

인터넷에서의 지능형 서비스의 전망

정의현*

◆ 목 차 ◆

- | | |
|--------------------|------------------|
| 1. 서론 | 3. 지능형 웹 서비스의 현황 |
| 2. 시맨틱 웹과 웹 서비스 기술 | 4. 결론 및 향후 전망 |

1. 서론

인터넷의 기술적인 발전과 양적인 팽창은 정보 환경의 급속한 변화를 가져왔으며, 사회 구조를 근본적으로 변화시키는 촉매가 되어왔다. 인터넷의 발전으로 과거 수십년에 걸친 변화가 불과 몇 달만에 일어나고 있으며, 이 속도는 점차 가속화되고 있다. 이러한 변화의 속도를 감안할 때 인터넷의 미래 모습을 예견하는 것은 쉽지 않은 일이지만, 많은 연구자들이 네트워크 자체의 고속화가 급속히 진행되고, 컴퓨터 네트워크를 넘어서 무선망과 유비쿼터스(Ubiquitous) 환경으로 확장될 것이라는 것에는 어느 정도 의견이 일치하고 있다. 그러나, 이러한 하드웨어적인 발전 예측에 비해서 정보 시스템(Information System) 관점에서의 인터넷의 발전 전망은 아직 불투명한 것이 현실이다.

유사이래로 가장 많은 정보가 저장되어 있는 매체인 인터넷은 문서, 이미지(image), 오디오(audio), 비디오(video) 등의 여러 정보들이 디지털화되어 저장되어 있는 가상적인 공간이며, 축적된 정보의 양은 기능하기 어려울 정도이다. 이러한 인터넷상의 정보들은 정보 시스템을 통해 만들어진 가상공간(Cyber Space)상에서 접근되며, 사용자들의 상호작용(interaction)에 의해 생산, 가공, 소비되게 된다. 현재, 인터넷의 가장 대표적인 정보 시스템은 웹(Web)이며, 웹이 소개된 후로 대부분의 정보 시스템이 웹으로 수렴하거나 최소한 웹

인터페이스를 제공하고 있어, 사용자들은 웹이라는 단일화된 인터페이스를 통해서 인터넷의 정보를 사용하고 있다.

가장 주목할 만한 문제점은 웹 상의 정보들은 사람에게 소비되기 위해서 만들어진 정보라는 것에 기인한다. 기본적으로 웹은 하이퍼텍스트(Hypertext) 구조를 이용하여 정보를 구조화하고 저장하며, 개별 사용자에 의한 문서의 생성과 그 문서간의 자유로운 연결에 근거하고 있다. 따라서, 웹 상의 정보에 접근하기 위해서는 검색 엔진의 사용이 필수적이다. 그러나, 기존의 검색 엔진은 모든 웹 문서의 인덱스를 저장할 수 없으며, 더 큰 문제는 사용자가 의도한 의미(semantic)에 적합한 문서를 정확히 찾을 수 없다는 점이다. 이러한 문제점은 웹이 저장하는 정보가 디지털 형태의 정보여서 컴퓨터가 자동으로 처리하고 이해할 수 있는 것처럼 보이지만, 실제로 정보의 의미를 이해하는 것은 사람(human)만이 할 수 있기 때문이다. 같은 단어도 사용하는 위치나 용도에 따라서 다른 의미(semantic)를 갖기 때문에, 정보의 양이 늘어나면 늘어날수록, 다양한 사용자가 참여하면 참여할수록 의도한 정보에 접근이 더욱 어려워지는 역효과가 생기는 것이다[1]. 이러한 문제는 단순히 정보의 접근에 관련된 문제를 떠나서, 웹에서 제공되는 서비스를 이용하고자 할 때, 자동화된 처리가 어렵다는 문제점을 야기시킨다. 이런 문제점을 해결하기 위해 제안된 것이 바로 시맨틱 웹(Semantic Web)[2][3]이다. 시맨틱 웹은 웹 리소스에 의미기반 정보를 태깅(tagging)하여 기계가

* 가톨릭대학교 컴퓨터정보공학부 초빙교수

리소스의 의미를 이해하여 사용자의 의도에 가장 적합한 것을 찾는 기능을 담당하고, 이와 관련하여 자동화된 정보처리 및 추론을 가능하게 하는 기술이다.

두 번째의 문제점은 정보 시스템간의 연동 문제이다. 일반적으로 인터넷의 정보 시스템들은 SNMP(Simple Network Management Protocol)나 SMTP(Simple Mail Transfer Protocol), CGI(Common Gateway Interface), CORBA(Common Object Request Broker Architecture) 등의 여러 프로토콜을 이용해서 결합되어 있다. 그러나, 기존의 프로토콜 결합(binding)은 종단 시스템간에 사전에 정의된 바이너리(binary) 결합을 하는 모습을 갖고 있으며, 다른 프로토콜 간에는 상호운용(interoperability)이 원칙적으로 불가능한 상황을 만들고 있다. 이것은 큰 관점에서 정보 시스템들이 서로 종류가 다른 파이프로 얼기설기 연결한 것으로 볼 수 있으며, 여러 기관에서 개별적으로 제공되는 서비스들을 통합적으로 연결해 사용하거나, 이미 구성된 통합 서비스에 새로운 서비스를 결합하는 것은 매우 어려운 문제가 되고 있다. 이를 해결하기 위한 방안으로 현재 각광받고 있는 것이 바로 웹 서비스(Web Services)[4][5]이다. 웹 서비스는 기존의 웹에서 제공하는 서비스를 하나의 실행 가능한 실행 모듈로 보는 관점으로, 전송 프로토콜 위에 문자 형태의 XML 형태로 서비스를 요청하고, 그 응답을 다시 문자 형태의 XML로 받게 된다. 이러한 구조에서는 어떤 종류의 서비스라도 통합이 가능한 특징이 있기 때문에 웹서비스는 다수의 연구소와 기업에서 차세대 분산 시스템의 정보 버스(Information Bus)로 생각하여, 많은 연구 투자를 하고 있다.

