

시구간 기반 시간 정보의 표현과 추론을 위한 시간 온톨로지 언어

(A Temporal Ontology Language for Representing and Reasoning about Interval-based Temporal Information)

김 상 균[†] 이 규 철^{**} 송 미 영^{***}
(Sang-Kyun Kim) (Kyu-Chul Lee) (Mi-Young Song)

요 약 W3C 웹 온톨로지 워킹그룹에서는 시맨틱웹을 위한 온톨로지 언어를 위해서 OWL을 개발하였다. 하지만 OWL은 시간에 대한 의미체계를 표현할 수 없기 때문에 시간 정보에 기반한 추론을 수행할 수 없다. 실세계의 개체들은 시간의 흐름에 따라 변하고 어떤 사건의 발생으로 인해 새로운 사실들이 생겨나고 있기 때문에 만약 지식베이스에 존재하는 지식이 시간 정보를 가지지 않는다면 불완전하고 부정확해질 수 밖에 없다. 따라서 본 논문에서는 시맨틱웹에서 시간 정보를 표현하고 추론할 수 있도록 OWL을 시간에 대한 의미체계를 가지도록 확장한 온톨로지 언어인 TL-OWL을 제안한다.

키워드 : TL-OWL, 시맨틱웹, 시간 온톨로지 언어, 시간 추론

Abstract The W3C Ontology Working Group has recently developed OWL as an ontology language for the Semantic Web. OWL, however, fails to perform the process of reasoning about temporal knowledge because it lacks full-pledged semantics for temporal language. Entities in the real world are changing as time passes, while new facts are being introduced as new events occur. KBs without temporal information are incomplete and incorrect. In this paper, we propose an extended temporal ontology language called TL-OWL which provides an abstract syntax and semantics for representing and reasoning about temporal information in the Semantic Web.

Key words : TL-OWL, Semantic Web, Temporal Ontology Language, Temporal Reasoning

1. 서 론

시맨틱웹(Semantic Web)은 웹에 의미체계를 기술함으로써 기계가 웹 상의 정보를 자동으로 이해하고 처리할 수 있도록 하는데 비전을 두고 있다. 기계가 정보를 자동으로 처리하기 위해서는 기계와 사람이 공유할 수

있는 지식을 명세해야 하는데 시맨틱웹에서는 온톨로지를 이용하여 웹 자원에 대한 지식을 기술한다. 최근 W3C의 웹 온톨로지 워킹 그룹에서는 시맨틱웹을 위한 온톨로지 언어로써 OWL(Web Ontology Language)[1]을 W3C 표준으로 제정하였다.

하지만 이러한 OWL은 시간 정보를 가지고 있지 않기 때문에 시간에 의존적인 질의를 정확히 처리할 수 없다. 실세계의 개체들은 시간의 흐름에 따라 계속 변화하며 또한 어떤 사건들로 인해 새로운 결과 또는 사실들이 생겨나기 때문에 OWL로 만들어진 지식베이스에 존재하는 지식들이 시간 정보를 가지지 않는다면 불완전하고 부정확해질 수 밖에 없다.

예를 들어 어떤 회사에 대해서 “XXX 회장 구속”, “경영 공백 위기”, “내수 판매 부진”, “주가 하락” 이렇게 4

- † 정 회 원 : 한국한의학연구원 정보연구센터 선임연구원
skkim@kiom.re.kr
 - ** 중 심 회 원 : 충남대학교 컴퓨터공학과 교수
kcllee@cnu.ac.kr
 - *** 정 회 원 : 한국한의학연구원 정보연구센터 책임연구원
smyoung@kiom.re.kr
(Corresponding author ip)
- 논문접수 : 2009년 3월 20일
심사완료 : 2009년 5월 22일

Copyright©2009 한국정보과학회: 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 소프트웨어 및 응용 제36권 제7호(2009.7)

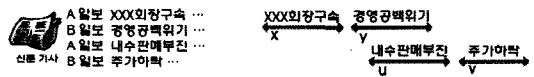


그림 1 신문 기사 사건 예제

개의 사건이 발생했고 이에 대한 신문 기사가 존재한다고 하자. 각각의 사건은 사건이 발생한 시구간 x, y, u, v 를 가지며 시구간들간의 시간 관계는 위 그림처럼 된다고 가정한다.

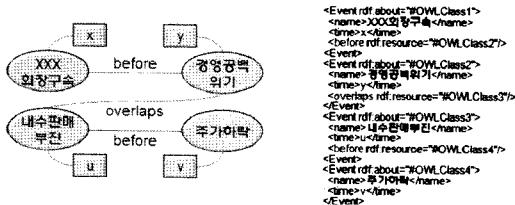


그림 2 신문 기사의 OWL 표현

위 그림은 그림 1의 신문 기사를 OWL을 이용해 온톨로지로 표현한 예제이다. OWL 온톨로지에서 4개의 사건은 4개의 개체로 기술되고 각 개체는 시구간 변수 x, y, u, v 와 연결되어 있다. 또한 각 사건은 신문 기사의 순서 관계에 따라 x before y, y overlaps u, u before v 가 된다고 기술하였다.

이러한 온톨로지가 있는 경우 사용자가 "XXX 회장 구속 이후 주가가 어떻게 되었나?"라는 질의를 한다고 하자. 그런데 OWL 온톨로지에는 XXX 회장 구속 이후 경영 공백 위기(x before y)가 생기고 경영공백 위기로 내수 판매가 부진(y overlaps u)하며 내수 판매 부진으로 주가가 하락(u before v)되었다라는 관계만 있을 뿐 XXX 회장 구속으로 주가가 하락(x before v)라는 관계가 명시적으로 존재하지 않기 때문에 질의의 답을 알 수 없다. 대신 OWL 기반의 온톨로지 추론뿐만 아니라 $before(x,y) :- before(x,y) \ \& \ overlaps(y,u) \ \& \ before(u,v)$ 와 같이 룰(Rule)을 이용한 규칙기반 추론을 이용한다면 질의의 답을 얻을 수 있다. 하지만 일반적으로 규칙기반 추론은 반결정가능한(semi-decidable)¹⁾ 문제가 있다.

반면에 위 그림처럼 시구간 변수를 선언하고 시구간

변수들간의 시간 관계를 지정함으로써 온톨로지에 시간에 대한 의미체계를 부여할 수 있다면 규칙기반 추론을 이용하지 않아도 x before $y \ \wedge \ y$ overlaps $u \ \wedge \ u$ before $v \ \wedge \ x$ before v 와 같이 온톨로지 기반의 시간 추론을 수행함으로써 XXX회장 구속 이후 주가 하락이라는 결과를 얻을 수 있게 된다.

OWL과 같은 온톨로지에서는 시간에 대한 정보가 없기 때문에 시간 추론을 할 수 없으며 룰추론이나 다른 방법을 이용해야 한다. 하지만 이 방법들은 결정가능하지 않는 문제가 있다. 뿐만 아니라 현재의 OWL은 시간 정보와 비시간 정보를 구별하지 않고 모두 일반 데이터로 취급하기 때문에 시간 정보에 기반하여 복잡한 시간 계산을 하기 어려운 문제가 있다.

옛날부터 인공 지능 분야에서는 일차 논리에 기반하여 시간 정보를 다루는 시간 논리(Temporal Logics)에 대한 연구가 수행되어 왔지만 최근에는 기술 논리(Description Logics)에 기반한 시간 기술 논리(Temporal Description Logics)을 통해 시간 관련 개념을 표현하고 추론하기 위한 연구[2]들이 많이 제안되고 있다.

