

RDF 모델을 나타내는 Petri Net 모형 구축

임재걸*, 이강재**, 정승환***, 심문주*

Construction of a Petri Net Model for a RDF Model

Jaegol Yim*, Kangjai Lee**, Seunghwan Jeong***, Moonjoo Shim*

요약

본 논문은 온톨로지(ontology) 작성에 사용되는 RDF(Resource Description Framework) 모델을 CPN(Colored Petri Net) 모델로 변환하여 표현하는 방법을 제안하고, 페트리 넷 시뮬레이션으로 RDF의 의미를 해석하여 RDF 질의문에 답할 수 있음을 보인다. 또한, 어떠한 RDF 모델도 CPN 모델로의 변환이 가능하다는 제안 방법의 완전성과 무결성을 소개하고, 제안한 방법으로 구축된 CPN 모델에서 RDF 질의에 대해 정확하게 결과를 구하는 방법을 제시한다.

▶ Keyword : 온톨로지(ontology), RDF(Resource Description Framework), CPN(Colored Petri Net), 질의(query)

• 제1저자 : 임재걸

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

I. 서론

본 논문은 RDF(Resource Description Framework)를 페트리 넷으로 변환하여 표현하는 방법을 제안한다. 또한, 페트리 넷 시뮬레이션으로 RDF의 의미를 해석하여 RDF 질의문에 답할 수 있음을 보인다. 즉, RDF 질의를 페트리 넷 시뮬레이션으로 치환하는 방법을 본 논문에서 제안한다.

따라서 본 논문의 결과를 바탕으로 이미 발견된 다양한 수학적 페트리 넷 분석 방법을 통하여 RDF를 분석하는 것이 가능하게 된다.

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 연구되었기 때문이다.

RDF(Resource Description Framework)는 웹 상의 자원을 표현하는, 월드와이드웹(World Wide Web) 컨소시엄이 정의한 규격이다. 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 있는지 표현할 수도 있다.

따라서, RDF로 표현된 정보에는 도메인 지식이 포함됨으로 기계적인 해석이 가능하다. 즉, 잘 작성된 RDF 모델은 일종의 온톨로지(ontology)이다.

II. 관련 연구

페트리 넷의 정의는 [표 1]과 같이 정의되는 이분그래프이다[1]. 여타의 그래프가 그렇듯이 페트리 넷도 그림으로 표현하면 더 직관적이다. 그림에서 P의 원소 장소(place)는 원이나 타원으로, T의 원소 변천(transition)은 사각형으로 표현한다.

표 1. 페트리 넷의 정의

<p>페트리 넷 N은 5가지 요소로 구성된다.</p> <p>$N = (P, T, F, W, M_0)$ where:</p> <p>$P = \{p_1, p_2, \dots, p_n\}$은 장소 (place)라는 유한 집합,</p> <p>$T = \{t_1, t_2, \dots, t_m\}$은 변천 (transition)이라는 유한 집합,</p> <p>$F : \text{유향간선}$이라는 $(P \times T)$와 $(T \times P)$의 합집합의 부분집합,</p> <p>$W : F \rightarrow \{1, 2, 3, \dots\}$은 유향간선의 가중치를 결정하는 함수,</p> <p>$M_0 : P \rightarrow \{0, 1, 2, 3, \dots\}$은 초기에 각 장소에 놓인 토큰의 수를 표현하며, 초기 마킹(marking)이라고 함.</p> <p>단, P와 T의 교집합은 공집합이고, P와 T의 합집합은 공집합이 아님.</p>
--

F의 정의에서 보듯이 간선은 장소와 변천, 혹은 변천과 장소를 연결하고, 장소끼리 혹은 변천끼리는 연결하지 않기 때문에 페트리 넷은 이분그래프이다. 페트리 넷의 역동적 성질에 대해서 모의실험을 가능케 하여 주는 변천 격발(fire)의 정의는 다음과 같다.

- 가. 하나의 변천 t 에 대하여, 입력 장소 p 가 최소한 $W(p, t)$ 만큼의 토큰을 갖고 있으면, t 는 장전되었다고 한다.
- 나. 장전된 t 는 격발될 수도 있고, 안 될 수도 있다.
- 다. t 의 격발은 $W(p, t)$ 만큼의 토큰을 각 입력 장소 p 에서 제거하고, $W(t, p)$ 만큼의 토큰을 각 출력 장소에 더하여 준다.