그러나, 상기한 두 가지 기술은 그 장점에도 불구하고 각각의 한계를 가지고 있다. 먼저, 웹 서비스는 콘텐츠 중심의 인터넷을 서비스 중심의 인터넷으로 바꿀 수 있는 장점을 갖고 있다. 그러나, 서비스의 사용을 위해서는 해당 인터페이스의 의미적 이해를 반드시 사람이 해주어야 한다는 단점을 갖고 있어, 아직까지 개발자에 의한 하드 코딩(hard-coding)에 의존하고 있다. 또한, 사전에 결합된 서비스의 운용 도중에 서비스의 인터페이스가 변경되거나 한다면, 전체 시스템을 새롭게 바꾸어주어야 한다. 그리고, 웹서비스를 이용하여 복합적인 서비스는 가능할 지 모르지만, 기계가 자동으로 사용자를 대행하여 필요한 기능을 찾

아서 업무를 처리하는 지능형 서비스로의 접근은 아직까지 어려운 실정이다. 이에 비해, 시맨틱 웹은 시맨틱 웹 공간에서 정보를 찾거나 추론할 수 있지만, 사용자를 위한 실제적인 서비스를 제공할 수 없다는 단점을 갖고 있다.

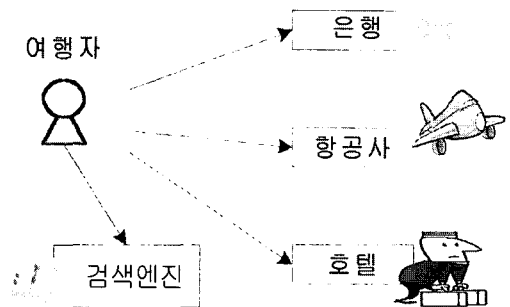
이러한 문제를 해결하기 위해 시맨틱 웹 기술을 웹 서비스와 결합하고자 하는 연구가 행해지고 있고, 이러한 연구 흐름을 통칭하여 지능형 웹서비스(Intelligent Web Service)[6][7]라 한다. 본 기고에서는 시맨틱 웹을 중심으로 하는 지능형 웹서비스, 특히 DAML-S를 중심으로 인터넷에서의 지능형 서비스의 연구 동향과 향후 인터넷에 미칠 영향에 대해서 살펴보기로 한다. 2장에서는 지능형 웹서비스를 이해할 수 있는 수준에서 시맨틱 웹과 웹서비스 기술의 소개 및 동향에 대해서 간략히 설명하고, 3장에서 지능형 웹서비스의 기술 구조 및 전망에 대해 설명하고, 4장에서 결론을 맺도록 하겠다.

2. 시맨틱 웹과 웹 서비스 기술

2.1. 웹 서비스(Web Services)

2.1.1. 웹 서비스의 개념

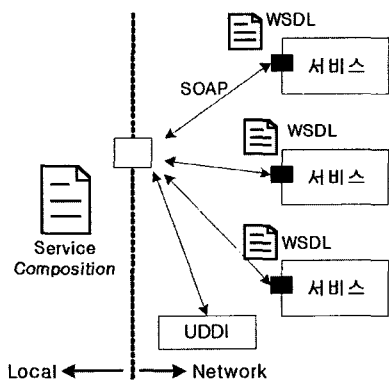
기존의 웹은 하이퍼링크(Hyperlink)로 연결된 HTML 문서들의 집합체라고 볼 수 있다. 이러한 콘텐츠(Contents) 중심의 인터넷에서는 서비스의 사용이 사람에게 의해서 이루어졌다. 예를 들어, 그림 1과 같은 여행사 시나리오를 살펴보자. 일반적으로 여행을 가기 위해 웹사이트를 사용하는 경우에는 사용자가 일일이



(그림 1) 기존의 웹 구조

해당 웹사이트를 찾고, 은행에서 환전 정보를 확인하고, 항공사 사이트에서 비행기의 날짜와 시간을 확인한 후에, 호텔을 예약한다. 이를 위해서, 사용자는 관련 웹사이트를 검색 엔진을 통해서 찾아야 하고, 웹사이트의 설명을 읽은 후, 입력 사항을 모두 입력한 후에야 비로서 원하는 업무를 처리할 수 있는 것이다. 즉, 사람이 해당 서비스를 보고, 판단하고 직접 해당 서비스와 상호작용을 하는 것인데, 지금의 웹사이트의 서비스는 사람(human)이 사용하도록 맞추어져 있기 때문이다. 이것이 지금까지 당연한 사용법처럼 여겨져 왔지만, 수백만개의 사이트가 있는 상황에서 사용자가 서비스를 일일이 확인하고, 다른 방식으로 되어 있는 서비스를 사용한다는 것은 많은 시간과 노력을 필요로 한다.