시간 기술 논리에 대한 연구들은 시간 정보를 어떻게 정형화하느냐에 따라 크게 시점 기반(Point-based) 기술 논리[3] 또는 시구간 기반(Interval-based) 기술 논리로 나뉘어진다. 시구간 기반 연구는 Allen의 시구간 개념[4]을 기반으로 개념을 표현하며, 시점 기반 기술 논리는 시간에 대한 명제들을 표현하고 추론하는 시제 논리(Tense Logics)에 기반을 두고 있는데 일반적으로 시제 논리는 모달 논리 형태로 시간 개념을 표현한다. 하지만 시맨틱웹에서 시간 기반 추론을 위해서는 OWL DL의 표현력을 가져야 하는데 기존의 시간 기술 논리들은 모두 OWL DL의 표현력을 가지지 못하는 문제가 있다.

따라서 본 연구에서는 위와 같은 OWL과 시간 기술 논리에서의 문제를 해결하고 시맨틱웹에서 결정가능한 시간 추론이 가능하도록 하기 위해서 시간, 특히 시구간

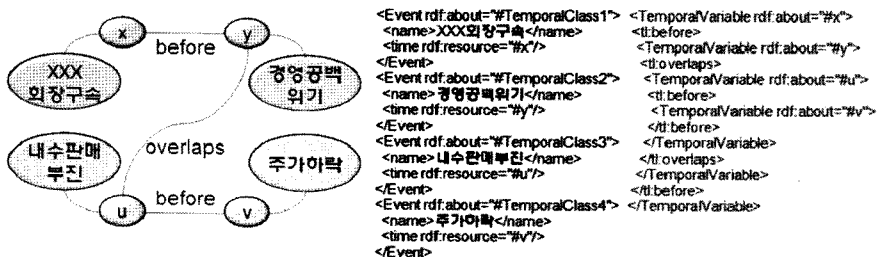


그림 3 신문 기사의 시구간 온톨로지 표현

1) 반결정가능한 알고리즘은 M에 존재하는 모든 x에 대해서 결정가능하며 M에 존재하지 않는 x에 대해서는 결정가능하지 않는 알고리즘을 말한다.

에 대한 의미체계를 가지도록 OWL을 확장한 언어인 TL-OWL을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 시간 온톨로지와 관련된 연구에 대해서 소개하고 3장에서는 본 논문에서 사용하는 시구간 의미체계의 근간이 되는 시구간 논리를 소개한다. 4장에서는 본 논문에서 제안하는 시구간 온톨로지 언어인 TL-OWL을 기술하며, 5장에서 결론을 맺는다.

2. 선행 연구

시간 마크업 언어 또는 시간 온톨로지와 관련해서 기존에 수행된 연구는 다음과 같은 연구들이 존재한다. 하지만 모두 본 연구에서와 같은 시간 추론 기능을 지원하지는 못한다.

시간 마크업 언어는 시간을 XML 기반으로 표현함으로써 시간 정보를 공유할 수 있도록 한다. 시간 마크업 언어의 대표적인 것으로는 TimeML(Time Markup Language)[5] 버전 1.2.1이 있다. TimeML은 웹 상의 자연어 텍스트에서 시간과 사건, 그리고 이들간의 관계들을 XML로 표현함으로써 기존의 질의 응답 시스템(question answering system)의 자연어 처리 성능을 향상시키기 위해 만들어졌다. 특히 TimeML에는 시간 함수 및 시간정량자등을 지정할 수 있는 TIMEX3 태그가 있어서 다양한 시간 표현을 만들 수 있다. 또한 최근 ISO에서는 TimeML을 확장하여 ISO-TimeML[6]을 만들었다. ISO-TimeML은 기존의 TimML에 다국어화를 위한 시맨틱을 추가하여 서로 다른 언어의 텍스트에서 시간 정보들간의 변환이 가능하도록 하였으며, 일차 로직을 기반으로 시구간 시맨틱을 표현하였다. 하지만 기술 논리와 같이 정형화된 시맨틱 모델이 없어 시간 추론에 있어 완전한 추론 결과를 제공하는데 한계가 있다.

최근 미국 Southern California 대학에서는 OWL을 이용해 웹사이트의 시간 정보뿐만 아니라 웹서비스의 시간적 특성을 기술하는 OWL-Time 온톨로지[7]를 구축하였다. 이 온톨로지는 이전의 DARPA Agent Markup Language 프로젝트의 일부인 DAML-Time 프로젝트로부터 시작되었으며 현재는 2006년 9월 W3C의 워킹 드래프트 상태에 있다. OWL-Time 온톨로지는 시점(instant), 시구간(interval)들간의 계층적인 관계와 시간기간(duration), 날짜시간(datetime) 정보를 표현하는 어휘들을 제공한다. 또한 이 시간 온톨로지가 OWL-S에 나오는 예제를 통해 어떻게 OWL-S를 지원하는지를 보이고 있다. OWL-Time은 OWL을 이용하여 시간 정보를 표현하는 온톨로지으로써 본 연구에서 제안하는 TL-OWL과 같이 시간 정보를 표현하고 추론할 수 있다. 하지만 OWL-Time은 자체가 시간 온톨로지이며

온톨로지 언어가 아니다. 따라서 OWL-Time에서 정의된 어휘들만 사용해서 시간을 정의할 수 밖에 없기 때문에 다양한 시간 표현에 제약이 있을 뿐만 아니라, 시간관계에 대한 시맨틱이 없기 때문에 1장에서 소개한 예제와 같은 것을 추론하기 위해서는 역시 온톨로지 추론이 아닌 규칙기반 추론을 사용해야 한다.

Milea et al.[3]은 OWL에 기반하여 시간을 표현하는 TOWL이라는 온톨로지 언어를 제안하였다. TOWL은 특히 Welty et al.[8]가 제안한 언어에 여러 기능들을 추가하여 표현력을 확장하였다. 일반적으로 기술 논리 분야에서는 시간 의존적인 개념을 표현하고 추론하는 방법으로써 시간 차원을 어떻게 처리하느냐에 따라 내적 방법(internal method)와 외적 방법(external method) [2]으로 나뉘어진다. 본 연구에서 제안하는 TL-OWL은 TL-ALCF의 경우와 같이 외적 방법을 사용하며 반면에 TOWL은 내적 방법을 사용한다. 외적 방법에서는 하나의 개체가 서로 다른 상태를 가지는 시점들에서 서로 다른 정보를 가질 수 있지만, 내적 방법에서는 하나의 개체가 서로 다른 상태를 가지는 시점들에서 서로 다른 개체로 인식되며, 따라서 하나의 개체는 서로 다른 정보를 가지는 집합으로 처리된다. TOWL에서는 이러한 내적 방법을 사용하기 위해 구체적 영역(concrete domain)의 개념을 소개하고 시간을 표현하는 방법을 제안하였다. 이와 같이 본 연구의 TL-OWL과 TOWL은 시간 차원의 표현 방법은 다르지만 모두 시간에 대한 구문과 시맨틱을 정의함으로써 시간 추론이 가능하게 하는 점에서는 유사하다. 하지만 TOWL에서는 아직 온톨로지 추론에서 중요한 포함문제나 합의성에 대한 정의를 제시하지 않고 있기 때문에 언의의 완전성을 확인할 수 없다.

3. 기본개념과 원칙

TL-ALCF는 Artale과 Franconi가 제안한 시구간 기반 기술 논리다. [9]에서 TL-ALCF의 포함(subsumption) 문제에 대한 사운드(sound)하고 완전(complete) 추론 절차가 제안되었으며 또한 이 절차가 결정 가능(decidable)하다는 것을 보였다. TL-ALCF는 크게 시간을 표현할 수 있는 시간 논리(Temporal Logics)인 TL과, ALC에 함수 롤(functional role)을 표현하는 특징 논리(Feature Logics)가 추가된 비시간 논리(Non-temporal Logics)인 ALCF[10]로 구성된다.

