

(1) 토픽(Topic)

토픽맵에서 기본이 되는 맵아이템으로 토픽은 수집된 정보 자원으로부터 명세화된 주제를 기술하거나 구체화한다. 토픽의 각 인스턴스는 정확히 하나의 주제와 연관된다. 토픽은 주제를 명백하게 식별하고자 참조되는 정보 리소스인 주제지시자와, 주제지시자를 참조하는 주제식별자, 그리고 정보 리소스를 참조하는 주제위치자와 관계한다.

각각의 토픽은 일련의 속성(Characteristic)들을 가지며, 관계역할(AssociationRole), 어커런스(Occurrence), 토픽이름(TopicName)을 포함한다.

(2) 관계(Association)

관계는 하나 또는 그 이상의 토픽들 간에 다중방식의(multi-way) 관련성이다. 관계는 타입(type)을 가지며 특정 범위(scope)로 정의될 수 있다. 관계에 의하여 정의된 관련성은 토픽 자신들이기 보다는 오히려 포함된 토픽의 주제 사이의 관련성이다.

(3) 관계역할(AssociationRole)

토픽은 관계내에서 특정한 부분 또는 특정한 역할을 수행하는데, 관계역할로 구체화된다. 관계는 토픽에 의해 수행되는 역할들의 집합으로 구성된다. 관계역할은 관계에서의 주제의 참여에 대한 표현이다. 관계역할은 역할을 수행하는 토픽을 가지며 관계에서 역할자 참여의 속성을 정의하는 타입을 갖는다. 역할과 관계는 토픽 자체가 아니라 참여하고 있는 토픽 주제 사이의 관련성을 표현한다.

(4) 어커런스(Occurrence)

어커런스는 어트리뷰트(attribute)와 매우 유사한 속성이다. 어커런스는 범위지정이 가능(scop_able)하고, 타입지정이 가능(type_able)하다. 어커런스의 값은 추상 메타클래스 리소스와 더불어 관계에서의 위치자 역할에 의해서 구체화된다. 리소스의 해석은 어커런스의 구체화에 의해서 정의된다. 어커런스는 인스턴스를 포함하고 있는 토픽에 대한 서술적 정보가 될 수 있고, 어커런스나 또는 주제의 인스턴스를 포함하고 있는 토픽의 속성을 표현할 수 있다.

(5) 토픽명(TopicName)과 변수명(Variant)

토픽명과 변수명은 토픽에 대해서 인간이 이해 가능하도록 한 라벨(label)이다. 우선적인 기본이름으로 명칭된 토픽명이 유니코드 문자열로 요구되는 반면에, 변수명은 아이콘, 이미지, 오디오와 같은 명칭과 같이 흔히 고려되지 않은 많은 데이터 타입을 포함할 수 있다. 변수명은 구체화된 토픽에 대한 선택적인 이름이다. 따라서 기본 토픽명의 환경조건에만 사용되어야 하는 제약은 받는 어커런스의 구체화라 할 수 있다.

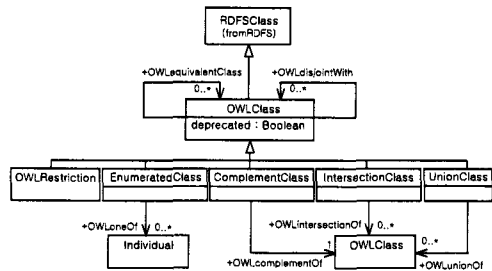
3. OWL의 메타모델 구조

OWL은 W3C에서 온톨로지를 공표하고 공유함을 위한 의미적 마크업 언어이다. OWL은 웹 구조속으로 적합시키기 위하여 XML 기반 인코딩과 그 배경이 되는 RDF(S)를 포함하는 RDF의 확장이다. OWL은 DL 이론에 적용될 수 있는 의미구조를 갖는다. DL은 인터프리테이션을 추상적인 도메인으로 매핑한다. 따라서 DL의 강건하고 완전한 추론 절차가 OWL 모델을 위한 추론 지원을 제공하는데 사용될 수 있다.[8] OWL의 서브셋으로 제안하고 있는 유형에는 OWL Lite, OWL DL, OWL Full이 있다. OWL Lite는 시소러스에 접근이 용이하고 단순한 면을 강조하여 웹 응용에 강점을 갖고 있으며, OWL DL은 Lite보다 좀더 논리적인 표현을 한다. OWL Full은 표현력에 있어서 가장 풍부하며 RDF의 자유로운 문법을 모두 허용한다. OWL Full은 OWL DL과 OWL Lite의 모든 기능을 포함하는 관계이며, 유효성 및 호환성에 있어서 가장 완벽하다.[9] 본 논문에서의 메타모델은 OWL Full을 기준한다.