웹서비스 기술은 웹을 단순한 HTML 문서의 집합으로 보는 것이 아니라, 실행이 가능한 여러 서비스들이 결합되어 있는 형태로 보는 것이다. 그림 2에서처럼, 웹서비스에서 제공하는 서비스는 SOAP(Simple Object Access Protocol)이라는 프로토콜을 통해서 서비스의 원격호출과 그 응답을 받는 것이 가능하다. 개별 서비스의 인터페이스는 모두 WSDL(Web Services Description Language) 문서형태로 기술(description)되며, 해당 서비스의 규격화가 이루어지기 때문에, 기계에 의해서 호출되는 것이 가능하다. 또한, 필요로 하는 서비스는 UDDI(Universal Description, Discovery, and Integration)를 통해서 검색할 수 있다. 그림 2에서 볼 수 있는 것처럼, 웹에서 제공되는 기능이 웹서비스 형태를 가지면, 여러 종류의 서비스를 결합하여 새로운



(그림 2) 웹 서비스의 구성

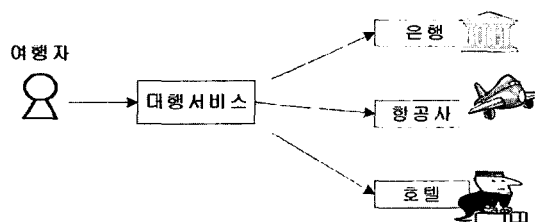
서비스를 만드는 것이 가능해지며, 여러 기관에서 개별적으로 나온 서비스를 동일한 방식으로 사용할 수 있어, 서비스 중심의 인터넷 사용이 가능해진다. 이것은 웹서비스가 단일한 인터페이스를 갖고 있는 실행 모듈이기 때문에, 인터넷에 연결된 어떤 컴퓨터나 장치, 그리고 기술적 기반에 상관없이 접근이 가능하기 때문이다. 웹사이트가 처음에는 사용자에게 단순히 정보만을 제공했지만, 웹서비스 관점에서는 상호간에 협업이 가능한 실행 모듈로의 역할을 하게 되는 것이다.

2.1.2. 웹 서비스의 이용

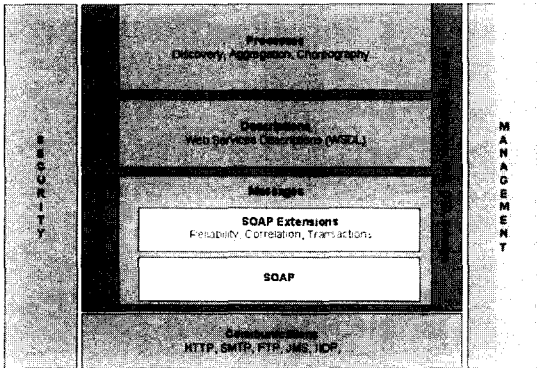
앞의 여행자 시나리오를 웹서비스를 이용한 형태로 다시 한번 생각해보자. 그림 1에서는 사용자가 개별 웹사이트를 수작업으로 접속하여 자신의 업무를 처리하였다. 그러나, 웹서비스를 이용하면, 해당 시나리오는 그림 3과 같은 형태가 된다. 즉, 웹서비스를 결합한 중간의 대행 서비스가 있고, 이 대행 서비스는 몇 가지 조건을 여행자로부터 받아들여서, 이것을 다른 웹서비스를 주어진 조건에 맞추어 결합을 하여 서비스를 제공하게 된다. 이러한 시나리오는 사용자의 시간을 절약할 수 있으며, 부가적인 서비스의 제공이 가능해진다는 장점을 갖게 된다. 이것은 단순하게 사용자들의 시간을 절약할 수 있을 뿐 아니라, B2B에서의 서비스 조합이나 e-business 트랜잭션에 사용이 가능하다는 점이 더욱 기대되며, 웹 페이지 자체가 하나의 컴퓨팅 파워나 서비스 조합으로 사용이 가능하다는 것을 제시하고 있다.

2.1.3. 웹 서비스의 구성요소

웹서비스를 한마디로 정의하면 인터넷을 통해 기술하고 배포하며, 실행이 가능한 모듈화된 애플리케이션을 의미한다. 다시 말하면, 웹서비스는 컴파일된 통한



(그림 3) 웹서비스를 이용한 인터넷 사용



(그림 4) 웹 서비스 계층 구조

정적 바인딩을 하던 기존의 분산객체 구조에서, 실행할 때 동적으로 서비스를 검색하고 이를 XML 메시지를 통하여 호출하는 구조를 갖고 있다. 이러한 구조에서는 코드 상에서 결합되어 수정이 복잡한 기존의 e-business를 동적으로 결합/분리/수정 할 수 있는 구조를 갖게 된다. 웹서비스는 실제로 활용이 가능한 기술이며, IBM, 마이크로소프트 등에서 개발되고 있으며, W3C에서 활발한 표준화 활동이 일어나고 있다[8]. W3C에서 제시한 웹서비스의 계층 구조는 그림 4[8]와 같다.

기본적으로 웹서비스는 하위에 여러 전송(Transport)계층을 사용할 수 있다. 지금까지는 HTTP가 주요한 전송계층을 이루고 있지만, TCP, UDP 같은 프로토콜로의 적용도 가능하다. 이렇게 다양한 전송계층이 사용 가능한 이유는 메시지 교환에 사용되는 SOAP 프로토콜이 XML로 만들어진 문자 프로토콜이기 때문이다. SOAP은 실제적인 웹서비스의 메시지 교환 프로토콜에 해당된다. SOAP을 이용해서 서비스 제공자와 서비스 사용자가 통신을 하게 된다. WSDL은 SOAP 통신을 위해서 웹서비스에서 제공되는 오퍼레이션(Operation)과 여러 인자들을 사전에 명시해놓은 서비스 프로파일(profile)이다. 이 서비스 프로파일을 이용해서 SOAP은 서비스 결합(binding)을 할 수 있다. 그리고, UDDI는 배포된 WSDL의 레지스트리(Registry) 서비스 역할을 하게 된다.

