

# 형식개념분석기법을 이용한 온톨로지 분석도구

김동순\*, 황석형\*\*, 김홍기\*\*\*

\*선문대학교 전자계산학과, \*\*선문대학교 컴퓨터정보학부, \*\*\*서울대학교 의생명 지식공학연구실  
e-mail:ds8025@gmail.com

## A tool for analysis of Ontology using Formal Concept Analysis

Dong-Soon Kim\*, Suk-Hyung Hwang\*\*, Hong-Gee Kim\*\*\*

\*Dept of Computer Science, SunMoon University

\*\*Division of Computer and Information Science, SunMoon University

\*\*\*Biomedical Knowledge Engineering Laboratory, Seoul National University

### 요 약

온톨로지는 의미적인 지식을 표현하고, 공유하기 위해서 여러 분야에서 널리 사용되고 있으며, 특히 시멘틱 웹의 상호운용성을 해결할 수 있는 중요한 요소로 자리 잡고 있다. 현재 온톨로지의 구축을 지원하는 다양한 도구들과 방법론들이 제시되고 있지만, 실제 구축되어진 온톨로지가 실용적이고, 도메인의 정보를 정확하게 반영했는지 검증할 수 있는 도구나 방법은 매우 미비하다. 따라서 본 연구에서는 OWL언어로 구축되어진 온톨로지의 소스로부터 온톨로지의 주요 요소들을 추출하여 형식개념분석(Formal Concept Analysis)모델로 변환하고 분석함으로써, 온톨로지에 포함된 구조적인 문제점을 파악할 수 있는 방법과 도구에 대해서 설명한다. 본 연구에서 제시한 온톨로지 분석도구를 사용하면, 구축되어진 온톨로지에 포함된 구조적 오류 및 온톨로지의 구조를 수월하게 파악할 수 있고, 온톨로지 구축작업 및 수정보완에 있어서 더 좋은 개념계층구조를 갖는 온톨로지를 제안할 수 있다.

### 1. 서론

온톨로지는 특정 도메인의 지식이나 정보를 의미적으로 표현하기 위한 정형화 되고 명시적으로 기술한 모델을 의미한다. 즉, 특정 영역이나 도메인 내에 있는 실체를 개념화하고, 그들 사이의 다양한 관계를 표현하고 있다[1]. 이러한 온톨로지는 여러 분야에서 지식공유 및 지식의 재사용성을 위해 사용되고 있으며, 특히 Tim Berners-Lee에 의해서 제안된 시멘틱 웹의 상호운용성을 해결할 수 있는 핵심요소로 자리 잡고 있다[2].

현재, 온톨로지를 기술할 수 있는 다양한 언어(RDF, OWL, DAML+OIL)[3-4]들이 제안되었고, 온톨로지 구축도구[5]나 구축방법론[6]등 다양한 연구[7]들이 활발하게 이루어지고 있다. 그러나 온톨로지를 만들기 위한 많은 제반연구들이 활발하게 제안되고 있음에도 불구하고 현재 존재하는 많은 온톨로지들은 잘못된 정보를 포함하고 있거나, 도메인의 정보를 제대로 반영하지 못하고 있을 가능성이 있다[8]. 온톨로지 구축에 있어서 위와 같은 문제는 빈번하게 발생할 수 있으므로, 이와 같은 문제점을 찾아내기 위한 온톨로지 분석이 필요하다.

기존에 개발된 온톨로지 분석도구로서, OWLDoc[9]은 OWL로 작성된 온톨로지 소스코드로부터 온톨로지의 구

조적 정보들을 JavaDoc과 유사한 html문서형태로 변환하여준다. 그러나 온톨로지내에 포함된 구조적인 오류를 파악 할 수 없다는 문제점이 있다. 한편 FCATViewTab [10]에서는 형식개념분석을 통하여 온톨로지를 분석하는 기능을 제공하고 있지만, 온톨로지의 인스턴스와 boolean type의 slot들에 한정되어 있기 때문에 온톨로지의 전체적인 구조파악 및 분석에는 한계가 있다.