RDF 모델을 나타내는 페트리 넷으로 본 논문에서는 칼라 페트리 넷(CPN: Colored Petri Net)을 사용한다. CPN의 토큰은 색깔이 있기 때문에 그런 이름으로 불린다. CPN을 용이하게 작성할 수 있는 환경을 제공하고 시뮬레이션을 수행하여 주는 소프트웨어 도구 중에 CPNTools[2]라는 소프트웨어가 있다. 본 논문에서는 이 도구를 이용하여 주어진 RDF 모델을 나타내는 CPN을 작성한다고 가정한다. CPN에 대한 설명은 3장에서 CPN의 구축 예를 다룰 때 자세히 설명한다.

근래에 다양한 분야에서 페트리 넷이 이용되고 있다. 본 논문과 관련이 있는 연구로서 [3]은 규칙기반 시스템을 나타내고 분석하기에 용이한 '일반화 페트리 넷'이라는 일종의 페트리 넷 변종을 제안하였다. 또한, Petri net을 표현하는 정형화된 방법 중에 metamodel, UML 등이 있는데, [4]는 UML,

RDFS, OWL을 이용한 페트리 넷 온톨로지를 제공한다.

III. 변환 방법

본 장에서는 RDFS가 제공하는 RDF 모델 작성에 사용될 기초 요소들을 차례로 살펴보고, 이들 각각을 CPN 모델로 나타내는 방법을 제시한다.

우선 core class에는 rdfs:Resource, rdfs:Class, rdfs:Literal, rdfs:Property와 rdf:Statement가 있다. CPN에서 이들 각각은 하나의 place로 표현된다. 또한, 각 place에 놓일 수 있는 토큰의 유형을 CPN declaration에서 정의한다.

일반적으로 RDF의 property는 CPN의 place로 표현한다. 그러나 core properties는 토큰의 전이를 가능하게 함으로써 나타낸다. 즉, rdf:type은 자원을 나타내는 토큰을 class로 전이하도록 [그림 1]처럼 나타낸다. [그림 1]에서는 rdf:about으로 자원을 지정한 경우와 rdf:ID로 자원을 지정한 경우 각각에 대한 CPN 모델을 보여주고 있다. rdf:about으로 지정한 경우가 먼저 다루어지는데 그림의 rdf:Description에서 보듯이 지정된 자원은 "'CIT1111'"로 가정하였다. 실제에서는 "'http://www.mydomain.org/uni-ns/#CIT1111'"처럼 URL이 사용된다.

[그림 1]에서 보이는 courseID = "'CIT1111'"은 CPN의 declaration에 포함될 문장이며, courseID라는 데이터 유형을 정의하는 의미가 있다. 또한, 그림에 보이는 CS_course와 course는 place를 지정하는 이름이고, <courseID>는 해당 place에 놓일 토큰의 유형을 지정한다.

따라서, 이 CPN에서는 <courseID>가 취할 수 있는 사례는 "'CIT1111'"뿐이다. 페트리 넷의 Transition은 그림 표기에서 사각형으로 표현된다. Transition에는 격발 조건을 더욱 제한하는 guard가 연합될 수 있는데, [그림 1]에서는 [x = "'CIT1111'"]이 guard이다. 즉, 변수 x가 "'CIT1111'"로 치환될 경우에만 이 transition이 격발하여 토큰 <'CIT1111'>을 course place로 전이시킬 수 있다. rdf:about은 이미 존재하는 자원을 지칭하기 때문에 본 CPN에서는 이 자원이 CS_course에 존재한다고 가정한다.

반면에 rdf:ID로 자원이 지정된 경우에는 자원을 새로 생성하는 것이므로 [그림 1]의 두 번째 모양처럼 CPN을 구축한다. 입력이 없는 transition을 source transition이라고 하며, 이러한 transition은 토큰을 생성하여 공급하는 역할을 한다. 즉, 여기서는 토큰 <'CIT1111'>을 생성하여 공급한다.

