구에서, 고인영의 연구[11]는 온톨로지를 통해 웹 서비스를 합성하고 있으며, 서비스 브로커는 서비스 매니저, 서비스 툴, 하부 서비스 등의 연계를 통해 합성정보를 서비스 요청자에게 지원하였다. 그러나 서비스의 실제 연결 과정이나 프로세스의 흐름에 대해서는 명시하지 못했다. 이강찬의 연구[12]는 브로커 역할을 수행하는 중재자를 XML 기반의 XMF 통합중재방법에 의해 연동하였다. 그러나 XML 기반의 정보만을 연결하는 방식으로 서비스를 연동하는 중재자 방식에는 실제 이를 수행하는 서비스 프로세스나 처리 등의 합성에 대한 역할을 수행하지 못하였다. 이러닝 분야에서 적용된 사례에서, Judith의 연구[4]는 이러닝 웹 서비스의 합성을 위한 서비스 브로커를 설계하였다. 그러나 웹 서비스를 컴포넌트의 측면으로만 보았으며, 웹 서비스의 합성에서도 서비스 제공자와 서비스 사용자의 측면만을 고려함으로서 정적이고 단편적인 방법을 제시하고 있으며, 각 연결과정에서도 정형적인 명세가 이루어지지 않았다. Jie Wu의 연구[13]에서는 이러닝에서 필요한 각 서비스들을 UDDI 저장소를 통하여 연결 및 구성하였다. UDDI와 서버 호스팅 웹 서비스 등에 대한 연결만 명시하였을 뿐 웹 서비스의 체계적인 합성을 위한 서비스 브로커를 사용하지 못하였다. Judith의 연구[14]에서는 이러닝 웹 서비스를 위한 서비스 브로커의 구조를 설계하였다. 그러나 이는 서비스 제공자와 서비스 사용자사이에서 필요한 구조적 항목만을 체계적으로 기술하였을 뿐 이들의 합성 및 운용을 위한 서비스 로직의 연결이나 운용에 대해서는 설계하지 못하였다.

Ⅲ. 이러닝 웹 서비스를 위한 서비스 브로커

3.1 웹 서비스 합성

본 논문에서는 이러닝 웹 서비스 합성을 위한 웹 서비스 브로커를 제시하고자 한다. 각 웹 서비스들은 LMS(Learning Management System)에 연결되어 필요한 로직을 수행하고 있으며, 웹 서비스 제공자는 이러한 웹 서비스를 구성할 때 해당 LMS에서 요구하거나 적용되어야 하는 항목들을 수용할 수 있도록 하여야 한다. 그림 3은 웹 서비스 브로커의 배경도를 나타낸다.

또한 그림 4는 웹 서비스 합성을 위한 웹 서비스 브로커에서의 서비스 합성 순서 다이어그램을 나타낸다.

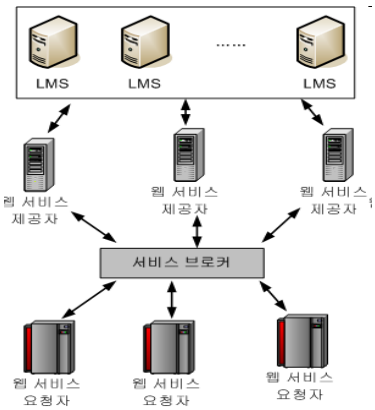

그림 3. 웹 서비스 브로커의 배경도

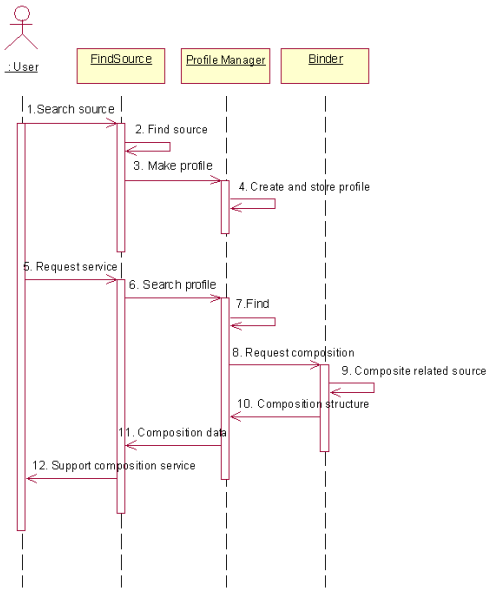

그림 4. 웹 서비스 브로커의 순차 다이어그램

즉, 브로커에 있는 Findsource는 웹 서비스의 위치와 정보를 찾고 ProfileManager를 통해 서비스 프로파일을 작성한다. 이후 사용자는 검색된 웹 서비스 정보를 기반으로 프로세스 구성을 요청하게 되면 ProfileManager를 통한 Binder가 관련된 웹 서비스들을 자동으로 합성한 후 합성구조를 사용자에게 제공함으로서 웹 서비스의 합성이 이루어지게 된다.

