

시맨틱 웹을 위한 객체지향의 마크업 언어

유명환*, 정희준*, 이강찬**, 김성한**, 민재홍**, 정인정*

* 고려대학교 전산학과 {myong, joonny96, chung}@korea.ac.kr

** 한국전자통신연구원 표준연구센터 {chan, sh-kim, jhmin}@etri.re.kr

Object Oriented Markup Language for the Semantic Web

Myong-Hwan Yoo*, Hee-Joon Chung*, Kang-Chan Lee**,
Sung-Han Kim**, Jae-Hong Min**, In-Jeong Chung*

* Dept of Computer Science, Korea University

** Protocol Engineering Center, Electronics and Telecommunications Research Institute

요 약

현재의 웹은 시각적인 표현을 위한 것으로써 정보를 사람에게 어떻게 보여줄 것인가에 대해서 초점을 두고 개발되었다. 따라서 폭발적으로 증가하는 웹 데이터에서 사용자가 원하는 정보를 신속·정확하게 찾는 것은 점점 어렵게 되었다. 이를 개선하기 위해 자연언어처리, 에이전트, 검색엔진 등과 같은 기술을 개발하였으나 정보와 표현을 위한 태그의 혼합으로 컴퓨터가 정보를 효과적인 추출 및 이해하는데 한계가 있다. 이는 지금까지의 웹 기술로써는 다양한 표현과 사용하기 쉽지만 정보의 의미표현이 부족하기 때문이다. 이러한 문제점을 해결하기 위해 정보를 은트로지로서 개념화하고 이를 컴퓨터가 이해하며 이기종 컴퓨터간의 자유로운 정보접근을 위해 1990년 대 말에 시맨틱 웹이 제안되었다. 현재 시맨틱 웹은 RDF(S), OIL, DAML, SHOE 등과 같은 마크업 언어가 연구·개발 중에 있으나 이 역시 지식표현 분야 위주의 연구로 그 한계가 있다. 이에 본 논문에서는 시맨틱 웹을 위한 지금까지의 마크업 언어에 대한 분석을 하고, 효과적인 시맨틱 웹의 구현을 위한 객체지향의 마크업 언어를 제안한다. 본 논문에서 제안하는 마크업 언어는 이기종의 분산환경에 적합하고 재사용성 및 확장성에 용이하는 등의 장점들을 갖고 있다.

1. 서론

초기의 웹은 정보를 어떻게 보여줄 것인가에 대해서 초점을 두고 개발하였다. 일반적인 문서가 텍스트 기반의 문서인 것에 반해 웹 문서는 이미지, 멀티미디어, 하이퍼링크와 같은 여러 형태의 정보를 다각적으로 보여주기 위해 개발된 것으로써 웹을 통한 정보의 교환을 주목적으로 하고 있다. 초기의 이러한 시도는 정보의 원천으로써 웹을 이끌어 내었고 이는 HTML이라는 사용하기 쉽고 다양한 형태의 데이터를 시각적으로 보여주기 위한 마크업 언어의 개발에서 비롯된 것이다.

시각적 측면에 초점을 둔 현재의 웹은 사람의 노력에 의해 정보를 찾고 추출한다. 그러나 이러한 방법은 정보의 양이 기하급수적으로 늘어나는 현실에서는 한계가 있다. 이를 개선하기 위해 자연언어처리, 에이전트, 검색엔진 등과 같은 기술을 연구되었으나 HTML이 정보와 표현을 위한 태그가 섞여 있는 현재의 웹 환경에서는 그 한계점이 있다. 이를 개선하기 위해 구조적인 문서 표현과 정보와 표현을 분리시켜

정보와 표현의 수정이 용이한 XML이 제안되었다. 그러나 XML도 역시 정보의 정확한 의미표현이 불가능하고 다양한 형태의 표현으로 인하여 정보의 애매성을 증가시키는 단점을 지니고 있다.

지금까지의 웹 기술은 인간과 컴퓨터간의 정보교환에 치중하고 정보와 태그의 혼합으로 정보추출의 어려움 그리고 컴퓨터의 웹 데이터에 대한 의미적 해석 및 처리의 취약 등의 어려운 문제점들이 있었다.

이를 해결하고 컴퓨터간의 정보교환이 가능하게 하며 웹 상의 데이터의 의미를 사람이 아닌 컴퓨터가 이해·처리할 수 있는 새로운 기술인 시맨틱 웹이 1990년 대 말에 소개되었다. 시맨틱 웹은 분산처리 환경 하에서 서로 다른 데이터베이스 및 이기종 컴퓨터마다 상이한 의미를 갖는 데이터에게 각각의 컴퓨터 및 응용 프로그램들에서도 자유롭게 접근이 가능하도록 한다.[1,2].