본 연구에서는, 형식개념분석(Formal Concept Analysis)[11]기법을 토대로, OWL로 작성된 온톨로지의 구조를 분석하여 포함된 구조적인 오류를 파악하고 수정 보완할 수 있는 방법을 제안하고자 한다. 본 논문은 다음과 같이 구성되어 있다. 제2장에서는 형식개념분석기법과 multicontexts[12]를 소개하고, 3장에서는 형식개념분석기법을 기반으로 하는 온톨로지 분석도구 및 분석사례를 대하여 설명한다. 제4장에서는 결론 및 향후과제를 논의한다.

### 2. Basic Notation

#### 2.1 형식개념분석기법(Formal Concept Analysis)

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

**[정의 1]** One-valued context  $K=(G, M, I)$ 는 객체들 (Objects)의 집합  $G$ 와 속성들(Attributes)의 집합  $M$ ,

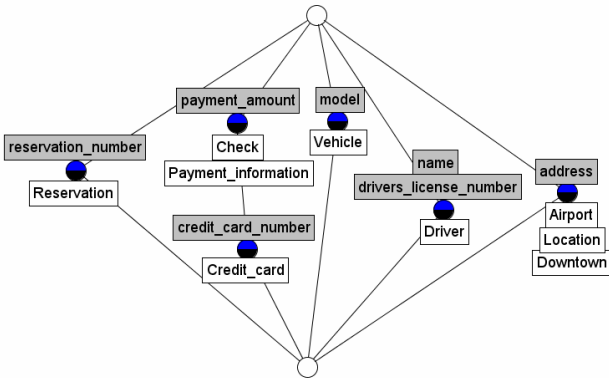
그리고 G와 M사이의 이항관계  $I \subseteq G \times M$ 로 구성된다.

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

<표 1> Car Reservation에 관한 context

	reservation_number	credit_card_number	payment_amount	drivers_license_number	name	model	address
Airport							X
Location							X
Driver				X	X		
Reservation	X						
Check			X				
Payment_information			X				
Vehicle						X	
Credit_card		X	X				
Downtown							X

[정의 2] context  $K=(G, M, I)$ 에 대하여,  $O \subseteq G, A \subseteq M$  일 때,  $intent(O)=A \wedge extent(A)=O$ 를 만족하는  $(O, A)$ 을 개념(formal concept)이라고 한다. 단,  
 $intent(O) := \{a \in M \mid \forall o \in O: (o, a) \in I\}$ ,  
 $extent(A) := \{o \in G \mid \forall a \in A: (o, a) \in I\}$ .



<그림 1> 표1에 대한 context의 개념격자

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-하위개념관계는 일종의 반순서관계에 해당하며, 개념들과 그들 사이의 상위-하위개념관계에 의해 만들어진 계층적 개념구조를 개념격자(Concept Lattice)라고 부르며, Hasse Diagram을 사용하여 가시화할 수 있다(그림1).

2.2 Multicontexts

Multicontexts[13]는 다양한 지식 및 정보를 표현하기 위한 context들의 집합을 의미하며, 각각의 context들의 개념구조들 사이의 개념적 결합관계를 표현함으로써 온톨로지, conceptual modeling, Description Logic등 다양한 지식을 표현할 수 있다.

[정의 3] multicontexts  $K = (S_I, R_P)$ 는 context들의 객체들(objects)과 속성들(attributes)을 의미하는 집합  $S_I$ , 그리고 각각의 context들이 가지는 이항관계들의 집합  $R_P$ 로 구성된다( $P$ 는 context의 index집합,  $I$ 는 set들의 index집합). 그리고 각각의 context가 구성하는 객체집합, 속성집합을 반환하는 함수  $\tau : P = I^2$ 가 존재한다. 따라서 multicontexts는 다음과 같이 formal context들의 집합  $K_P = \{ (S_i, S_j, R_p) \mid S_i, S_j \in S_I, R_p \in R_P \text{ and } R_p \subseteq S_i \times S_j \text{ if } \tau : p = (i, j) \}$ 으로 표현될 수 있다.