2.2. 시맨틱 웹(Semantic Web)

2.2.1 시맨틱 웹의 배경

지금까지 많은 웹 페이지가 만들어져 있지만, 그

웹 페이지들의 정보는 모두 사람에게 소비되는 것을 목적으로 하고 있다. 사람들에 의해 쉽게 만들어지고, 쉽게 소비되는 구조는 지금까지 웹이 발전하는데 큰 역할을 하였다. 그러나, 사람에게 보여지기 위한 웹 페이지가 많아지면서 문제가 발생하고 있는데, 무엇보다 사용자가 의도한 정보에 정확하게 접근하는 것이 힘들어지고 있다는 점이다. 예를 들면, 날짜 정보는 사람들이나 관습마다 모두 다르게 표현하고 있다. 예를 들어, 2003년 12월 10일이라는 이라는 날짜가 주어질 때, 어떤 문서에서는 “YYYY-MM-DD”라고 표현하지만, 어떤 문서에서는 “YYYY-DD-MM”으로 기술될 수 있기 때문에, 기계가 문서에서 정확하게 의미를 파악하는 것은 어렵다. 이렇게 같은 의미를 다른 형식으로 표현하는 문제 때문에, 렉시컬(lexical) 검색에 의존하여 정보에 접근하는 현재의 컴퓨팅 방식에서는 원하는 정보를 쉽게 찾을 수 없게 된다. 이와 반대로 여러 표현이 1개의 의미를 나타내는 경우도 많은데, “나비”와 “고양이”는 다른 동물이지만, 일부 사람들은 “고양이”를 “나비”라고 부르는 경우도 있다. 즉, 우리가 정보를 표현하고, 전달하는 것은 문법적인 양식과 단어에 그 의미를 담아서 전달하지만, 그 단어나 양식이 사용되는 컨텍스트(context)에 따라서 전달하고자 하는 의미(semantic)가 달라진다는 것이다. 그러나, 현재의 HTML 기반의 웹은 이러한 의미 정보를 수용할 수 없는 단점을 갖고 있다.

이를 해결하기 위해 Tim Berners Lee에 의해 W3C에서 주창한 기술분야가 바로 시맨틱 웹이다. 시맨틱 웹은 현재의 웹 기술을 확장하는 새로운 기술로서 기존의 웹 기술과 달리 웹에 올려져 있는 리소스를 의미론적(semantic)으로 기술(description)하여 리소스에 접근할 때, 단순한 키워드의 매칭에 근거하지 않고 의미론적인 접근이 가능하도록 하는 기술이다. 이 기술은 인간의 웹 접근에도 많은 편리함을 제공하지만, 더 가능성이 큰 것은 기계가 의미론적인 정보 처리(information processing)가 가능하다는 점이다[2]. 이렇게 되면, 웹상의 정보를 결합하거나 의미를 추론하는 것이 가능하게 된다. 이 기술은 현재 W3C에서 전폭적인 지원을 받고 있으며, RDF[9], DAML+OIL[10], OWL[11] 등의 기술(description) 방식에 대한 연구와 이 기술을 적용한 온토로지(Ontology)와 응용 프로그램 등이 활발

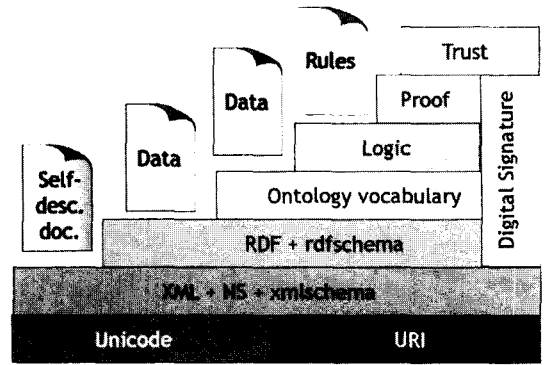
히 개발되고 있는 상태이다.

2.2.2 시맨틱 웹의 구조

시맨틱 웹을 이해하기 위해서 RDF(Resource Description Framework)를 이용해서 시맨틱 태깅에 대한 개념을 간단히 살펴보자. 시맨틱 태깅이란 웹 상의 리소스에 의미 정보를 기술하는 방식이다. 즉, 특정 리소스에 대한 여러 속성이 주어질 때, 그 속성의 값을 명시하여 해당 리소스에 대해 기계가 이해할 수 있는 시맨틱 정보를 추가하는 것이다. RDF 그래프 형태로 특정 리소스에 대한 정보를 나타내면, 일반적으로 그림 5-(a)와 같은 형태로 표현이 가능하다.

예를 들어 설명하기 위해, 그림 5-(b)처럼 본 기고에 시맨틱 태깅을 해보자. 먼저, 본 기고의 URI를 "http://cse.catholic.ac.kr/Article/sw_status.doc"이라 했을 때, 이 리소스를 기술하기 위해서, "저자"와 "게재학회"라는 가상의 프로퍼티를 정의하고, 그 프로퍼티가 각각 "정의현", "인터넷 정보학회"라는 값을 갖도록 하겠다. 이러한 경우에, 기계에게 "저자"가 "정의현"이라는 정보를 찾으라는 의미기반 명령을 하면, 기계는 저자라는 단순한 텍스트의 패턴을 찾는 것이 아니라, "저자"라는 프로퍼티에 대해서 웹을 검색하게 된다. 이때, 해당 문서에 사람이 읽는 정보는 저술자나 지은이 등으로 되어 있다 하더라도, 문서에 "저자"라는 프로퍼티의 값이 "정의현"이라는 것을 뒤져서, 정확히 해당 정보에 접근할 수 있게 된다. 이러한 방식으로 웹의 리소스가 기술되어 있다면, 기계가 의미론적으로 정보에 접근할 수 있다는 장점이 생긴다.