TL-ALCF의 시간 개념(C와 B로 표현)은 그림 4의 구문 규칙에 따라 만들어지는데, 시간 개념(C, B)는 비시간 개념(E, F)와 구별된다. 원자 특징(atomic feature)과 원자 매개변수 특징(atomic parametric feature)의 이름은 똑같은 알파벳을 이용하지만 원자 매개변수 특

TL	$E, F \rightarrow C$	(비시간 개념)
	$E \cap F$	(연언)
	$E @ X$	(한정자)
	$E[Y] @ X$	(대체 한정자)
	$\diamond(X) \overline{Tc}.E$	(존재 정량자)
	$Tc \rightarrow (X(V)Y) (X(V)\#) (\#(V)Y)$	(시간 제약조건)
	$\overline{Tc} \rightarrow Tc Tc \overline{Tc}$	
	$V, W \rightarrow V, W$	(선언)
	$b a m mi o oi s si d di f fi =$	(Allen의 시간 관계)
	$X, Y \rightarrow x y z \dots$	(시간 변수)
$\overline{X} \rightarrow X X \overline{X}$		
ACCF	$C, B \rightarrow A$	(원자 개념)
	$T \perp \neg C C \cap B C \cup B \forall R.C \exists R.C$	
	$p \downarrow q$	(동의)
	$p \uparrow q$	(비동의)
	$p \uparrow\uparrow$	(미정의)
	$p : C$	(선택)
	$p, q \rightarrow f$	(원자 특징)
	$*g$	(원자 매개변수 특징)
$p \circ q$	(패스)	

그림 4 TL-ALCF 의 구문²⁾

징은 앞에 기호가 붙는다. p와 q는 패스(path)로써 $\gamma_1 \circ \dots \circ \gamma_k$ 와 같은 원자 특징 또는 원자 매개변수 특징 γ_i 의 유한한 시퀀스이다($1 \leq i \leq k$).

TL-ALCF에서 사용하는 시간 관계는 그림 5와 같이 Allen[4]의 시간 관계를 따른다. Allen의 시간 관계는 그림과 같이 6개의 시간관계와 이의 역관계 그리고 equal이 포함되는데, equal의 역관계가 없기 때문에 그림에서는 equal을 생략하였다. 시간 변수들은 정의된 행동이 발생하는 현재 시간을 나타내는 특별한 시간 변수인 # (now라고 부른다)를 제외하고는, 시간 존재 정량자(temporal existential quantifier)인 “ \diamond ”와 함께 표현된다.

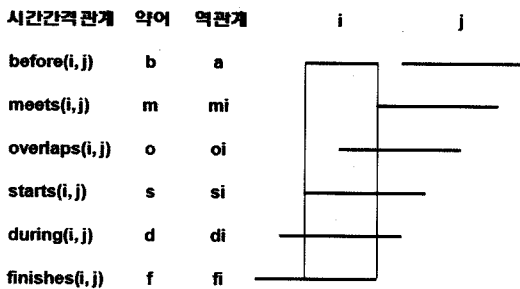


그림 5 Allen의 시구간 관계

TL-ALCF가 실제 어떻게 사용되는지 설명하기 위해 다음과 같이 비행기 표를 예약하는 간단한 개념을 살펴본다.

Reserve-Flight-1 $\hat{=}$ $\diamond(x\ y)(x\ fi\ \#)(y\ mi\ \#)$.
 $((*TICKET1:Unreserved)@x \ \cap$
 $(*TICKET1:Reserved)@y)$

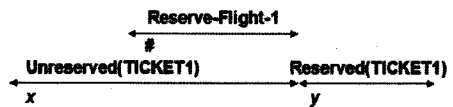


그림 6 Reserve-Flight-1 개념이 가지는 시구간들의 시간 의존 관계

위 그림은 Reserve-Flight-1 개념이 가지고 있는 시구간들의 시간 의존 관계를 보인 것으로, Reserve-Flight-1은 특정 시구간 동안 Unreserved 상태에 있다가 Reserved 상태에 있게 되는 행동을 표현한다. $(x\ fi\ \#)(y\ mi\ \#)$ 은 시간 제약으로써 Allen의 시간관계를 이용해서 시구간 x가 시구간 #과 같이 끝나고 시구간 y는 시구간 #와 만난다는 것을 나타낸다. @연산자는 bindable이라고 부르며, $(*TICKET:Unreserved)@x$ 와 $(*TICKET:Reserved)@y$ 에서 *TICKET: Unreserved가 시간 x에 일어나고 *TICKET:Reserved가 시간 y에 일어난다는 것을 나타낸다. 또한 개념을 기술할 때 특정

2) 알파벳 위에 선을 가지는 글자는 해당 알파벳의 집합을 의미한다. 하지만 본 논문에서는 이러한 기호 표현의 제약으로 인해 이를 (X), (Tc)와 같이 중괄호({})를 이용하는 표기법을 병행한다.

선택 연산자 : 를 사용하는데 이 연산자는 존재 정량자 3와 유사하지만 부분 함수(partial function)로 해석되는 의미를 가진다.

4. 시구간 온톨로지 언어

본 절에서는 OWL에 시구간 의미체계를 추가한 시구간 온톨로지 언어를 제안한다. 이 언어를 기술하기 위해서 다음과 같이 OWL 표준안[1]과 같은 방법을 사용한다.

4.1 요구조건과 표현언어

최근 W3C 웹 온톨로지 워킹 그룹에서는 시맨틱웹을 위한 온톨로지 언어인 OWL을 만들었다. 하지만 1절에서 언급한 것과 같이 OWL은 시간에 대한 의미체계를 가지지 않기 때문에 시간에 기반한 추론을 수행할 수 없는 문제가 있다. 뿐만 아니라 기존의 시간 기술 논리에 대한 연구[2]들을 보면 시간 추론을 제공하지만 모두 OWL의 표현력에는 많이 미치지 못한다. 따라서 본 연구에서는 OWL에 시간 의미체계를 추가한 온톨로지 언어인 TL-OWL(TemporaL Web Ontology Language)을 제안한다.

일반적으로 시간 기술 논리에 대한 연구들은 1절에서 기술한 것과 같이 시점 기반과 시구간 기반으로 나뉘어 있는데 이들은 시간을 어떻게 정형화하느냐에 따라서 달라지는 것이기 때문에 어떤 방법이 좋다고 판단하기는 어렵다. 하지만 본 연구에서는 이 두 방법 중 시구간에 기반한 시간 온톨로지 언어를 제안한다. TL-OWL을 정형화를 위해 사용하는 기술 논리는 TL-SHOIN(D)[11]로써, 이는 [9]의 TL-ALCF를 OWL DL의 표현력을 가지도록 확장한 언어이다. 이 시간 기술 논리에 대한 구문과 의미체계 그리고 결정가능한 추론 알고리즘은 [11]에서 이미 소개하였다. 특히, TL-SHOIN(D)에서의 포함문제 추론은 TL-ALCF의 추론 방법과 유사하다. 즉, TL-SHOIN(D) 개념의 일반 형태(normal form)을 구하고 두 개념간의 s-매핑이 존재함을 보임으로써 포함 여부를 결정할 수 있다.

본 장에서는 이와 같은 요구사항을 만족하는 시구간 온톨로지 언어를 정의하기 위해서 OWL의 표준안과 같은 방법을 이용한다. 특히, OWL 표준안에서는 온톨로지를 두 가지 형태의 구문과 의미체제로 표현하는데 이는 다음과 같다.

