

RDF 모델을 컬러 페트리 넷으로 변환하는 알고리즘

임재걸*, 권기용*, 주재훈**, 이강재***

An Algorithm to Transform RDF Models into Colored Petri Nets

Jaegel Yim *, Kiyong Gwon *, Jaehun Joo **, Kangjai Lee ***

요약

본 논문은 온톨로지 작성에 사용되는 RDF(Resource Description Framework) 모델을 컬러 페트리 넷 모델로 변환하는 알고리즘을 제안한다. 제안하는 알고리즘은 RDF 모델의 클래스와 프로퍼티들을 컬러 페트리 넷의 플레이스로 매핑하여 RDF 모델의 의미를 컬러 페트리 넷의 토폴로지로 변환한 다음 클래스와 프로퍼티들 간의 관계를 토큰의 전이로 나타냄으로써 RDF의 문장들을 컬러 페트리 넷에 반영한다. RDF 문장들을 반영하는 기본적인 방법은 주어와 객체를 나타내는 토큰들의 순서쌍으로 구성된 토큰을 생성하여, 술어를 나타내는 자리로 전이하는 방법이다. 주어진 RDF 모델을 제안하는 방법으로 실제 CPNTools를 이용하여 컬러 페트리 넷 모델로 변환하고, RDF 질의에 대한 추론과 답을 CPNTools에서 구하는 사례를 보였다.

Abstract

This paper proposes an algorithm to transform RDF(Resource Description Framework) models for ontology into CPN(Colored Petri Net) models. The algorithm transforms the semantics of the RDF model into the topology of the CPN by mapping the classes and the properties of the RDF onto the places of the CPN model then reflects the RDF statements on the CPN by representing the relationships between them as token transitions on the CPN. The basic idea of reflecting the RDF statements on the CPN is to generate a token, which is an ordered pair consisting of two tokens (one from the place mapped into the subject and the other one from the place mapped into the object) and transfer it to the place mapped into the predicate.

We have actually built CPN models for given RDF models on the CPNTools and inferred and extracted answers to the RDF queries on the CPNTools.

▶ Keyword : ontology, RDF(Resource Description Framework), RDFS(RDF Schema), Colored Petri Net

• 제1저자 : 임재걸 교신저자 : 이강재

• 투고일 : 2008. 10. 29, 심사일 : 2008. 10. 30, 게재확정일 : 2008. 12. 12.

* 동국대학교(경주) 컴퓨터멀티미디어학과 ** 동국대학교(경주) 전자상거래학과 *** 수원과학대학 컴퓨터정보과

※ 이 논문은 2008년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음. (KRF 2008 314 D00362)

I. 서론

본 논문은 RDF(Resource Description Framework) 모델을 컬러 페트리 넷(CPN, Colored Petri Net)으로 변환하는 방법을 소개한다. 또한 RDF 질의를 컬러 페트리 넷 시뮬레이션으로 치환하여 RDF 질의에 적절하게 답할 수 있음을 보인다. 즉, RDF 질의를 컬러 페트리 넷 시뮬레이션으로 치환하는 방법을 본 논문은 제공한다.

RDF(Resource Description Framework)는 월드와이드웹(World Wide Web) 컨소시엄이 정의한 웹 상의 자원을 표현하는 규격이다. 따라서, RDF로 표현된 정보는 기계적인 해석이 가능하다. 사용자는 RDFS(RDF Schema)라는 언어로 자신이 정의한 용어를 사용하여 RDF를 작성한다. 나아가서 RDFS로 어떤 성질이 어떤 사물에 적용되고 어떤 값을 가질 수 있는지, 그리고 사물 간에 어떤 관계가 있는지도 표현한다. 즉, RDFS로 도메인 지식을 표현한다. 따라서, 잘 작성된 RDF 모델은 일종의 온톨로지(ontology)이다.

페트리 넷은 1962년에 처음 소개된 이후 시스템 성능 테스트와 통신 규약의 일관성 및 타당성 테스트 등을 비롯한 컴퓨터 관련 전 분야에서 시스템 모델과 분석 도구로 널리 사용되고 있다. 이와 같이 페트리 넷을 이용한 연구들이 활발히 진행되는 이유는 페트리 넷 모델은 구축하기가 용이하고, 구축된 모델을 분석하는 수학적 방법이 널리 연구되고 있기 때문이다[1].

컬러 페트리 넷은 페트리 넷의 일종으로 복잡한 현상을 표현할 때 페트리 넷을 구성하는 요소들의 수가 급격히 증가하는 현상을 피하는 수단으로 등장하였다. 현재 컬러 페트리 넷을 구축하고, 시뮬레이션을 실행 할 수 있는 환경을 제공하는 소프트웨어 도구도 다수가 개발되어 있다. 이들 중 가장 널리 사용되는 도구 중에 개인용 컴퓨터용 도구인 CPNTools[2]가 있다. 본 논문에서는 RDF 모델을 컬러 페트리 넷으로 표현하는 방법을 정의하고, 제안하는 방법을 적용하여 CPNTools로 컬러 페트리 넷 모델을 구축한 사례를 보인다.