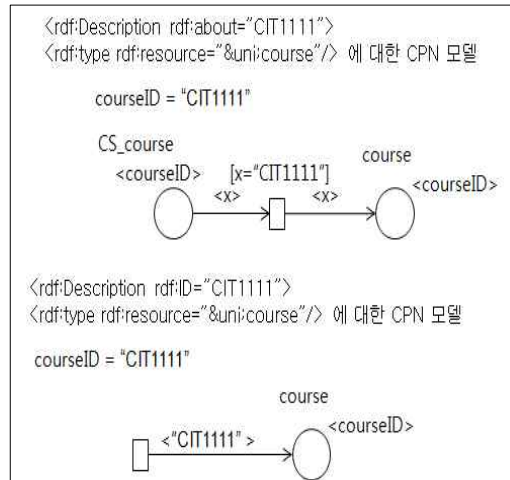


그림 1. rdf:type의 CPN 모델

RDF의 property는 object의 property가 value라는 문장을 의미하며 이때 object는 자신이고 value는 property의 attribute에 쓰인다.

예를 들어, [그림 2]에 보이는 <rdfs:Class ……>는 lecturer라는 class가 staffMember class의 subclass임을 의미하며, 이 경우의 object는lecturer이고 property는 subClassOf이며 value는 "'staffMember'"이다. core properties 중 rdfs:subClassOf는 object에 해당하는 place의 모든 토큰을 value에 해당하는 place로 전이하도록 [그림 2]의 CPN처럼 변환한다.

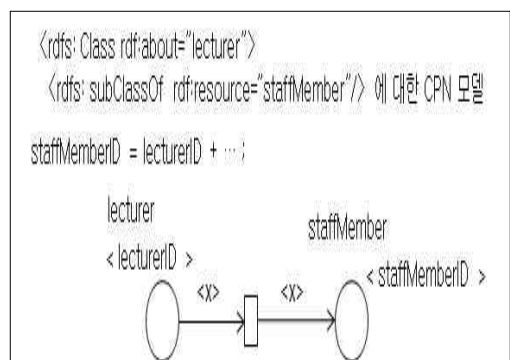


그림 2. rdfs:subClassOf의 CPN 모형

rdfs:subPropertyOf를 나타내는 CPN을 구축하는 방법도 rdfs:subClassOf의 경우처럼 object에 해당하는 place의 모든 토큰을 value에 해당하는 place로 전이하도록 구축한다.

properties의 조건을 제한하는 core property에 rdfs:domain, rdfs:range 등이 있다. property는 CPN에서 place로 나타내고, domain과 range는 property를 나타내는 place의 토큰 유형에 나타낸다. 즉, domain을 토큰 유형의 1번 원소, range를 2번 원소로 지정한다. 이러한 변환 예를 종합적으로 보이기 위하여 [그림 3]의 RDF 모델을 [그림 4]와 같이 CPN으로 변환하면, place phone이 이러한 변환 예를 보인다.

RDF에서 reification은 문장 A에 대한 문장을 만들고 싶을 때 문장 A를 지칭하는 이름을 만드는 것이다.

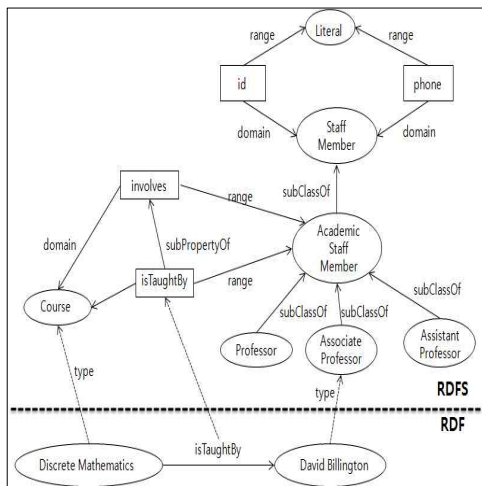


그림 3. RDF 모델의 예

예를 들어, 다음과 같은 description은

```
<rdf:Description rdf:about = ""949352"">
<uni:name>Grigoris Antoniou</uni:name>
</rdf: Description>
```

다음과 같이 reify된다.

```
<rdf:Statement
rdf:about = ""StatementAbout949352"">
<rdf:subject rdf:resource = ""949352""/>
<rdf:predicate rdf:resource =
""&uni:name""/>
<rdf:object>Grigoris Antoniou</rdf:object>
</rdf:Statement>
</rdf:Statement>
```