3.1 OWL 메타모델의 주요 구성요소[10,11]

(1) 클래스(Class)

클래스는 개체의 분류를 위한 추상 메카니즘을 제공한다. 모든 클래스는 클래스 확장이라 하는 일련의 개체들과 관계한다. 클래스는 공리가 되는 '클래스 기술'을 통해서 표현된다. 클래스 기술은 여섯 가지 타입으로 식별자(identifier), 열거(enumeration), 프로퍼티제한(restriction), 여집합(complement), 교집합(intersection), 합집합(union) 등이 있다. 개체의 공통 속성은 프로퍼티 제한을 이용하는 클래스에 대해서 명세화 될 수 있다. 클래스의 개체는 그 클래스에 적용할 수 있는 각각의 프로퍼티에 대한 값을 가질 필요가 없다. 또한 개체는 하나 이상의 클래스의 인스턴스가 될 수 있다. 그림 2는 OWL 메타모델의 클래스 다이어그램을 보인다.



(그림 2) OWL 메타모델의 클래스 다이어그램

(2) 개체(Individual)

개체는 클래스의 인스턴스이고, 0 또는 그 이상의 클래스의 인스턴스가 될 수 있다. 개체의 표기형식은 RDFS 리소스로부터 상속된다. 개체 클래스에는 추가적 속성그룹으로 데이터타입 슬롯(DatatypeSlot)과 객체슬롯(ObjectSlot)이 집합될 수 있다. 리스트에 있는 모든 개체를 각각이 상호간에 모두 다르다는 의미로 AllDifferent와 관계한다. 또한 개체들 사이에 differentFrom 관계를 표현하는 간편한 방법을 제공한다.

(3) 프로퍼티(Property)

프로퍼티는 추상 클래스로서 개체와 데이터 값을 연관시키거나, 개체와 개체 사이에 관계를 형성해 주며, 하위 클래스로서 데이터타입 프로퍼티와 객체프로퍼티로 나누어진다.

데이터타입 프로퍼티(DatatypeProperty)는 개체를 데이터 값과 연관시킨다. 데이터타입 프로퍼티는 클래스의 인스턴스와 데이터범주의 인스턴스 사이의 관련성을 제공한다. OWL DL의 논리상 데이터타입 프로퍼티는 객체 프로퍼티와 주석 프로퍼티와 온톨로지 프로퍼티와 별개(disjoint)이어야 한다.

객체프로퍼티(ObjectProperty)는 개체와 개체를 연관시키며 두 클래스 인스턴스 사이의 관련성을 제공한다. 객체 프로퍼티의 값은 개체이다. 대칭형 프로퍼티는 그 자신의 역이 된다.

(4) 제한사항(Restriction)

OWL 제한사항은 특별한 클래스로서, 특정 프로퍼티 제한사항을 만족시키는 모든 개체의 클래스를 표현한다. 두 가지 유형의 프로퍼티 제한사항이 있다. 즉 값(value) 제한과 카디널리티(cardinality) 제한이다. 값 제한은 프로퍼티의 범위에 제한을 두고, 카디널리티 제한은 프로퍼티가 취할 수 있는 값의 수에 제한을 둔다.

4. 매핑을 위한 분석

매핑변환 연구에서 해결의 주요 문제점은 이질적인 환경의 모델간의 관련성이다. 구조나 의미, 또는 기술적인 차이를 이해하기 위해서 먼저 각 모델의 특성과 개념, 프로퍼티 등을 분석한다.