첫째, OWL 표준안에서는 사람이 OWL을 읽고 이해하기 쉽도록 하기 위해서 프레임 언어 스타일을 가지는 추상 구문(abstract syntax)으로 표현한다. 또한 이 추상 구문에 대한 의미체계, 즉 직접 모델 의미체계(Direct Model-Theoretic Semantics)을 정의한다. 둘째, OWL이 본래 RDF를 확장하여 정의되었기 때문에 RDF/XML로 기술되는 교환 구문(exchange syntax)을

정의하고 교환 구문을 파싱하여 RDF 그래프로 표현할 수 있도록 한다. 또한 교환 구문에 대한 의미체계인 RDF 호환 모델 의미체계(RDF-Compatible Model-Theoretic Semantics)을 정의한다.

OWL에서는 이 두 의미체계간에 변환이 가능하도록 하기 위해서 RDF 매핑을 통해 추상 구문을 RDF 트리플로 변환하는 방법을 제공하고 있으며, 추상구문 형태의 온톨로지가 RDF 변환을 통해 RDF 그래프로 변환되더라도 온톨로지 함의성이 보존되어야 한다는 것을 명시하고 있다. 함의성은 RDF 모델 의미체계의 기본 해석 방법으로써 온톨로지가 어떤 RDF 표현 또는 어떤 모델이 가지고 있는 모든 RDF 표현에 대한 의미를 포함하는지에 대한 것이다. 따라서 본 논문에서도 TL-OWL에 대한 함의성을 정의하였다.

4.2 추상 구문

OWL은 RDF를 확장한 언어이기 때문에 RDF 트리플로 처리되지만 RDF트리플 표현은 사람이 읽기에 너무 어렵다. 이 때문에 OWL 표준안에서는 OIL과 같은 프레임(frame) 언어 스타일의 추상 구문을 이용하여 OWL을 기술하는데, 추상 구문은 EBNF(Extended Backus-Naur Form) 표기법으로 표현된다. 이 절에서는 OWL과 같은 형태로 TL-OWL의 추상구문을 정의한다. 특히 OWL 구문을 TL-OWL의 구문과 구별하기 위해 이탤릭체로 표기하였다.

[9]에서는 시간 개념을 정의하기 위해 $(\{X\})(Tc).(Q_0 \sqcap Q_1 @ X_1 \sqcap \dots \sqcap Q_n @ X_n)$ 와 같은 일반 형태를 사용한다. 여기에서 $\{X\}$ 는 시간 변수 집합, $\{Tc\}$ 는 이 시간 변수들의 제약조건 그리고 Qualification $(Q_0 \sqcap Q_1 @ X_1 \sqcap \dots \sqcap Q_n @ X_n)$ 는 한정자(qualifier)들의 연언이 된다. 하지만 온톨로지에서는 모든 정보가 RDF의 Subject, Property, Object와 같은 SPO 형태로 표현하는데 이 일반 형태는 RDF 트리플로 표현할 수 없는 문제가 있다. 따라서 본 논문에서는 다음 표 1과 같이 모든 클래스와 구성자에 대해서 시구간 기반 논리의 의미체계는 유지하면서 RDF 트리플로 표현될 수 있도록 추상 구문과 의미체계를 정의한다.

TL-OWL은 클래스와 프로퍼티에 대한 정보를 제공하는 TemporalClass, TemporalVariable, Datatype-Property, ObjectProperty의 4개 공리를 가지는데, 각각의 공리는 공리 이름과 deprecated 정보, 그리고 주석(annotation) 정보를 가진다.

TemporalClass는 시간 개념을 표현하기 위한 클래스로써 하나 이상의 temporalDescription들을 가지며 다시 temporalDescription은 시간 변수와 비시간 개념을 연결하는 Qualification들의 연언을 가진다. Temporal-Variable은 시간 제약 네트워크를 RDF 트리플로 표현

표 1 TL-OWL 추상 구문

VID (variableID) ::= URireference featureID ::= URireference
axiom ::= 'TemporalClass(classID [Deprecated] {annotation} temporalDescription)' axiom ::= 'TemporalVariable(VID [Deprecated] {annotation} { description } (temporalRelation))' temporalDescription ::= 'intersectionOf(Qualification {Qualification})' Qualification ::= 'Qualification(onVariable(VID) bindVariable(description))' 'Qualification(onVariable(VID) onSubstitutiveVariable(VID) bindSubstitutiveVariable(description))' temporalRelation ::= 'before(VID) 'after(VID) 'meets(VID) 'metBy(VID) 'overlaps(VID) 'overlappedBy(VID) 'starts(VID) 'startedBy(VID) 'during(VID) 'contains(VID) 'finish(VID) 'finishedBy(VID) 'equal(VID)'
axiom ::= 'DatatypeProperty(datavaluedPropertyID [Deprecated] {annotation} ... ['Functional' 'ParametricFunctional'] ...)' 'ObjectProperty(individualvaluedPropertyID [Deprecated] {annotation} ... ['Functional' 'InverseFunctional' 'ParametricFunctional' 'InverseParametricFunctional'] ... ['pathOf(featureID featureID)])'
description ::= classID / restriction / feature 'unionOf(description))' 'intersectionOf(description))' 'complementOf(description))' 'oneOf(individualID))'
restriction ::= 'restriction(datavaluedPropertyID dataRestrictionComponent (dataRestrictionComponent))' 'restriction(individualvaluedPropertyID individualRestrictionComponent (individualRestrictionComponent))' dataRestrictionComponent ::= 'allValuesFrom(dataRange)' 'someValuesFrom(dataRange)' 'selectValuesFrom(dataRange)' 'value(individualID)' 'cardinality' individualRestrictionComponent ::= 'allValuesFrom(description)' 'someValuesFrom(description)' 'selectValuesFrom(description)' 'value(individualID)' 'cardinality'
feature ::= 'agreementOf(featureID featureID) 'disagreementOf(featureID featureID)' 'undefinednessOf(featureID))'

하기 위한 클래스이다. TemporalVariable은 시간 변수를 가지고 선언이 되며 하위 요소로 Allen의 시간 관계에 대한 프로퍼티를 가지고 이 프로퍼티가 다른 시간 변수를 참조함으로써 시간 변수들간의 시간 관계를 표현할 수 있다. 나머지 두 개의 프로퍼티 공리는 OWL의 프로퍼티 공리의 정의에 해당하는데, TL-OWL에서의 프로퍼티 공리는 OWL의 타입에서 추가적으로 ParametricFunctionl, InverseParametricFunctional 타입을 가질 수 있다. 또한 특정 논리(feature logic)[10]에서 사용하는 패스 연산자를 위한 pathOf를 프로퍼티를 가진다.

TL-OWL의 description에는 특정 로직 연산자들을 위한 feature가 추가되었으며, restriction에는 선택 연산자를 위한 selectValuesFrom이 추가되었다. 아래 그림은 그림 7의 Reserve-Flight-1 개념을 위의 추상 구문으로 표현한 것이다.

4.3 직접 모델 의미체계

TL-OWL 온톨로지에서 온톨로지 어휘(vocabulary)는 URI 참조와 리터럴(literal)을 가진다. 다음은 이러한 TL-OWL의 어휘를 정의한 것이다.

