

CBD에 의한 온톨로지 기반 시맨틱 웹 서비스 생성

하 안[†]

요 약

본 연구는 자바와 C++ 컴포넌트로부터 동적으로 웹 서비스를 생성하기 위한 연구로써, 온톨로지를 이용하여 시맨틱 웹 서비스를 지원하고자 한다. 시맨틱 웹 서비스는 웹의 내용뿐만 아니라 웹 서비스의 접근을 용이하게 하는 장점이 있다. 이를 위해 컴포넌트를 위한 시맨틱 서비스 발견이 필요하며, 컴포넌트로부터 웹서비스를 생성하기 위해 온톨로지 기반 프레임워크를 활용한다. 특히, 본 연구에서는 컴포넌트와 온톨로지를 UML 모델로 변환시키고, 이를 다시 WSDL 문서로 사상시키므로써, 객체 모델링을 이용한 동적 웹 서비스 생성을 한다. 본 연구의 주요 의의는 컴포넌트로부터 웹서비스를 동적으로 생성하는 것과 온톨로지를 사용하므로써, 시맨틱 웹 환경을 제공하는 것이다. 다시 말해, 본 연구는 온톨로지 기반 시맨틱 웹 서비스와 CBD 방법론을 통합하고자 한다.

키워드 : 자바컴포넌트, C++ 컴포넌트, VML, 온톨로지, WSDL, CBD

Generation of semantic Web service based on Ontology by CBD

Yan Ha[†]

ABSTRACT

This study suggests that it dynamically generates semantic Web services from Java and C++ components in order to supporting semantic Web service by using ontology. And the semantic Web should enable greater access not only to content but also to services on the Web. It needs semantic service discovery for components. So, we add ontology-based framework to Web service generation system from components. Especially, components and ontologies are transformed UML model so that it makes WSDL documents to support a generation of dynamic Web service using object modeling. The main contribution of this study is to generate web service dynamically from components and to support semantic Web environment by using ontology. In other words, this study integrates semantic Web service based on ontology and CBD method.

Key Words : Java Component, C++ Component, VML, Ontology, WSDL, CBD

1. 서 론

시맨틱 웹은 웹 데이터에 대한 의미적인 해석 및 처리의 취약 등의 문제점을 해결하고, 컴퓨터 간의 정보 교환이 가능하고 웹 상의 데이터의 의미를 사람이 아닌 컴퓨터가 이해, 처리할 수 있는 새로운 정보 기술이다. 온톨로지는 도메인 내에 공유되는 데이터들을 개념화하여 형식적이고 명백한 규정으로 만든 것으로, 특정 분야에서 사용되는 어휘들을 모은 것을 일컫는다. 최근 W3C에서는 시맨틱 웹의 표준 웹 온톨로지 언어로 OWL(Ontology Web Language)을 채택하였고, OWL을 이용한 서비스의 시맨틱 명세를 위해 OWL-S가 정의된 바 있다. 따라서, 최근에 이 온톨로지 개념을 이용한 시맨틱 웹 서비스가 많이 활성화되고 있는 추세이다. 이

에 따라, 본 연구는 온톨로지 개념을 적용하여, 기존의 웹 서비스에 시맨틱 기능을 추가하고, 컴포넌트 기반 웹 서비스를 생성하고자 한다. 이는 서비스에 따른 컴포넌트 저장 시스템과 컴포넌트의 기능을 구현한 웹 서비스와 직접적으로 상호작용하는 시맨틱 기반의 사용자 응용을 모두 통합한 형태가 되는 것으로, 본 연구의 가장 큰 의의는 e-비즈니스의 효율적인 환경을 위한 시맨틱 웹 서비스의 지원과 컴포넌트의 활용에 있다.

2. 관련 연구

시맨틱 웹의 활용은 기계 가독형 정의에 기반한 정보 연계를 통해 웹 자원을 지식화함으로써 정보의 효율적 검색, 통합, 재사용을 구현하는 종합적인 기술로 최근 많이 각광받고 있다. OWL은 가장 널리 알려진 웹 온톨로지 언어로써, 시맨틱 웹 형성의 기본이 되는 온톨로지 개발에 필요한 언어 구조를 재정의하였다. 이는 DAML+OIL에 기반을 두고 있다.