3.2 웹 서비스 브로커의 ADL 명세 및 구현

웹 서비스 브로커의 구현을 위한 Wright 명세에서 컴포넌트는 웹 서비스 프로세스로 보며, 웹 서비스 프로세스간의 연결은 커넥터에서 처리하였다. 또

www.dbpia.co.kr

한 웹 서비스의 포트는 웹 서비스의 호출 및 응답에 관련한 프로세스 인터페이스이다.

```
Style ServiceBroker
    WebService Service1
        SourceURL = test001.hanto.kr□§
        Port p = request → reply → p⊓§
        Computation =
            FindSource → p.findsource →
                p.makeprofile → Computation⊓§
            □ ProfileManager → p.makeprofile →
                p.createNstoreprofile →
                Computation⊓§
    WebService Service2
        SourceURL = test002.hanto.kr□§
        Port p = request → reply → p⊓§
        Computation =
            FindSource → p.findsource →
                p.makeprofile → Computation⊓§
            □ ProfileManager → p.makeprofile →
                p.createNstoreprofile →
                Computation⊓§
                    :    :
                    :    :
    WebService Servicen
        SourceURL = test00n.hanto.kr□§
        Port p = request → reply → p⊓§
        Computation =
            FindSource → p.findsource →
                p.makeprofile → Computation⊓§
            □ ProfileManager → p.makeprofile →
                p.createNstoreprofile →
                Computation⊓§
    ServiceBroker Binder
        Role s1 = request → reply → s1⊓§
        Role s2 = request → reply → s2□§
                    :    :
                    :    :
        Role sn = request → reply → sn□§
        Glue = s1.request → s1.getsourceURL →
                    s2.request → Glue
               □ s2.reply → s2.getsourceURL →
                    s2.connectURL → s1.reply
                    → Glue □
                    :    :
                    :    :
               □ sn.reply → sn.getsourceURL →
                    sn.connectURL → sn-1.reply
                    → Glue □ §
        Constraints
            ∃! s1 ∈ WebService1,
               ∀s2 ∈ WebService2 :
                    :    :
                    :    :
               ∀sn-1 ∈ WebServicen-1 :
               ∀sn ∈ WebServicen :
               TypeServer(s1)∧TypeClient(s2)
               ...TypeServer(sn-1)∧TypeClient(sn)
                   ⇒ connected(s1,s2) ; ... ;
                     connected(sn-1,sn)
EndStyle
Configuration Broker
    Style ServiceBroker
    Instances S1 : WebsService1 ; B : Broker ;
              S2 : WebsService2 ;
                    :    :
                    :    :
              Sn-1 : WebServicen-1
              Sn : WebServicen
    Attachments S1 as B ; B as S2 ; ... ;
              Sn-1 as B ; B as Sn
EndConfiguration
```

이는 각 웹서비스들은 $Service_1$, $Service_2$, ... $Service_n$ 으로 명시되며, 이들에 대한 정보는 모두 바인더에서 웹 서비스 합성에 이용된다. 바인더는 각 웹서비스의 프로세스 수행 요구 및 수행결과 응답을 Role에서 명세하고 있으며, Glue에서는 이에 대한 연결을 나타낸다. 이때 합성의 대상이 웹 서비스이기 때문에 먼저 서비스의 위치를 찾는 getsourceURL처리가 필요하다.

Ⅳ. 적용 및 분석

본 연구에서는 영어 학습 시스템에 적용하여 제안된 방법이 운용 가능함을 나타내었다. 그림 5는 적용된 예제 시스템의 배경도를 나타낸다.

