

자동적 상황인지를 위한 동사의 표현

손종수, 정인정

고려대학교 과학기술대학 전산학과
충청남도 연기군 조치원읍 서창리 208 고려대학교 과학기술대학 202호
Tel: +82-41-860-1342, E-mail: {mis026, chung}@korea.ac.kr

요약

인터넷의 보급과 정보통신기술의 발달로 인해 사용자가 접할 수 있는 자료의 양은 절대적으로 방대해졌지만 그에 비해 정보를 효율적으로 가공하는 방법은 여전히 키워드 검색에 의존하고 있다. 이 문제를 해결하기 위하여 기계가 자료의 의미를 이해하고 가공할 수 있는 시맨틱 웹이 소개되었다. 또한, 사용 가능한 정보기기가 다양화 되어 유비쿼터스 컴퓨팅 환경에서 서로 다른 언어와 시스템에서의 호환성 문제를 해결하는 것이 중요한 과제가 되었다.

본 논문에서는 시맨틱 웹 기술과 퍼지 집합을 이용하여 유비쿼터스 컴퓨팅 환경 하에서 자동적 상황인지를 위한 동사(Verb)를 표현한다. 사용자의 상황을 자동적으로 인지하고 처리하기 위해서는 적용될 정보기기들의 언어와 기종에 독립적인 시스템을 구축해야 하는데 이 문제를 해결하기 위하여 표준 웹 온톨로지 언어인 OWL을 사용하였다. 그리고 사용자가 접할 수 있는 현실의 문제들은 일반적인 집합(Crisp set)으로 표현하기 힘들기 때문에 퍼지 집합이 적용 가능한 형태로 확장된 OWL언어를 이용하였다. 표현한 동사들은 서로 다른 여러 객체와의 상관관계로서 표현이 되며 그 조합을 이용하여 서비스가 취할 행동을 추론할 수 있다.

키워드

Semantic web; Ontology; Ubiquitous; Context; Fuzzy

서론

1990년대 중반 이후 인터넷이 본격적으로 보급되기 시작했다. 인터넷의 보급은 정보통신 기술의 발달로 인한 것 이었으며 또한 인터넷의 보급으로 인해 정보통신 기술의 발전 속도가 급속도로 올라갈 수 있었다. 이에 따라 사용자가 취득할 수 있는 각종 자료의 절대적인 양은 급증하게 되었으며 활용 빈도도 높아지게 되었다.

그러나 사용자가 접할 수 있는 많은 양의 자료를 수집하여 그것을 정보로서 가공하는 작업은 여전히 쉽지 않다.

취득할 수 있는 방대한 양의 자료 가운데 필요한 자료만을 골라내는 것이 쉽지 않은 이유는 사용자가 입력한 단어에 대한 의미적인 해석을 컴퓨터 스스로 하지 못하여 자료를 수작업으로 분류 관리해야 하기 때문이다. 이러한 문제를 컴퓨터 스스로 해결할 수 있도록 하기 위하여 주창된 기술은 Tim Berners Lee에 의해 제기된 시맨틱 웹이다. 시맨틱 웹은 기계가 이해하고 처리할 수 있는 형태로 자료를 표현하는 웹을 의미한다. 이는 사람의 도움을 최소화하여 기계 스스로 자료처리가 가능한 형태로 자료를 표현한다. 시맨틱 웹을 구현하기 위해서는 하나의 메타 데이터(Meta data)로서, 기계가 지식을 이해하고 처리할 수 있도록 데이터를 표현하는 온톨로지 기술이 필수적이다. 온톨로지는 자료와 자료사이의 유사관계 및 포함관계 등을 기술하며 자료를 이용한 추론의 기반을 제공한다.