* 이 논문은 2005년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임.(KRF-2005-214-D00345)

† 정 회 원 : 경인여자대학 정보미디어학부 부교수
논문접수 : 2007년 3월 15일, 심사완료 : 2007년 5월 18일

이와 관련하여, 현재까지 국내외 연구진들에 의해 연구된 연구의 주요 내용은 다음과 같다. [1]은 OWL에 의료지식을 포함시킨 것으로, 의료 온톨로지를 e-Health 웹 서비스 배치를 가능하게 한다. 이는 다른 엔터티에 의해 만들어진 자원이나 다른 도메인으로부터 생긴 독립적인 자원의 반복을 포함한다. [2]는 시맨틱 웹을 위한 웹 서비스와 에이전트를 위한 개발 방법론과 소프트웨어 구조에 관한 연구를 제시하였으며, [3]은 온톨로지 언어를 통해 시맨틱을 웹에서 기술하는 장점에 대해 기술하고, 어떻게 시맨틱 웹이 레지스트리에서 정의되는 가를 제시하였다. [4, 5]에서는 WSDL(Web Service Definition Language)와 DAML-S 그라운드링 관계를 정의하고, WSDL을 DAML-S Grounding로 변환하는 방법을 제안하였다. 또, [6]에서는 온톨로지를 레지스트리에 발행하는 방법을 제안한 것으로 WSDL 문서를 온톨로지 언어인 DAML+OIL의 tMODEL로서, 레지스트리에 저장한다. 이는 웹 서비스 기술에서 사용하는 오퍼레이션을 온톨로지 개념으로 사상한다. 이는 템플릿에 기반한 기능과 질의 검색을 표현한다. 이 연구는 DAML-OIL 온톨로지 언어에 기반하여 사상 방법을 제안한 것으로 다른 온톨로지 언어에 대해 적용하기가 어려운 단점이 있다. 따라서, 본 연구에서는 여러 온톨로지 언어를 대표할 수 있는 메타 모델을 제안하고, 이를 위해 잘 알려지고 현재 널리 사용되고 있는 UML(Unified Modeling Language) 다이어그램을 사용한다. [7]은 본 연구의 선행 연구로서, UML의 관점에서 UDDI(Universal Description Discovery & Integration Specification)와 서비스 온톨로지를 UML로 어떻게 표현하는가에 초점이 맞추어져 있다. 이에 비해 본 연구는 선행 연구를 기반으로 이를 확장하여, 동적으로 웹 서비스를 생성하는 전체 시스템 구성에 초점을 맞추고 있다.

3. UML, XMI, WSDL

3.1 UML

UML은 프로그램 개발 과정을 일관되게 진행하고 업무 담당자들 간의 의사소통을 용이하게 하여 프로젝트가 효율적으로 진행될 수 있는 공통적인 객체지향적 설계 언어이다. UML의 구조는 온톨로지를 비롯한 웹 서비스를 모델링하기에 가장 적합한 언어이다[8]. UML에는 각종 다이어그램이 포함되어 있어 정적인 구조 및 동적인 구조에 대한 표현이 모두 가능하다. 이 중 클래스 다이어그램은 클래스들 간의 관련성을 나타내는 것으로 객체 모델링에서 가장 공통적으로 쓰이는 다이어그램이다. 이것은 시스템의 정적인 설계 관점을 표현하는 것으로, 클래스는 동일한 애틀리뷰트와 행위를 가진 객체들의 관계와 그들의 의미를 설명하는 객체들의 집합을 기술한다. 클래스들 간의 관련성에는 의존 관계, 결합 관계, 집합 관계 그리고, 일반화 관계가 있다.