서로 다른 유형의 메타모델을 하나의 구조로 통합시키는 데는 통합하고자 하는 각각의 메타모델(목표모델/소스모델)들을 별도로 유지하고, 중간 매개의 역할을 하는 모델(매개모델)안으로 양쪽의 메타모델들을 매핑시키는 것이 유리하다. 왜냐하면 소스모델 요소들과 목표 모델 요소들 사이에 관계를 유지할 수 있기 때문에 매핑의 결과는 무결성을 유지할 수 있다. 본 연구에서 매개 모델 역할은 토픽맵과 OWL의 기반을 이루는 DL이다.[1]

이질적인 환경의 메타모델 사이에 변환의 근본 문제는 각 메타 모델에 포함되어 있는 대응 요소들이 임의적이고 불규칙한데 있다. 즉, 메타모델간의 매핑시에 모델을 구성하는 각 요소들이 단일 요소가 아니거나(1:1 대응 아님), 또는 한 모델에서는 두 가지 유형의 개념으로 표현되는 요소가 다른 모델에서는 한 개의 통합된 개념으로 존재할 수 있는 경우이다. 따라서 매핑 과정에서 요소들의 손실이 발생하지 않도록 하기 위한 방법이 필요하다.

본 연구에서는 소스모델의 요소들과 목표모델의 요소들 사이에 손실 방지를 위하여 해당 요소의 발생시 고유 모델의 정보를 파악할 수 있는 태그를 첨부하는 스타일로 매핑을 수행한다. 이는 소스모델과 목표모델의 요소들의 이동경로를 추적할 수 있고, 후에 역추적도 가능하다.

본 논문에서 조사된 메타모델 매핑의 요소는 뚜렷한 개념의 주제에 속하여 단순매핑이 가능한 토픽이나 클래스 또는 객체의 매핑보다는 이들을 제외한 속성 항목이나 계층구조를 중심으로 이루어졌다. <표 1참조>

(표 1) 메타모델의 매핑요소

TopicMaps	OWL
-관계(Association)	-프로퍼티(Property) : 객체, 데이터
-관계역할(AssociationRole)	타입
-어커런스(Occurrence)	-프로퍼티제한(PropertyRestriction)
-타입계층구조(Hierarchy)	-클래스계층구조(Hierarchy)

본 논문은 모델간의 매핑을 처리하기 위해서, DSTC (Distributed Systems Technology Centre : www.dstc.edu.au)의 QVT(Query/View/Transform) 변환 언어를 사용한다. 그리고 QVT 변환 결과의 평가를 위해서 DSTC의 프로토타입 변환 엔진 Tefkat를 사용한다.

5. 토픽맵에서 OWL로 매핑

5.1 관계 및 관계역할의 매핑

토픽맵에서 관계와 관계역할은 OWL의 객체 프로퍼티와 매핑된다. 만일 토픽맵에서 가능한 식별자가 없다면, 객체 프로퍼티를 사용하기 위한 식별자를 생성한다.<그림 3, 4참조>

```

RULE Assoc_ObjProp (as, oop)
FORALL
  Association as
WHERE
  assoc.identifier() != NULL
MAKE
  OWLObjectProperty oop
  oop.type = OWLObjectProperty
  oop.ID = as.identifier()

RULE Assoc_ObjProp_Anon (as, oop) //ObjProp_Anonym
FORALL
  association as
WHERE
  assoc.identifier() == NULL
MAKE
  OWLObjectProperty oop
  oop.type = OWLObjectProperty
    
```

```
oop.ID = genid()
```

(그림 3) 관계와 객체 프로퍼티 매핑규칙

```

RULE Role_ObjProp (ar, oop)
FORALL
  AssociationRole ar
WHERE
  ar.identifier() != NULL
MAKE
  OWLObjectProperty oop
  oop.type = OWLObjectProperty
  oop.ID = ar.identifier()

RULE Role_ObjProp_Anon (as, oop) //ObjProp_Anonym
FORALL
  AssociationRole ar
WHERE
  ar.identifier() == NULL
MAKE
  OWLObjectProperty oop
  oop.type = OWLObjectProperty
  oop.ID = genid()
    
```

