형식개념분석기법 기반의 온톨로지 분석도구(OWL Analyzer)의 개발1)

김동순*, 황석형**, 김홍기***, 양경모**
*선문대학교 전자계산학과, **선문대학교 컴퓨터정보학부,
***서울대학교 의생명 지식공학 연구실
e-mail: ds8025@gmail.com

On developing OWL Analyzer based on Formal Concept Analysis

Dong-Soon Kim*, Suk-Hyung Hwang**, Hong-Gee Kim***, Kyung-Mo Yang**
*Dept of Computer Science, SunMoon University
**Division of Computer and Information Science, SunMoon University
***Biomedical Knowledge Engineering Laboratory, Seoul National University

요 약

온톨로지는 시멘틱 웹의 상호운용성에 있어서 가장 중요한 역할을 하고 있으며, 다양한 분야에서 지식의 공유 및 재사용을 목적으로 사용되고 있다. 현재 대부분의 온톨로지들은 도메인 전문가나 온톨로지 개발자들이 Protégé와 같은 도구를 사용하여 수작업으로 구축되어 지고 있다. 비록 전문가들이 Protégé와 같은 도구를 사용할지라도, OWL등과 같은 언어로 구축된 온톨로지가 실용적이고 도메인의 정보를 정확하게 반영하였음을 검증하는 것은 쉽지 않다. 따라서 본 연구에서는 형식개념분석기법(Formal Concept Analysis)을 사용하여, OWL로 구축된 온톨로지의 소스로부터 온톨로지의 주요 요소들을 추출, 분석하여 구조적 문제점을 파악 할 수 있는 OWL온톨로지 분석도구(OWL Analyzer)의 개발에 대하여 설명한다. 본 연구에서 개발된 OWL Analyzer를 사용함으로써, 구축된 온톨로지내에 포함된 오류를 수월하게 파악할 수 있고, 온톨로지 개발자에게 보다 좋은 개념계층구조를 갖는 온톨로지를 제안할 수 있다.

1. 서론

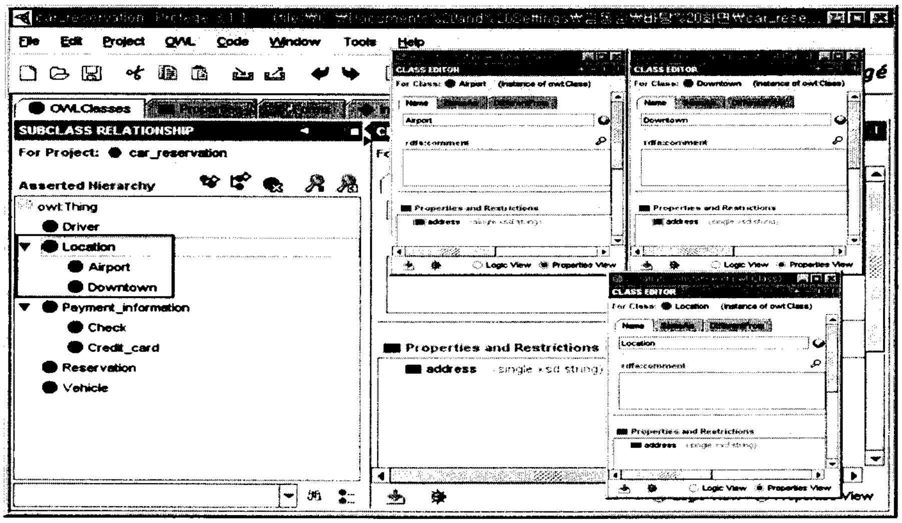
온톨로지는 공유된 개념화(shared conceptualization)에 대한 정형화되고 명시적으로 기술해놓은 지식을 말한다. 즉, 특정 도메인의 개념들을 정의하고, 그들 사이의 관계를 계층적으로 표현하고 있다[1]. 이러한 온톨로지는 여러 분야에서 지식공유 및 지식의 재사용성을 위해 사용되고 있으며, 특히 Tim Berners-Lee에 의해서 제안된 시멘틱 웹에서의 온톨로지는 상호운용성의 토대가 되는 핵심요소로 자리 잡고 있다[2].

