

시맨틱 웹 기반의 비교 구매 시스템

이호경*, 유영훈*, 조근식*

*인하대학교 컴퓨터공학부

machi@eslab.inha.ac.kr, yhyu@eslab.inha.ac.kr, gsjo@inha.ac.kr

Comparison Shopping System based on Semantic Web

Ho-Kyoung Lee*, Young-Hoon Yu*, Geun-Sik Jo*

*School of Computer Science & Engineering, Inha University

요 약

기존 쇼핑물의 컨텐츠는 확장 및 통합이 고려되지 않은 기반 기술로 구성되어 있어서, 비교 구매 등록시, 별도의 제품 메타 데이터, 상품 목록 리스트를 생성해야 한다. 이렇게 작성된 메타 데이터 역시 의미 (Semantics)를 지니고 있지 않기 때문에, 단순한 텍스트 검색 이상을 제공하기 힘들고, 이로 인해 구매자는 불필요한 검색 결과 속에서 많은 시간을 낭비해야 하는 실정이다. 또한, 각자 나름의 정보 체계를 지닌 비교 구매 사이트들 간의 정보 공유가 되지 않기 때문에 보다 많은 제품을 비교 하고 싶어 하는 구매자의 요구를 수용하지 못하고 있다.

본 논문에서는 시맨틱 웹 기반 기술인 daml+oil을 사용해서 개별적인 쇼핑물의 도메인 온톨로지를 구성하여 개념 정의 및 구조를 명시하고, 이 기반위에 DAML 마크업 인스턴스 데이터를 작성하여, DAMLJessKB를 통해 시맨틱적 요소를 추출하고, 추론엔진인 JESS에 fact로 넘겨 시맨틱 추론을 수행하도록 하였다. 그 결과 별도의 메타 데이터 생성 없이 분산된 쇼핑몰간의 데이터를 통합할 수 있고, 컴퓨터가 개념들 간의 의미를 이해할 수 있게 됨으로써 추론을 통한 검색 서비스를 제공하는 시스템을 설계 구현하였다.

1. 서론

기존 쇼핑물의 컨텐츠는 확장 및 통합이 고려되지 않은 기반 기술로 구성되어 있어서 비교 구매 등록시 별도의 제품 메타 데이터, 상품 목록 리스트를 생성해야 한다. 이렇게 작성된 메타 데이터 역시 의미를 지니고 있지 않기 때문에 단순한 텍스트 검색 이상을 제공하기 힘들고 이로 인해 구매자는 불필요한 검색 결과 속에서 많은 시간을 허비해야 하는 실정이다. 또한, 각자 나름의 정보 체계를 지닌 비교 구매 사이트들 간의 정보 공유가 안되기 때문에 보다 많은 제품을 비교 하고 싶어 하는 구매자의 요구를 수용하고 있지 못하고 있다.

같은 맥락으로 웹상에서 도메인간 정보 공유의 문제점을 해결하기 위해서 시맨틱 웹이 제안되었고, 온톨로지, 메타 데이터, 에이전트, 추론 엔진 등의 구성 요소를 통해 웹상에서 의미의 이해를 수반한 정보 공유가 가능해 졌다 [6].

비교 구매 시스템에서도 쇼핑물간의 정보 공유를 위해 지식을 이루는 개념의 명세인 온톨로지 구성과 이들 개념을 기반으로 데이터를 표현함으로써 컴퓨터가 의미를 이해하고 추론을 비롯한 응용을 할 수 있게 되어 기존의

HTML을 비롯한 텍스트 정보 표현, 검색, 공유, 관리의 문제점을 극복할 수 있다 [3][7].

본 논문은 이러한 시맨틱 웹 기반의 비교 구매 사이트를 제안함으로써 기존의 비교 구매 사이트의 문제점을 해결해 나가고자 한다.