(그림 4) 관계역할과 객체프로퍼티 매핑규칙

5.2 어커런스의 매핑

토픽맵의 어커런스는 OWL의 객체프로퍼티 또는 데이터타입 프로퍼티와 매핑될 수 있다. 위치자의 인스턴스 형태의 자원을 갖는 어커런스는 객체 프로퍼티가 된다. 데이터의 인스턴스 형태의 자원을 갖는 어커런스는 데이터타입 프로퍼티이다. 즉 어커런스 데이터 타입이 'uri' 프로퍼티이면 OWL에서 객체 프로퍼티이고, 그렇지 않으면 데이터 프로퍼티이다. 만일 토픽맵 구조에서 가능한 식별자가 없으면, OWL에서 프로퍼티 구조에 사용될 식별자를 생성한다.<그림 5, 6참조>

```

RULE Occr_ObjProp (ar, oop)
FORALL
  Occurrence ocr
WHERE
  ocr.identifier() != NULL AND
  ocr.resource instanceof(Locator)
MAKE
  OWLObjectProperty oop
  oop.type = OWLObjectProperty
  oop.ID = ocr.identifier();

RULE Occr_ObjProp_Anon (as, oop) //ObjProp_Anonym
FORALL
  Occurrence ocr
WHERE
  ocr.identifier() == NULL AND
  ocr.resource instanceof(Locator)
MAKE
  OWLObjectProperty oop
  oop.type = OWLObjectProperty
  oop.ID = genid();
    
```

(그림 5) 어커런스와 객체프로퍼티 매핑규칙

```

RULE Occr_DTyProp (ar, dtp)
FORALL
  Occurrence ocr
WHERE
  ocr.identifier() != NULL AND
  ocr.resource instanceof(Data)
MAKE
  OWLDatatypeProperty dtp
  dtp.type = OWLDatatypeProperty
    
```

```

    dtp.ID = ocr.identifier();
RULE Ocrr_DTyProp_Anon (as, dtp) //DTyProp_Anonym
FORALL
  Occurrence ocr
WHERE
  ocr.identifier() == NULL AND
  ocr.resource instanceof(Data)
MAKE
  OWLDatatypeProperty dtp
  dtp.type = OWLDatatypeProperty
  dtp.ID = genid();
    
```

(그림 6) 어커런스와 데이터타입프로퍼티 매핑규칙

```

RULE Ocr_DTPropRestriction(top, top_ocr, ocls, dtp,
rdfsr, res)
FORALL
  Topic top
  Occurrence top_ocr
  OWLClass ocls
  OWLDatatypeProperty dtp
  RDFSResource rdfs
WHERE
  top.occurrences.contains(top_ocr) AND
  top_ocr.resource instanceof(Data) AND
  top.identifier() == ocls.ID AND
  top_ocr.identifier() == dtp.ID AND
  top_ocr.type.identifier() == rdfsr.ID
MAKE
  OWLRestriction res
  res.type = OWLRestriction
  ocls.subClassOf = res
  res.onProperty = dtp
  res.someValuesFrom = rdfs
    
```

(그림 9) 어커런스와 데이터타입프로퍼티제한 매핑규칙

5.3 관계역할과 프로퍼티제한, 어커런스와 프로퍼티제한의 매핑

토픽맵에서 관계역할은 OWL에서 프로퍼티제한과 매핑될 수 있다. 토픽맵에서 어커런스타입의 토픽은 OWL에서 객체프로퍼티 제한 또는 데이터프로퍼티 제한과 매핑될 수 있다. 어커런스타입의 토픽이 위치자의 인스턴스로서 리소스이면 객체프로퍼티 제한과 매핑되고, 데이터의 인스턴스로서 리소스이면 데이터프로퍼티 제한과 매핑된다.<그림 7, 8, 9참조>

```

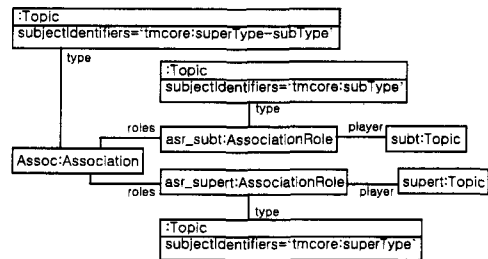
RULE Role_PropRestriction(as, asr_s, asr_o, ocls, oop_s,
rdfs_o, res)
FORALL
  Association as
  AssociationRole asr_s //asr_subject
  AssociationRole asr_o //asr_object
  OWLClass ocls
  OWLObjectProperty oop_s //oop_subject
  RDFSResource rdfs_o //rdfs_object
WHERE
  as.roles.contains(asr_s) AND
  as.roles.contains(asr_o) AND
  as.type.identifier() == ocls.ID AND
  asr_s.type.identifier() == oop_s.ID AND
  asr_o.type.identifier() == rdfs_o.ID
MAKE
  OWLRestriction res
  res.type = OWLRestriction
  ocls.subClassOf = res
  res.onProperty = oop_s
  res.someValuesFrom = rdfs_o
    
```

(그림 7) 관계역할과 프로퍼티제한 매핑규칙

5.4 토픽맵 계층구조의 매핑

토픽맵은 슈퍼타입과 서브타입의 계층구조를 갖는다. 이러한 supertype-subtype의 관계는 OWL에서는 subClassOf 관계와 매핑된다.