현재 연구·개발된 시맨틱 웹을 위한 마크업 언어로는 RDF(S)[4], OIL[5], DAML[6,7], SHOE[8] 등이 있다. 이러한 마크업 언어는 주로 지식표현 분야에서 주로 연구가 진행되고 있다. 그러나 지식표현 분야에

서의 연구는 현실세계를 모델링하는데 있어서 그 한계점이 있다. 이에 본 논문에서는 현재 연구·개발된 마크업 언어를 비교 및 분석을 통하여 그 단점을 개선하고 분산환경에 적합하며 정보의 재사용성과 확장성이 뛰어난 마크업 언어를 제안하고자 한다.

본 논문은 다음과 같은 구성으로 이루어져 있다. 2장에서는 관련 연구로써 시맨틱 웹의 핵심이 되는 온톨로지에 대한 설명과 기존의 마크업 언어들의 비교 분석을 한다. 3장에서는 기존의 마크업 언어의 문제점을 해결하기 위한 객체지향의 마크업 언어를 제안하고, 마지막으로 결론과 향후 과제를 언급한다.

2. 관련연구

2.1. 온톨로지

컴퓨터가 정보를 이해할 수 없는 현재 웹 환경의 문제점들을 해결하기 위하여 컴퓨터가 이해할 수 있는 지식을 웹 문서에 삽입해야 한다. 이를 위해 공통적이고 표준적인 지식표현 방법이 필요하다. 따라서 효과적인 웹 환경을 구현하기 위해서는 다음과 같은 온톨로지가 중요하다.

온톨로지는 공유되는 데이터들의 개념화한 형식적이고 명백한 규정이다. 이는 특정 분야에서 사용되는 표준 어휘들의 모음이라고 할 수 있다. 에이전트의 수행을 위한 정보를 만들고 질의어를 이용한 정보검색을 위해서는 도메인의 개념화를 해야 한다. 즉, 도메인 내의 지식을 표현하고 의사소통 하기 위한 단어를 제공함으로써 이기종 간의 상호운용성을 보장하고 지식의 통일화를 시킨다[1,2].

온톨로지는 어휘 사전의 역할 이외에 지식을 효과적으로 표현하기 위해 정보의 의미를 부여하고, 또 정보들간의 관계를 설정해 준다. 즉, 온톨로지는 광범위한 도메인에 적용이 가능하도록 표준을 제시함으로써 웹 문서에 나타난 지식을 표현, 공유와 재사용을 그 목적으로 하고 있다. 더 나아가 시맨틱 웹의 목적인 자동적인 실행과 추론을 하기 위해 그 중요성이 부각되고 있다[3].

시맨틱 웹을 위한 마크업 언어는 웹 문서에 이러한 온톨로지를 삽입함으로써 웹 문서에 정보의 의미성과 정보간의 관계를 설정해 주고 정보의 수집을 용이하게 하며 추론과 정보의 재조합과 자동적인 실행을 하기 위한 조건을 만들어 준다.

2.2. 시맨틱 웹을 위한 마크업 언어의 비교·분석

현재 시맨틱 웹을 위한 마크업 언어로는 RDF(S)[4], OIL[5], DAML[6,7], SHOE[8] 등이 있다. RDF는 XML의 의미표현 부족을 개선하기 위해 시맨틱 네트워크에 근간하여 자원의 의미를 표현하고 이는 후에 OIL과 DAML의 기본 문법으로 사용된다[4,5,6]. OIL과 DAML은 프레임 시스템과 기술 로직에 근간하여

제약조건과 분류체계를 제공한다. DAML은 현재 OIL과 결합된 형태로 DAML+OIL이라는 언어로 사용된다[5,6,7]. SHOE는 이와 별개로 HTML확장 형태의 언어로써 ISA 계층구조와 FOL(First-Order Logic), Horn-clause Logic을 기반으로 하여 개발되었고 추론규칙을 제공한다. 각 언어는 표1과 같은 특징이 있다. 각 기호는 요소들의 표현능력을 나타낸 것으로 ○는 잘 정의가 되어 있는 것을 나타내며 △는 해당 요소를 제공은 하나 그 기능이 미약함을 나타내며, ×는 제공하지 않음을 나타낸다.