3. 온톨로지 분석도구 및 분석사례

3.1 온톨로지 분석도구의 개요와 구조

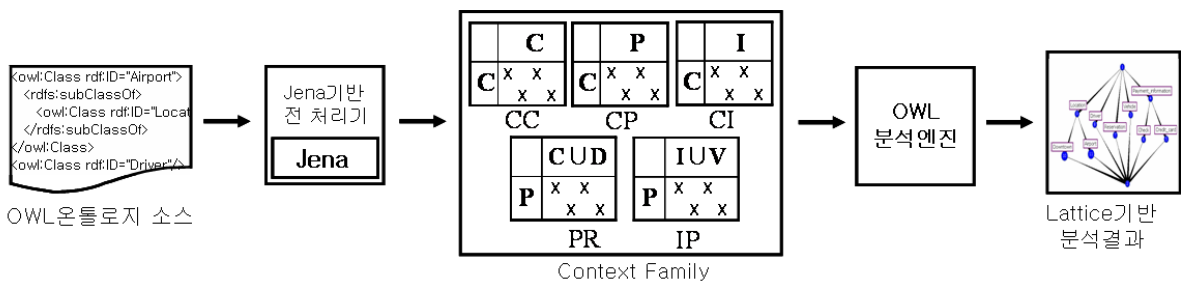
본 연구에서 개발된 온톨로지 분석도구는 그림2와 같은 구조로 구성되어 있다. OWL로 작성된 온톨로지 소스코드를 Jena[13]를 기반으로 온톨로지의 주요요소들을 추출하는 모듈인 Jena기반전처리기에 입력하여, 5종류의 context들로 구성된 Context Family를 추출한다. Context Family를 OWL 분석엔진에 적용함으로써, OWL로 작성된 온톨로지 소스코드로부터 개념격자를 생성할 수 있다. 이와 같은 개념격자에 의해 온톨로지의 상세정보의 추출과 구조의 분석이 가능하며, 온톨로지의 구조적인 문제점을 자동적으로 파악하여 가시화할 수 있다.

3.2 Context Family

Context Family는 온톨로지의 핵심요소들(클래스, 클래스들 사이의 is-a hierarchy, 프로퍼티, 프로퍼티의 domain과 range, individual, individual의 값)을 형식개념분석기법으로 분석하기 위한 context들의 집합으로서 multicontexts개념을 적용하여 온톨로지 모델을 다음과 같이 context의 집합으로 정의할 수 있다.

[정의 4] Context Family  $CF = \{S_K, B_J\}$ 는 온톨로지의 주요요소들의 집합인  $S_K$ , 와 온톨로지의 요소들 사이의 이항관계들의 집합  $B_J$ 로 구성된다.

- $S_K = \{ S_c, S_p, S_i, S_d, S_v \}$   
 $S_c$  : 온톨로지의 클래스들의 집합  
 $S_p$  : 온톨로지의 프로퍼티들의 집합  
 $S_i$  : 온톨로지의 individual들의 집합  
 $S_d$  : 온톨로지의 primitive datatype들의 집합  
 $S_v$  : 온톨로지의 primitive data value들의 집합



<그림 2> 온톨로지 분석도구의 전체구조

- $B_J = \{ B_{cc}, B_{cp}, B_{ci}, B_{pr}, B_{ip} \}$   
 $B_{cc} : \tau (cc) = (c, c) B_{cc} \subseteq S_c \times S_c$   
 $B_{cp} : \tau (cp) = (c, p) B_{cp} \subseteq S_c \times S_p$   
 $B_{ci} : \tau (ci) = (c, i) B_{ci} \subseteq S_c \times S_i$   
 $B_{pr} : \tau (pr) = (p, c \cup d) B_{pr} \subseteq S_p \times (S_c \cup S_d)$   
 $B_{ip} : \tau (ip) = (i, i \cup v) B_{ip} \subseteq S_c \times (S_i \cup S_v)$

위와 같은 온톨로지의 요소집합, 그들 사이의 이항 관계 집합을 통하여, 온톨로지 정보를 표현하는 5개의 context를 다음과 같이 정의 할 수 있다.