2. 관련연구

2.1. 비교 구매 시스템

비교 쇼핑 시스템은 사용자가 특정 상품을 찾고자 할 때 에이전트가 사용자를 대신하여 인터넷에 존재하는 상점을 검색하여 상품별로 해당 상품을 비교하는 시스템을 말한다. 이 기능으로 인해 사용자는 인터넷을 직접 검색하는 수고와 시간을 줄일 수 있는 장점을 얻을 수 있다. 개발된 시스템의 방식으로는 BargainFinder 와 Jango 같은 시스템을 예로 들 수 있다 [9]. BargainFinder 는 비교 쇼핑 시스템의 원조라 할 수 있는 시스템으로 미리 등록된 9개의 인터넷 CD 상점에서 사용자가 원하는 상품의 가격을 비교하는 기능을 제공한다. 하지만 이 시스템은 미리 정해진 인터넷 상점만 비교 할 수 있다는 단점과 함께 별도의 메타 정보를 생성해야 하는 단점, 또한 가격에 중심을 둔 비교를 제공하기 때문에 제품 사양에 대한 구매자 욕구를 충족하기 어렵다는 문제점을 가지고 있다.

한편 Jango는 상점을 등록할 필요 없이 사용자의 요구

※ 본 연구는 한국과학재단 지정 인천대학교 동북아전자물류연구센터의 지원에 의한 것임.

에 따라 상점을 분석할 수 있는 기능을 추가하여 이러한 문제를 해결하려고 하였으나 상점의 분석 성공률이 50% 이하로 실질적으로 문제를 해결하지는 못해서 실제적인 사용이 어렵다. HTML source를 분석을 통한 정보 추출의 대안으로 XML을 제시하기도 하지만, 단순한 구조 정보 속에서 의미를 이해할 수 없기 때문에 다양한 개념상의 상품 정보를 공유하는데 한계가 있다.

현재 운영되는 국내외 비교 구매 사이트는 BargainFinder와 같은 방법으로 서비스를 제공하고 있다.

2.2. Semantic web

시맨틱 웹이란 데이터에 의미를 부여하여 컴퓨터가 이해할 수 있는 언어로 만들어 컴퓨터에 의해 처리 될 수 있도록 고안된 차세대 웹의 비전이다 [1]. 시맨틱 웹을 이루는 핵심적인 구성요소는 도메인 지식을 공유하기 위해서 개념의 명세를 위한 온톨로지 (RDFs, daml+oil, OWL), 그에 따라 구조화된 데이터를 표현하기 위한 메타 데이터 (RDF, DAML), 자동적이고 지능적인 기능을 수행하기 위한 에이전트, 개념화 되고 구조적인 데이터 기반의 추론 엔진이라고 할 수 있다.

2.2.1. DAML

XML 인코딩에 기반한 시맨틱 웹 언어인 RDF와 RDFs는 실제 복잡한 지식 표현 언어로서는 제한을 갖는다. 예를 들어 속성의 속성이라고 할 수 있는 unique, transitive, inverse, disjoint classes 등의 정의와 같은 추가적인 기능이 미흡하다. DAML (DARPA Agent Markup Language)은 인터넷 상의 기계들간 정보를 이해가능하도록 하기 위한 시맨틱 마크업 언어이다. DAML은 기존의 RDF와 RDFs에 추가적인 시맨틱 기능을 결합함으로써 단점을 극복하고자 하는 목적으로 시작되었다.

2.2.2. Ontology

시맨틱 웹은 컴퓨터가 공유되는 데이터들의 광범위한 개념을 이해하기 위해 정형적인 (formal) 온톨로지에 기반한 구조적인 데이터에 크게 의존하게 된다. 특정 도메인내의 지식들을 개념화 하고 이를 명세화 (specification)함으로써 애플리케이션간의 정보 공유와 재사용을 도울 수 있다.