더욱이 웹의 시맨틱 정보를 이용한 기계의 추론도 가능해지는데, 예를 들면 오늘 부산으로 출장을 가려고 하는데, 교통편을 예약하려고 한다면, 시맨틱 웹이 제공되는 상황에서는 다음과 같은 가상적인 시나리오가 가능하다. 먼저, 사람은 지능형 기계에게 가장 적



(그림 6) 시맨틱 웹 요소의 계층 구조

합한 교통편에 대한 정보를 요구하면서, 선호하는 교통 수단은 기차라는 인지를 추가해서 질의한다. 그러면, 기계는 철도청 사이트에서 "파업중"이라는 시맨틱 정보를 얻고, "다른 교통편"이라는 시맨틱 정보를 이용하여, 고속버스와 비행기가 가능하다는 것을 얻는다. 그리고, 날씨 사이트에서 "폭설"이라는 시맨틱 정보를 얻어내어, 고속버스보다는 비행기가 좋다는 정보를 사용자에게 제시할 수 있게 된다.

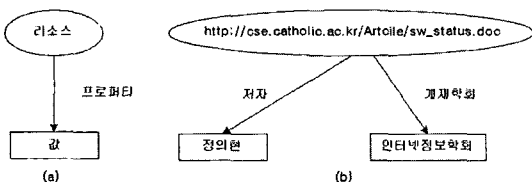
2.2.3 시맨틱 웹의 구성

시맨틱 웹은 기본적으로 그림 6과 같은 계층 구조를 갖고 있다[12]. 계층 구조를 간단히 살펴보면, 모든 리소스는 URI(Uniform Resource Identifier)로 지정된다. 그리고, XML은 URI로 지정된 모든 리소스를 기술하는 기본 태그 언어이며, RDF는 의미를 기술하는데 사용된다. 최근에는 DAML+OIL과 OWL이 새로운 기술 언어로 주목받고 있다. 그리고, 그 위에 이러한 기술 언어를 이용하여 구축된 온토로지(Ontology)들이 있으며, 지능형 소프트웨어를 위한 계층이 그 위에 위치하고 있다.

3. 지능형 웹 서비스의 현황

3.1. 지능형 웹 서비스의 정의

웹서비스의 뛰어난 상호운용성과 장점에도 불구하고, 웹서비스의 경우에는 해당 오퍼레이션과 입력/출력 정보에 대한 의미 정보가 없다는 단점을 가지고



(그림 5) 본 기고의 RDF 그래프 표현

있다. 웹서비스의 경우에는 기본적인 통신 결합이 XML을 기반으로 되어 있다. 그러나, XML은 상호운용을 위한 문법(syntax) 기반의 결합이지, 메시지 자체의 의미를 결합 주체가 이해하여 결합을 하는 것은 아니라는 점이다. 즉, 그림 3에서 볼 수 있는 것처럼 프로그래머가 대항 서비스를 만들거나 서비스의 결합을 위해서는 반드시 사전에 해당 WSDL이 명시하는 결합점(binding point)에 대한 명세를 이해하고 있어야 한다는 점이다.

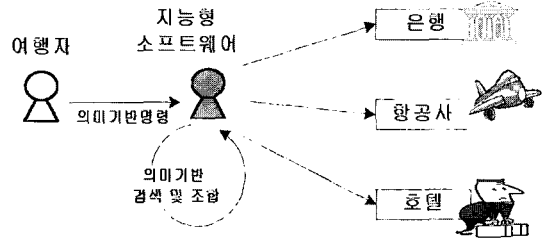
예를 들어, 다음과 같이 항공사의 WSDL에 명시된 인자에 <sample:ReturnDate>라는 인자가 있을 경우, 이것이 여행객의 귀환 날짜를 의미하는지, 예약을 취소할 때 돈을 받는 날짜를 의미하는지 기계는 알 수 없다는 점이다.

```

<operation name= GetInfo >
  <input message= sample:GetInfoRequest />
  <output message= sample:ReturnDate />
</operation>
<ReturnDate>
    
```

단지, 프로그래머가 사전에 그 인자가 의미하는 바를 파악하여, 하드 코딩하는 방법밖에는 없다는 점이다. 이러한 것이 문제가 되는 이유는 웹서비스로 서비스의 동적인 결합이 가능한 반면에, 의미에 맞는 서비스를 동적으로 발견하고, 사람의 개입없이 자동으로 호출하는 것은 어렵기 때문이다. 또한, 이런 구조에서는 서비스의 결합점이 변경되는 경우에 소스 레벨에서 다시 코딩을 해주어야 한다[6]. 즉, 웹서비스에서 지적되는 문제는 웹서비스의 호출자가 의미적으로 웹서비스를 이해하거나 사용할 수 없다는 점이다. 이에 비해 지능형 웹서비스의 목표는 기계가 사용자를 대항하여 자동으로 서비스를 검색하고, 자동으로 입력과 출력에 대한 정보를 이해하는 것이다. 이 목표가 제대로 이루어진다면, 정보를 의미기반으로 찾거나 추론하는 것보다 더 다양한 분야, 사용자의 의미기반 명령에 어울리는 서비스 응용의 결합이 가능하다는 점이다.

이해를 돕기 위해 여행자 시나리오를 다시 한번 생각해보자. 그림 3의 경우에는 여행자에게 대항 서비스



(그림 7) 지능형 웹 서비스의 구조

를 하기 위해, 프로그래머가 사전에 웹 서비스에 대한 의미를 모두 파악한 후에 하드코딩을 해주었다. 그러나, 지능형 웹서비스가 제공되는 시점에서는 그림 7에서 볼 수 있는 것처럼 에이전트에게 의미기반 명령을 주면, 에이전트가 어떤 웹서비스가 그것을 제공하는지를 시맨틱 검색을 통해서, 필요한 웹서비스를 동적으로 조합할 수 있게 된다.