- $CF_J = \{ (S_m, S_n, B_j) \mid S_m, S_n \in S_K, B_j \in B_J \text{ and } B_j \subseteq S_m \times S_n \text{ if } \tau : j = (m, n) \}$   
 $CF_{cc} = \{ (S_c, S_c, B_{cc}) \mid S_c \in S_K, B_{cc} \in B_J \}$   
 $CF_{cp} = \{ (S_c, S_p, B_{cp}) \mid S_c, S_p \in S_K, B_{cp} \in B_J \}$   
 $CF_{ci} = \{ (S_c, S_i, B_{ci}) \mid S_c, S_i \in S_K, B_{ci} \in B_J \}$   
 $CF_{pr} = \{ (S_p, (S_c \cup S_d), B_{pr}) \mid S_p, S_c, S_d \in S_K, B_{pr} \in B_J \}$   
 $CF_{ip} = \{ (S_i, (S_i \cup S_v), B_{ip}) \mid S_i, S_v \in S_K, B_{ip} \in B_J \}$

### 3.3 온톨로지의 구조적인 문제

OWL로 작성되어진 온톨로지를 context family로 변환하여, lattice형태로 가시화하고 분석함으로써, 온톨로지에 존재하는 구조적인 문제점(클래스 계층 구조의 문제, 프로퍼티의 문제, individual의 문제)들을 수월하게 파악할 수 있다.

#### (1) 클래스 계층구조 확인 알고리즘

온톨로지에서 가장 중요하고, 기초가 되는 요소는 클래스이다. 클래스는 클래스가 가지는 속성에 의해서 표현되어진다. 따라서 특정 개념을 클래스로 표현한다면 개념이 가지고 있는 속성들을 잘 파악하여, 클래스에 정의함으로써 도메인의 정보를 제대로 반영하고, 유용한 온톨로지가 될 수 있다.

```
//input: 온톨로지서 subClassOf관계를 이용하여 상하위관계가 정의된 2개의 클래스
//output: 문제가 없을시 true, 문제 발생시 false
Is-A_Check(superClass, subClass) {
  if(subClass.numberofProperties==superClass.numberofProperties){
    while(numberofProperties){
      if(subClass.Property.range >= superClass.Property.range)
        return false;
    }
  }
  else if(subClass.numberofPropertes < superClass.numberofProperties)
    return false;
  else
    return true;
}
```

<표 3> 클래스 계층구조 문제를 체크하기 위한 알고리즘

클래스 계층구조의 문제란 온톨로지의 핵심요소인 클래스들 사이의 상하위 관계를 클래스가 갖는 속성을 고려하지 않고 형식적으로만 subClassOf관계를 명시한 상태를 말한다. 하위클래스는 상위클래스의 프로퍼티를 상속받고, 상위클래스와 구분이 되는 프로퍼티를 추가적으로 선언하거나, 프로퍼티의 range를 더 상위 개념으로 선언함으로써 프로퍼티 중심적인 디자인 관점에서 상하위 클래스사이에 엄밀한 포섭관계(subsumption)가 성립되어야 한다. 하지만 대부분의 온톨로지서 매우 복잡하고, 사람에 의해서 수작업으로 구축되어지므로, 클래스의 계층구조 문제가 빈번하게 발생한다. 따라서 표3의 알고리즘은 온톨로지서 subClassOf를 이용하여 상하위 관계가 정의된 2개의 클래스를 입력함으로써 이들 사이에 적절하게 is-a관

계가 정의되었는지 확인할 수 있다.