정의 3.1 **TL-OWL 어휘** V는 리터럴 집합 V_L 과 V_C, V_{TC} ,

```

TemporalClass ( ex:Reserve-Flight-1
intersectionOf (
  Qualification (
    onVariable ( ex:x )
    bindVariable (
      restriction (
        onProperty ( ex:TICKET1 )
        selectValuesFrom ( ex:Unreserved )
      )))
  Qualification (
    onVariable ( ex:y )
    bindVariable (
      restriction (
        onProperty ( ex:TICKET1 )
        selectValuesFrom ( ex:Reserved )
      )))
  ))))
TemporalVariable(ex:x finishedBy NOW)
TemporalVariable(ex:y metBy NOW)
    
```

그림 7 Reserve-Flight-1 개념의 추상 구문 표현

$V_{TV}, V_D, V_I, V_{DP}, V_{IP}, V_{AP}, V_O, V_{OP}$ 이렇게 10개의 URI 참조 집합으로 구성된다. V_C, V_{TC}, V_{TV}, V_D 는 서로소(disjoint)이며, $V_{DP}, V_{IP}, V_{AP}, V_{OP}$ 는 상호 서로소(pairwise disjoint)이다. V_C 는 클래스 이름, V_{TC} 는 시간 클래스 이름, V_D 는 데이터타입 이름, V_{AP} 는 주석 프로퍼티 이름, V_{IP} 는 개체값 프로퍼티 이름, V_{DP} 는 데이터값 프로퍼티 이름, V_I 는 개체 이름, V_{TV} 는 시간 변수 이름, V_O 는 온톨로지 이름, V_{OP} 는 빌트인 OWL 온톨로지 프로퍼티이다.

정의 3.2 D를 데이터타입 맵이라고 하자. D와 관련해서 $V_L, V_C, V_{TC}, V_{TV}, V_D, V_I, V_{DP}, V_{IP}, V_{AP}, V_O$ 어휘를 가지는 추상 TL-OWL 해석은 $I = \langle R, T, EC, ER, L, S, LV \rangle$ 형태의 튜플이 된다. (P 는 멱집합 연산자)

- R : 비어있지 않은 I의 리소스 집합
- LV : I의 리터럴 값. 유니코드 문자열, 언어 태그, D의 각 데이터타입에 대한 데이터값 공간을 포함하는 R의 부분집합
- T : I에서 시간에 대한 시구간 집합
- $EC : V_C \rightarrow P(O), V_{TC} \rightarrow P(T \times O), V_{TV} \rightarrow P(T), V_D \rightarrow P(LV)$
- $ER : V_{DP} \rightarrow P(O \times LV), V_{IP} \rightarrow P(O \times O), V_{AP} \cup \{ rdfs:type \} \rightarrow P(R \times R), V_{OP} \rightarrow P(R \times R)$
- $L : TL \rightarrow LV$ (TL 은 V_L 에서 타입을 가지는 리터럴의 집합)
- $S : V_I \cup V_{TC} \cup V_{TV} \cup V_C \cup V_D \cup V_{DP} \cup V_{IP} \cup V_{AP} \cup V_O \cup \{ owl:Ontology, owl:DeprecatedClass, owl:DeprecatedProperty \} \rightarrow R$

- $S(V_I) \subseteq O$
- $EC(owl:Thing) = O \subseteq R$ (O 는 LV 와 서로소)
- $EC(owl:Nothing) = \{ \}$
- $EC(rdfs:Literal) = LV$
- If $D(d') = d$ then $EC(d') = V(d)$
- If $D(d') = d$ then $L("v"^{d'}) \in V(d)$
- If $D(d') = d$ and $v \in L(d)$ then $L("v"^{d'}) = L2V(d)(v)$
- If $D(d') = d$ and $v \notin L(d)$ then $L("v"^{d'}) \in R - LV$

위 정의는 TL-OWL의 추상 구문에 대한 해석으로써 데이터타입 맵에 대한 정의는 [1]의 3.1절의 것을 따른다. EC는 TL-OWL 클래스와 데이터타입을 나타내는 URI 참조에 의미를 부여하며, ER은 TL-OWL 프로퍼티를 나타내는 URI 참조에 의미를 부여한다. 특히 TL-OWL에서의 시간 개념은 3절에서 소개한 시구간 기반 기술논리와 마찬가지로 시간에 대한 해석을 가진다. 따라서 본래 EC와 ER은 해석 함수 $EC_{V,L,H}$ 와 $ER_{V,L,H}$ 와 같이 표현되어야 하지만 본 논문에서는 의미체계 표현을 간단하게 하기 위해서 아래 첨자를 생략하도록 한다. L은 타입을 가진 리터럴에 의미를 부여하며, S는 OWL 개체를 나타내는 URI 참조와 주석에 의미를 부여한다. 또한 S는 일반 리터럴과 타입을 가지는 리터럴로 확장될 수도 있다.

4.3.1 구성자 의미체계

EC는 TL-OWL의 클래스를 구성하는 여러 구성요소들로 확장이 된다. 다음은 이러한 EC 확장을 나타낸 테이블으로써, OWL에 대한 구성자는 [1]의 3.2절에 나와 있기 때문에 생략하였으며 여기에서는 TL-OWL에만 나오는 구성자들을 기술하였다.

TL-OWL에는 NOW라는 빌트인 TL-OWL 시간 변수 하나가 존재하는데, 이 시간 변수는 시간 개념이 발생하는 현재의 시간을 나타낸다. 또한 13개의 Allen 시간관계는 빌트인 TL-OWL 개체값 프로퍼티로 정의되며 이를 통해 시간 변수들간의 시간 관계가 표현된다. qualification은 시간 변수와 비시간 개념을 연결해 주는데, 특히 bindSubstitutiveVariable은 대체 한정자를 표현하는 것으로써 변수 Y를 변수 X로 바꾸기 위해 사용하며, 두 시간 변수간에 상호 참조를 만들어 준다. 또한 qualification에서는 EC가 내포하고 있는 해석 함수의 의미체계에서 시구간 t와 제약조건 H의 표현을 가진다. 마지막 세 개의 구성자는 특정 논리에 대한 구성자로서 p와 q는 부분 함수가 되며 이 구성자의 의미체계에 나오는 dom은 부분 함수의 도메인을 나타낸다.

4.3.2 공리 의미체계

추상 TL-OWL 해석 I는 OWL의 공리와 사실 그리

표 2 EC 확장 테이블

추상 구분	의미체계
NOW	EC(NOW)는 현재 시구간
before(x), x는 시간 변수	$\{[u_1, v_1] \in T \mid EC(x)=[u_2, v_2] \text{ implies } v_1 < u_2\}$
after(x), x는 시간 변수	$\{[u_1, v_1] \in T \mid EC(x)=[u_2, v_2] \text{ implies } v_2 < u_1\}$
다른 시간 관계들..	...
qualification(x bindVariable(c)), x는 시간변수	$EC(c), t = V(x)$
qualification(x y bindSubstitutive-Variable(c)), x와 y는 시간변수	$EC(c), H = HU\{y \rightarrow V(x)\}$
restriction(p selectValuesFrom(e)), p는 부분함수	$\{x \in \text{dom}p \mid ER(p)(x) \in EC(e)\}$
agreementOf(p q), p와 q는 부분함수	$\{x \in \text{dom}p \cap \text{dom}q \mid ER(p)(x) = ER(q)(x) \}$
disagreementOf(p q), p와 q는 부분함수	$\{x \in \text{dom}p \cap \text{dom}q \mid ER(p)(x) \neq ER(q)(x) \}$
undefinednessOf(p), p와 q는 부분함수	$O - \text{dom}p$

고 다음 테이블에서 소개하는 4개의 공리를 만족한다. 이 테이블에 나오는 공리는 OWL의 공리에서 추가되는 부분만을 기술한 것으로 OWL의 공리는 [1]의 3.2절을

참고한다. 또한 주석과 deprecated에 대한 부분은 생략하였다. 아래의 표 3에서 대괄호[...]는 옵션 항목을 나타낸다.