3.2 XMI(XML Metadata Interchange)

XMI는 분산 환경에서 모델링 도구와 메타데이터 저장소

와의 교환을 쉽게 하기 위한 목적으로 OMG에서 제안되었다. 또한, OMG의 모델링 표준으로 채택된 UML을 OMG의 메타 모델링과 레퍼지토리 표준인 MOF(Meta Object Facility)로 옮기기 위한 목적으로 XMI가 사용된다. 다시 말해, XMI은 분산 이종 환경에서 모델링 틀들과 메타데이터 레퍼지토리(Repository)들 간에 메타데이터의 교환을 쉽게 할 수 있도록 하는 것이다. 대표적인 상용 도구들은 XMI 형식으로 저장된 UML 모델을 읽어오거나 작성된 UML 모델을 XMI 형식으로 저장할 수 있다[9]. 따라서, 본 연구에서는 UML 다이어그램을 XML 문서 인스턴스인 XMI로 변환해서 사용한다.

3.3 WSDL

UDDI는 인터넷에서 전 세계 비즈니스 목록에 자신을 등재하고, e-비즈니스의 상호 운용을 가능하게 하는 것이며[10], WSDL은 특정 비즈니스가 제공하는 서비스를 설명하고, 개인이나 다른 회사들이 그러한 서비스에 전자적으로 접근할 수 있는 방법을 제공하기 위해 사용되는 XML 언어이다. 즉, UDDI는 기업들이 자신들의 서비스 내용을 웹에 스스로 등록할 수 있게 해주는 XML 기반의 등록처이며, WSDL은 그렇게 하기 위한 언어이다.

웹 서비스에서 발생하는 WSDL과 UDDI의 관계를 설명하면 다음과 같다. 첫째, WSDL 서비스 인터페이스는 서비스에 대한 재사용이 가능하게 하는 정의이며, UDDI 레지스트리에서 tModel형태로 발행된다. 둘째, WSDL 구현은 서비스의 인스턴스를 기술한 것으로 각 서비스 엘리먼트는 UDDI의 businessService를 발행하는 데 사용한다. 셋째, 서비스 구현 문서는 UDDI 레지스트리에 하나 이상의 bindingTemplate을 포함한 businessService로 발행하고 businessService는 서비스 제공자가 발행한다. 넷째, WSDL 서비스 인터페이스 검색은 UDDI의 tModel을 검색한다. 다섯째, WSDL 구현 문서는 UDDI 레지스트리에 businessService로 발행되고 이는 하나 이상의 bindingTemplate을 포함한다. WSDL 문서의 네트워크 서비스에 대한 정의에 사용되는 주요 엘리먼트는 Types, Message, Operation, Port Type, Binding, Service이다.

4. 웹 서비스 생성

4.1 온톨로지

온톨로지는 특정 도메인에 중요한 개념을 계층적으로 기술한 것을 일컫는 것으로 각 개념은 프로퍼티의 기술을 갖는다. 형식적으로 온톨로지는 개념들의 집합을 갖는데, 이는 클래스라고 말한다. 온톨로지는 클래스와 이들 간의 관계로 구성되는데, 관계의 종류는 "subclassof", "superclassof" 등 상위클래스와 하위클래스 간의 상속의 관계 등을 표현할 수 있다. 온톨로지 언어로는 Resource Description Framework(RDF), RDF Schema, OWL, DAML+OWL 등이 있다.

온톨로지 언어 OWL은 매우 잘 알려진 시맨틱 웹 활용 언어로서, 4개의 주요 클래스로 구성되어 있다. 각 클래스에 대한 간략한 설명은 다음과 같다. SERVICE는 웹 서비스를 선언하

는 참조 클래스들의 기본적인 중심점이 된다. SERVICE 클래스의 인스턴스는 각 구별되는 서비스를 위해 존재한다. 프로퍼티 describeBy는 서비스 클래스의 프로퍼티를 지원한다. 각 클래스 SERVICEPROFILE, SERVICEPROFILE, SERVICEMODEL, SERVICEGROUNDING는 SERVICE 클래스 인스턴스에 각 프로퍼티를 갖는다. 클래스 SERVICEMODEL의 인스턴스는 SERVICEGrounding의 자식 클래스를 지원한다.

4.2 UML로부터 OWL로의 사상

온톨로지를 포함한 컴포넌트로부터 웹 서비스를 생성하기 위해 온톨로지 언어 OWL와 웹 서비스 표준인 WSDL간의 바인딩이 필요하다. 이를 위해 2단계의 사상이 필요한데, 첫째는 온톨로지로부터 UML로의 사상이고, 둘째는 WSDL로부터 UML로의 사상이다. 여기서 UML은 OWL로부터 웹 서비스로의 바인딩과정에서 Syntax 및 Semantic을 일치시키기 위한 모델링 도구이다. 각 단계별 사상 내용은 다음과 같다.