RDF 모델은 월드와이드웹 자원을 기계가 이해할 수 있는 형태로 표현한다. 따라서, RDF 모델을 해석하고 필요한 정보를 유추할 수 있는 소프트웨어 개발이 가능하다. 달리 말하면, RDF 모델과 이를 해석하는 소프트웨어는 별개로 개발된다. 이에 반하여 컬러 페트리 넷의 경우에는 RDFS로 표현된 클래스(class)들과 프로퍼티(property)들 간의 관계가 컬러 페트리 넷의 토폴로지에 반영되어 컬러 페트리 넷 상의 시뮬레이션으로 필요한 정보를 유추할 수 있다. 즉, 컬러 페트리

넷 모델에는 지식 표현과 추론 엔진이 통합되어 있다.

본 논문은 RDF를 컬러 페트리 넷으로 변환하는 알고리즘을 제안하고, 제안한 방법을 적용하여 실제 RDF 모델에 대한 컬러 페트리 넷 모델을 CPNTools를 이용하여 구축하고, 시뮬레이션을 실행하여 RDF 질의에 대한 답을 구함으로써 변환 알고리즘의 적합성을 보였다.

II. 관련 연구

1. RDF 모델

XML은 마크업(markup)을 정의하는 목적으로 사용할 수 있는 일반적인 언어이다. 그래서 응용 프로그램 간에 데이터를 교환하는 목적으로 XML 문서가 일반적으로 사용된다. 그러나 XML은 데이터의 의미를 표현하는 기능이 없다. RDF는 이러한 XML의 부족한 부분이 보강된 데이터 모델이다. 즉, 사용자는 RDFS라고 불리는 언어를 이용하여 필요한 용어를 정의하고, 어떤 종류의 개체에 어떤 프로퍼티가 적용되는지, 그리고 개체들 간에 어떤 관계가 있는지 정의할 수 있다.

RDF의 목적은 월드와이드웹의 자원들을 묘사하는 것으로, 자원을 묘사하는 기본 형식으로 주어(subject), 술어(predicate), 객체(object)로 구성된 문장(statement)을 사용한다.

예를 들어, "352로 지칭되는 자원의 이름은 Grigoris Antoniou이다"라는 문장을 나타내는 rdf의 예는 다음과 같다.

```
<rdf:Description rdf:about="352">
  <uni:name>Grigoris Antoniou</uni:name>
</rdf:Description>
```

RDF 모델은 RDFS 레이어와 RDF 레이어로 구성되며, RDFS 레이어는 RDF 데이터 모델에 사용되는 용어를 정의한다. RDFS에 사용되는 기초적인 용어에 핵심 클래스와 핵심 프로퍼티(property)가 있다. 핵심 클래스에는 rdf:Resource, rdfs:Class, rdfs:Literal, rdf:Property, rdf:Statement 등이 있으며 lecturer라는 단어가 클래스 이름으로 사용된다. 고 정의하는 rdfs:Class의 사용 예는 다음과 같다:

```
<rdfs:Class rdf:ID="lecturer" > ...
```

핵심 프로퍼티의 종류로 rdf:type, rdfs:subClassOf,

rdfs:subPropertyOf가 있으며, rdf:type은 주어가 지칭하는 자원이 객체 자원(클래스임)의 인스턴스(instance)임을 정의한다. rdfs:subClassOf는 주어가 지칭하는 자원들이 모두 객체가 지칭하는 자원에 속함을 의미한다. 또한 subPropertyOf의 의미와 사용 예도 subClassOf의 경우와 비슷하다.

프로퍼티를 제한하는 핵심 프로퍼티에는 rdfs:domain과 rdfs:range가 있다. 다음은 phone이라는 성질이 스태프 멤버에 적용되고(즉, phone의 domain이 스태프 멤버이고), 값은 literal(range는 literal)임을 정의하는 사용 예이다.

```
<rdf:Property rdf:ID="phone">
  <rdfs:domain rdf:resource="#staffMember"/>
  <rdfs:range rdf:resource="#&rdf:Literal"/>
</rdf:Property>
```

RDF의 목적은 월드와이드웹의 자원들을 묘사하는 것으로, 자원을 묘사하는 기본 형식으로 자원을 나타내는 주어, 속성을 나타내는 술어, 속성의 값을 나타내는 객체로 구성된 삼형식 구조를 갖는 문장을 사용한다. RDF에서는 문장으로 응용 영역의 모든 자원들을 묘사할 수 있지만, 경제성을 높이기 위하여 문장의 수를 줄이거나 사과의 단위를 적절히 반영하기 위하여 다음과 같은 여러 가지 기본 단어를 이용한다.