표 3 공리의 해석

추상 구분	의미체계
TemporalClass(c q ₁ ... q _n)	$EC(c) = EC(q_1) \cap \dots \cap EC(q_n)$
TemporalVariable(c [descr ₁ ... descr _n] ([before(v ₁)] [after(v ₁)] [meets(v ₁)] [metBy(v ₁)] [overlaps(v ₁)] [overlappedBy(v ₁)] [starts(v ₁)] [startedBy(v ₁)] [during(v ₁)] [contains(v ₁)] [finish(v ₁)] [finishedBy(v ₁)] [equal(v ₁)]) ... ([before(v _k)] [after(v _k)] [meets(v _k)] [metBy(v _k)] [overlaps(v _k)] [overlappedBy(v _k)] [starts(v _k)] [startedBy(v _k)] [during(v _k)] [contains(v _k)] [finish(v _k)] [finishedBy(v _k)] [equal(v _k)])	$EC(c) = [EC(\text{descr}_1) \cap \dots \cap EC(\text{descr}_n)] \cap$ $([ER(\text{before}(v_1))] [ER(\text{after}(v_1))] \cup$ $[ER(\text{meets}(v_1))] [ER(\text{metBy}(v_1))] \cup$ $[ER(\text{overlaps}(v_1))] [ER(\text{overlappedBy}(v_1))] \cup$ $[ER(\text{starts}(v_1))] [ER(\text{startedBy}(v_1))] \cup$ $[ER(\text{during}(v_1))] [ER(\text{contains}(v_1))] \cup$ $[ER(\text{finish}(v_1))] [ER(\text{finishedBy}(v_1))] \cup$ $[ER(\text{equal}(v_1))])$ $\cap \dots \cap$ $([ER(\text{before}(v_k))] [ER(\text{after}(v_k))] \cup$ $[ER(\text{meets}(v_k))] [ER(\text{metBy}(v_k))] \cup$ $[ER(\text{overlaps}(v_k))] [ER(\text{overlappedBy}(v_k))] \cup$ $[ER(\text{starts}(v_k))] [ER(\text{startedBy}(v_k))] \cup$ $[ER(\text{during}(v_k))] [ER(\text{contains}(v_k))] \cup$ $[ER(\text{finish}(v_k))] [ER(\text{finishedBy}(v_k))] \cup$ $[ER(\text{equal}(v_k))])$
DatatypeProperty(p super(s ₁) ... super(s _n) domain(d ₁) ... domain(d _n) range(r ₁) ... range(r _n) [Functional] [ParametricFunctional])	$ER(p) \subseteq O \times LV \cap ER(s_1) \cap \dots \cap ER(s_n) \cap$ $EC(d_1) \times LV \cap \dots \cap EC(d_n) \times LV \cap$ $O \times EC(r_1) \cap \dots \cap O \times EC(r_n)$ [ER(p)가 functional] [ER(p)가 parametric functional]
ObjectProperty(p super(s ₁) ... super(s _n) domain(d ₁) ... domain(d _n) range(r ₁) ... range(r _n) [inverse(i)] [pathOf(x y)] [Symmetric] [Functional] [ParametricFunctional] [InverseFunctional] [InverseParametricFunctional] [Transitive])	$ER(p) \subseteq O \times O \cap ER(s_1) \cap \dots \cap ER(s_n) \cap$ $EC(d_1) \times O \cap \dots \cap EC(d_n) \times O \cap$ $O \times EC(r_1) \cap \dots \cap O \times EC(r_n)$ [ER(p)는 ER(i)의 역] [<u,v> ∈ ER(x) ∩ <v,w> ∈ ER(y) implies <u,w> ∈ ER(p), x,y가 부분함수이고, u ∈ domx, v ∈ domy] [ER(p)가 symmetric] [ER(p)가 functional] [ER(p)가 parametric functional] [ER(p)가 inverse functional] [ER(p)가 inverse parametric functional] [ER(p)가 transitive]

4.3.3 온톨로지 해석

다음 정의는 TL-OWL의 추상 구문과 의미 체계에 따라 TL-OWL 온톨로지를 해석하고 추론하는 방법을 기술한 것이다. 특히 기술 논리의 만족가능성(정의 3.3)과 일관성(정의 3.4), 온톨로지들간의 합의성(정의 3.5)을 정의하였다.

정의 3.3 D를 데이터타입 맵이라고 하자. D와 관련해서 $V_L, V_C, V_{TC}, V_{TV}, V_D, V_I, V_{DP}, V_{IP}, V_{AP}, V_O$ 어휘를 가지는 추상 TL-OWL 해석이 TL-OWL 온톨로지 O를 만족한다는 것은 다음의 조건들에 필요충분조건이 된다.

- O에서 클래스 아이디(시간 클래스 아이디, 시간변수 아이디, 데이터타입 아이디, 개체 아이디, 데이터값 프로퍼티 아이디, 개체값 프로퍼티 아이디, 주석 프로퍼티 아이디, 주석 아이디, 온톨로지 아이디)로 사용되는 URI 참조는 각각 $V_C(V_{TC}, V_{TV}, V_D, V_I, V_{DP}, V_{IP}, V_{AP}, V_O)$ 에 속한다.
- O에서 각각의 리터럴은 V_L 에 속한다.
- I는 온톨로지 주석을 제외하고 O의 각 지시자(directive)를 만족한다.
- $\langle o, S(owl:Ontology) \rangle \in ER(rdf:type)$ 를 가지는 어떤 $o \in R$ 가 존재. 이 때 $Annotation(p \ v), \langle o, S(v) \rangle \in ER(p)$ 의 형태를 가지는 온톨로지 주석이 있고, 만약 O가 이름 n을 가지면 $S(n) = o$ 가 성립.
- I가 O의 *owl:imports* 주석 지시자에서 나온 각각의 온톨로지를 만족한다.

정의 3.4 추상 TL-OWL 온톨로지, 공리, 사실의 집합이 데이터타입 맵 D와 관련해서 **일관성(consistency)**을 가지는 것은 D와 관련해서 집합의 각 온톨로지, 공리, 사실을 만족하는 어떤 해석 I가 존재하는 것에 필요충분조건이다.

정의 3.5 만약 데이터타입 맵 D와 관련해서 추상 TL-OWL 온톨로지, 공리, 사실의 집합 O를 만족하는 해석이 다른 추상 TL-OWL 온톨로지, 공리, 사실의 집합 O'를 만족한다면, 데이터타입 맵 D와 관련해서 O가 O'를 **합의(entail)**한다고 한다.

4.4 RDF 그래프 변환

이 절에서는 TL-OWL의 추상 구문을 TL-OWL의 교환 구문으로 변환하는 방법을 기술한다. TL-OWL의 교환 구문은 RDF/XML 형태를 가지는데 이 RDF/XML 형태의 TL-OWL 온톨로지는 RDF/XML 문서를 파싱해 만들어지는 RDF 그래프가 된다. 따라서 추상 구문의 TL-OWL 온톨로지를 교환 구문으로 바꾸는 것은 온톨로지의 각 지시자를 RDF 트리플 집합으로 변환하는 것이 된다.

표 4는 표 2와 표 3에서 소개한 TL-OWL의 추상 구

문을 교환 구문으로 변환하는 변환 규칙을 기술한 것이다. 테이블의 왼쪽 열은 추상 구문(S), 가운데 열은 추상 구문의 트리플 변환 (T(S)), 오른쪽 열은 트리플 변환의 메인노드 (M(T(S)))에 대한 식별자를 나타낸다. 반복되는 것들은 Qualification₁... Qualification_n와 같이 생략기호를 사용하였으며, 추상 구문에서 옵션인 것은 대괄호([])를 사용하였고, 변환 규칙에서 어떤 트리플은 생성될 수도 있고 생성되지 않을 수 있는데 이러한 트리플은 [opt]로 표시하였다.