이러한 경우에 중간에 서비스의 변화가 있어도 지능형 소프트웨어가 자동으로 대체할 서비스를 찾게 되며, 이런 구조는 동적인 B2B 결합에도 큰 장점이 될 수 있다. 그러나, 시맨틱 웹 기술과 웹서비스는 서로 다른 목적을 갖고 시작되었기 때문에, 두 기술의 연동에 관한 연구는 아직 시작점에 있다. 기본적으로 시맨틱 웹은 인터넷을 기계가 이해할 수 있는 정보 저장소에 대한 연구인데 반해, 웹서비스는 분산책체 기술에 근거하기 때문이다. 이 두 가지 기술을 결합하고자 하는 연구 중 가장 주목할 만한 것은 DAML-S[13]이다.

3.2 DAML-S

3.2.1 DAML-S의 목표

DAML-S는 DARPA Agent Markup Language for Services의 약어로 카네기멜론, 스탠포드, 예일, BBN Technology, SRI 등이 공동으로 진행하는 연구이다. 기존 시맨틱 웹이 웹 콘텐츠에 시맨틱 태깅을 하여 지식 구조를 만들려는 시도인데 반해, DAML-S는 콘텐츠보다는 웹에서 접근가능한 프로그램, 센서, 디바이스 등에 시맨틱 태깅을 하는 것을 목표로 한다. 즉, 서비스에 대한 의미를 태깅하는 것이라고 정의할 수 있다. 이것이 현실화되면 기계가 서비스를 의미론적으로 이해할 수 있기 때문에, 사용자의 의도에 가장 적합한 서비스

처리를 자동으로 찾고(discover), 실행하고(invocation), 조합하고(composition), 모니터링(monitoring) 하는 것이 가능하게 된다. DAML-S가 추구하는 기능은 다음과 같다.

· 자동화된 웹 서비스 발견(Automatic Web service discovery)

현재 웹에서 제공되는 서비스를 사용하기 위해서는 사용자가 검색 엔진을 통해서 서비스를 제공하는 사이트를 찾아야 했다. 그러나, DAML-S는 기계가 서비스들이 제공한 시맨틱 정보를 이용해서 자동으로 필요한 서비스를 찾아내게 된다.

· 자동화된 웹서비스 실행(Automatic Web service invocation)

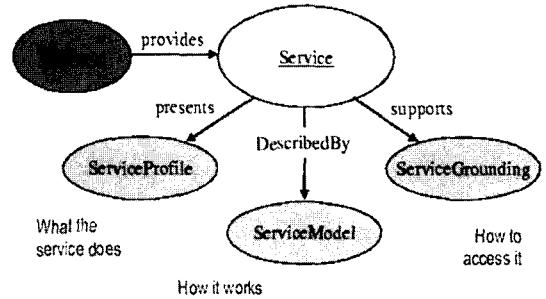
서비스를 사용하기 위해서는 사용자가 해당 웹사이트에 직접 가서, 폼에 여러 조건을 입력하고 서비스를 수작업으로 실행해야 한다. 이것을 DAML-S에서는 이러한 서비스를 함수 호출 집합이라는 관점에서 파악하고, 기계가 이해할 수 있는 시맨틱 정보를 제공한다. 기계는 이 시맨틱 정보를 이용하여 해당 서비스를 사용하기 위해 어떤 입력과 출력이 이해할 수 있어, 자동화된 실행을 할 수 있게 된다.

· 자동화된 웹서비스 조합(Automatic Web service composition and interoperation)

만일 여러 개의 서비스를 사용해야 하는 경우, 사용자가 여러 조건에 맞추어 사이트를 옮겨 다니며 조합을 해주어야 한다. DAML-S에서는 기계가 서비스에 대한 이해를 하고 있기 때문에, 자동으로 필요한 서비스들을 조합할 수 있도록 지원한다.

· 자동화된 웹서비스 모니터링(Automatic Web service execution monitoring)

개별 서비스를 사용자가 사용하는 경우에는 사용자가 해당 서비스에 대한 모니터링을 직접 하는 것이 가능하며, 자동화된 서비스 호출에서도 이런 부분들에 대한 모니터링을 제공해야 한다. 하지만, 이 부분은 DAML-S의 목표이기는 하지만, 아직까지는 제공되지 못하고 있다.



(그림 8) 서비스 온토로지의 관계

3.2.2 DAML-S의 구조

DAML-S의 온토로지 구조는 다음과 같은 3가지 정보에 대한 것을 제공하기 위해 구성되어 있다. 첫째, 해당 서비스는 무엇이고, 무엇을 할 수 있는가? 둘째, 해당 서비스는 어떻게 동작하는가? 셋째, 서비스를 어떻게 사용하는가? 그림 8[13]은 RDF 그래프 형태로 서비스에 대한 클래스의 관계를 보여준다.

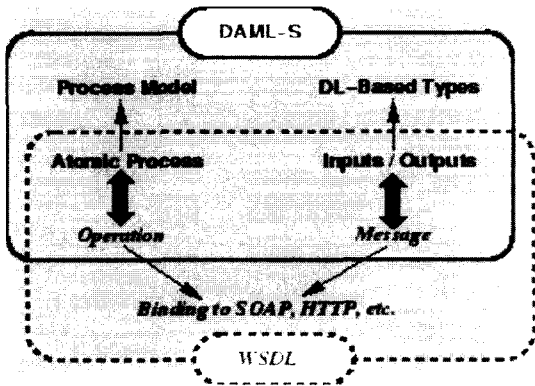
Service 클래스는 특정 웹 서비스를 선언하는 구조적 진입점을 나타내고, Service 클래스의 인스턴스는 구별 가능한 단위 서비스를 의미한다. 특정 Service 클래스는 ServiceProfile, ServiceModel, ServiceGrounding 클래스에 의해서 각각의 속성이 기술된다.