1.1 rdf:Resource

rdf에서는 동일한 이름이 동일한 사람을 지칭한다는 보장이 없다. 그래서 유일한 식별자를 사용하는 방법을 채용한다. 다음은 "CIT1111"이 스태프 번호 "949318"에 의하여 가르쳐지고, 그의 이름은 "David Billington"이라고 정의하는 rdf:resource의 예이다.

```
<rdf:Description rdf:about="CIT1111">
  <uni:isTaughtBy rdf:resource="949318"/>
</rdf:Description>
```

```
<rdf:Description rdf:about="949318">
  <uni:name>David Billington</uni:name>
</rdf:Description>
```

1.2 Reification

RDF에서는 다음과 같이 문장에 대한 문장을 작성하는 것이 가능하다: "철수는 '영희가 xxx 웹 페이지의 주인이다'이라고 생각한다." 이러한 기능을 reification이라 하며, 언급되는

문장을 의미하는 보조 객체를 생성하고, 이 객체의 주어, 술어, 객체를 명시하는 방법을 이용한다. 위의 문장은 RDF에서 다음과 같이 정의된다.

```
<rdf:Description rdf:about="철수">
  <rdf:believesIn rdf:resource="#Statement123"/>
</rdf:Description>
```

```
<rdf:Statement rdf:about="Statement123">
  <rdf:subject>영희</rdf:subject>
  <rdf:predicate>owner</rdf:predicate>
  <rdf:object>xxx</rdf:object>
```

1.3 Container

여러 자원들의 집합을 하나의 단위로 참조할 수 있도록 하기 위하여, RDF는 container라는 요소를 제공한다. 세 가지 유형의 container가 있는데 이들은 rdf:Bag, rdf:Seq, rdf:Alt이다. rdf:Bag은 여러 자원들로 구성되며, 구성 원소들에 우선순위가 없다. 이에 반하여 rdf:Seq의 경우에는 구성 원소들에 순위가 있다는 점이 rdf:Bag과 다른 점이다. 한편 rdf:Alt의 경우에는 구성 원소 중 하나만 선택된다. 식별자가 "949352"인 강사가 가르치는 과목이 "CIT1112"와 "CIT3116"임을 정의하는 container의 예는 다음과 같다.

```
<uni:lecturer rdf:about="949352">
  <uni:coursesTaught>
    <rdf:Bag>
      <rdf:_1 rdf:resource="CIT1112"/>
      <rdf:_2 rdf:resource="CIT3116" .../>
```

1.4 Collection

Container의 구성 원소를 지정한 원소만으로 한정하는 방법으로 collection이 사용된다. 다음은 "CIT2112"가 "949111", "949352", "949315"에 의하여 가르쳐지고 다른 사람은 가르치는 사람이 없다고 정의하는 문장이다.

```
<rdf:Description rdf:about="CIT2112">
  <uni:isTaughtBy rdf:parseType="collection">
    <rdf:Description rdf:about="949111"/>
    <rdf:Description rdf:about="949352"/>
    <rdf:Description rdf:about="949318"/>
  </uni:isTaughtBy>
</rdf:Description>
```

RDF 모델은 위에서 보인 4가지 예처럼 XML 문법으로 기술할 수 있지만, RDFS를 나타내는 부분과 RDF를 나타내는 부분으로 구분된 그림으로도 표현할 수 있다.

RDFS 그림 부분에서는 프로퍼티를 사각형으로, 클래스를 타원으로 나타내고, RDF 그림 부분에서는 인스턴스를 타원으로 표현한다.

아래 <그림 1>은 RDF 모델의 예이다[3]. 이 예는 RDFS 레이어에서 교수는 학문적 스태프들의 일부를 "professor는 academic staff member의 subclass"라고 표현하였으며, 이것은 바로 professor라는 용어를 정의하는 효과를 나타낸다.

비슷한 방법으로 isTaughtBy를 비롯한 다양한 용어들이 RDFS 레이어에 정의되어 있음을 볼 수 있다. 그리고 RDF 레이어에서는 실제로 "David"가 이산수학을 담당한다는 사실을 "Discrete Mathematics isTaughtBy David Billington"이라고 기술한다.

문장의 주어와 객체는 그림에서 유향 간선의 시작 정점과 끝점으로 매핑되고, 술어는 그림에서 유향 간선의 레이블로 표현되어 있으므로, 이 그림을 일상 언어나 XML 형식의 문장으로 쉽게 변환할 수 있음을 알 수 있다.

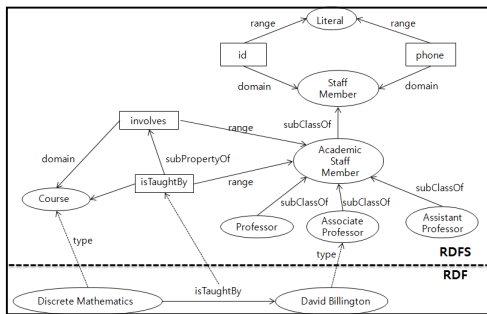


그림 1. RDF 레이어와 RDFS 레이어
Figure 1. An example RDF model consisting of RDF layer and RDFS layer

2. 컬러 페트리 넷

페트리 넷은 1962년 칼 아담 페트리의 박사학위 논문[4]에서 태동되었으며, 1970년부터 1975년까지 미국 MIT에서 가장 활발하게 페트리 넷 이론을 연구하였다. 1970년 후반부터는 유럽에서 페트리 넷 연구가 활발히 전개되어 1980년부터 매년 페트리 넷 이론과 응용(Application and Theory of Petri Nets)이라는 세계 학술대회가 열리고 있다.