온톨로지로부터 UML로의 사상을 시키는 연구를 기반으로 다음과 같은 <표 1>를 제시한다[11].

<표 1> OWL와 UML 간의 사상

OWL	UML
Class	Class, property, ownedAttribute, type
Individual	Instance
Property	ownedAttribute, binary association
Subclass, Subproperty	Subclass, generalization
Class, property	N-ary association
One of	Enumeration
Domain, Range	Navigable.non-navigable
disjointWith, unionOf	disjointWith, cover
minCardinality, maxCardinality, inverseOf	Multiplicity
No equivalent	Derived
Ontology	Package
RDF:properties	Dependency
...	...

4.3 WSDL로부터 UML로의 사상

본 연구에서는 웹 서비스를 동적으로 생성하기 위해 UML 요소에 대한 WSDL로의 변환이 필요하다. 따라서, 이를 위해 WSDL과 UML 요소 간의 매핑이 필요하며, 이를 위해 <표 2>와 같이 사상한다.

<표 2> WSDL로부터 UML로의 사상

WSDL	UML
Description, Interface, Binding, Service, TypeDefinition, elementDeclaration	Class, Property, N-ary association
Interface, interfaceOperation, interfaceFault, interfaceMessageReference, interfaceFaultReference	Class, Property, N-ary association
Binding, BindingOperation, BindingFault	Class, Property, N-ary association
Service, Endpoint	Class, Property, N-ary association
...	...

4.4 UML을 이용한 WSDL 생성

본 연구는 온톨로지를 적용하며, 이에 대한 내용을 UML 모델로 변환시키며, 이를 다시 WSDL의 요소로 사상, 웹 서비스를 생성한다. 다시말해, OWL의 구성요소가 UML로 사상된 이후, 웹 서비스를 지원하기 위해 UDDI 내의 WSDL 구성요소 Operation 등으로 바인딩이 된다. 이에 대한 생성 규칙들은 다음과 같다.

[규칙 1] 각 서비스를 의미하는 상위클래스는 Data와 Operations을 하위클래스로 갖으며, 이들 간의 일반화 관계를 갖는다.

[규칙 2] 클래스 Data는 WSDL의 input과 output이되는 클래스들을 하위클래스로 갖으며, 이들 간의 일반화 관계를 갖는다.

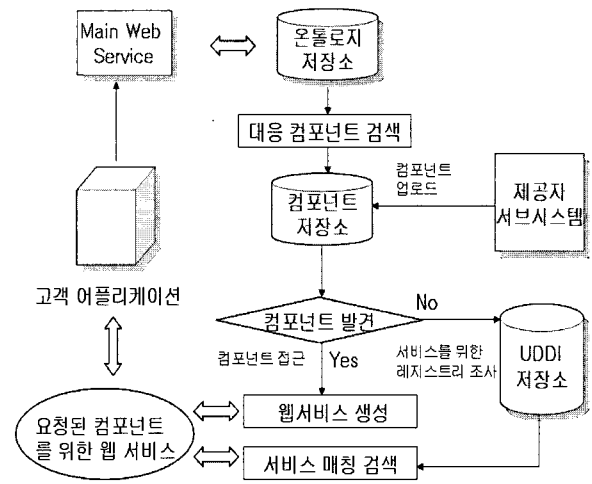
[규칙 3] 클래스 Operations는 WSDL의 operations에 해당하는 클래스들을 하위클래스로 갖으며, 이들 간의 일반화 관계를 갖는다.

5. 시스템

시맨틱 웹 서비스와 컴포넌트 기반 개발 방법과의 통합을 위해 제안되는 본 시스템에 대한 전체적인 시스템 및 해당 알고리즘은 다음과 같다.