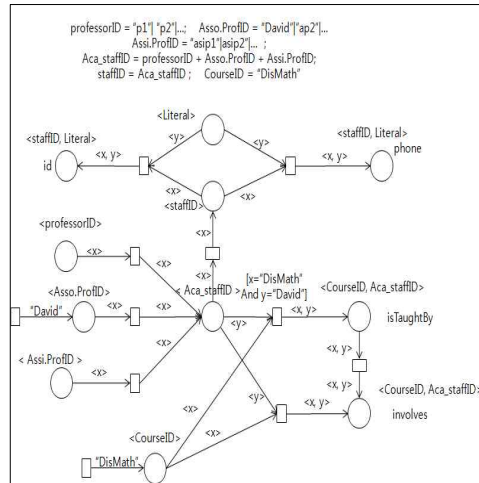


그림 4. [그림 3]을 나타내는 CPN 모델

이러한 reification은 predicate를 place로 나타내는 일반적인 방법을 적용하여 [그림 5]와 같은 CPN으로 변환한다. Transition의 guard에 쓰인 ""Sta...352""는 ""StatementAbout949352""의 약자이고 ""352""는 ""949352""의 약자이다.

[그림 5]의 CPN 모델로부터 원래의 문장을 복원하는 CPN 모델은 [그림 6]과 같다.

어떤 교수가 담당하는 과목들을 모두 나열하는 경우와 같이 하나의 문장에 여러 자원들을 집합적으로 사용하고 싶을 때 rdf에서 container class를 사용한다. container class는 각 요소마다 하나의 토큰을 생성하도록 CPN을 구축한다.

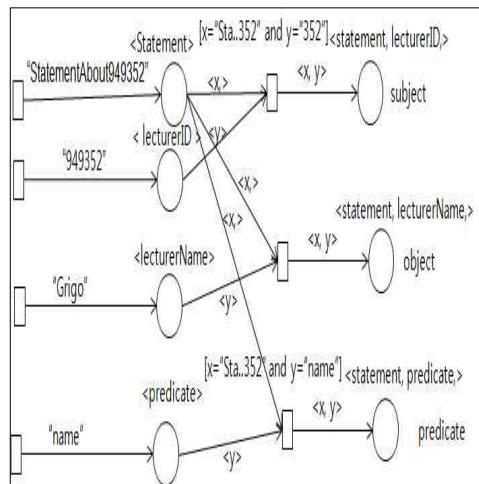


그림 5. reification의 CPN 모델

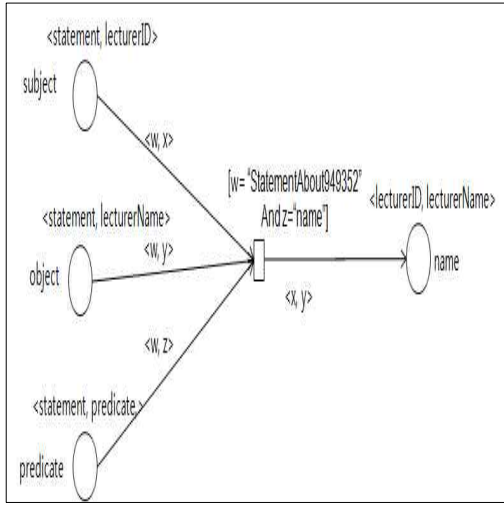


그림 6. 원래 statement를 복원하는 CPN

IV. 완전성과 무결성

RDFS를 구성하는 모든 class와 properties에 대하여 CPN 모델을 구축하는 방법을 보였으므로 3장에 소개한 방법으로 하면 어떠한 RDF 모델도 CPN 모델로 변환이 가능하다.

이렇게 구축된 CPN이 RDF와 RDFS의 axiomatic semantics를 만족하는지 살펴보자. 우선 다음 axiom을 만족해야 한다.

$$Type(?r, ?t) \leftrightarrow PropVal(tpye, ?r, ?t)$$

좌변이 참이면 [그림 1]처럼 ?r을 ?t place로 전이하는 CPN으로 표현된다. 따라서, 제안하는 CPN 구축 방법은 좌변이면 우변을 만족한다. 또한, CPN에서는 place가 property를 나타냄으로 CPN에서 좌변이 참이면 의미상 우변도 참이다.