표 4에서 어떤 변환은 온톨로지의 지시어(Directive)를 위한 것이며, 어떤 변환은 이 지시어의 일부를 위한 것이다. 따라서 지시어를 변환할 때 지시어가 가지고 있는 컴포넌트를 변환하기 위해 다른 변환 규칙을 호출하게 된다.

그림 7에서 추상 구문으로 표현된 Reserve-Flight-1 개념은 표 4의 변환 규칙에 따라 RDF 트리플로 변환될 수 있으며, 이를 RDF/XML 교환 구문으로 표현하면 다음과 같다.

4.5 RDF 호환 모델 의미체계

TL-OWL에 대한 의미체계는 RDF 의미체계를 확장한 것으로 정의된다. RDF와 RDFS 어휘를 가지는 URI 참조와 리터럴의 집합을 V 그리고 데이터타입 맵을 D 라고 하면 V에 대한 D-해석 I는 $I = \langle R_I, T_I, P_I, EXT_I, S_I, L_I, LV_I \rangle$ 가 된다. 여기에서 R_I 는 논의의 영역(domain of discourse 또는 universe)으로써 V에 있는 URI 참조 표시와 리터럴의 표시를 가지고 있는 집합이다. T_I 는 I의 시공간 집합, P_I 는 I의 프로퍼티이며, EXT_I 는 P_I 에서 $P(R_I \times R_I)$ 로의 매핑으로 프로퍼티에 의미를 부여하기 위해 사용된다. S_I 는 V가 가지는 URI 참조를 R_I 에서의 표시자로 매핑하고, L_I 은 V에서 타입을 가지는 리터럴을 R_I 에서의 표시자로 매핑하는 역할을 하는 것으로써, 이들은 각각 TL-OWL의 URI 참조와 리터럴을 실제 값으로 매핑하는 역할을 하게 된다. LV_I 는 R_I 의 부분집합으로써, 유니코드 문자열 집합, 유니코드 문자열 쌍의 집합, 언어 태그, D의 각 데이터타입에 대한 값의 공간을 포함한다.

다음은 TL-OWL의 각각의 어휘에 대한 RDF 호환 모델 의미체계로써, 여기에서는 OWL의 것을 제외한 TL-OWL의 의미체계만을 기술한다. 클래스의 집합 C_I 는 $C_I = \{x \in R_I \mid \langle x, S_I(rdfs:Class) \rangle \in EXT_I(S_I(rdf:type)) \wedge \langle T, x \rangle \in C_I\}$ 와 같이 정의된다. 이는 TL-OWL 클래스는 rdfs:Class의 하위클래스로 정의됨을 의미한다. 또한 C_I 에서 $P(R_I)$ 로의 매핑 $CEXT_I$ 는 $CEXT_I(c) = \{x \in R_I \mid \langle x, c \rangle \in EXT_I(S_I(rdf:type)) \wedge \langle T, x \rangle \in C_I\}$ 와 같이 정의되는데 이는 클래스 c의 인스턴스이며 시간 T 정보를 가지는 리소스 x들의 집합을 나타낸다.

표 4 트리플 변환

추상 구문 - S	변환 - T(S)	메인노트 - M(T(S))
vID	vID rdf:type tl:TemporalVariable.	vID
featureID	featureID rdf:type owl:FunctionalProperty. OR featureID rdf:type tl:ParametricFunctionalProperty. OR featureID rdf:type owl:InverseFunctionalProperty. OR featureID rdf:type tl:InverseParametricFunctionalProperty. featureID rdf:type rdf:Property [opt].	featureID
Qualification(vID C)	_:x rdf:type tl:Qualification. _:x rdf:type rdfs:Class [opt]. _:x tl:onVariable T(vID). _:x tl:bindVariable T(C).	_:x
Qualification(vID ₁ , vID ₂ , C)	_:x rdf:type tl:Qualification. _:x rdf:type rdfs:Class [opt]. _:x tl:onVariable T(vID ₁). _:x tl:onSubstitutiveVariable T(vID ₂). _:x tl:bindSubstitutiveVariable T(C).	_:x
Restriction(ID selectValuesFrom(selection))	_:x rdf:type tl:Restriction. _:x rdf:type rdfs:Class [opt]. _:x owl:onProperty T(ID). _:x tl:selectValuesFrom T(selection).	_:x
TemporalClass(classID Qualification ₁ ... Qualification _n)	classID rdf:type tl:TemporalClass. classID rdf:type rdfs:Class [opt]. classID owl:intersectionOf T(SEQ Qualification ₁ ... Qualification _n)	
TemporalVariable(classID temporalRelation ₁ ... temporalRelation _n)	classID rdf:type tl:TemporalVariable. classID rdf:type rdfs:Class [opt]. classID owl:unionOf T(SEQ temporalRelation ₁ ... temporalRelation _n)	
DatatypeProperty(ID [ParametricFunctional] ...)	ID rdf:type owl:DatatypeProperty. ID rdf:type rdf:Property. [opt] [ID rdf:type tl:ParametricFunctional.] ...	
ObjectProperty(ID [ParametricFunctional InverseParametricFunctional] ... [pathOf(featureID ₁ featureID ₂)] ...)	ID rdf:type owl:ObjectProperty. ... ID rdf:type rdf:Property. [opt] [ID rdf:type tl:ParametricFunctional.] ... [ID rdf:type tl:InverseParametricFunctional.] ... [ID tl:pathOf T(SEQ featureID ₁ featureID ₂)] ...	
agreementOf(featureID ₁ featureID ₂)	_:x rdf:type rdfs:Class. _:x tl:agreementOf T(SEQ featureID ₁ , featureID ₂).	_:x
disagreementOf(featureID ₁ featureID ₂)	_:x rdf:type rdfs:Class. _:x tl:disagreementOf T(SEQ featureID ₁ , featureID ₂).	_:x
undefinednessOf(featureID)	_:x rdf:type rdfs:Class. _:x tl:undefinednessOf T(featureID).	_:x
before(vID)	_:x rdf:type tl:TemporalVariable. vID rdf:type tl:TemporalVariable. _:x tl:before vID.	_:x
after(vID)	_:x rdf:type tl:TemporalVariable. vID rdf:type tl:TemporalVariable. _:x tl:after vID.	_:x
다른 시간 관계들..

```

<tl:TemporalClass rdf:about="#Reserve-Flight-1">
  <owl:intersectionOf rdf:parseType="Collection">
    <tl:Qualification>
      <tl:onVariable rdf:resource="#x"/>
      <tl:bindVariable>
        <owl:Restriction>
          <owl:onProperty rdf:resource="#TICKET1"/>
          <tl:selectValuesFrom rdf:resource="#Unreserved"/>
        </owl:Restriction>
      </tl:bindVariable>
    </tl:Qualification>
    <tl:Qualification>
      <tl:onVariable rdf:resource="#y"/>
      <tl:bindVariable>
        <owl:Restriction>
          <owl:onProperty rdf:resource="#TICKET1"/>
          <tl:selectValuesFrom rdf:resource="#Reserved"/>
        </owl:Restriction>
      </tl:bindVariable>
    </tl:Qualification>
  </owl:intersectionOf>
</tl:TemporalClass>
<tl:TemporalVariable rdf:about="#x">
  <tl:finishedBy rdf:resource="#tl:NOW"/>
</tl:TemporalVariable>
<tl:TemporalVariable rdf:about="#y">
  <tl:metBy rdf:resource="#tl:NOW"/>
</tl:TemporalVariable>

```

그림 8 Reserve-Flight-1 개념의 RDF/XML 표현

아래 표 11은 TL-OWL의 위의 RDF 호환 모델의 미체계에 추가적으로 가지는 조건들을 기술한 것이다.