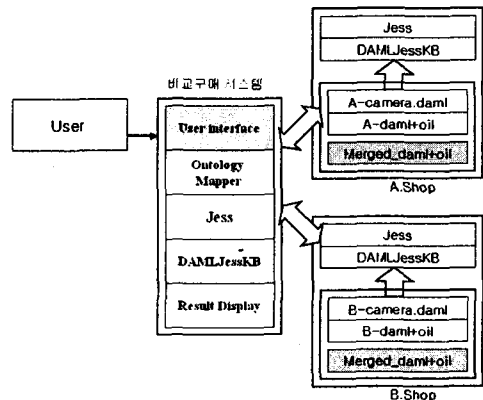
DAML은 OIL (Ontology Inference Layer)와 함께 온톨로지를 기술하는데 사용 될 수 있다. daml+oil은 분류, 특성 제한, W3C XML Schema Definition Language가 제공하는 타입 정의 기반의 데이터 타입 편의 등의 지원을 포함한다 [3].

2.2.3. DAMLJessKB

DAMLJessKB는 시맨틱 웹 추론을 위한 도구 (reasoning Tool)이다. DAMLJessKB는 DAML 파서인 Jena를 통해서 DAML 문서를 로딩하고 파싱하게 된다 [2]. 이렇게 생성된 DAML Triples 스트림은 DAMLJessKB를 통해서 추론엔진인 JESS (Java Expert System Shell)의 fact로 넘겨지게 되고 JESS는 미리 정의해둔 daml+oil rules, domain rules을 가지고 추론을 수행하게 된다 [5].

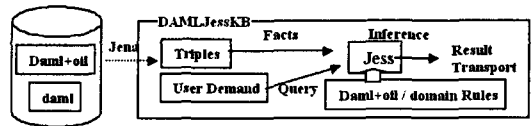
3. 시스템 구조

본 논문이 제시하는 시맨틱 웹 기반의 비교 구매 시스템 구조도는 [그림 1]과 같다. 각 도메인에서는 daml+oil기반으로 각각의 상품에 맞는 온톨로지를 생성하고 그에 따른 인스턴스 DAML문서를 만들게 된다. 시맨틱 비교 구매 시스템으로 정보를 통합 할 때 비교 구매 시스템의 Ontology Mapper를 통해 여러 도메인상의 온톨로지를 매핑하게 되고 해당 상품 개념간의 관계를 정의한 통합된 온톨로지 (Merged_daml+oil)를 얻어내게 된다.



[그림 1] 시맨틱 웹 기반의 비교 구매 시스템 구조도

구매자는 비교 구매 시스템의 사용자 인터페이스를 통해 원하는 조건의 상품을 검색하게 되고 이 요구사항 항목을 각 도메인에 보내어 통합된 온톨로지 (Merged_daml+oil)와 인스턴스 문서를 DAMLJessKB를 통해 JESS의 fact로 집어넣고 daml+oil rules/domain rules을 기반으로 추론을 수행하게 된다. 그 결과는 다시 DAMLJessKB를 통해 DAML포맷 형태로 얻게 된다.



[그림 2] DAMLJessKB를 통한 JESS 추론 과정

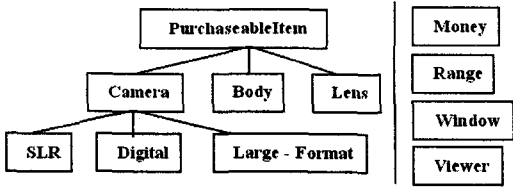
여러 도메인에서 수행한 결과는 비교 구매 시스템의 Result Display 모듈을 통해 최종 구매자에게 전송 된다.

3.1. 사용자 인터페이스

구매자는 웹이나 애플리케이션의 사용자 인터페이스를 통해서 자신의 구매 제품, 검색 항목을 비롯한 세부 검색 정보를 입력하게 된다. 이들 항목을 통해 JESS Query 생성기에서 쿼리를 생성 및 실행하게 되고 그 결과를 구매자에게 제공하게 된다.

3.2. 온톨로지 및 메타 데이터 생성

본 논문에서는 비교 구매 도메인으로써 카메라에 대한 daml+oil기반의 온톨로지를 구성하였다[4].



[그림 3] Camera 온톨로지 구조

위의 [그림 3] 구조도에서 SLR은 Camera의 subClass라고 정의되어 있다. 쿼리시 Camera 클래스를 선택한다면 SLR 클래스 역시 검색 대상이 된다. 따라서 SLR을 검색시 Digital, Large-Format과 같은 카메라의 subClass 항목들도 선택될 것이다.