온톨로지를 작성하는 언어로서 표현력과 출현 순서에 따라 XML, RDF, OWL 등이 존재한다. XML은 문서를 구조적으로 기술할 수 있는 문법을 제공하지만 문서의 의미를 제한할 수 있는 수단은 제공하지 않는다. RDF는 객체 및 객체 사이의 관계를 표현하는 자료의 모델이다. 이는 자원의 속성과 클래스를 표현할 수 있는 계층구조의 의미론을 제공한다. OWL은 DAML-OIL을 발전시킨 언어로서 객체에 대한 속성과 클래스에 대하여 기술할 수 있는 더 많은 어휘를 제공한다.

유비쿼터스 컴퓨팅이란 언제(Anytime) 어디서든(Anywhere) 누구나(Anyone) - (3A) - 자료를 처리할 수 있는 컴퓨팅 환경을 의미한다. 3A를 만족시키기 위하여 자동적인 상황 인지 및 처리기술이 필수적이라고 할 수 있다. 자동적인 상황인지 및 처리를 위하여 Context 인지 기술이 대두되고 있는데 Context란 사람, 장소, 논리·물리적 객체를 포함하는 객체의 상황에 대한

특징을 기술한 모든 정보라고 정의 할 수 있다[1]. Context의 인지를 통하여 컴퓨팅 능력이 없는 사물에 대하여 컴퓨팅이 가능하도록 논리적 공간으로의 정보 이양이 가능하다.

위에서 언급한 두 개의 기술을 이용하면 객체에 대한 정보와 객체에 대한 상황 정보를 취득할 수 있다. 그러나 주어진 정보를 가지고 기계가 자동적으로 처리하게 하는 것은 여전히 숙제로 남아있다. 전통적인 프로그래밍 언어를 이용하여 조건에 의한 반사적인 처리를 시킬 수 있는 것은 사실이지만 추론에 의해 지능적으로 처리하는 방법에 비해 적용 및 처리 범위가 좁고 서로 다른 기종의 컴퓨팅 환경에서 작동이 불가능한 단점이 존재한다.

본 논문에서는 온톨로지 기술 언어 중 하나인 OWL을 확장하여 상황인지를 위한 동사(Verb)를 표현하는 방법을 제안한다. 동사의 표현은 몇 개의 Context값의 집합으로 표현된다. 읽어 들인 Context 값들은 온톨로지로 표현된 상황인지 동사를 참조하여 조합되고 일어나고 있는 일에 대해서 추론이 가능하다.

기반 기술

OWL

본 논문에서는 여러 가지 생성언어 가운데 OWL(Web Ontology Language)를 확장하여 상황인지에 필요한 동사를 표현하였다. OWL은 XML 기반의 W3C 표준 온톨로지 생성언어로서 기계가 처리할 수 없는 정보를 단순 표현하는데 그치지 않고 정보의 내용을 기계가 직접 이해하고 처리할 수 있는 형태로 표현하도록 설계되었다[3].

OWL은 이전의 생성언어들보다 풍부한 어휘(Vocabulary)와 형식적 의미론(Formal semantics)을 포함하고 있기 때문에 기계 해석이 가능한 웹 자료를 작성하는데 뛰어나다. OWL은 표현력에 따라 OWL Lite, OWL DL, OWL Full 세 가지 형태로 나뉘어져 있다. 순서에 따라 후자는 전자의 어휘와 문법을 포함하는 관계를 가진다.

그러나 OWL은 온톨로지의 정의에서 내려진 범위 - 객체에 대한 기술 - 이상을 표현하지 않는다. 따라서 동사(verb)에 대한 표현 및 그를 통한 추론 기반이 부족하다.

Context

유비쿼터스 컴퓨팅 환경 하에서 상황인지를 위한 핵심적 키워드는 Context Awareness 이다. Context란 객체의 상황에 대한 특징을 기술한 모든 정보를 의미한다[4]. 여기서 객체는 사용자와

응용소프트웨어를 포함하며 사용자와 응용소프트웨어와의 상호작용도 포함한다. Context는 크게 컴퓨팅 Context, 사용자 Context, 물리적 Context로 나눌 수 있으며 개념적으로는 순서에 따라 논리적 공간에서의 Context, 사람에 대한 Context 그리고 물리적 공간에서의 Context로 분류할 수 있다.