5.1 전체 시스템 구성도

본 연구에서 제안하는 컴포넌트와 시맨틱 웹을 통합하여, 동적 웹 서비스를 생성하는 전체 시스템에 관한 설계는 (그림 1)과 같다. (그림 1)에서 보면, CBD와 웹 서비스를 통합하는 데 있어서 주요 구성요소는 제공자 서브시스템, 고객 어플리케이션, 컴포넌트 저장소, 웹 서비스, UDDI 저장소 5 부분이 있다.



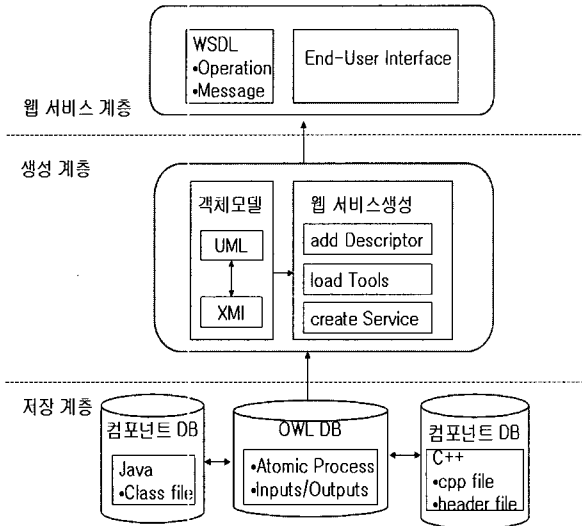
(그림 1) CBD와 웹 서비스 통합 설계

본 연구는 온톨로지 저장소를 추가하고 이에 대한 검색 및 추론 과정을 수행한다. 이 과정을 통해 추출된 객체는 웹 서비스 동적 생성 알고리즘에 의해 사용자에게 전달되며, 이

때, 자바 및 C++ 컴포넌트에 해당하는 틀이 각각 적용되어 웹 서비스를 생성한다.

5.2 알고리즘

특정 도메인에 관한 온톨로지를 포함한 컴포넌트로부터 시맨틱 웹 서비스 컴포넌트로 변환, 생성을 위한 프로세스는 (그림 2)와 같다.



(그림 2) 웹 서비스 생성 과정

위의 (그림 2)를 바탕으로 자바와 C++ 컴포넌트에 해당하는 웹 서비스를 생성하기 위한 세부적인 알고리즘은 다음과 같다.

```

begin(
  client_request();//클라이언트의 요청을 받음
  map_ontology_UML();//온톨로지로부터 UML로 사상
  convert_UML_XMI();//UML로부터 XMI로 변환
  while (subclass in operations' Class)
  { //오퍼레이션 클래스 안에 서브클래스가 있는 동안
    make_operation_name();//오퍼레이션 이름 생성
    get_dataclass_input();//데이터클래스를 입력받음
    get_dataclass_output();//데이터클래스를 출력받음
  }
  if ( java component)
  { // 자바 컴포넌트의 경우
    use_java2WSDLAPI();
    convert_component_WSDL();
    add_descriptor();
  } else // C++ 컴포넌트의 경우
  { edit_headerfile();
    compile_file();
    generate_WSDL();
  }
  deploy_component();//컴포넌트서비스를 웹 서버에 배치한다.
  client_access_component();//클라이언트가 컴포넌트를 접근한다.
}end;
    
```

5.3 결과

① Ontology

해당 도메인에 대한 온톨로지를 클래스와 그들 간의 관계성으로 표시하면 <표 3>과 같다. 대표적인 클래스로는 RegistrationService, Data, Operations을 꼽을 수 있다.

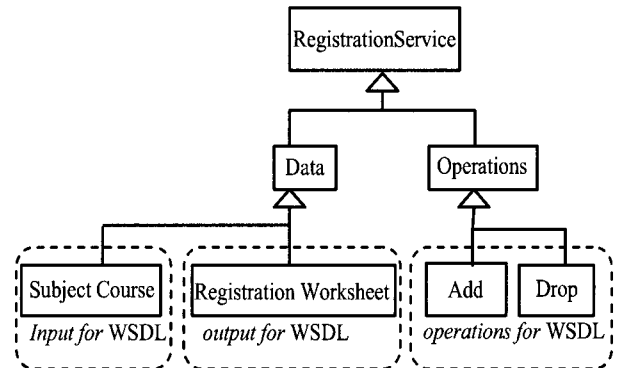
<표 3> Ontology 예제

```

<owl:Class rdf:ID="RegistrationService">
  <rdfs:comment>This is an example of
    ontology</rdfs:comment>
  <owl:Class rdf:ID="Data">
    <rdf:subclassOf
      rdf:resource="#RegistrationService"/>
  </owl:Class>
  <owl:Class rdf:ID="Operations">
    <rdfs:subclassOf
      rdf:resource="#RegistrationService"/>
  </owl:Class>
  :
</owl:Class>
    
```