· 서비스 프로파일(Service Profile)

서비스 프로파일은 해당 서비스가 무엇인지를 설명하는 것이다. 즉, 서비스를 찾는 기계에게 필요한 정보로서 해당 서비스가 요구에 적합한지를 알아내는데 사용한다. 서비스 프로파일은 크게 3가지 정보를 제공하는데, 첫째, 서비스를 제공하는 서비스 제공자에 대한 정보, 둘째, 서비스가 제공하는 기능, 셋째, 해당 서비스의 속성(등급이나 평가)에 관한 정보이다. 이러한 정보들은 레지스트리 서비스를 통해서 기계가 필요로 하는 서비스를 찾는데 이용된다.

· 프로세스 모델(Process Model)

DAML-S에서는 서비스를 정확히 표현하기 위해, 각 서비스를 프로세스(process)라는 관점에서 파악한다. 이 모델을 이용해서 복합 서비스 등의 서비스 호출을 하고자 하는 기계가 해당 서비스에 대한 흐름, 입력값, 출력값에 대한 정보들을 알 수 있다. 그리고, 이러



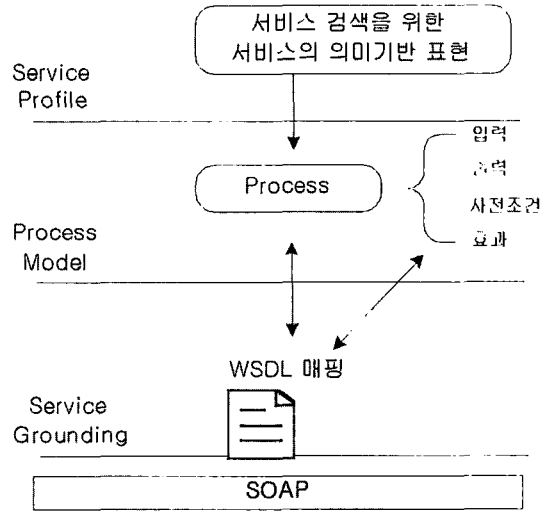
(그림 9) WSDL과 서비스 기반의 관계

한 관점을 만족시키기 위해서, ServiceModel의 하위 클래스인 ProcessModel를 정의하였다. 프로세스 모델은 크게 process와 process control model의 두 가지로 구성된다. process는 입력, 출력, 사전조건 등에 대한 것을 명시하도록 구성되어 있으며, Process Ontology에 기술되어 있다. process control model은 서비스 요청의 모니터링에 대한 것이지만, 아직 완전히 명시되어 있지 않다.

· 서비스 기반(Service Grounding)

서비스 기반은 하위 통신 프로토콜, 메시지 포맷 등의 서비스의 실제 호출과 관련된 부분에 대한 정보를 갖고 있다. 일반적으로 서비스 프로파일과 프로세스 모델에서 정의한 프로세스, 그리고 입출력 정보 등을 그림 9[13]와 같이 하위의 WSDL을 이용해서, 매핑 작업을 해주게 된다. 일반적으로 프로세스는 WSDL의 개별 서비스에 해당되며, 입출력 정보는 WSDL의 메시지에 대응되게 된다. 이러한 부분은 상위 온토로지 에서 정의한 기술을 실제 하위의 웹 서비스 기술과 연결시켜주는 기능을 하게 된다.

이 세가지 구성 요소가 지능형 웹서비스의 요구 사항에 어떻게 대응되는지 살펴보자. 먼저, 자동화된 웹 서비스 발견은 서비스 프로파일에 의해서 지원된다. 둘째, 자동화된 웹서비스 실행과 조합은 프로세스 모델에 의해서 가능하다. 그리고, 하위 웹 서비스와의 매핑은 서비스 기반이 담당하게 된다. DAML-S의 구성요소의 계층 구조는 그림 10과 같다.



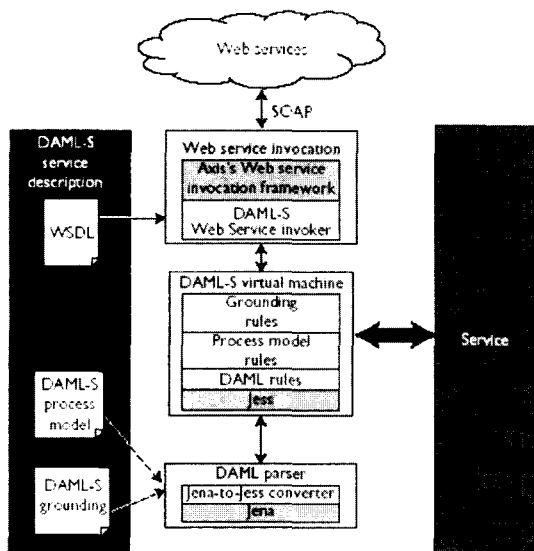
(그림 10) 계층별 모델과 구조

3.2.3 DAML-S의 응용

DAML-S의 가능성에도 불구하고, DAML-S의 실제 응용 사례는 아직 많지 않은 편이다. 기본적으로 DAML-S를 다루는 틀은 서비스 조합을 그래픽하게 처리할 수 있는 웹서비스 컴포저(Web Services Composer)[14], 기존 웹 서비스 표준과의 상호운용을 위한 WSDL2DAML-S 컨버터(Converter)[15], DAML-S matchmaker[16] 등이 있다. 앞으로 지능형 웹 서비스가 좀더 활발해지면, 틀 분야에 대한 연구도 활발해질 것으로 기대한다. 그리고, DAML-S를 채택하여 자신의 웹 서비스를 제공하는 곳이 아직 없기 때문에, 아마존(Amazon)에서 발표된 웹 서비스를 DAML-S 버전으로 구성한 것이 카네기멜론의 Atlas 팀에 의해서 제공되고 있다[17]. 이 외에도 카네기 멜론의 Kasahr[6]에 의해 제시된 DAML-S의 아키텍처는 그림 11과 같다.