언제 어디서든 누구나 정보기술을 활용할 수 있는 유비쿼터스 환경 하에서 Context는 컴퓨팅의 영역을 논리적인 공간(Cyber world)에서 물리적인 공간(Real world)까지 확대시키는 역할을 한다. 상황에 대한 특징을 담고 있기 때문에 그 정보들을 활용하여 현재 일어나고 있는 상황에 대한 추론이 가능한 기반을 제공한다고 할 수 있다.

대표적인 Context 인지 기술로서 GAIA프로젝트를 들 수 있는데 GAIA프로젝트는 Ontology서버를 이용하여 객체에 대한 정보를 수집하고 센서 네트워크를 이용하여 상황정보를 수집한다. 그리고 수집한 객체 정보 및 상황정보를 기반으로 추론하여 상위 Context(Higher level context)를 추출한다[5]. GAIA의 Context 모델은 주어 목적어 사이에 서술어 모델을 넣는 방식이지만 서술어 자체에 대한 정의 및 표현은 없기 때문에 의미적으로 완벽하게 상황인지를 한다고 보기는 어렵다.

관련 연구

CoOL

CoOL[6]은 독일 Ludwig-Maximilians 대학교에서 진행 중인 Context 인지 프로젝트의 일환으로 연구되고 있는 언어이다. CoOL은 온톨로지 언어인 DAML-S와 객체 표현이 가능한 로직 언어인 F-Logic으로 이루어져 있으며 온톨로지 언어를 이용하여 객체에 대한 정보 및 상황을 기술하고 F-Logic을 이용하여 추론엔진에 질의어(Query)를 전달하는 방식을 채택하였다. 이 연구의 주요한 목적은 추론의 기반이 뛰어난 온톨로지 언어를 이용하여 상황정보를 기술하고 상대적으로 부족한 질의어 처리[18]를 F-Logic으로 대체하는 것이다.

CoOL에서 Context 정보를 처리하기 위하여 ASC (Aspect-Scale-Context)모델을 사용하는데 ASC 모델은 객체에 대한 정보를 구분시켜 처리를 명확하게 해준다. Context를 DAML-S로 표현하기 위하여 CoOL은 DAML-S에서 기본적으로 제공하는 3개의 서비스 기술(Service description - Service profile, Service model, Service grounding) 외에 Service context를 추가하여 ASC 모델 기반으로 하는 정규적인 기술(Description)을 제공한다.

CoOL 은 온톨로지를 기반으로 관계값들을

추론하는 방법이 표준화된 웹 온톨로지 언어와 크게 다르지 않으며 온톨로지 질의 언어 또한 표준화되고 있어 효용성 면에서 탁월하지는 않다.

Fuzzy OWL

Fuzzy OWL은 OWL에서 기본적으로 제공하지 못하는 퍼지 집합을 웹 온톨로지로 표현하는데 목적을 둔 OWL의 확장 언어[8] 중 하나이다. Fuzzy OWL은 OWL로 기술된 온톨로지를 Fuzzy OWL로 변환하는 방법과 OWL로 기술할 수 없는 멤버십 함수를 기술하는 방법을 제공한다. OWL로 기술된 온톨로지를 Fuzzy OWL로 변환하는 4가지의 원칙은 다음과 같다.

- Rule1) 모든 OWL의 클래스는 Fuzzy OWL에 매핑(Mapping)된다.
- Rule2) 모든 Subsumption과 Equivalent는 Fuzzy OWL에서 Fuzzy Subsumption과 Equivalent이다.
- Rule3) OWL에서의 모든 클래스는 Fuzzy OWL에서 Restriction value = 1로 매핑된다.
- Rule4) OWL에서의 속성은 Fuzzy OWL에서 Restriction value = 1로 매핑된다.

Fuzzy OWL은 FDL이라는 이름의 네임스페이스를 독자적으로 정의하였는데 FDL을 이용하여 Fuzzy constraint를 표현하면 다음과 같다.