다음은 property, resource, class 등의 포함관계를 나타내는 axiom이다. [그림 2]의 subClassOf를 나타내는 CPN 표현 방법을 적용하였으므로 이 axiom들은 CPN에서 참이다.

$$Type(?p, Property) \rightarrow Type(?p, Resource)$$

$$Type(?c, Class) \leftarrow Type(?c, Resource)$$

reified statement에 대하여 다음이 성립하는데, [그림 5]에 보이는 바와 같이, 제안하는 방법으로 구축된 CPN도 이를 만족한다.

$$Type(?st, Statement) \rightarrow$$

$$\exists ?p \exists ?r \exists ?v (PropVal(Predicate, ?st, ?p) \wedge$$

$$PropVal(Subject, ?st, ?p) \wedge PropVal(Object, ?st, ?v))$$

다음은 domain과 관련된 axiom인데, 제안하는 방법으로 구축된 CPN은 이 axiom을 만족한다. 예를 들어 [그림 3]에서 phone의 domain은 staffMember이고 phone의 토큰 유형은 <staffID, Literal>임으로 이 axiom을 만족함을 알 수 있다. 비슷한 이유로 range 관련 axiom도 만족한다.

$$PropVal(domain, ?p, ?d) \rightarrow$$

$$\forall ?x \forall ?y (PropVal(?p, ?x, ?y) \rightarrow Type(?x, ?d))$$

CPN에서는 질의에 대한 답도 용이하게 구할 수 있다. ^associateProfessor와 같이 associateProfessor로 정의된 원소들을 구하는 질의에 대한 답은 해당 place에 입력 중 source transition들만 격발하여 답을 구한다. 예를 들어 [그림 4]에서, 이 질의에 대한 답은 "David"가 된다. 그러나 ^staffMember에 대한 답은 staffMember place의 입력 중 source transition이 없으므로 empty이다.

inherit된 instance까지 모두 구하는 질의, 예를 들면 staffMember와 같은 질의에 대한 답은 staffMember에 토큰을 생성하는 모든 격발 순서를 실행하여 구한다. [그림 4]의 경우에는 ""David""를 생성하는 source transition을 격발한 후, 이 토큰을 Academic_staff place로 전이하고, 또 다시 staffMember place로 전이함으로써 답을 구한다.

다음과 같은 SQL 질의에 대한 답도 용이하게 구할 수 있다.

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

이러한 질의에 대하여 CPN은 위와 같은 방법으로 <""David"">를 생성하고 <""DisMath"">도 생성한다. 다음으로 [그림 4]에서 guard가 [x = ""DisMath"" and y = ""David""]인 transition을 격발하여 <""DisMath"", ""David"">를 생성하여 DisMath와 David를 출력한다. 마찬가지로 방법으로 하면

```
Select Y
From ""DisMath"" involves {Y}
```

에 대한 SQL 질의의 답도 쉽게 구할 수 있다.

V. 결 론

본 논문은 RDF 모델을 CPN 모델로 변환하는 방법을 제안하였다. 제안하는 방법이 모든 RDF description을 CPN으로 변환하며, 이렇게 구한 CPN이 RDF axiom들을 만족함을 보였다. 나아가서 RDF 질의에도 정확한 답을 구할 수 있음 보였다.

본 연구 결과는 RDF 모델을 CPN으로 변환하는 방법을 제안함으로써, RDF 모델 분석에 다양한 CPN 분석 방법을 적용할 수 있는 초석을 마련하였다.

향후 연구 방향은 두 가지이다. 하나는 OWL과 같이 더 일반적인 방법으로 작성된 온톨로지를 CPN으로 변환하는 방법을 연구하는 것이다. 또 다른 하나는 RDF 모델 분석에 CPN 분석 방법을 이용하는 방법을 연구하는 것이다.

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/cpntools/cpntools.wiki>
- [3] 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.
- [4] Dragan Gasevic, "Petri Nets on the Semantic Web Guidelines and Infrastructure", ComSIS Vol. 1, No.2, Nov. 2004, pp.127-151.
- [5] Gioris Antoniou, Frank van Harmelen, A Semantic Web Primer, The MIT Press, 2004.