지금까지의 시맨틱 웹을 위한 마크업 언어는 다음과 같은 문제점이 있다. 모든 언어들이 의미표현에 있어서 잘 정의가 되어 있으나 RDF와 SHOE의 경우 온톨로지의 제약조건을 제시할 수 없다. 추론규칙은 OIL과 DAML+OIL의 경우 언어 상에서 명시적으로 표현하지 않고 분류체계나 "union", "intersection" "disjoint" 등과 같은 조건으로써 추론규칙을 정의한다. SHOE는 Horn-clause Logic에 기반한 IF-THEN

	RDF(S)	OIL	DAML+OIL	SHOE
의미표현	○	○	○	○
제약조건	X	△	○	X
추론규칙	X	△	△	○
추상클래스	X	△	△	X
다중상속	○	○	○	X
자료구조	△	△	△	X
분류체계	○	○	○	△
재명명	X	X	○	○
충돌회피	X	X	△	○

표 1 시맨틱 웹을 위한 마크업 언어의 비교

형태의 명시적인 추론규칙을 정의한다. 모든 언어들이 추상클래스는 지원하지 않으나 OIL과 DAML은 상위클래스가 추상클래스로써의 역할을 대신한다. 분류체계는 기술 로직에서 제공하는 일반화(generalization)와 상세화(specialization)로 분류체계를 제공하고 SHOE는 ISA 관계로 이를 대신한다. 재명명은 DAML+OIL은 "sameClassAs"과 "samePropertyAs", SHOE는 "rename"을 사용하여 클래스나 속성의 이름을 다른 이름으로 대신한다.

재명명은 서로 다른 온톨로지를 импорт(import)시켰을 경우 발생하는 충돌과 관련된 것으로 DAML+OIL은 이러한 재명명을 통하여 충돌발생을 억제할 수 있으나 상속받는 클래스의 속성들의 충돌을 방지할 수 없다. 이에 비해 SHOE는 이런 경우에 "PREFIX"를 사용하여 효과적인 충돌회피를 할 수 있다.

```
<ooml:Class rdf:ID="Hermaphrodite">
  <ooml:unionOfInheritance rdf:parseType="ooml:Inheritance"/>
  <rdfs:subClassOf rdf:about="#Male"/>
  <rdfs:subClassOf rdf:about="#Female"/>
</ooml:unionOfInheritance>
</ooml:Class>
```

그림 1 제안하는 객체지향의 마크업 언어로 표현된 자동분류 샘플의 온톨로지

```
<daml:Class rdf:ID="Male">
  <rdfs:subClassOf rdf:resource="#Animal"/>
</daml:Class>

<daml:Class rdf:ID="Female">
  <rdfs:subClassOf rdf:resource="#Animal"/>
  <daml:disjointWith rdf:resource="#Male"/>
</daml:Class>
```

그림 2 DAML+OIL로 표현된 생물학적 암수 구분 온톨로지

```
<daml:Class rdf:ID="Hermaphrodite">
  <daml:unionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#Male"/>
    <daml:Class rdf:about="#Female"/>
  </daml:unionOf>
</daml:Class>
```

그림 3 DAML+OIL로 표현된 자동동체 생물의 온톨로지(unionOf사용)

```
<daml:Class rdf:ID="Hermaphrodite">
  <rdfs:subClassOf rdf:resource="#Male"/>
  <rdfs:subClassOf rdf:resource="#Female"/>
</daml:Class>
```

그림 4 DAML+OIL로 표현된 자동동체 생물의 온톨로지(다중 상속사용)

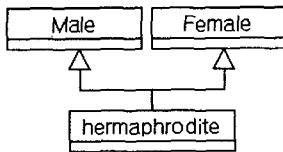


그림 5 UML로 표현된 자동동체 생물의 온톨로지

3. 객체지향의 마크업 언어의 제안

시맨틱 웹을 위한 마크업 언어는 온톨로지를 효과적으로 표현해야 한다. 그리고 마크업 언어는 온톨로지 설계자를 위해 직관적인 지식표현 방법을 제공해야 하고 분산환경에 적합하도록 온톨로지를 공유 및 재사용할 수 있어야 한다. 또한 변화하는 실세계에 적합하도록 온톨로지 역시 변화를 대응할 수 있어야 할 뿐만 아니라 기존의 웹 환경을 수용할 수 있어야 한다. 본 논문에서는 객체지향의 마크업 언어를 제안함으로써 위의 조건에 적합하고 온톨로지의 재사용성과 확장성을 제고하고자 한다.

시맨틱 웹을 위한 마크업 언어에 객체지향 패러다임을 사용함으로써 기존의 마크업 언어에서 얻을 수 없었던 추상 클래스, 정보 은닉 등의 기법이 사용되어 모델링하려는 도메인의 개념을 단순화시키고 온톨로지의 유연성과 재사용성을 높일 수 있다.