예를 들어, 렌즈를 통한 화상 그대로 뷰파인더에 비친다는 개념은 Camera클래스의 viewFinder속성 값이 ThroughTheLens란 값인 경우 <daml:intersectionOf>를 사용하여 [그림 4]와 같이 SLR을 정의하였다.

```
<daml:Class rdf:ID="SLR">
  <daml:intersectionOf rdf:parseType="Collection">
    <daml:Class rdf:about="#Camera"/>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#viewFinder"/>
      <daml:hasValue rdf:resource="#ThroughTheLens"/>
    </daml:Restriction>
  </daml:intersectionOf>
</daml:Class>
```

[그림 4] Camera 온톨로지 (daml+oil) 의 예

이상으로 작성된 온톨로지의 개념 및 구조를 기반으로 DAML 인스턴스 메타 데이터를 작성하였다.

3.3. 온톨로지 매핑

각 도메인의 콘텐츠 통합과정에서 용어의 충돌을 막기 위해 서로 다른 도메인 온톨로지상의 개념간 관계를 매핑하는 작업이 필요하다. 본 논문에서는 온톨로지 매핑시 KAON(The Karlsruhe Ontology and Semantic Web Tool Suite) Ontology Mapper를 사용하여 각 도메인간의 용어의 관계를 설정하였다.

예를 들어, A도메인의 경우 카메라 렌즈의 크기에 대한 속성을 "size"라고 사용하고 동일한 개념을 B도메인의 경우 "focal-length"를 사용한다면 JESS의 추론과정에서 이들 속성을 동일한 개념으로 간주할 수 있어야 한다. daml+oil 스펙에서는 "samePropertyAs"를 사용해서 아래와 같이 "focal-length"와 "size"간의 관계가 동일함을 표현할 수 있다. 마찬가지로 조리개의 수치를 나타내는 "f-stop"과 "aperture" 역시 마찬가지이다.

```
<daml:DatatypeProperty rdf:ID="focal-length">
  <daml:samePropertyAs rdf:resource="#size"/>
  <rdfs:domain rdf:resource="#Lens"/>
  <rdfs:range rdf:resource="#xsd:string"/>
</daml:DatatypeProperty>
<daml:DatatypeProperty rdf:ID="f-stop">
  <daml:samePropertyAs rdf:resource="#aperture"/>
  <rdfs:domain rdf:resource="#Lens"/>
  <rdfs:range rdf:resource="#xsd:string"/>
</daml:DatatypeProperty>
```

[그림 5] 매핑된 온톨로지

3.4. DAMLJessKB

3.4.1. daml+oil Rules

JESS는 daml+oil 표기법을 이해하지 못하기 때문에 DAMLJessKB를 통해서 Rule로 변환해 주어야 한다.

DAMLJessKB는 daml+oil로 구성된 온톨로지상의 의미를 fact로 JESS에 제공하게 된다. [그림 6]은 자식의 인스턴스는 부모의 인스턴스라는 daml+oil의 시멘틱을 rule로 표현한 것이다. 아래는 subclass의 관계이며 자식의 인스턴스가 발견되면 해당 rule이 활성화 되어 부모의 인스턴스이기도 하다는 fact를 추가해 주는 rule이다.

```
(defrule subclass-instances
  (PropertyValue &rdfs:SubClassOf ?child ?parent)
  (PropertyValue &rdfs:type ?instance ?child)
  =>
  (assert
    (PropertyValue &rdfs:type ?instance ?parent)
  )
)
```

[그림 6] subclass-instances Rule

3.4.2. Domain Rules

DAMLJessKB는 특정한 도메인의 rule을 적용할 수 있다. 도메인 rule을 통해서 카메라 쇼핑몰을 보다 세부적으로 정의할 수 있다.