#### (2) 프로퍼티의 domain, range 확인 알고리즘

온톨로지의 프로퍼티는 기본적으로 domain과 range 정보로 구성된다. 따라서 프로퍼티는 domain과 range 정보를 반드시 기술되어야 한다. domain에는 해당 프로퍼티가 선언되어진 온톨로지나 참조되어지는 온톨로지의 클래스가 기술되어야 하며, range에는 클래스나 온톨로지를 지원하는 datatype을 선언해야 한다. 표4의 알고리즘은 하나의 프로퍼티를 입력으로 domain이나 range가 잘 정의되었는지 확인할 수 있다.

```
// input : 온톨로지에 정의되어진 프로퍼티
// output : 문제가 없을시 true, 문제 발생시 false
PropertyCheck ( property ) {
  boolean do == false;
  boolean ran == false;
  if ( property.domain == null || property.range == null ) {
    return false;
  }
  while ( numberOfClass ) {
    if ( property.domain.type == class )
      do = true;
    if ( property.range.type == class )
      ran = true;
  }
  while ( numberOfDatatype ) {
    if ( property.range.type == datatype )
      ran = true;
  }
  if ( do == false || ran == false )
    return false;
  return true;
}
```

<표 4> 프로퍼티 domain, range를 체크하기 위한 알고리즘

#### (3) 클래스-Individual 확인 알고리즘

온톨로지에 선언되어진 individual은 기본적으로 individual이 해당되는 클래스가 갖는 프로퍼티들의 range타입에 맞는 실제 데이터 값이나, 다른 individual들이 선언되어져야 한다. 표5의 알고리즘은 하나의 individual을 입력으로 individual에 속하는 클래스의 프로퍼티를 반영하여 실제 데이터 값을 갖는지 확인할 수 있다.

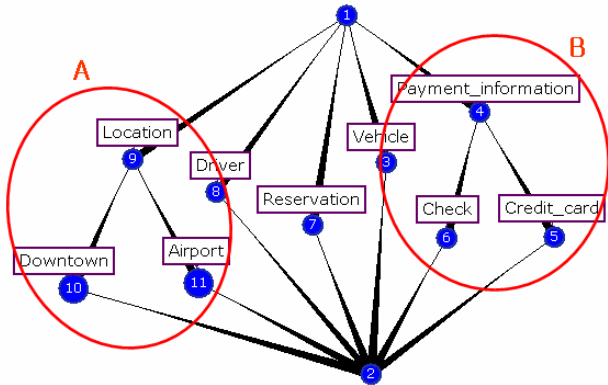
```
// input : 온톨로지에 정의되어진 individual
// output : 문제가 없을시 true, 문제 발생시 false
IndividualCheck ( individual ) {
  class = individual.getClass();
  while ( class.numberofProperties ) {
    if ( individual.property.value == null ||
        individual.property.type != property.range.type ) {
      return false;
    }
  }
  return true;
}
```

<표 5> 클래스-individual을 체크하기 위한 알고리즘

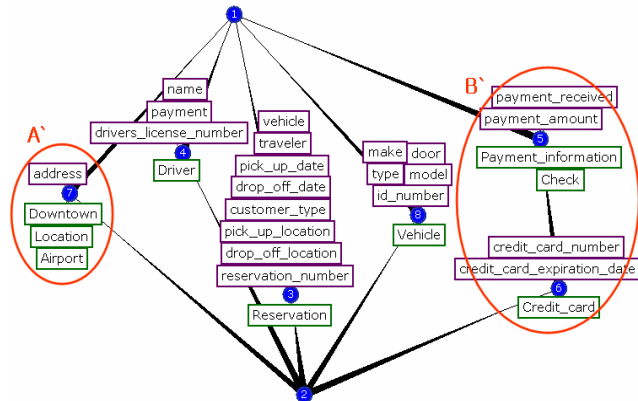
### 3.4 온톨로지 분석사례

본 연구에서 개발한 온톨로지 분석도구를 사용하여 Prompt의 튜토리얼 예제로 제공되는 car reservation 온톨로지를 분석해 보았다. 앞서 정의한 클래스 계층구조문제는 CF<sub>cc</sub>의 개념격자(그림3)와 CF<sub>cp</sub>의 개념격자(그림4)를 비교함으로써 문제점을 파악 할 수 있다. 그림