그림 10은 토픽맵에서의 계층구조를 나타내고, 그림 11은 토픽맵의 타입 계층구조와 OWL에서 subClassOf와의 매핑규칙을 보인다.



(그림 10) 토픽맵의 subtype-supertype 구조

```

RULE Ocr_ObjPropRestriction(top, top_ocr, ocls, oop,
rdfsr, res)
FORALL
  Topic top
  Occurrence top_ocr
  OWLClass ocls
  OWLObjectProperty oop
  RDFSResource rdfs
WHERE
  top.occurrences.contains(top_ocr) AND
  top_ocr.resource instanceof(Locator) AND
  top.identifier() == ocls.ID AND
  top_ocr.identifier() == oop.ID AND
  top_ocr.type.identifier() == rdfsr.ID
MAKE
  OWLRestriction res
  res.type = OWLRestriction
  ocls.subClassOf = res
  res.onProperty = oop
  res.someValuesFrom = rdfs
    
```

(그림 8) 어커런스와 객체프로퍼티제한 매핑규칙

```

RULE ClassHierarchy(as, asr_sup, asr_sub, sup_ocls,
sub_ocls)
FORALL
  Association as
  AssociationRole asr_sup //asr_supertype
  AssociationRole asr_sub //asr_subtype
  OWLClass sup_ocls //supertype_ocls
  OWLClass sub_ocls //subtype_ocls
WHERE
  as.subjectIdentifier=='tmcore:superType-subType'
AND
  as.roles.contains(asr_sup) AND
  asr_sup.subjectIdentifier=='tmcore:superType' AND
  as.roles.contains(asr_sub) AND
  asr_sub.subjectIdentifier=='tmcore:subType' AND
  asr_sup.identifier()==sup_ocls.ID AND
  asr_sub.identifier()==sub_ocls.ID
MAKE
  asr_sub.subClassOf = asr_sup;
    
```

(그림 11) supertype-subtype과 subClassOf의 매핑규칙

6. OWL에서 토픽맵으로 매핑

6.1 프로퍼티의 매핑

OWL 객체프로퍼티는 토픽맵의 관계역할과 매핑된다. 반면, OWL 데이터타입 프로퍼티는 토픽맵의 어커런스와 매핑된다. 이것은 왕복의(round trip) 매핑에서 약간의 비대칭 결과를 가져올 수 있다. 따라서 리소스 값과 같은 위치자를 갖는 어커런스는 OWL을 통하여 왕복매핑의 결과로서 관계가 될 것이다.<그림 12, 13참조>

```

RULE ObjProp_Role (ar, oop)
FORALL
  OWLObjectProperty oop
WHERE
  oop.type.contains(OWLObjectProperty)
MAKE
  AssociationRole ar
  ar.sourceLocator.value = oop.ID
  ar.sourceLocator.notation = 'URI';
    
```

(그림 12) 객체프로퍼티와 관계역할의 매핑규칙

```

RULE DTyProp_Occr (ocr, dtp)
FORALL
  OWLDatatypeProperty dtp
WHERE
  oop.type.contains(OWLDatatypeProperty)
MAKE
  Occurrence ocr
  ocr.sourceLocator.value = dtp.ID
  ocr.sourceLocator.notation = 'URI';
    
```

(그림 13) 데이터타입 프로퍼티와 어커런스의 매핑규칙

6.2 프로퍼티 제한사항의 매핑

OWL 프로퍼티 제한(Restrictions)은 토픽맵에서 관계역할과 어커런스를 각각 토픽과 관련시키는데 사용된다. 따라서 OWL의 객체프로퍼티 제한은 토픽맵의 관계역할과 매핑되고, OWL의 데이터타입 프로퍼티 제한은 토픽맵의 어커런스와 매핑된다.<그림 14, 15참조>

```

RULE ObjPropRest_Role (res, ocls, oop_s, rdfs_o,
ocls_top, as, asr_s, asr_o)
FORALL
  OWLRestriction res
  OWLClass ocls
  OWLObjectProperty oop_s
  RDFSResource rdfs_o
  Topic ocls_top
WHERE
  ocls.subClassOf.contains(res) AND
  ocls_top.identifier() == ocls.ID AND
  res.onProperty == oop_s AND
  res.somValuesFrom == rdfs_o
MAKE
  Association as
  AssociationRole asr_s
  AssociationRole asr_o
  as.type = ocls_top
  asr_s.subjectIdentifier = oop_s.ID
  as.roles = asr_s
  asr_o.subjectIdentifier = rdfs_o.ID
  as.roles = asr_o;
    
```