시맨틱 웹을 위한 마크업 언어는 도메인 분석에 있어서 정확하고 직관적인 방법을 제공한다. 이를 위

한 메타 언어로써 소프트웨어 공학 쪽에서 널리 사용되는 UML[11]을 사용할 수 있다. UML은 소프트웨어 설계·디자인 부분에서 사용되는 모델링 언어로써 그래픽 표기 방법을 사용하여 직관적인 방법을 제공한다. 시맨틱 웹을 위한 마크업 언어는 궁극적으로 컴퓨터가 이해할 수 있어야 한다. 이에 UML은 컴퓨터 응용프로그램에 적합한 메타 모델을 제공한다. DAML에서도 UML의 사용을 모색하고 있으나 전역 범위(global scope)와 같은 문제[9,10]로 인하여 UML을 충분히 활용할 수 없다.

위 그림에서 알 수 있듯이 그림 5의 온톨로지를 표현하는데 있어서 DAML+OIL의 경우 "unionOf"(그림2)와 다중 상속(그림 3)의 두 가지가 있다. 이는 의미 혼동을 가중시킨다. 또한 그림 2와 같이 "unionOf"를 사용하여 새로운 온톨로지를 만들 경우 그림 1[7]에서 "disjoint"로써 "Male"과 "Female"가 양립할 수 없다고 정의를 하였기 때문에 일관성이 없다.

SHOE는 시맨틱 네트워크를 사용하여 개발된 언어로써 그 구조가 단순하고 다른 언어에 비해 사용하기 쉽다는 장점을 지니고 있다[8]. 그러나 이는 객체-속성-값이라는 단순한 형태로 정보를 표현하기 때문에 복잡하고 여러 속성이 필요한 정보를 온톨로지로 표현하는 데는 한계를 지니고 있다. 이에 비해 RDF(S), OIL, DAML 그리고 본 논문에서 제안하고자 하는 객체지향의 마크업 언어는 표현하고자 하는 개념을 여러 속성을 가질 수 있는 클래스로 나타냄으로써 정보를 온톨로지로 효과적으로 나타낼 수 있다.

OIL과 DAML은 프레임시스템, 기술로직에 바탕으로 하기 때문에 전역 범위에 유효한 다음과 같은 요소들이 있다[5,6,7]. OIL의 경우 추상 상위클래스와 유사한 기능으로 다른 클래스들의 대표성을 나타내고 이를 사용하기 위해 정의한 "cover A by B C D", 클래스에 필요한 슬롯을 선정의하는 "slot-def", 이를 다시 상속받는 형태로 재정의하는 "subslot-of", DAML의 경우 데이터의 타입속성을 정의하는 "Datatype Property", OIL의 "slot-def"나 "subslot-of"와 유사한 형태로 각 속성을 정의하는 "Object Property", OIL과 DAML에 공통적으로 나타나는 "domain", "range"는 전역 범위에 작용한다[5,6,7]. 이로 인하여 분산된 웹 환경에서 온톨로지 정의를 위해 다른 온톨로지를 사용할 경우 충돌을 야기시킬 수 있다. 이러한 전역 범위는 온톨로지 구조가 거대해지고 복잡해질수록 잦은 충돌이 발생하고 온톨로지의 확장이나 유지보수를 어렵게 함으로 온톨로지의 재사용성을 떨어뜨린다.

객체지향의 마크업 언어는 이러한 전역 범위 문제를 효과적으로 해결할 수 있다[9,10]. 객체지향 패러다임은 유효범위를 클래스 내의 지역 범위(local scope)로 한정하고 각 클래스의 속성들을 캡슐화(encapsulation)시킴으로써 각 속성간의 충돌 방지하여 다른 시스템간의 온톨로지 교환을 효과적으로 할 수 있을 뿐

만 아니라 온톨로지 확장과 재사용성을 높여준다. 그 외에도 온톨로지 설계자들 간에 전역 범위로 인한 설계 제약을 줄임으로써 병렬적 설계가 가능하게 한다.