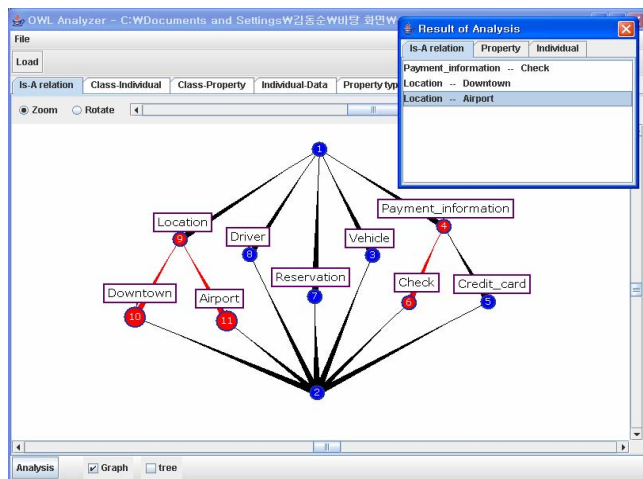
3의 A부분에서 Location클래스의 하위클래스로서 Downtown과 Airport클래스가 존재하지만, 그림4의 A' 부분은 3개의 클래스가 모두 address 프로퍼티만 정의되어 있기 때문에 같은 개념에 속한다. 따라서 온톨로지 개발자가 의도한대로 온톨로지의 구조가 성립하려면, Airport, Downtown클래스들에 각 클래스가 갖는 새로운 프로퍼티가 정의되어야 한다.



<그림 3> CF<sub>cc</sub>의 개념격자



<그림 4> CF<sub>cp</sub>의 개념격자



<그림 5> 파악된 클래스 계층구조의 문제 가시화

또한, 그림5의 B부분과 그림6의 B'부분을 비교해봄으로써 구조적 문제점을 파악할 수 있다. 본 연구에서 개발한 온톨로지 분석도구(그림5)는 클래스 계층구조의 문제점을 자동으로 파악하여 CF<sub>cc</sub>의 개념격자에 붉은 색으로 체크해주거나, 팝업창을 통하여 warning해줄 수 있다.

그 외에도 CF<sub>ci</sub>로부터 변환된 개념격자에서는 온톨

로지의 모든 individual들을 클래스별로 클러스터링한 정보를 파악할 수 있으며, CF<sub>id</sub>를 토대로 변환된 개념격자에서는 individual들이 가지는 실제 프로퍼티의 값과 individual들이 갖는 문제들을 파악할 수 있다. CF<sub>pr</sub>의 개념격자는 프로퍼티를 type별로 분류하고 있으며, CF<sub>pr</sub>와 CF<sub>cp</sub>의 개념격자를 통하여 프로퍼티의 문제점을 파악할 수 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 온톨로지 분석도구를 개발하였다. 이와 같은 분석도구는 온톨로지의 개발 및 구현뿐만 아니라, 구조적인 오류를 찾아내고 수정 보완하는 제반 관리활동에 도움이 될 수 있다.

현재 온톨로지 분석도구에서는 OWL소스코드로부터 온톨로지의 문제점을 파악하는 기능만 제공되고 있으나, 향후연구에서는 발견된 온톨로지의 문제점들을 수정/변경함으로써 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] Klyne, D., Guha, R.V. Resource Description Framework Concepts and Abstract Syntax. <http://www.w3.org/TR/rdf-schema/>

[4] M. Smith, C. Welty, D. McGuinness, OWL Web Ontology Language Guide <http://www.w3.org/TR/2003/WD-owl-guide-20030331>

[5] Protégé : <http://protege.stanford.edu/>

[6] M. Cristani, R. Cuel, A survey on ontology creation methodologies International Journal on Semantic Web and Information System, Vol 1, No.2, 49-69, 2005

[7] 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

[8] 류광택, 이현중, 이상학, 김은주, 이승한, 정현철, 김흥기, 김학래, 웹 온톨로지 개발지침 연구, 한국전산원 기관발행물, 2004

[9] Horridge, M. :OWLDoc (2004): <http://www.co-ode.org/downloads/owldoc/co-ode-index.php>

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

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

[12] R. Wille. Conceptual structures of multicontexts. Proceedings of the 4th ICCS, Sidney(AU), Volume 1115, pages 23-39, August 1996

[13] Jena : <http://jena.sourceforge.net/>