웹 서비스는 학습자의 전 학습정보를 LMS에 요청하여 분석하는 “전 학습정보 분석“과 현재의 학습된 결과를 분석하여 학습자에게 제공하는 ”학습결과 분석“으로 구성하였으며, 이는 하나의 LMS를

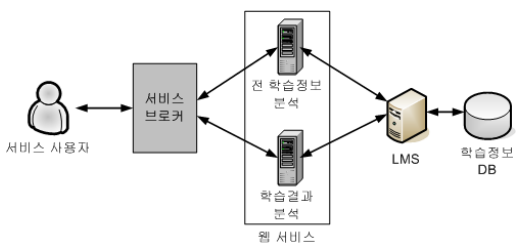

그림 5. 예제 웹 서비스 브로커 배경도

www.dbpia.co.kr

통해 서비스를 제공하였다. 이를 통하여 구현된 웹 서비스는 다음과 같다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<definitions name="AnalysisService"
    targetNamespace="http://jhy.hanto.kr/sr/
        LearningAnalysisService.wsdl"
    xmlns:tns="http://jhy.hanto.kr/sr/
        LearningAnalysisService.wsdl"
    xmlns:xsd="http://www.w3.org/2001
            /XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/
            wsdl/soap/"
    xmlns="http://schemas.xmlsoap.org/wsdl/">
    <message name="getLearnerIDRequest">
    <part name="learner_id"
            type="xsd:string"/>
    </message>
    <message name=
            "getCurrentLearnerNameResponse">
    <part name="return" type="xsd:string"/>
    </message>
    <message name="getBeforeScoreRequest">
    <part name="learning_category"
            type="xsd:string"/>
    </message>
    <message name=
            "getScoreResponse">
    <part name="return" type="xsd:float"/>
    </message>
                :                    :
                :                    :
    <message name="getCurrentScoreRequest">
    <part name="currentlearning_category"
            type="xsd:string"/>
    </message>
    <message name=
            "getCurrentScoreResponse">
    <part name="return" type="xsd:float"/>
    </message>
    <message name="getLearningResultRequest">
    <part name="score_value"
            type="xsd:float"/>
    </message>
    <message name=
            "getLearningResultResponse">
    <part name="return" type="xsd:float"/>
    </message>
    <portType name="Role_s1">
    <operation name="getLearnerName">
    <input message="tns:getLearnerIDRequest"/>
    <output message=
            "tns:getCurrentLearnerNameResponse"/>
    </operation>
    <operation name="getBeforeScore">
    <input message="tns:getBeforeScoreRequest"/>
    <output message=   "tns:getScoreResponse"/>
    </operation>
    </portType>
    <portType name="Role_s2">
    <operation name="getCurrentScore">
    <input message="tns:getCurrentScoreRequest"/>
    <output message=
            "tns:getCurrentScoreResponse"/>
    </operation>
    <operation name="getLearningResult">
    <input message=
            "tns:getLearningResultRequest"/>
    <output message=
            "tns:getLearningResultResponse"/>
    </operation>
    </portType>
    <binding name="ServiceBroker1" type=
            "tns:Role_s1">
    <binding name="ServiceBroker2" type=
            "tns:Role_s2">
    <service name="AnalysisService"/>
</definitions>
```