현재, 온톨로지를 기술할 수 있는 다양한 언어(RDF, OWL, DAML+OIL)[3-5]들이 제안되었고, 이와 같은 언어들을 사용하여 수월하게 온톨로지를 개발할 수 있는 도구(Protégé, OntoEdit, WebODE)[6-8]들과 온톨로지를 구축하기 위한 가이드라인을 제공하는 방법론[9-11]등 다양한 연구들이 활발하게 이루어지고 있다. 그러나, 온톨로지를 만들기 위한 제반연구들이 제안되고 있음에도 불구하고 현재 존재하는 많은 온톨로지들은 잘못된 정보를 포함하고 있거나, 도메인의 정보를 제대로 반영하지 못하고 있을 가능성 있다[10].

예를 들어 그림1은 Protégé의 확장기능으로 구현된 prompt에서 예제로 제공되고 있는 Car Reservation 온톨로지를 나타내고 있다. Car Reservation 온톨로지에서는 Location클래스의 하위클래스로 Airport와 Downtown클래스들이 선언되어 있다. 그러나 Location은 1개의 프로퍼티(address)가 정의되어 있으며, Airport와 Downtown에 상속되고 있다. 이러한 경우, Location의 하위클래스로 선언된 Airport와 Downtown은 Location과 비교하여 새로운 프로퍼티가 추가정의 되어있지 않기 때문에, 프로퍼티 중심적인 디자인관점에서 엄밀한 포섭관계(subsumption)가 성립하지 않는다.

Protégé를 이용한 수작업에 의한 온톨로지 구축에 있어서 위와 같은 문제는 빈번하게 발생할 수 있으므로, 이와

1) 본 연구는 보건복지부 용어 표준화 지원도구 및 온톨로지 기반의 EHR 상호운용 기술개발 과제(과제번호: A05-0909-A80405-05N1-00050B)의 지원에 의해 이루어진 것임.



<그림 1> Protégé에서 작성된 CC Reservation 온톨로지

같은 문제점을 찾아내기 위한 분석이 필요하다. 기존에 개발된 온톨로지 분석도구로서, OWLDoc[12]은 OWL로 작성된 온톨로지 소스코드로부터 온톨로지의 구조적 정보들을 JavaDoc과 유사한 html문서형태로 변환하여준다. 그러나, 온톨로지내에 포함된 구조적인 오류를 파악 할 수 없다는 문제점이 있다. 한편 FCAViewTab[13]에서는 형식개념분석을 통하여 온톨로지를 분석하는 기능을 제공하고 있지만, 온톨로지의 인스턴스와 boolean type의 slot 들에 한정되어 있기 때문에 온톨로지의 전체적인 구조파악 및 분석에는 한계가 있다.

본 연구에서는, 이와 같은 문제점을 해결하기 위하여 형식개념분석(Formal Concept Analysis)[14]을 토대로, OWL로 작성된 온톨로지의 구조를 분석하여 포함된 오류를 파악하고 수정 보완할 수 있는 방법을 제안하고자 한다. 본 논문은 다음과 같이 구성되어 있다. 제2장에서는 형식개념분석기법을 소개하고, 3장에서는 형식개념분석기법을 기반으로 하는 온톨로지 분석도구(OWL Analyzer)에 대하여 설명한다. 제4장에서는 결론 및 향후과제를 논의한다.

2. 형식개념분석기법

형식개념분석기법의 기본이 되는 구조는 Formal Context이며, 정형화된 정의는 다음과 같다.

[정의 1] One-valued context K=(G, M, I)는 객체들(Objects)의 집합 G와 속성들(Attributes)의 집합 M, 그리고 G와 M사이의 이항관계 $I \subseteq G \times M$로 구성된다.■

어떤 객체 g가 속성 m을 가지고 있을 경우, gIm 또는 $(g, m) \in I$ 로 나타낸다. 표 1은 Car Reservation에 관한 context의 예이다.

[정의 2] context K=(G, M, I)에 대하여, $O \subseteq G$, $A \subseteq M$ 일 때, intent(O)=A ∧ extent(A)=O를 만족하는 (O, A)을 개념(formal concept)이라고 한다. 단,

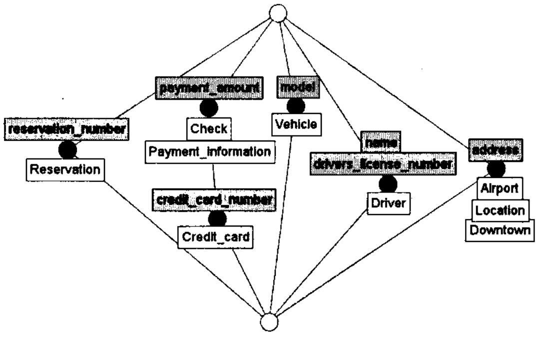
$intent(O) := \{a \in M \mid \forall o \in O: (o, a) \in I\}$,