페트리 넷은 다양한 시스템에 적용되는 그래프 모델이며, 또한 수학적 모델이기도 하다. 그래서 순서도(flowchart)나

구역도표(block diagram)처럼 시스템을 시각적으로 표현하여 의사소통을 원활화하는 목적으로 사용되고, 시스템의 동적인 성질을 시물레이션하는 도구로도 사용되었으며, 시스템의 성질을 수학적으로 분석하는 방편으로도 사용되어 왔다.

컬러 페트리 넷은 페트리 넷을 구성하는 토큰의 종류가 증가함에 따라 플레이스(place)와 트랜지션(transition)의 수가 폭발적으로 증가하는 것을 피하기 위하여 고안된 변종 페트리 넷이다[1, 5].

컬러 페트리 넷을 용이하게 작성할 수 있는 환경을 제공하고 시물레이션을 수행하여주는 소프트웨어 도구 중에 CPNTools[2]라는 소프트웨어가 있다. CPNTools를 이용하여 병렬시스템의 모델을 구축하고, 타당성을 검증하는 연구는 무수히 많다. 본 논문에서는 이 도구를 이용하여 주어진 RDF 모델을 나타내는 컬러 페트리 넷을 작성하고, 질의에 답하는 예를 보인다.

이처럼 다양한 분야에서 페트리 넷이 이용되어 왔으며, [6]은 규칙기반 시스템을 나타내고 분석하기에 용이한 '일반화 페트리 넷'이라는 일종의 페트리 넷 변종을 제안하였다. 페트리 넷을 표현하는 정형화된 방법 중에는 메타 모델(metamodel), 통합 모델 언어(UML: Unified Modeling Language) 등이 있는데, [7]은 통합 모델 언어, RDFS, 웹 온톨로지 언어(OWL: Web Ontology Language)를 이용한 페트리 넷 온톨로지를 제공하였다. 특히 [8]에서는 웹 온톨로지 언어 논리 기술(OWL DL)을 Predicate-Transition Net으로 변환하는 방법을 소개하였다. 기타 관련 연구로 RDF와 같은 언어로 기술된 온톨로지의 스키마의 버전을 관리하는 기법을 제안한 연구[9]와 RDF와 RDF 스키마로 기술된 시멘틱 웹 언어에 대한 효율적인 검색 기법을 소개한 연구[10] 등 무수히 많다.

본 논문에서는 RDF 모델을 컬러 페트리 넷으로 변환하는 방법을 새롭게 정의하고, 또한 CPNTools로 변환 컬러 페트리 넷에서 시물레이션을 수행하여 RDF 질의에 대한 답을 구함을 보임으로써 해서 변환 방법의 적절성을 입증한다는 점에서 기존의 연구와는 다르며, 지금까지 이런 형태의 연구는 없었다.

III. 변환 방법

이 장에서는 RDF 모델을 컬러 페트리 넷으로 변환하는 방법을 제안한다.

제안하는 방법은 RDFS 부분에서 핵심 클래스들을 식별하여 각각을 컬러 페트리 넷의 플레이스로 매핑한 다음, 관계를

나타내는 핵심 프로퍼티들인 subClassOf, subPropertyOf 를 식별하여 주어 나타내는 플레이스의 토큰을 객체를 나타내는 플레이스로 전이하도록 트랜지션에 나타낸다. 이때, 클래스와 프로퍼티가 resource의 subClass임도 나타낸다. 다음은 프로퍼티의 영역(domain)과 치역(range)을 가려내, 영역을 나타내는 플레이스의 토큰과 치역을 나타내는 플레이스의 토큰으로 구성된 순서쌍 토큰을 프로퍼티를 나타내는 플레이스로 전이하는 트랜지션을 도입한다.

RDFS로 컬러 페트리 넷의 구조를 구축한 다음, RDF 문장들을 다음과 같이 컬러 페트리 넷에 반영한다. 첫째, 문장의 주어, 술어, 그리고 객체를 가려낸 다음 술어가 핵심 프로퍼티의 타입이면 주어가 객체의 한 유형임을 반영하기 위하여 주어를 나타내는 토큰을 객체를 나타내는 플레이스에 놓도록 트랜지션을 만든다.

둘째, rdf:container의 경우에는 Bag, Seq, Alt의 세 가지 유형 각각에 대하여 다음과 같이 컬러 페트리 넷으로 표현한다.

Bag는 여러 문장을 하나로 표현한 것임으로 다시 여러 문장으로 나누어 각각을 표현하면 된다.

예를 들어,

```

<rdf:Description rdf:about="318">
<uni:coursesTaught>
  <rdf:Bag>
    <rdf:_1 rdf:resource="C111"/>
    <rdf:_2 rdf:resource="C112"/>
  </rdf:Bag>
  ...

```

이것은 "318은 C111을 가르친다."와 "318은 C112를 가르친다."라는 두 문장으로 분리할 수 있다. 컬러 페트리 넷에서는 도메인이 항상 제한적으로 주어짐으로 이러한 방법은 곧 RDF의 collection을 표현한 것과 동일한 효력을 나타낸다.