카메라의 가격은 몸체의 가격과 렌즈의 가격의 합이라는 rule의 기본이 되는 카메라란 한개의 몸체를 갖는다는 domain rule 정의는 [그림 7]과 같다.

```
(defrule rule01
  (PropertyValue &rdfs: type ?n &machi:#Body)
  ?t1 <- (PropertyValue &machi:Camera ?n ?a)
  ?t2 <- (PropertyValue &machi:#Camera ?n ?b)
  ?t3 <- (PropertyValue &rdfs:type ?a &xmlns:string)
  ?t4 <- (PropertyValue &rdfs:#type ?b &xmlns:string)
  ?t5 <- (PropertyValue &rdfs:value ?a ?name1)
  ?t6 <- (PropertyValue &rdfs:value ?b ?name2)
  (test (> (call ?t2 getFactId) (call ?t1 getFactId)))
  (test (> (call ?t4 getFactId) (call ?t3 getFactId)))
  (test (> (call ?t6 getFactId) (call ?t5 getFactId)))
  =>
  (printout WSTDOUT "카메라는 한개의 Body를 갖는다." crlf)
  (printout WSTDOUT ?n "의 부분 " ?name2 crlf)
  (printout WSTDOUT "Retracting fact:" (call ?t2 getFactId) crlf)
  (printout WSTDOUT "Retracting fact:" (call ?t4 getFactId) crlf)
  (printout WSTDOUT "Retracting fact:" (call ?t6 getFactId) crlf)
  (retract ?t2)
  (retract ?t4)
  (retract ?t6) )
```

[그림 7] JESS Domain Rule 의 예

3.4.3. JESS Query

지금까지의 facts, rules를 기반(Knowledge Base)으로 JESS의 defquery를 통해서 원하는 정보를 얻을 수 있다.

예를 들어, 카메라 몸체중 가격이 20만원 이상인 리스트를 얻기 위한 JESS Query 는 [그림 8]과 같다.

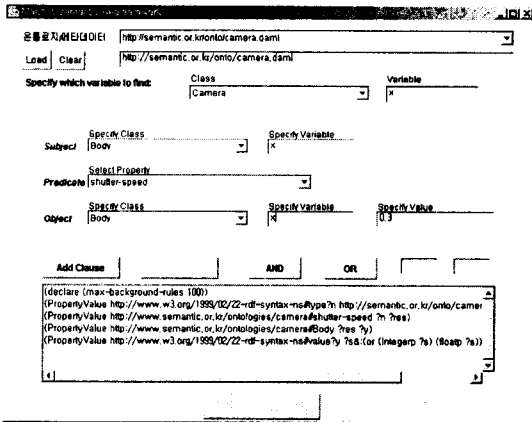
```
(defquery search2 (declare (max-background-rules 100))
(PropertyValue &rdf:type ?n &mach:Body)
(PropertyValue &mach:cost ?n ?res)
(PropertyValue &rdf:value ?res
?s&:(or (integerp ?s) (floatp ?s))&(>= (float ?s) 200000))
)
```

[그림 8] JESS Query의 예

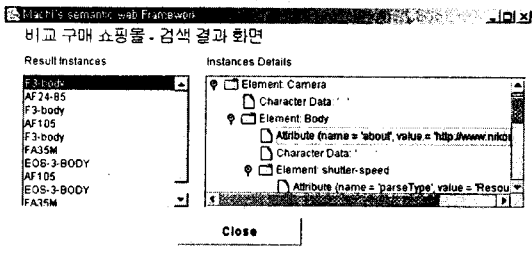
4. 구현 및 실험 평가

4.1. 구현

지금까지 구축된 시스템을 통해서 구매자가 조리개 4.5 -5.6, 셔플 스피드 1/500 sec -1.0 sec 이면서 삼성 제품이 아닌 150만원이하의 카메라 검색하고자 할때 [그림 9]는 쿼리 생성 인터페이스이다.



[그림 9] 쿼리 생성 인터페이스



[그림 10] 결과 인터페이스

위의 [그림 10]은 해당 쿼리의 결과 set이 왼쪽 리스트로 보여지고 해당 항목 선택시 제품의 세부 정보를 트리 구조로 보여주고 있다.