Fuzzy constraints	Examples for FOWL
$A(a) \geq n$	<code><fdl:individual fdl:name="a"> <fdl:membershipOf rdf:resource="#A"/> <fdl:moreOrEquivalent fdl:value=n/> </fdl:individual></code>
$A(a) \leq n$	<code><fdl:individual fdl:name="a"> <fdl:membershipOf rdf:resource="#A"/> <fdl:lessOrEquivalent fdl:value=n/> </fdl:individual></code>
$R(a, b) \geq n$	<code><fdl:individual fdl:name="a" fdl:name="b"> <fdl:membershipOf rdf:resource="#R"/> <fdl:moreOrEquivalent fdl:value=n/> </fdl:individual></code>
$A(a) > n$	<code><fdl:individual fdl:name="a"> <fdl:membershipOf rdf:resource="#A"/> <fdl:moreThan fdl:value=n/> </fdl:individual></code>
$A(a) < n$	<code><fdl:individual fdl:name="a"> <fdl:membershipOf rdf:resource="#A"/> <fdl:lessThan fdl:value=n/> </fdl:individual></code>
$R(a, b) \leq n$	<code><fdl:individual fdl:name="a" fdl:name="b"> <fdl:membershipOf rdf:resource="#R"/> <fdl:lessOrEquivalent fdl:value=n/> </fdl:individual></code>
$R(a, b) > n$	<code><fdl:individual fdl:name="a" fdl:name="b"> <fdl:membershipOf rdf:resource="#R"/> <fdl:moreThan fdl:value=n/> </fdl:individual></code>
$R(a, b) < n$	<code><fdl:individual fdl:name="a" fdl:name="b"> <fdl:membershipOf rdf:resource="#R"/> <fdl:lessThan fdl:value=n/> </fdl:individual></code>

그림 1) Fuzzy constraints

기본 개념

전통적인 프로그래밍 방법으로 표현할 수 있는 동사(Verb)는 대입(Is), 동치(Equivalence), 대소관계(More than, less than)등이다. 현재 이와 같은 논리적인 관계를 이용하여 전산학적으로 대두되는 거의 모든 문제를 해결하고 있다. 그러나 이 표현들은 객체의 행동 자체를 정의할 수는 없다. 다시 말해, 지금까지의 프로그래밍 방법은 입력데이터와 명령이 저장은 되지만 컴퓨터가 의미적으로 해석하지 못하는 명령을 기반으로 하고 있다.

사람은 현실세계에서 일어나는 일들을 인지하고 취해야 할 행동을 추론하는 시스템을 갖추고 있다. 예를 들어 ‘불이 났다’ 라는 상황 인식을 하는 과정을 통해 살펴보면, ‘불’ 이 무엇인지 알고 있고 그 불의 크기가 얼마인지 어디에서 났는지 등을 파악하여 사람에게 위험하게 난 불 만을 ‘불이 났다’ 라고 판단한다. 그리고 ‘불이 났다’ 라는 정보를 접하게 되면 가장 효율적인 해결방법을 구하여 그에 맞춘 행동을 취한다. 즉, ‘불이 나다’ 라는 동사의 개념은 불이란 객체에 대한 의미와 그를 둘러싼 상황정보의 집합관계로서 표현이 가능하며 표현된 ‘불이 났다’ 와 그와 관계된 취할 수 있는 행동의 상관관계를 이용해 특정 명령을 추론해낼 수 있다.

본 논문에서 제시하는 상황인지를 위한 동사의 표현은 위에서 설명한 바와 같이 온톨로지로 표현한 객체 정보와 Context인지 기술을 이용한 상황정보의 집합관계로서 나타냈다.

시나리오

지능형 가정 안전 관리 제품인 A가 있다고 가정하자. A의 기능은 실시간으로 가정에서 생기는 일들을 파악하고 안전관리를 위해 설치된 장치들에 그에 따르는 명령을 내리는 것으로 한다. 집주인 K씨가 가스렌지를 실수로 켜놓고 외출을 나갔고 그로 인해 가정에 불이 났을 때 제품 A는 가정에 불이 난 것을 자동적으로 파악 해야 하고 또한 불을 끄는 장치를 가동 해야 한다.