OIL과 DAML은 RDF(S)를 수용하였기 때문에 RDF에서 제공하는 "Bag", "Sequence", "Alternative"와 같은 간단한 자료구조를 제공하고 SHOE는 제공하지 않는다. 현실세계는 다양한 자료구조가 필요하며 현재 제공하는 자료구조 외에도 집합, 다차원 배열, 객체 등의 다양한 자료구조를 제공해야 한다. 그림 6은 2×3의 배열에서 2행 2열에 값이 7인 최소행렬을 나타낸 것이다. 이러한 자료구조의 제공은 실제계에 근접한 온톨로지의 모델링에 중요한 역할을 담당한다.

```
<ooml:Array rdf:parseType="ooml:array">
  <ooml:Scale="0" rdf:parseType="ooml:arrayScale">
    <ooml:Index="1">2</ooml:Index/>
    <ooml:Index="2">3</ooml:Index/>
    <ooml:Default>0</ooml:Default/>
  </ooml:Scale/>
  <ooml:Position="7" rdf:parseType="ooml:arrayPosition">
    <ooml:Index="1">2</ooml:Index/>
    <ooml:Index="2">2</ooml:Index/>
    <ooml:Value>7</ooml:Value/>
  </ooml:Position/>
</ooml:Array/>
```

그림 6 제한하는 객체지향의 마크업 언어로 표현된 최소행렬

이외에도 객체지향 패러다임을 마크업 언어에 이용함으로써 추상 상위클래스[11]를 사용할 수 있다. 현재 하향방식의 온톨로지 설계방법인데 반해 추상 상위클래스를 사용함으로써 상하향의 양방향 설계가 가능하다. 또한 하위 클래스에서 중복되는 값의 불필요한 입력을 피할 수 있을 뿐만 아니라 치환(overriding)을 통하여 클래스의 상세화에 적합한 방법을 제공한다.

4. 결론 및 향후 과제

시맨틱 웹은 컴퓨터가 이해할 수 있도록 정보를 온톨로지로서 웹 문서에 삽입하여 자동적인 저오의 추출·재조합·자동실행이 가능하도록 하는 차세대 웹 환경이다. 시맨틱 웹의 이상적인 기능을 수행하기 위해서는 가장 먼저 웹 문서에 온톨로지를 삽입할 수 있는 마크업 언어에 대한 연구가 필요하다. 본 논문은 개발·연구 중인 RDF(S), OIL, DAML, SHOE와 같은 기존의 마크업 언어를 비교·분석하고 이를 바탕으로 단점을 해결할 수 있는 객체지향 방식의 마크업 언어를 제시하였다.

시맨틱 웹은 독립된 새로운 웹 환경이 아닌 기존의 웹 환경을 수용하고 자동화되고 지능적인 웹 환경을 제공하는데 그 목적이 있다. 이를 위해서는 기존의 HTML, XML 등과 같은 마크업 언어들을 수용할 수 있어야 한다. 따라서 독자적인 마크업 언어가 아닌 기존의 마크업 언어를 기반으로 하여 발전해 나가야 할 것이다. 마크업 언어의 연구와 함께 마크업 언어로 표

현된 지식을 효과적으로 사용하여 시맨틱 웹을 구현하기 위해 에이전트 등의 기술 연구가 병행되어야 한다. 또한 객체의 의미 표현, 재사용성, 확장성 등의 연구가 오랫동안 진행해 온 소프트웨어 공학측면에서의 연구도 필요하다.

참고문헌

- [1] McIlraith, S.A., Son, T.C., Honglei Zeng. Semantic Web services. IEEE Intelligent Systems. Vol. 16 Issue 2, p.46-53 March-April 2001.
- [2] J. Hendler. Agents and the Semantic Web. IEEE Intelligent Systems. Vol.16 Issue 2, p.30-37 March-April 2001.
- [3] Introduction to Ontologies on the Semantic Web <http://www.cs.umd.edu/users/hendler/ontologies.html>
- [4] S. Decker, P. Mitra, S. Melnik. Framework for the semantic Web: an RDF tutorial. IEEE Internet Computing , Vol. 4 Issue 6 p.68-73, Nov.-Dec. 2000
- [5] D. Fensel, I. Horrocks, F. Van Harmelen, S. Decker, M. Erdmann, M. Klein. OIL in a nutshell. Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling, and Management. p.1-16. 2000.
- [6] About DAML. <http://www.daml.org/about.html>
- [7] Annotated DAML+OIL Ontology Markup <http://www.w3.org/TR/daml+oil-walkthru/>
- [8] The SHOE Specification <http://www.cs.umd.edu/projects/plus/SHOE/spec.html>
- [9] S. Cranefield, S. Haustein, M. Purvis. UML-Based Ontology Modelling for Software Agents Proceedings of the Workshop on Ontologies in Agent Systems, 5th International Conference on Autonomous Agents. 2001
- [10] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, Markus Stumptner, Markus Zanker. Transforming UML domain descriptions into Configuration Knowledge Bases for the Semantic Web. Proceedings of the Workshop on Knowledge Transformation for the Semantic Web p. 11-18. 2002
- [11] Jesse Liberty. Beginning Object-Oriented Analysis and Design with C++. WROX 1998