$extent(A) := \{o \in G \mid \forall a \in A: (o, a) \in I\}$.■

<표 1> Car Reservation에 관한 context

	reservation _number	credit_card _number	payment _amount	drivers _license _number	name	model	address
Airport							×
Location							×
Driver				×	×		
Reservation	×						
Check			×				
Payment_ information			×				
Vehicle						×	
Credit_card		×	×				
Downtown							×

임의의 개념 (O_1, A_1), (O_2, A_2)에 대하여, $O_1 \subseteq O_2$ ($\Leftrightarrow A_1 \supseteq A_2$)인 경우, 개념 (O_1, A_1), (O_2, A_2)은 상위-하위개념관계이며 $(O_1, A_1) \leq (O_2, A_2)$과 같이 표현한다. context K=(G, M, I)로부터 만들어진 모든 개념들 간의 상위-하위개념관계≤는 일종의 반순서관계에 해당하며, 개념들과 그들 사이의 상위-하위개념관계에 의해 만들어진 계층적 개념구조를 개념격자(Concept Lattice)라고 부르며, Hasse Diagram을 사용하여 가시화할 수 있다(그림2).



<그림 2>표1에 대한 context의 개념격자

특히, 형식개념분석 기법은 여러 가지 다양한 값을 가지는 속성들(many-valued attributes)과 객체들, 그리고 객

체와 속성 사이의 관계를 나타내는 데이터에도 적용될 수 있다. many-valued attributes로 구성된 context를 many-valued context라고하며 정의는 다음과 같다.

[정의 3] Many-valued context K =(G, M, W, I)는 객체들(Objects)의 집합 G와 속성들의 집합 M, 속성의 값 W, 그리고 G와 M과 W사이의 관계 $I \subseteq G \times M \times W$로 구성된다.■

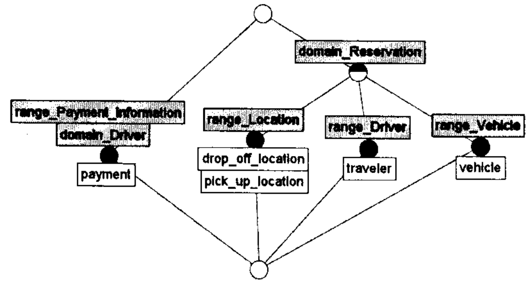
Many-valued context는 각 셀에 해당 객체가 갖는 속성들의 값을 기입한 형태의 테이블로 나타낼 수 있다(표2).

<표 2> Car Reservation에 관한 many-valued context

	domain	range
payment	Driver	Payment_information
traveler	Reservation	Driver
vehicle	Reservation	Vehicle
drop_off_location	Reservation	Location
pick_up_location	Reservation	Location

<표 3> 표2에 대한 Scaling 적용 결과

	domain		range			
	Driver	Reservation	payment_information	Driver	Vehicle	Location
payment	X		X			
traveler		X		X		
vehicle		X			X	
drop_off_location		X				X
pick_up_location		X				X



<그림 3>표3 context에 대한 개념격자

Many-valued context로부터 개념들을 추출하고 개념격자를 구성하기위해서는 특정한 규칙에 따라 Many-valued context를 One-valued context로 변환할 필요가 있으며, 이러한 변환과정을 스케일링(scaling)이라고 한다. 스케일링을 수행하기 위해서, Many-valued context의 각 속성들에 대하여 scale context를 토대로 해석하여, 손실된 정보 없이 표3과 같은 One-valued context로 변환 시킬 수 있으며, 이를 토대로 그림 3과 같은 개념격자를 구축할 수 있다.

3. OWL분석도구의 개발

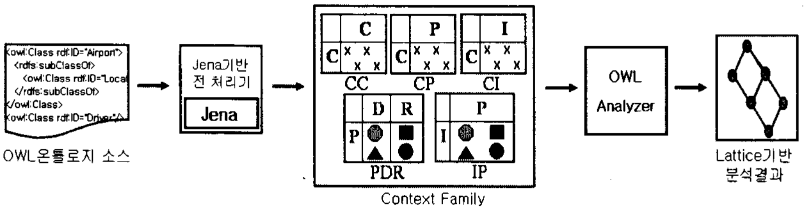
본 연구에서 개발된 OWL 분석도구는 그림4와 같은 구조로 구성되어 있다. OWL로 작성된 온톨로지 소스코드를 Jena[15]를 기반으로 온톨로지의 주요요소들을 추출하는 모듈인 Jena기반전처리기에 입력하여, 5종류의 context들로 구성된 Context Family를 추출한다. Context Family를 OWL Analyzer에 적용함으로서, OWL로 작성된 온톨로지 소스코드로부터 개념격자를 생성할 수 있다. 이와 같은 개념격자에 의해 온톨로지의 상세정보의 추출과 구조의 분석이 가능하다.