이 때, A가 센서를 통해 받아들인 정보는 다음과 같다.

```

home.mainWindow = closed
home.gasRange.gas = outFlow
home.gasRange.fire = on
home.TV = on
home.family.daughter = in
home.temperature = 72.3
home.air.pollutionLevel = 7

```

그림 2) 센서를 통해 받아들인 정보

위 정보를 종합해 볼 때 가스 렌지에서 가스가 흘러나오고 있고 내부의 온도가 상당히 높으며 공기오염 정도가 크므로 불이 난 것으로 판단하는 것이 가능하다.

전통적인 방법을 이용한 해결

전통적인 프로그래밍 기법을 이용하여 위 시나리오에서 제시한 문제를 해결하려면 각 센서를 통해 받아들인 정보를 변수에 대입하고 변수 값의 논리적인 관계를 이용하여 해당 장치에 명령을 내리는 방법을 택할 수 있다. 그 예시는 아래와 같다.

```

.. 중략 ..

IF (home.temperature >= 70 and
home.air.pollutionLevel > 7)
    do(fireExtinguisher);
ELSE IF (home.air.pollutionLevel >= 5)
    do.open(home.mainWindow);

.. 중략..

```

그림 3) 전통적인 방법을 이용한 해결

위와 같은 방법은 사용하기 간단한 장점이 있지만 새로운 논리를 세울 때마다 새로 컴파일해야 하기 때문에 가정에서 일어날 수 있는 수많은 변수와 변화에 대처하기 힘들다는 단점이 있으며 이용될 기기마다 같은 언어, 같은 시스템을 사용해야 하기 때문에 확장성, 호환성의 문제가 생길 수밖에 없다.

제안하는 방법을 이용한 해결

본 논문에서 제안하는 방법은 상황정보처리를 위한 동사를 전산적으로 처리 가능하도록 온톨로지 기술언어인 OWL을 이용해서 표현하고 추론엔진을 통해 취해야 할 행위를 추론하는 방법이다. 이 방법을 사용하면 전통적인 해결방법을 이용한 방법과 비교했을 때 컴파일을 그 때 그 때 할 필요가 없기 때문에 변화에 능동적이며 서로 다른 기종 간에 생길 수 있는 호환성의 문제를 효과적으로 해결할 수 있다.

제시된 상황정보들이 불이 난 것과 관계된 정도를 0~1사이의 값으로 표현하면 다음과 같다.

```

home.mainWindow      : 0.1
home.gasRange.gas    : 0.3
home.gasRange.fire   : 0.3
home.TV              : 0
home.family.daughter : 0
home.temperature     : 0.8
home.air.pollutionLevel : 0.9

```

그림 4) 0~1 사이의 값으로 표현

위에서 표현된 불이 난 것과 관계된 정도를 이용하여 불이 나다(Fire)라는 동사를 표현 가능하다. 그를 OWL의 Fuzzy 확장 언어인 FOWL[11]로서 표현하면 다음과 같다.

```

....중략....

<fdl:individual fdl:name="fire">
  <fdl:membershipOf>
    <fdl:Restriction/>
    <fdl:onProperty rdf:resource="#homeContext"/>
    <fdl:someValuesFrom>
      <fdl:class>
        <fdl:unionOf rdf:parseType="Collection">
          <fdl:Class rdf:about="#homeTemperature"/>
            <fdl:moreOrEquivalent fdl:value=0.8/>
          <fdl:Class rdf:about="#airPolluteLevel"/>
            <fdl:moreOrEquivalent fdl:value=0.9/>
          .....
        </fdl:intersectionOf>
      </fdl:class>
    </fdl:someValuesFrom>
  </fdl:membershipOf>
</fdl:individual>

....중략....