Alt 유형은 다음에 보이는 OK 함수를 사용하여 균등한 비율로 토큰을 생성하게 한다.

```

color Ten0 = int with 0..10;
color Ten1 = int with 1..10;
Fun OK(s:Ten0, r:Ten1)=(r <= s);

```

예를 들어, 다음과 같은 경우에는 OK 함수가 참일 확률을 50%로 만들어(OK함수의 s값을 5로 하여 줌) 함수 값이 참

이면 "352"를 아니면 "318"로 토큰을 만들도록 한다.

```

<rdf:Alt>
  <rdf:li rdf:resource="352"/>
  <rdf:li rdf:resource="318"/>
</rdf:Alt>

```

Seq는 순서화된 container이다. 컬러 페트리 넷에서 순서가 있는 원소들을 표현하기에 적당한 자료구조는 리스트이다.

예를 들어 다음과 같은 rdf:seq는 <그림 2>와 같은 컬러 페트리 넷으로 변환된다. <그림 2>에서 1' []은 빈 리스트를 의미한다. <그림 1>의 트랜지션이 격발하려면 Seq가 이 빈 리스트로, i가 1로, (i, CID)가 (1, "C111")로 대체되어야 한다. 격발결과 i+1, 즉 2가 i 자리로 되돌아가고, Seq^{^(CID)}, 즉 ["C111"]이 빈 리스트 자리로 되돌아간다. 이 트랜지션이 또 격발하면 비슷한 방법으로 ["C111" "C112"]라는 리스트가 생성된다.

```

<rdf:Seq>
  <rdf:li rdf:resource="352"/>
  <rdf:li rdf:resource="318"/>
</rdf:Seq>

```

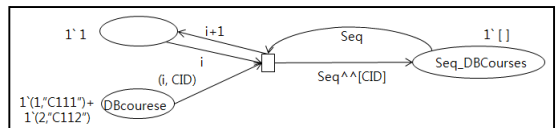


그림 2. rdf:Seq의 컬러 페트리 넷 모델
Figure 2. A colored Petri net model for rdf:Seq

셋째, 나머지 rdf 모든 문장은 주어, 술어, 객체를 각각 플레이스에 매핑하고, 주어와 객체 플레이스를 입력으로 하고 술어 플레이스를 출력으로 하는 트랜지션의 guard로 문장을 매핑하여 주어와 객체를 나타내는 토큰의 순서쌍인 새로운 토큰을 술어를 나타내는 플레이스로 전이하도록 컬러 페트리 넷에 나타낸다.

이와 같은 방법으로 rdf:resource 속성이나 reification 등도 컬러 페트리 넷에 나타낼 수 있다. 이러한 컬러 페트리 넷 변환 방법을 기술한 알고리즘이 <표 1>이다.

표 1. RDF/RDFS의 컬러 페트리 넷 변환 알고리즘
Table 1. Algorithm to transform a RDF model into a CPN

<p>RDF/RDFS 모델의 CPN 변환 알고리즘</p> <ul style="list-style-type: none"> - 입력 : RDF/RDFS 모델 - 출력 : 컬러 페트리 넷 모델 <ol style="list-style-type: none"> 1. RDFS에서 resource, class, literal, property, statement 등 핵심 클래스들을 가려내 컬러 페트리 넷의 플레이스로 매핑한다. 2. RDFS에서 subClassOf 관계와 subPropertyOf 관계를 subject class/property를 나타내는 플레이스의 토큰을 object class/property를 나타내는 플레이스로 전이하는 트랜지션을 도입하여 컬러 페트리 넷에 나타낸다. 3. RDFS에서 프로퍼티의 영역과 치역을 가려내어, 영역을 나타내는 플레이스의 토큰과 치역을 나타내는 플레이스의 토큰으로 구성된 순서쌍 토큰을 프로퍼티를 나타내는 플레이스로 전이하는 트랜지션을 도입하여 컬러 페트리 넷에 나타낸다. 4. 주어, 술어, 객체로 구성된 RDF 문장 각각에 대하여 주어, 술어, 객체를 각각 플레이스에 매핑하고, 주어와 객체를 나타내는 토큰으로 구성된 순서쌍 토큰을 술어를 나타내는 플레이스로 전이하는 트랜지션을 컬러 페트리 넷에 도입한다. <p>(가) 술어가 rdf:type이면 술어를 플레이스에 매핑하지 않고, 주어를 나타내는 토큰을 객체 클래스를 나타내는 플레이스로 전이</p> <p>(나) container의 경우에는 세 가지 유형 Bag, Seq, Alt 각각에 대하여 컬러 페트리 넷으로 표현</p>

IV. 변환 예

앞 장에서 제안한 알고리즘을 적용하여, RDF 모델을 컬러 페트리 넷으로 변환하는 사례를 살펴본다.