② 클래스 다이어그램

<표 3>을 살펴보면, UML 클래스 다이어그램의 주요 클래스로는 Data와 Operations이 있으며, 이는 상위 클래스인 RegistrationService 클래스로부터 일반화 관계를 갖으며, 상속이 일어난다. 이에 대한 내용을 클래스 다이어그램으로 나타내면 (그림 3)과 같다.



(그림 3) <표 3>에 대한 클래스 다이어그램

③ WSDL 문서의 예

다음은 (그림 3)으로부터 생성된 WSDL 문서의 예이다. Data 클래스의 하위 클래스들은 각각 WSDL의 input과 output이 되고, 클래스 Operations에 상속되는 하위 클래스들은 역시 WSDL의 operation이 된다. 이에 대한 내용은 <표 4>와 같다.

〈표 4〉 (그림 3)에 해당하는 WSDL 문서

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="RegistrationService"
targetNamespace="http://www.seiti.com/~white/Registration.wsdl20"
xmlns=http://www.w3.org/2004/03/wsdl
xmlns:seiti=
http://www.seiti.com/~white/registration.owl#
xmlns:mep=http://www.w3.org/TR/wsdl20-patterns>
<interface name="Registration">
<operation name="Add" pattern="mep:in-out">
<input messageLabel="seiti:OperationRequest" name="AddClassRequest"/>
<output messageLabel="seiti:OperationResponse" name="AddClassRequest"/>
<operation name="Drop" pattern="mep:in-out">
<input messageLabel="seiti:OperationRequest" name="DropClassRequest"/>
<output messageLabel="seiti:OperationResponse" name="DropClassRequest"/>
:
</interface>
</definitions>
```

④ UDDI 문서

UDDI는 웹 서비스를 통해 검색을 하는 고객을 위한 메카니즘을 제공한다. 본 시스템에 적용하여 생성되는 UDDI는 <표 5>와 같다.

〈표 5〉 UDDI 문서의 일부