(그림 14) 객체프로퍼티제한과 관계역할의 매핑규칙

```

RULE DTyPropRest_Occr (res, ocls, dtp_s, rdfs_o,
    
```

```

ocls_top, ocr)
FORALL
  OWLRestriction res
  OWLClass ocls
  OWLDatatypeProperty dtp_s
  RDFSResource rdfs_o
  Topic ocls_top
WHERE
  ocls.subClassOf.contains(res) AND
  ocls_top.identifier() == ocls.ID AND
  res.onProperty == dtp_s AND
  res.somValuesFrom == rdfs_o
MAKE
  Occurrence ocr
  ocr.type = rdfs_o
  ocr.subjectIdentifier = dtp_s.ID
  ocr.parent = ocls_top;
    
```

(그림 15) 데이터타입제한과 어커런스의 매핑규칙

6.3 OWL 계층구조의 매핑

OWL은 클래스 계층구조를 갖는다. OWL의 클래스 계층구조는 토픽맵에서 타입 계층구조로 매핑된다. 즉, OWL에서 subClassOf 관계는 토픽맵에서 그림 16과 같이 subtype-supertype 관계로 나타낼 수 있다.

```

RULE ClassHierarchy(as, asr_sup, asr_sub, sup_ocls,
sub_ocls)
FORALL
  OWLClass sup_ocls
  OWLClass sub_ocls
  Topic sup_top
  Topic sub_top
WHERE
  sub_ocls.subClassOf.contains(sup_ocls) AND
  sup_top.identifier() == sup_ocls.ID AND
  sub_top.identifier() == sub_ocls.ID
MAKE
  Association as
  AssociationRole asr_sup
  AssociationRole asr_sub
  as.subjectIdentifier = 'tmcore:superType-subType'
  as.roles = asr_sup
  asr_sup.subjectIdentifier = 'tmcore:superType'
  asr_sup.topicPlayingRole = sup_top
  as.roles = asr_sub
  asr_sub.subjectIdentifier = 'tmcore:subType'
  asr_sub.topicPlayingRole = sub_top;
    
```

(그림 16) subClassOf 구조와 subtype-supertype구조의 매핑규칙

7. 결 론

웹 기술이 컴퓨터의 기반기술이 됨에 따라, 웹은 온톨로지의 개념 표현과 추론기능을 기반으로한 시맨틱 웹으로 전환되고 있으며, 온톨로지는 정보의 표현과 처리를 지식 표현과 추론 처리로 고도화하여 차세대 컴퓨터 기술의 핵심 역할을 하고 있다. 현재 온톨로지 구축에 양대 주류로서 토픽맵과 OWL이 있다. 토픽맵과 OWL은 많은 유사점을 갖지만, 역사적으로 기술적으로 의도하는 목적에서 차이가 있고, 제약언어, 데이터모델, 구문에 있어서 각기 다른 표현을 가지고 있다. 하지만 지식 표현 온톨로지의 구체화에 사용되고, DL의 이론에 기반한 의미구조를 가지며, XML 상호교환 포맷을 갖는 공통의 특성이 있다.

따라서 본 논문은 이질적인 온톨로지 언어인 토픽맵과 OWL의 구성 요소를 분석하고 서로 다른 온톨로지 언어에서 구축된 지

식정보를 웹상에서 공유하고 상호 커뮤니케이션 및 통합이 가능하도록 메타모델의 매핑을 수행하였다. 매핑의 수행을 위하여 먼저 각 메타모델내의 매핑에 적용될 개념 추출 및 요소 사이의 관계를 분석하였다. 이질적인 환경의 메타모델 사이의 매핑에서 해결해야할 근본적인 문제는 각 메타모델에 포함되어 있는 대응 요소들이 임의적이고 불규칙한 점에 있다. 따라서 매핑 과정에서 요소들의 손실이 발생할 수 있는데 이를 방지하기 위하여 소스 모델에서 목표모델로의 매핑 과정을 추적할 수 있는 트래킹 시스템을 적용시키는 것이다.