1. RDF 모델의 컬러 페트리 넷 표현

〈그림 1〉에 보이는 RDF 모델을 컬러 페트리 넷으로 변환하면 〈그림 3〉과 같다. 〈그림 3〉에서 소스 트랜지션 t1, t2을 제외한 나머지 부분은 알고리즘의 단계 1부터 단계 3에서 처리된다.

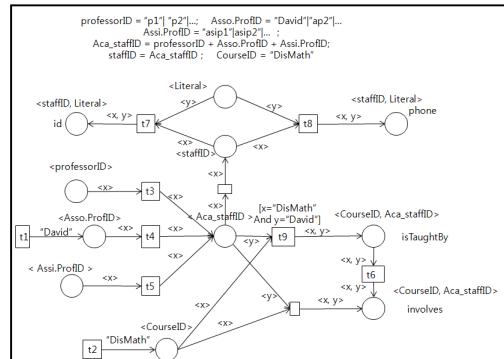


그림 3. 〈그림 1〉의 RDF 모델의 컬러 페트리 넷 표현
Figure 3. A CPN representing the RDF model in Figure 1

단계 1에서 id, Literal 등 모든 클래스와 프로퍼티 각각에 매핑할 플레이스들을 도입한다. 단계 2에서는 subClassOf (t3, t4, t5 등)와 subPropertyOf(t6)들을 트랜지션으로 나타낸다. 단계 3에서는 치역과 영역을 t7과 t8처럼 컬러 페트리 넷에 반영한다. 단계 4에서는 t1과 t2를 도입하여 David와 DisMath가 각각 professor와 course 유형임을 나타낸다. 그리고 "DisMath isTaughtBy David"를 t6의 가드에 표현한다.

2. rdf:Resource 속성

rdf:Resource 속성을 표현하는 방법은 일반 문장을 다루는 방법과 다르지 않다. "CIT1111"이 "Discrete Mathematics"이고 resource 949318이 가르치며, 949318이 "David"이고 Associate professor라는 문장을 컬러 페트리 넷으로 표현하면 〈그림 4〉와 같다.

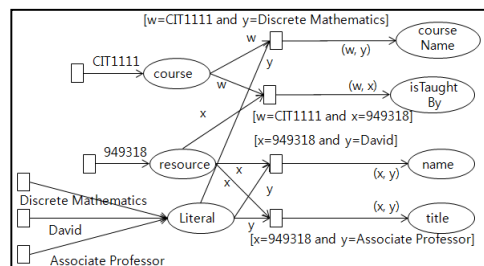


그림 4. rdf:Resource의 컬러 페트리 넷 표현 사례
Figure 4. An example CPN representation of rdf:Resource

3. 객체의 변환

〈그림 5〉는 객체가 구조체인 예를 보이는 RDF 모델이다 [11].

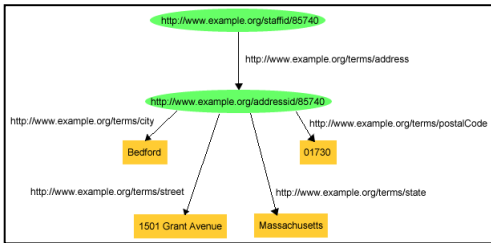


그림 5. RDF 문장의 객체가 구조체인 경우의 예
Figure 5. An example structured object in an RDF statement

이 그림은 고유번호가 86740인 직원(staff)의 주소가 시(city), 거리(street), 주(state), 우편번호(postal code)로 구성되었음을 보인다. 이러한 경우에, 구조체로 구성된 이 주소 값이 다른 곳에서도 쓰이면 이 그림처럼 구조체 자원을 만들어 둘 필요가 있으나, 이 구조체가 여기에서만 사용된다면 특정 자원을 생성하지 않고 주소 객체에 직접 반영할 수도 있다.

〈그림 6〉은 〈그림 5〉의 컬러 페트리 넷 모델로, (a)는 주소가 다른 곳에서 참조된 경우이고 (b)는 그렇지 않은 경우이다.

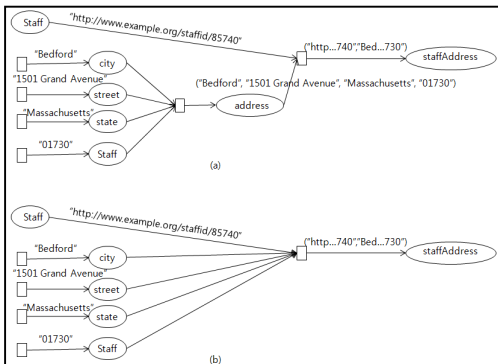


그림 6. 〈그림 5〉의 컬러 페트리 넷 모델. (a)주소가 다른 곳에서 참조될 경우 (b)그렇지 않은 경우
Figure 6. A CPN representation of the RDF statement shown in Figure 5, (a)adequate if 'address' is referred by some other statements, (b)adequate if 'address' is not referred by any other statements

4. Reification

reification은 술어를 플레이스로 나타내는 일반적인 방법을 적용하여 변환한다. 2장에서 소개한 reification문을 변환하면 〈그림 7〉과 같은 컬러 페트리 넷으로 변환할 수 있다. 그림에서 트랜지션의 가드에 쓰인 "Sta..352"는 "Statement About949352"의 약자이고, "352"는 "949352"의 약자이다.