```
...
<business businessKey=
"uddi:seiti.Registration.example" serviceKey="...">
...
<categoryBag>
<KeyedReferenceGroup tModelKey=
"uddi:ubr.uddi.org:categorizationGroup:MAPPINGGROUP">
<KeyReference tModelKey=
"uddi:ubr.uddi.org:categorization:
OPERATION_CONCEPTS" KeyName="Add"
KeyValue="AddClass"/>
<KeyedReference tModelKey=
"uddi:ubr.uddi.org:categorization:INPUT_CONCEPTS" KeyName="Input"
KeyValue="SubjectCourse"/>
<KeyedReference tModelKey=
"ubr.uddi.org:categorization:OUTPUT_CONCEPTS" KeyName="Output"
KeyValue="RegistrationWorksheet"/>
...
</KeyReferenceGroup>
...
</categoryBag>
</businessService>
...
```

5.4 평가

본 연구에서는 기존에 연구된 온톨로지 관련 연구들 중 온톨로지 모델링 방법에 대해 비교 분석해 보았다. [10]은 온톨로지를 컴포넌트 방법으로 모델링한 바 있다. 따라서, 온톨로지의 클래스와 프로퍼티에 대해 컴포넌트로 변환시키고, 인스턴스에 대해 Is-a 관계를 갖도록 하며, 클래스와 서브클래스 간에는 Part-of 관계를 갖도록 하였다. 다시 말해,

이는 컴포넌트 모델링 방법을 사용하므로 컴포넌트 모델링 생성 전의 다양한 클래스 및 관계성의 표현이 어려운 실정이다. 또한, [12]의 Schemagraph는 노드와 선(edge)으로 온톨로지를 분석하였는데 선에 대해서는 hasInstance, hasProperty, hasSubClass와 같은 이름을 사용할 수 있다. 이와 같이 특정 모델링 방법을 제안하고 활용하기 때문에 이를 사용하기 위해서는 새로운 모델링 표기법을 익혀야 하는 불편함이 있다. 이에 비해 본 연구는 UML 클래스 다이어그램의 활용으로 정적인 구조의 클래스 및 관계성 뿐 만 아니라 이를 활용하여, 컴포넌트 다이어그램 생성 및 코드 활용이 무척 용이한 장점이 있다. 또한, 현재 많이 개발되어 있는 UML 상용 도구를 그대로 활용할 수 있는 장점이 있다.

6. 결론

본 연구는 시맨틱 웹 서비스와 컴포넌트 방법론을 통합하여, 웹 환경에서 효율적인 컴포넌트를 재사용시키는 것을 목표로 한다. 현재까지 개발된 기존 연구가 컴포넌트로부터 웹 서비스를 동적으로 생성하는 연구에 초점을 맞추는데 비해, 본 연구는 웹 서비스에 온톨로지 개념을 적용하여 시맨틱 웹 서비스를 지원하고, 이와 동시에 컴포넌트 동적 생성 및 지원을 가능하게 한다. 본 연구는 현재까지 개발된 웹 서비스 시스템의 설계를 바탕으로 시스템을 구현하고, 이에 대한 성능평가 기준을 마련하여 이와 유사한 연구들과 비교하여 정량적, 정성적인 평가가 이루어져야 할 것이다.

참고 문헌

- [1] Carl Mattocks Esq, "Managing Medical Ontologies using OWL and e-business Registry/Repository."
- [2] Holger Knublauch, "Ontology-Driven Software Development in the Context of the Semantic Web: An Example Scenario with Protégé/OWL."
- [3] Dogac, A., Laleci, G., Kabak, Y., Cingil, I., "Exploiting Web Services Semantics: Taxonomies vs. Ontologies," IEEE Data Engineering Bulletin, Vol 25, No.4, 2002.
- [4] Paolucci, M., Srinivasan, N., et al. "Towards a Semantic Choreography of Web Services from WSDL to DAML-S," Proc. of ICWS, pp.22-26, 2003.
- [5] Martin, D., et al. "Describing Web Services using OWL-S and WSDL," DAML-S Coalition working document, http://www.daml.org/services/owl-s/1.0/owl-swsdl.html(2003)
- [6] Sivashanmugam K, et al., "Adding Semantics to Web Services standards," In proc. Of ICWS, 2003.
- [7] Yan Ha, Roger Lee, "Integration of Semantic Web Service and component-Based Development for e-business environment," SERA 2006, August 2006.
- [8] Jyotishman Pathak, Doina Caragea, Vasant G. Honavar, "Ontology-Extended Component-based Workflows: A framework for Constructing Complex Workflows from Semantically

Heterogeneous Software Components,” LNCS 3372, pp.42~56, 2005.

- [9] 하 안, “OSD, CDF 문서로부터 UML 클래스 다이어그램으로 변환시스템”, 정보처리학회 논문지 제10권제5호, pp.493~p.502, 2003년 10월.
- [10] Using UDDI to find ebXML Reg/regs, 2001.
- [11] World Wide Web Consortium. OWL Web Ontology Language Reference. W3C Recommendation 10 Feb, 2004.
- [12] Abhijit Patil, Swapna Oundhakar, Amit Sheth, Kunal Verma, “METER-S Web service Annotation Framework,” WWW2004, New York, May 2004.



하 안

e-mail : white@kic.ac.kr

1992년 덕성여자대학교 전산학과(학사)

1994년 이화여자대학교 전자계산교육전공 (석사)

2000년 전북대학교 대학원 전산통계학과 (이학박사)

2000년~2001년 중앙대학교 정보통신연구소 연구전담교수

2006년~2007년 Central Michigan University 방문교수

2001년~현재 경인여자대학 정보미디어학부 부교수

관심분야: XML 멀티미디어 응용, CBD, 시맨틱 웹 서비스 등