Context Family는 온톨로지의 핵심요소들(클래스, 클래스들 사이의 is-a hierarchy, 프로퍼티, 프로퍼티의 domain과 range, individual, individual의 값)을 형식개념분석기법으로 분석하기 위한 context들의 집합으로서 다음과 같이 정의된다.

[정의 4] Context Family CF = {CC, CP, CI, PDR, IP}는 다음과 같은 5개의 context들로 구성된다.

- **Class-Class Context**
  CC = (C, C, I_1)
  C : 클래스들의 집합, $I_1 \subseteq C \times C$
- **Class-Property Context**
  CP = (C, P, I_2)
  P : 프로퍼티(property)의 집합, $I_2 \subseteq C \times P$
- **Class-Individual Context**
  CI = (C, I, I_3)
  I : Individual의 집합, $I_3 \subseteq C \times I$
- **Property-Domain Range Many-valued Context**
  PDR = (P, {D, R}, (C∪T), I_4)
  D : 프로퍼티의 domain, R : 프로퍼티의 range,
  T : primitive 데이터타입들의 집합,
  $I_4 \subseteq P \times \{D, R\} \times (C \cup T)$
- **Individual-Property Many-valued Context**
  IP = (I, P, (I∪V), I_5)
  V : primitive 데이터 값들의 집합
  $I_5 \subseteq I \times P \times (I \cup V)$ ■

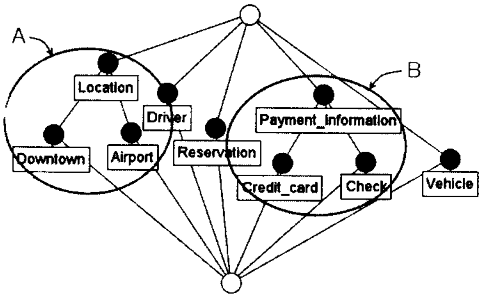
OWL Analyzer는 Context Family를 토대로 6개의 개념격자[2]를 표현하여 OWL로 작성된 온톨로지를 분석하고



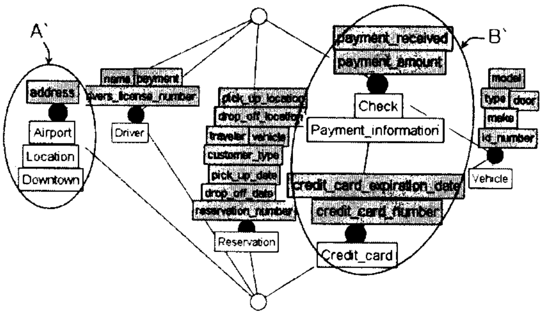
<그림 4> 형식개념분석을 이용한 OWL분석도구의 구조

2) PRD는 domain 및 range에 대한 스케일링을 각각 적용하여 2종류의 개념격자로 표현된다.

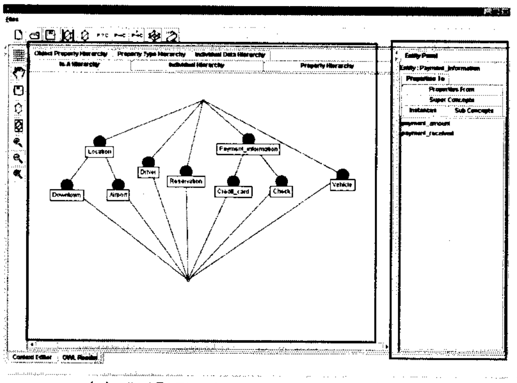
가시화한다. 서론에서 제기된 온톨로지의 구조적 문제점은 CC의 개념격자(그림5)와 CP의 개념격자(그림6)를 비교함으로써 문제점을 파악 할 수 있다. 그림5의 A부분에서 Location클래스의 하위클래스로서 Downtown과 Airport 클래스가 존재하지만, 그림6의 A'부분은 3개의 클래스가 모두 address 프로퍼티만 정의되어 있기 때문에 같은 개념에 속한다. 따라서 온톨로지 개발자가 의도한대로 온톨로지의 구조가 성립하려면, Airport, Downtown클래스들에 각 클래스가 갖는 새로운 프로퍼티가 정의되어야 한다. 또한, 그림5의 B부분과 그림6의 B'부분을 비교해봄으로써 구조적 문제점을 파악할 수 있다.