4.2. 실험 평가

본 논문에서는 제안하는 시스템으로 검색되어진 결과가 얼마나 정확하게 예측 되어졌는가를 알아보기 위해 정보 검색(Information Retrieval)에서 사용되어지는 재현율(Recall) 사용하였다 [8].

검사 데이터(test set)에 대한 매칭 데이터 (hit set)의 비율인 recall은 다음과 같다.

$$Recall = \frac{|Test \cap result|}{|Test|}$$

(Test : 관련있는 검사 데이터의 수, result : 검사 결과)

제안된 시스템은 두개의 소평물상의 분산된 샘플 데이터 200개에서 17개의 가능한 데이터 set중, 15개의 데이터 set을 검색할 수 있었고 특별한 메타 데이터 없이 0.88이 란 recall 수치를 나타내었다. 반면 별도의 메타 데이터 생성하지 않은 기존의 소평물 정보를 통합 시 용어의 충돌, 표현 양식의 차이로 인하여 쿼리 형태에 따라 recall 수치가 불규칙 적이었고, 평균적으로 현저히 낮았다. 이는 도메인 수가 증가 하거나 쿼리의 속성수가 증가 할수록 두드러지게 나타났다.

5. 결론 및 향후 연구

본 논문은 시멘틱 웹 기반의 비교 구매 시스템을 구현하기 위해서 DAML 메타데이터, daml+oil 온톨로지 그리고 추론 엔진 (JESS, DAMLJessKB)을 사용하여 별도의 메타 데이터를 생성하지 않고도 분산된 도메인간 데이터를 통합할 수 있었다. 또한, 컴퓨터가 이해 가능하고 개념간의 관계를 매칭할 수 있기 때문에 보다 세부적인 제품 특성을 표현가능 했으며 추론 검색을 통해 구매자가 원하는 제품을 정확히 보여줄 수 있었고 DAML을 이해하는 검색엔진의 경우 해당 제품을 정확히 검색해 줄 기반을 마련하였다.

향후 연구로는 보다 많은 도메인 통합시 지능형 에이전트를 통해 자동화 하고자 하며, Ontology Learning을 기반한 자동화된 온톨로지 매핑의 연구가 필요하다.

또한 RuleML을 사용해 JESS에 적용했던 Rule들을 유사한 다른 도메인과 공유하고자 하며, 최근 이슈화 되고 있는 Mobile Web 환경으로의 시스템 확장이 필요하다.

참고문헌

- [1] Tim Berners-Lee, James Hendler and Ora Lasilla, "The Semantic Web", The Scientific American, May 2001.
- [2] Kopena, J. and Regli, W., 'DAMLJessKB: A tool for rea-soning with the Semantic Web.' IEEE Intelligent Systems18(3), May/June, 2003
- [3] Cost, R. S., Finin, T., Joshi, A., Peng, Y., Nicholas, C.,Soboroff, I., Chen, H., Kagal, L., Perich, F., Zou, Y., and Tolia, S. 'ITTALKS: A Case Study in the Semantic Web and DAML+OIL.' IEEE Intelligent Systems 17(1):40-47,2002.
- [4] Roger L. Costello,David B. Jacobs "OWL tutorial" <http://www.xfront.com/owl/>
- [5] E. Friedman-Hill, Jess: The Rule Engine for the Java Platform, 1995, <http://herzberg.ca.sandia.gov/jess>.
- [6] U. Shah et al., "Information Retrieval on the Semantic Web," Proc. ACM Conf. Information and Knowledge Management,ACM Press, 2002.
- [7] Kapil Rajendra Dukle. A Prototype Query-Answering Engine Using Semantic Reasoning.
- [8] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl "Analysis of e-commendation Algorithms for E-Commerce" Minnesota, 2000
- [9] R. Doorenbos, O. Etzioni, and D. Weld. A Scalable Comparison-Shopping Agent for the World-Wide Web. In Proceedings of the First Internation Conference on Autonomous Agents, 1997