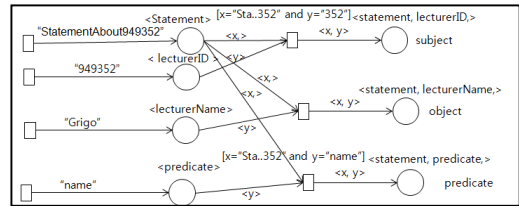


그림 7. reification의 컬러 페트리 넷 모델
Figure 7. A CPN model for 'reification'

V. 실험 결과

컬러 페트리 넷 모델의 실용성을 보이기 위하여 컬러 페트리 넷에서 RDF 질의에 대한 답을 구하는 방법을 알아본다.

우선, 질의 $\hat{}$ associateProfessor와 같이 원래 associate Professor로 정의된 원소들을 구하는 질의에 대한 답은 해당 플레이스에 입력 중, 소스 트랜지션들만 격발하여 답을 구한다. 예를 들어 〈그림 3〉에서, 이 질의에 대한 답은 "David"가 된다.

그러나, $\hat{}$ staffMember에 대한 답은 staffMember 플레이스의 입력 중 소스 트랜지션이 없으므로 공집합(empty)이다.

상속된(inherit) 인스턴스까지 모두 구하는 질의, 예를 들면 staffMember와 같은 질의에 대한 답은 staffMember에 토큰을 생성하는 모든 격발순서를 실행하여 구한다.

〈그림 3〉의 경우에는 "David"를 생성하는 소스 트랜지션을 격발한 후, 이 토큰을 Academic_staff 플레이스로 전이하고, 또 다시 staffMember 플레이스로 전이함으로써 답을 구한다.

다음과 같은 질의에 대한 답도 용이하게 구할 수 있다.

```
Select X, Y
From {X} isTaughtBy {Y}
```

이러한 질의에 대하여 컬러 페트리 넷은 소스 트랜지션을 격발하여 <"David">를 생성하고 <"DisMath">도 생성한다.

그리고, 가드가 $\{x="DisMath" \text{ and } y="David"\}$ 인 트랜잭션을 격발하여 $\langle\langle "DisMath", "David" \rangle\rangle$ 를 생성하여 "DisMath"와 "David"를 출력한다. 마찬가지로

Select Y

From "DisMath" involves $\{Y\}$:에 대한 답도 "David"라고 쉽게 구할 수 있다.

컬러 페트리 넷을 작성하고 시뮬레이션을 실행하는 환경을 제공하는 도구로 CPNTools가 널리 알려져 있다. <그림 1>의 RDF 모델을 표현하는 컬러 페트리 넷 모델을 CPNTools로 구축한 예가 <그림 8>이다.

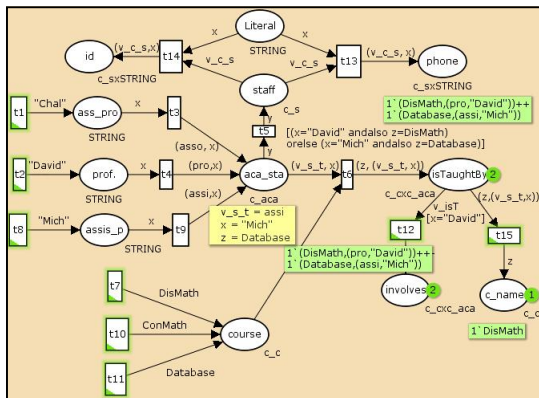


그림 8. CPNTools로 구축한 질의 결과를 보이는 컬러 페트리 넷 모델

Figure 8. A CPN model, built in CPNTools, showing the answers for some queries

구축한 모델 <그림 8>에서

Select X, Y

From $\{X\}$ isTaughtBy $\{Y\}$:에 대한 답이 플레이스 isTaughtBy의 토큰으로 보인다. 즉, "DisMath isTaughtBy professor David"이고 "Database isTaughtBy assistant professor Mich"임을 알 수 있다.

비슷한 방법으로 involves에 대한 답도 <그림 8>에 있을 수 있다.

Select N

from course $\{X\}$.isTaughtBy $\{Y\}$, $\{C\}$ name $\{N\}$
where $Y = "David"$ and $X = C$;

이에 대한 질의의 결과는 플레이스 c_name에 DisMath이다. <그림 8>의 트랜잭션 t15에는 가드 $\{x="David"\}$ 가 연합되어 있어, 결국 플레이스 c_name에 있는 토큰은 "David"가 가르치는 과목 이름 "DisMath"가 된다.

VI. 결론

본 논문은 RDF 모델을 컬러 페트리 넷 모델로 변환하는 알고리즘을 제안하였다. RDF 모델은 일종의 온톨로지인데, 지금까지 온톨로지를 컬러 페트리 넷으로 변환 하는 알고리즘을 소개한 연구는 전무하다. 제안하는 알고리즘은 RDFS 레이어를 컬러 페트리 넷으로 변환한 다음 RDF 문장들을 변환하기 때문에 알고리즘이 간단하며, 기존의 컬러 페트리 넷 구축 및 시뮬레이션 도구를 사용할 수 있다는 장점이 있다.