```

그림 5) FOWL 을 이용한 표현

위와 같이 표현된 동사들을 이용하여 문제를 해결할 때, 자료의 흐름을 계층적으로 나타내면 센서 등을 통해 상황정보를 받는 단계(Context), 상황정보들의 집합관계를 이용하여 현재 일어나고 있는 일을 추론하는 단계(Do), 일어난 일에 대해 취해야 할 행위를 추론하는 단계(Do'), 추론된 행위를 실행하는 단계(Service)로 나눌 수 있다.

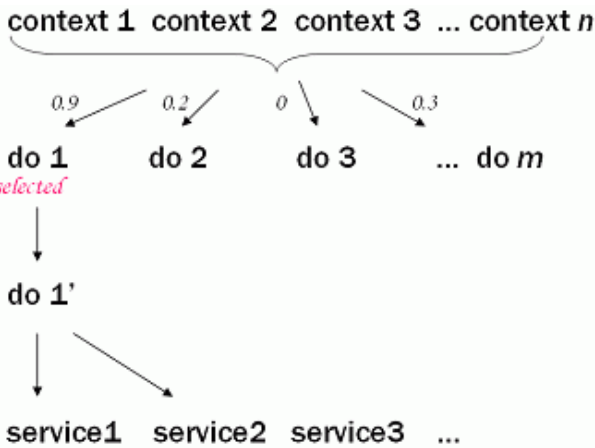


그림 6) Context 추론 단계

Context로서 받아들여지는 자료들은 현실세계 뿐 아니라 가상세계에서 받아들여지는 모든 상황정보를 포함 가능 하다. 이 자료들은 객체에 대한 기술이 된 온톨로지와 상황인지 동사에 대한 기술이 된 온톨로지를 가진 추론엔진에 의해 현재 일어나고 있는 상황정보로서 가공이 된다. 이 때 참조 규칙(Inference rule)을 가진 지식 저장소(Knowledge base)를 참조한다. 파악된 상황정보는 명령을 내릴 수 있는 서비스들과 상호 접속을 통해 현실/가상 세계에 자동적 처리를 실행한다.

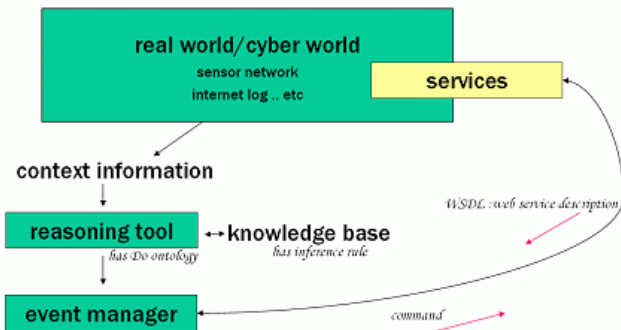


그림 8) 서비스 흐름

결론

유비쿼터스 컴퓨팅 환경이 점차 커져감에 따라서 Context 정보의 공급이 늘어나고 있다. 또한 서로 다른 언어와 기종간 호환성이 더더욱 중요해지고 있다. 이에 따라 웹 서비스의 보급 및 시맨틱 웹의 연구 및 보급이 중요한 문제로 대두되는 것이 사실이다. 그리고 그와 더불어 그를 기반으로 자동적으로 상황인지 및 처리하는 기술이 필요하다.

시맨틱 웹 및 온톨로지 기술은 객체에 대한 의미적인 정보를 컴퓨터가 직접적으로 처리할 수 있도록 고안되었다. 현재까지 제시된 온톨로지 표현법은 객체에 대한 표현을 그 목적으로 하고 있다. 그러나 객체에 대한 표현만으로는 모든 지식을 전산적으로 처리하는데 부족한 면이 있다. 따라서 본 논문에서는 시맨틱 웹 기술을 이용하여 상황인지에 필요한 동사를 기술하는 것에 목표를 두었으며 일반동사를 계산 가능한 형태로 표현하여 컴퓨터를 이용한 지식 표현의 범위를 넓히는데 노력하였다.