getLearnerIDRequest는 학습자의 ID를 LMS에 요청하며, getCurrentLearnerNameResponse는 학습자의 ID를 기반으로 학습자의 이름을 서비스로부터 응답받고, getBeforeScoreRequest는 학습단원의 전 학습점수를 요청하며, getScoreResponse은 요청한 학습단원의 학습점수를 응답받는다. 또한 getCurrentScoreRequest는 학습자의 현재 학습결과의 점수를 요청하며, getCurrentScoreResponse는 학습자의 학습점수를 응답받고, getLearningResult Request는 전 학습결과와 현재의 학습결과에 대한 학습결과 비교 값을 요청하며, getLearningResultResponse은 학습결과의 비교 값을 응답받는다. 포트 타입의 Role_s1과 Role_s2는 웹 서비스간의 커넥터 역할을 의미한다.

Ⅴ. 결 론

본 연구에서는 이러닝 웹 서비스 합성을 위한 ADL기반의 서비스 브로커를 제안하였다. 적용된 이러닝 웹 서비스는 전 학습 성적조회와 현재의 학습 결과 및 전 성적과의 비교 값을 처리하는 프로세스를 대상으로 하였으며, 이는 LMS와 연동하여 학습을 진행하였다. 특히 기존의 이러닝 프로세스인

www.dbpia.co.kr

전 학습 성적조회와 학습결과 비교 등을 웹 서비스로 제공함으로서 이러한 프로세스를 필요로 하는 다른 애플리케이션에서도 사용이 가능하며, 프로세스의 운용에 있어서도 상호 유동적으로 활용할 수 있었다. 또한 이러닝 웹 서비스의 연결을 담당하는 서비스 브로커를 정형명세인 CSP기반의 ADL중의 하나인 Wright를 이용함으로서 체계적인 프로세스 구축이 가능하였다. 이때 Wright 명세는 기존의 컴포넌트에 지향적이므로 웹 서비스 처리에 맞도록 변경하였다.

향후 연구과제로는 많은 이러닝 로직들을 웹 서비스로 전환하여 제공하고, 이를 효율적으로 연결 및 운용할 수 있는 세분화된 ADL 기반의 서비스 브로커가 필요하며, 이질적인 웹 서비스 환경도 고려되어야 한다.

참 고 문 헌

〔1〕 허정회 외 4인, "지능형 웹 서비스 표준 기술 동향 및 국내 도입 방안 연구", 한국전산원 연구보고서 NCA Ⅳ-RER-03077, 2003.

〔2〕 정수현, 염창선, "닷넷을 이용한 안정적 서비스를 위한 웹 기반 학습평가시스템 개발", Journal of the Society of Korea Industrial and Systems Engineering, Vol.30, No.4, 2007.

〔3〕 신성윤, 강일고, 이양원, "사용자 프로파일을 이용한 웹 기반 비디오 학습 평가 시스템의 구현", 한국컴퓨터정보학회 논문지 제10권 제6호, 2005.

〔4〕 Judith Rodriguez, et at al, "A Web Services Broker for E-learning", LNCS Volume 2657, 2003.

〔5〕 오지훈, 시대근, 정영식, 한성국, "Genie: 온톨로지 기반 시맨틱 웹 서비스 합성 시스템", 정보과학회논문지:컴퓨팅의 실제, 제10권 제5호, 2004.

〔6〕 이원석 와 3인, "웹서비스를 이용한 서비스 기반의 디바이스 연동 기술", 한국SI학회지 제4권 제2호, 2005.

〔7〕 이성규, 진찬욱, 김태석, "서비스 지향 아키텍처를 기반으로 한 웹서비스 시스템 모델링", 한국시뮬레이션학회 논문지 Vol.16, No.1, 2007.

〔8〕 Michael William Rennie, Vojislav B. Mi˘si c, Towards a Service-Based Architecture Description Language, University of Manitoba technical report TR 04/08, 2004.

〔9〕 Zhenhua Yu and Yuanli Cai, Object-Oriented Petri nets Based Architecture Description Language for Multi-agent Systems, International Journal of Computer Science and Network Security, VOL.6 No.1B, 2006.

〔10〕 Robert Allen, Rémi Douence, David Garlan, Specifying and Analyzing Dynamic Software Architectures, Proceedings of 1998 Conference on Fundamental Approaches to Software Engineering, 1998.

〔11〕 고인영, 전범준, "U-health 서비스 구축을 위한 온톨로지 관리기 및 서비스 브로커의 설계", 한국정보보호학회지 제17권 제1호, 2007.

〔12〕 이강찬, 이규철, "XML기반의 인터넷 정보 중재자의 설계 및 구현", 한국전자거래학회지 제6권 제2호, 2001.

〔13〕 Jie Wu, Shiyong Zhang, "Broadband Multimedia e-learning system using Web Service", 2004 APAN Network Research Workshop, 2004.

〔14〕 Judith Rodríguez, Luis Anido, Manuel J. Fernández, "How can the Web Services Paradigm improve the E-learning?", Proc of the The 3rd IEEE International Conference on Advanced Learning Technologies, IEEE Computer Society, 2003.

정 화 영 (Hwa-Young Jeong) 정회원

1994년 2월 경희대학교 전자계산공학과 공학석사

2004년 8월 경희대학교 전자계산공학과 공학박사

2000년 3월~2005년 2월 예원예술대학교 만화게임영상학부 조교수

2004년 5월~현재 (사)한국인터넷정보학회 논문지 편집위원

2005년 3월~현재 경희대학교 교양학부 조교수

<관심분야> 소프트웨어공학, CBD, 교육공학

www.dbpia.co.kr