RDF 모델은 RDF 레이어와 RDFS 레이어가 분리되어 있어, RDF 모델을 다루는 소프트웨어와 RDF 질의를 다루는 소프트웨어가 별개로 개발된다. 이에 반하여 컬러 페트리 넷 모델에는 단어와 뜻, 즉 RDF 레이어와 RDFS 레이어가 포함되어 있어 컬러 페트리 넷 모델을 분석함으로써 질의에 대한 답을 쉽게 구할 수 있다. 본 논문에서는 제안하는 방법의 실용성을 보이기 위하여 CPNTools를 이용하여 RDF 모델을 나타내는 컬러 페트리 넷을 구축하고, 컬러 페트리 넷의 동적 성질을 시뮬레이션하여 질의에 대한 답을 찾는 과정을 보았다. 본 연구 결과는 RDF 모델을 컬러 페트리 넷으로 변환하는 방법을 제안함으로써, RDF 모델 분석에 다양한 컬러 페트리 넷 분석 방법을 적용할 수 있는 초석을 제공하였다.

향후 연구 방향은 두 가지이다. 하나는 웹 온톨로지 언어인 OWL과 같이 더 일반적인 방법으로 작성된 온톨로지를 컬러 페트리 넷으로 변환하는 방법을 연구하는 것이고, 다른 하나는 RDF 모델 분석에 수학적 컬러 페트리 넷 분석 방법을 이용하는 방안을 더 연구하는 것이다.

<http://www.w3.org/TR/RECrdfsyntax/>

참고문헌

- [1] T. Murata, "Petri nets: Properties, analysis and applications," Proceedings of the IEEE, Vol. 77. no. 4, April 1989, pp.541-580.
- [2] <http://wiki.daimi.au.dk/컬러 페트리 넷tools/컬러 페트리 넷tools.wiki>
- [3] Grioris Antoniou, Frank van Harmelen, *A Semantic Web Primer* The MIT Press, 2004.
- [4] C.A. Petri, "Kommunikation mit Automaten", Bonn: Institut fur Instrumentelle Mathematik, Schriften des IIM Nr. 3, 1962. Also English translation, "Communication with Automata", New York:Griffiss Air Force Base, Tech. Rep. RADC TR 65 377, vol. 1, Suppl. 1, 1966.
- [5] K. Jensen, "An Introduction to the Practical Use of Coloured Petri Nets" Lecture Notes in Computer Science vol. 1492, Springer Verlag 1998, pp.237-292.
- [6] Dong Her Shih, Hsiu Sen Chiang, Binshan Lin, "A Generalized Associative Petri net for Reasoning," IEEE Transactions on Knowledge and Data Engineering, Sep. 2007 (Vol. 19, No. 9) pp. 1241-1251.
- [7] Dragan Gasevic, "Petri Nets on the Semantic Web Guidelines and Infrastructure" ComSIS Vol. 1, No.2, Nov. 2004, pp.127-151.
- [8] G. Zhang, F. Meng, C. Jiang, J. Pang, "Using Petri Net to Reason with Rule and OWL" Proceedings of the sixth IEEE International Conference on Computer and Information Technology (CIT'06), Seoul Korea, Sept. 20 22 2006, CIT 2006:42.
- [9] 김병곤, 오성균, "시맨틱 웹 구축을 위한 스카마 관리 기법 연구", 한국컴퓨터정보학회논문지, 제12권, 제1호, 9-15쪽, 2007년 3월
- [10] 김연희, 신혜연, 임해철, 정균락, "시맨틱 웹 데이터의 키워드 질의 처리를 위한 인덱싱 및 저장 기법" 한국컴퓨터정보학회 논문지, 제12권, 제5호, 93-101쪽, 2007년 11월
- [11] Frank Manola, Eric Miller, "RDF Primer", W3C Recommendation 10 Def. 2004,

저자 소개



임재걸

1981년 동국대학교 전자계산학과 학사
1987년 University of Illinois 석사
1990년 University of Illinois 박사
1992년 - 현재 동국대학교(경주) 컴퓨터멀티미디어학과 교수
<관심분야> 멀티미디어시스템, 컴퓨터 시스템 분석



권기용

2004년 - 현재 동국대학교(경주) 컴퓨터멀티미디어학과 재학
<관심분야> 멀티미디어시스템, 위치 기반서비스, 소프트웨어 설계 및 개발



주재훈

1982년 한국해양대학교 공학사
1987년 부산대학교 경영학석사
1992년 부산대학교 경영학박사
1993년 - 현재 동국대학교(경주) 전자상거래학과 교수
<관심분야> 시맨틱 웹, 전자상거래, 지식경영



이강재

1981년 동국대학교 전자계산학과 학사
1983년 동국대학교 전자계산학과 석사
1997년 동국대학교 컴퓨터공학과 박사
1986년 - 현재 수원과학대학 컴퓨터 정보과 교수
<관심분야> 데이터베이스 응용, 데이터마이닝