본 논문에서는 OWL의 퍼지 확장언어인 FOWL[8]을 이용하여 상황인지 동사를 표현하였다. 이 작업은 객체에 대한 정보로 국한되던 지식의 표현범위를 객체의 행동을 정의하는 동사에까지 그 범위를 확장한 것이다.

표현된 동사를 이용한 상황인지 시스템을 구축하기 위해서는 지식저장고(Knowledge base)에 참조 규칙을 어떻게 적용하고 구현하느냐에 따라 그 결과가 많이 달라질 수 있다. 향후 작업에서는 실제로 적용 가능한 예제를 더 풍부하게 만들고 만들어진 온톨로지를 활용한 추론 시스템을 구현 할 것이다. 추론 시스템이 완성되면 서로 다른 언어와 서로 다른 기종의 시스템 환경에서 능동적이고 확장 가능한 상황인지 및 대처 시스템을 구축하는 것이 가능하다.

References

- [1] Providing Architectural Support for Building Context Aware Applications. Anind K. Dey, GATECH, Nov. 2000.
- [2] Ontology-based methodology for e-service discovery. Devis Bianchinia, Valeria De Antonellisa, Barbara Pernicib, Pierluigi Plebani.
- [3] <http://www.w3c.org>
- [4] The Context Toolkit: Aiding the Development of Context-Enabled Applications Daniel Salber, Anind K. Dey and Gregory D. Abowd. In

- the Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI '99), Pittsburgh, PA, May 15-20, 1999. pp. 434-441.
- [5] Gaia: A Middleware Infrastructure to Enable Active Spaces. Manuel Román, Christopher K. Hess, Renato Cerqueira, Anand Ranganathan, Roy H. Campbell, and Klara Nahrstedt, In IEEE Pervasive Computing, pp. 74-83, Oct-Dec 2002.
- [6] CoOL: A Context Ontology Language to Enable Contextual Interoperability. Thomas Strang, Claudia Linnhoff-Popien, and Korbinian Frank. German Aerospace Center (DLR), Oberpfaffenhofen
- [7] On the Representation of Context ROBERT STALNAKER. Department of Linguistics and Philosophy, Massachusetts Institute of Technology, Cambridge
- [8] Extending OWL by Fuzzy Description Logic. Mingxia Gao, Proceeding of the 17th IEEE international conference on tools with artificial intelligent (ICTAI'05)
- [9] Fuzzy Querying. Giorgos Stamou. National Technical University of Athens, Greece
- [10] Well-Founded Semantics for Description Logic Programs in the SemanticWeb. Thomas Eiter, Thomas Lukasiewicz, Roman Schindlauer, and Hans Tompits
- [11] Extending OWL by Fuzzy Description Logic Mingxia Gao, Chunnian Liu. Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology. Beijing University of Technology
- [12] Description Logic Programs: Combining Logic Programs with Description Logic Benjamin N. Grosz, Ian Horrocks, Raphael Volz, Stefan Decker
- [13] Web Ontology Language OWL and Its Description Logic Foundation. Zuo zhihong & Zhou mingtian. School of Computer Science and Engineering. University of Electronic Science and Technology of China
- [14] Reducing OWL Entailment to Description Logic Satisfiability. Ian Horrocks. Department of Computer Science. University of Manchester, Manchester, UK
- [15] Fuzzy Matchmaking for Web Services. Kuo-Ming Chao, Muhammad Younas, Chi-Chun Lo, Tao-Hsin Tan
- [16] Extensible soft semantic web services agent. Haibin Wang, Yan-Qing Zhang. Rajshekhar Sunderraman. Soft Comput (2005)
- [17] Reaching consensus: A moderated fuzzy web services discovery method. Chun-Lung Huang a, Chi-Chun Lo a, Kuo-Ming Chao b, Muhammad Younas. Information and Software Technology xx (2006) 1- 14
- [18] Borgida, A : On the relative expressiveness of description logics and predicate logics. Artificial Intelligence 82 (1996) 353-367