기존의 매핑연구는 의미보다는 구조적인 측면에 무게를 두었고, 정보지향 관점보다는 통합 프로세스의 트랜잭션지향성을 추구하였지만, 본 연구는 구조적인 점 뿐만 아니라 의미적인 측면인 정보지향적인 매핑을 수행하였다는 점에서 기존 연구와의 차별을 둔다. 본 연구의 결과는 이질적인 구조기반에서 구축된 지식정보의 상호 커뮤니케이션이 가능하게 하는 토대가 됨으로서 온톨로지 언어의 유연성을 제고시킬 수 있고 온톨로지 기반으로 하여 개념적 표현 모델의 매핑에 근거하기 때문에 시맨틱 웹의 발전에 활용이 가능할 수 있다.

참고문헌

- [1] 홍현술, "DL기반에 의한 이질적 시스템간의 메타모델 매핑", 한국정보처리학회 추계학술대회 논문집, 제12권 제2호, 2005년 11월, pp.423-426.
- [2] Garshol, Lars Marius; Living With Topic Maps, <http://www.ontopia.net/topicmaps/materials/tmrdf.html>, 2003.
- [3] Rath, H. Holger; The Topic Maps Handbook, Empolis Arvato, Knowledge Management
- [5] Steve Pepper, "Navigating Haystacks and Discovering Needles: Introduce the New TopicMap Standard", Markup Languages 1(4), 1999, pp. 47~74
- [6] Michel Biezunski, Martin Bryan, Steve Newcomb, ISO/IEC 13250 TopicMaps
- [7] Steve Pepper, Graham Moore, "XML Topic Maps(XTM) 1.0", TopicMaps.Org.
- [8] F.M. Domini, M. Lenzerini, D. Nardi, and A. Schaerf, Reasoning in description logics, In Gerhard Brewka, editor, *Principles of Knowledge Representation, Studies in Logic, Language and Information*, pp.193-238, CSLI Publications, 1996.
- [9] Simith, M. K. C. Welth, and D. L. McGuinness, Web Ontology Language (OWL) Guide Version 1.0, 2003}
- [10] M. Dean, D. Connolly, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. Web Ontology Language(OWL) Reference Version 1.0. Working draft, W3C, November 2002.
- [11] W3C, "OWL Web Ontology Language Reference", <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>

<부 록>

토픽맵과 OWL의 매핑결과 예

Topic Map XTM (1)
<topic id="Cap"> <baseName><baseNameString>Cap</baseNameString>

</baseName> </topic>
OWL RDF/XML (1)
<Class rdf:ID="Cap"> <label>Cap</label> </Class>

Topic Map XTM (2)
<topic id="PersonalCap"> <baseName> <baseNameString>Personal Cap</baseNameString> </baseName> <instanceOf> <topicRef xlink:href="#Cap"/> </instanceOf> </topic>
OWL RDF/XML (2)
<Class rdf:ID="PersonalCap"> <label>Personal Cap</label> <subClassOf rdf:resource="#Cap"/> </Class>

Topic Map XTM (3)
<topic id="KimsWhiteCap"> <baseName> <baseNameString>Kim's White Cap</baseNameString> </baseName> <instanceOf> <topicRef xlink:href="#PersonalCap"/> </instanceOf> </topic>
OWL RDF/XML (3)
<personalCap rdf:ID="KimsWhiteCap"/>

Topic Map XTM (4)
<association id="KimsWhiteCapsIsWhite"> <instanceOf> <topicRef xlink:href="#hasColor"/> </instanceOf> <member> <roleSpec> <topicRef xlink:href="ColorOf"/> </roleSpec> <topicRef xlink:href="#KimsWhiteCap"/> </member> <member> <roleSpec> <topicRef xlink:href="#IsColor"/> </roleSpec> <topicRef xlink:href="#White"/> </member> </association>
OWL RDF/XML (4)
<ObjectProperty rdf:ID="hasColor"/> <Description rdf:about="KimsWhiteCap"> <subClassOf> <Restriction> <onProperty rdf:resource="hasColor"/> <hasValue> <Color rdf:ID="White"/> </hasValue> </Restriction> </subClassOf> </Description>