<그림 5> CC (Class-Class context)의 개념격자



<그림 6> CP (Class-Property context)의 개념격자



(a) 개념격자표시 panel (b) Entity panel

<그림 7> OWL Analyzer의 screen shot

한편, CI로부터 변환된 개념격자에서는 온톨로지의 모든 individual들을 클래스별로 분류한 정보를 파악할 수 있으며, IP를 토대로 변환된 개념격자에서는 individual들이 가지는 실제 프로퍼티의 값을 파악할 수 있다. PDR의 Range속성을 스케일링하여 변환된 개념격자는 프로퍼티를 type별로 분류하고 있으며, Object 프로퍼티의 Domain과 Range를 스케일링한 개념격자는, 클래스간의 관계(association)정보를 표현하고 있다. 또한, CC의 개념격자에서 한 개의 노드를 선택하여 클릭하면 그림7(b)와 같이 해당 클래스에 관련된 상세정보(individuals, 상위클래스, 하위클래스, 프로퍼티)를 text형태로 표현해주는 entity panel도 포함하고 있다.

4. 결론 및 향후과제

본 연구에서는 OWL로 표현된 온톨로지의 소스코드로부터 핵심요소들을 Context Family로 변환하여, 해당 온톨로지의 각 요소들의 상세한 정보를 추출, 분석하여 개념격자 형태로 가시화하고, 구조적 문제점을 파악할 수 있는 OWL Analyzer를 개발하였다. 이와 같은 분석도구는 온톨로지의 개발 및 구현뿐만 아니라, 구조적인 오류를 찾아내고 수정 보완하는 제반관리활동에 도움이 될 수 있다.

현재 OWL Analyzer에서는 OWL소스코드로부터 온톨로지의 구조적인 정보를 분석하는 기능만 제공되고 있으나, 앞으로 OWL Analyzer에 의해 분석된 내용을 토대로 해당 온톨로지의 문제점들을 파악함과 동시에 올바른 형태의 온톨로지로 재구성할 수 있도록 기능을 확장할 예정이다. 또한, 다양한 온톨로지 요소들(axiom, cardinality, constraint 등)을 고려하여 Context Family를 확장하고 제반기능을 추가할 필요가 있다.

참고문헌

[1] T. R. Gruber. Toward principles for the design of ontologies use for knowledge sharing. Presented at the Padua workshop on Formal Ontology, March 1993

[2] Heflin, J. and Hendler, J. Semantic Interoperability on the web. in proceedings of Extreme Markup Language 2000. Graphic Communications Association, 2000. pp.111-120

[3] RDF Primer : http://www.w3.org/TR/rdf-primer/

[4] OWL Overview : http://www.w3.org/TR/owl-features/

[5] DAML+OIL Reference Description : http://www.w3.org/TR/daml+oil-reference

[6] Protégé : http://protege.stanford.edu/

[7] GmbH., How to work with OntoEdit, OntoEdit Tutorial, September 2003

[8] WebODE : http://webode.dia.fi.upm.es/WebODEWeb/index.html

[9] M. Fernandez-Lopez. A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies. OntoWeb IST-2000-29243 Project Deliverable 1.4, 2002.

[10] 류광택, 이헌중, 이상학, 김은주, 이승한, 정현철, 김홍기, 김학래 ,웹 온톨로지 개발지침 연구, 한국전산원 기관발행물,2004

[11] Suk-Hyung Hwang, Hong-Gee Kim, Myeng-Ki Kim, Sung-Hee Choi, Hae-Sool Yang, A Data-Driven Approach to Constructing an Ontological Concept Hierarchy Based on the Formal Concept Analysis. ICCSA(4), 2006, 937-946

[12] OWLDoc : http://www.co-ode.org/downloads/owldoc/co-ode-index.php

[13] FCAViewTab : http://informatics.mayo.edu/LexGrid/index.php?page=fca

[14] B. Ganter, R. Wille, Formal Concept Analysis Mathematical Foundations, Springer-Verlag, 1999

[15] Jena : http://jena.sourceforge.net/