4. 결론 및 향후 전망

정보통신기술의 급속한 발전으로 인해서, 보다 빠르고 어디서나 접근할 수 있는 차세대 인터넷의 비전은 멀지 않은 장래에 실현될 수 있을 것으로 보인다. 그러나, 사용자들에게 정보를 보다 효율적으로 제공하는 정보 시스템에 관한 문제는 여러 문제점을 나타내고 있다. 본 기고에서는 정보 시스템적인 측면에서의



(그림 11) DAML-S의 구현 예

인터넷의 현황과 향후 주목할 기술중의 하나인 지능형 웹 서비스와 그 근간을 이루는 시맨틱 웹, 웹 서비스 기술에 대해서 살펴보았다. 시맨틱 웹은 기계가 정보를 자동으로 처리할 수 있다는 면에서 기계대 기계의 상호운용에 대한 가능성을 제시하고 있고, 웹 서비스는 기존 분산객체 기술을 웹 기반으로 확장한 분산 컴퓨팅 방식이다. 상기한 두 가지 기술이 결합되는 지능형 웹 서비스는 사용자의 서비스 요청을 의미기반으로 받아서, 사용자에게 제공하는 보다 지능화된 인터넷 서비스를 제공할 수 있을 것으로 기대된다. 이러한 기술은 기존의 e-business 구조에 지능을 부여하게 되어, 다양한 수준의 고부가 서비스 시장을 만들 수 있으며, 차세대 인터넷 정보 시스템의 근간으로 발전할 것으로 예상된다. 따라서, 향후의 인터넷 정보 시스템은 지능형 소프트웨어들이 사용자들의 의미적 요구를 받아들여, 사용자의 개입 없이 자동화한 부가 서비스를 제공하는 보다 인간 중심적인 정보 환경으로 진화할 것으로 예측된다. 또한, 이 분야는 여러 IT 기술 분야에 영향을 미칠 뿐 아니라, 향후 IT 기술의 향방을 가름할 중요한 연구 분야이므로 학계와 산업계에서 많은 연구와 관심을 가져야 할 것이다. 물론 이러한 시도가 지능형 소프트웨어가 모든 것을 대행해주는 식의 영화와 같

은 미래의 인터넷을 담보하지는 않지만, W3C에서 시맨틱 웹을 주도하고 있는 Eric Miller[1]의 언급대로, “웹을 이용해서 반 나절 걸릴 일을 1시간 안에 한다면 그것으로 가치가 있다”라고 얘기한 관점에서, 앞으로의 인터넷 정보 시스템의 한 흐름으로 관심을 기울여야 할 것이다.

참고문헌

- [1] S.M.Cherry, “Weaving a Web of Ideas,” IEEE Spectrum, 39(9):65-69, Sep., 2002.
- [2] T.Berners-Lee, J.Hendler, and O.Lassila, “The Semantic Web,” Scientific American, 284(5):34-43, May, 2001.
- [3] J.Heflin and J.Hendler, “A Portrait of the Semantic Web in Action,” IEEE Intelligent Systems, 16(2): 54-59, Mar./Apr. 2001.
- [4] J.Roy and A.Ramanujan, “Understanding Web services,” IEEE IT Pro, 3(6):69-73, Nov./Dec. 2001.
- [5] B.Meyer, “.Net is coming[Microsoft Web services platform],” IEEE Computer, 34(8):92-97, Aug. 2001.
- [6] M.Paolucci and K.Sycara, “Autonomous Semantic Web Services,” IEEE Internet Computing, 7(5):34-41, Sep/Oct. 2003.
- [7] S.A.McIlraith, T.C.Son, and Zeng Honglei, “Semantic Web services,” IEEE Intelligent System, 16(2):46-53, Mar./Apr. 2001.
- [8] W3C Working Draft, “Web Services Architecture,” <http://www.w3.org/TR/2003/WD-ws-arch-20030808/>, Aug., 2003.
- [9] RDF Home Page, <http://www.w3.org/RDF/>, 2003.
- [10] DAML Home Page, <http://www.daml.org/>, 2003.
- [11] W3C Proposed Recommendation, “OWL Web Ontology Language Overview,” <http://www.w3.org/TR/2003/PR-owl-features-20031215/>, 2003.
- [12] T.Berners-Lee, Keynote session, XML-2000 Conference, Washington DC, Dec., 2000.
- [13] The DAML Services Coalition, “DAML-S: Semantic Markup for Web Services,” May, 2003.

- [14] E.Sirin, J.Hendler, and B.Parsia, "Semi-automatic Composition of Web Services using Semantic Descriptions," ICEIS 2003, 2003.
- [15] WSDL2DAML-S Converter Home Page, Home Page, "http://www.daml.ri.cmu.edu/wsd2damls/", 2003.
- [16] M. Paolucci, "Semantic Matching of Web Services Capabilities," Proc. of ISWC, LNCS 2342, Springer Verlag, pp. 333~347, 2002.

○ 저 자 소 개 ○



정 의 현

1992년 한양대학교 전자공학과(학사)

1994년 한양대학교 전자공학과(석사)

1999년 한양대학교 전자공학과(박사)

1999년~2002년 (주) 대우통신 선임 연구원

2002년 (주) 스마트카드 테크놀로지 연구소장

2003~현재 : 가톨릭대학교 컴퓨터정보공학부 초빙교수

관심분야 : 네트워크 컴퓨팅, 지능형 에이전트, 시맨틱 웹, P2P