이는 모델이 RDF 트리플로 표현되기 때문에 트리플 구조에 의해 가질 수 있는 내포된 의미들으로써 일반적으로 프로퍼티를 중심으로 S 또는 O가 있을 때 존재하는 정보들을 기술한다.

정의 3.9 D를 rdf:XMLLiteral , xsd:integer , xsd:string 데이터타입을 포함하는 데이터타입 맵이라고 하자. RDF, RDFS, OWL, TL-OWL 어휘를 포함하는 어휘 V에 대한 **TL-OWL 해석** $I = \langle R, T_i, P_i, \text{EXT}_i, S_i, L_i, \text{LV}_i \rangle$ 는 표 5~표 11에 기술된 모든 제약조건을 만족하는 V의 D-해석이라고 한다.

4.6 TL-OWL 함의성

RDF에 기반한 온톨로지의 함의성이라는 것은 기술 논리의 포함문제와 비슷한 의미를 가진다. 즉, "온톨로지가 어떤 RDF 표현(statement) 또는 어떤 모델이 가지고 있는 모든 RDF 표현에 대한 의미를 포함하는가"에 대한 것이다. 이에 대한 정형화된 정의는 다음과 같이 기술될 수 있다.

정의 3.10 K와 Q를 RDF 그래프의 집합, D를 데이터타입 맵이라고 하자. 그러면, D와 관련해서 K가 Q를 **TL-OWL 함의(entail)**하는 것은 D와 관련해서 K에 있는 모든 RDF 그래프를 만족하는 모든 TL-OWL 해석이 또한 Q에 있는 모든 RDF 그래프를 만족하는 것에 필요충분조건이다. K가 **일관성(consistency)**을 가지는 것은 K에 있는 모든 RDF 그래프를 만족하는 TL-OWL 해석이 있다는 것에 필요충분조건이다.

다음은 TL-OWL의 함의성을 정리한 것이다. 즉, TL-OWL의 직접 모델 의미체계와 RDF 호환 모델의 미체계간에 RDF 변환을 수행하더라도 TL-OWL 온톨로지의 함의성은 유지된다.

정리 3.1 O와 O'를 임포트에 달려 있는 추상구문 형

표 5 TL-OWL 구조적 구분에 대한 조건

If E is	then			설 명
	$S_1(E) \in$	$\text{CEXT}_1(S_1(E)) =$	and	
<i>tl:TemporalClass</i>	C_1	ITC	$\text{ITC} \subseteq C_1$	ITC를 TL-OWL 클래스의 집합으로 정의
<i>tl:TemporalVariable</i>		ITV	$\text{ITV} \subseteq C_1$	ITV를 TL-OWL 시간 변수의 집합으로 정의
<i>tl:Qualification</i>	C_1	ITQ	$\text{ITQ} \subseteq C_1$	ITQ를 TL-OWL qualification 집합으로 정의

표 6 TL-OWL 클래스, 데이터타입, 프로퍼티 특징

If E is	then if $e \in \text{CEXT}_1(S_1(E))$ then	설 명
<i>tl:TemporalClass</i>	$\text{CEXT}_1(e) \subseteq \text{IOT}$	TL-OWL 클래스의 인스턴스는 TL-OWL 개체이다.
<i>tl:TemporalVariable</i>	$\text{CEXT}_1(e) \subseteq T$	TL-OWL 시간 변수는 시구간값들이다.
If E is	then $c \in \text{CEXT}_1(S_1(E))$ iff $c \in \text{IOOPUIODP}$ and	설 명
<i>tl:Parametric FunctionalProperty</i>	시간에 상관없이, $\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle \in \text{EXT}_1(c)$ implies $y_1 = y_2$.	개체값 프로퍼티와 데이터값 프로퍼티는 매개변수 함수 프로퍼티가 될 수 있다.
If E is	then $c \in \text{CEXT}_1(S_1(E))$ iff $c \in \text{IOOP}$ and	설 명
<i>tl:InverseParametric FunctionalProperty</i>	시간에 상관없이, $\langle x_1, y \rangle, \langle x_2, y \rangle \in \text{EXT}_1(c)$ implies $x_1 = x_2$.	개체값 프로퍼티만 역 매개변수 함수의 프로퍼티가 될 수 있다.

표 7 TL-OWL restriction에 대한 조건

If	then $x \in \text{IOR}, y \in \text{IOCIDC}$, 부분함수 $p \in \text{IOOPUIODP}$, and $\text{CEXT}_1(x) =$
$\langle x, y \rangle \in \text{EXT}_1(S_1(\text{tl:selectValuesFrom})) \wedge \langle x, p \rangle \in \text{EXT}_1(S_1(\text{owl:onProperty}))$	$\{u \in \text{domp} \mid \text{EXT}_1(p)(u) \in \text{CEXT}_1(y)\}$

표 8 TL-OWL qualification에 대한 조건

If	then $z \in \text{ITQ}, d \in \text{IOCIDC}, x, y \in \text{ITV}$, and $\text{CEXT}_1(z) =$
$\langle z, d \rangle \in \text{EXT}_1(S_1(\text{tl:bindVariable})) \wedge \langle z, x \rangle \in \text{EXT}_1(S_1(\text{tl:onVariable}))$	$\{u \in \text{IOT} \mid u \in \text{CEXT}_1(d), t = V(x)\}$
$\langle z, d \rangle \in \text{EXT}_1(S_1(\text{tl:bindSubstitutiveVariable})) \wedge \langle z, x \rangle \in \text{EXT}_1(S_1(\text{tl:onVariable})) \wedge \langle z, y \rangle \in \text{EXT}_1(S_1(\text{tl:onSubstitutiveVariable}))$	$\{u \in \text{IOT} \mid u \in \text{CEXT}(d), H = H\{y!V(x)\}\}$

표 9 TL-OWL 특징에 대한 조건

If E is	then $\langle x, y \rangle \in \text{EXT}_1(S_1(E))$ iff
<i>tl:agreementOf</i>	$x \in \text{IOC}, y \in \text{IOOPUIODP}$ 에 속하는 부분함수 p, q 의 시퀀스, $\text{CEXT}_1(x) = \{u \in \text{domp} \cap \text{dom}_q \mid \text{EXT}_1(p)(u) = \text{EXT}_1(q)(u)\}$
<i>tl:disagreementOf</i>	$x \in \text{IOC}, y \in \text{IOOPUIODP}$ 에 속하는 부분함수 p, q 의 시퀀스, $\text{CEXT}_1(x) = \{u \in \text{domp} \cap \text{dom}_q \mid \text{EXT}_1(p)(u) \neq \text{EXT}_1(q)(u)\}$
<i>tl:undefinednessOf</i>	$x \in \text{IOC}, y \in p \in \text{IOOPUIODP}$ 인 부분함수, $\text{CEXT}_1(x) = \text{IOT-domp}$
<i>tl:pathOf</i>	$x \in \text{IOOP}, y \in \text{IOOP}$ 에 속하는 부분함수 p, q 의 시퀀스, $u \in \text{domp}$ and $v \in \text{dom}_q$ 에 대해서 $\langle u, v \rangle \in \text{EXT}_1(p) \cap \langle v, w \rangle \in \text{EXT}_1(q)$ implies $\langle u, w \rangle \in \text{EXT}_1(x)$

표 10 TL-OWL 시간 관계에 대한 조건

If E is	then $x \in \text{CEXT}_1(S_1(E))$ iff
<i>tl:NOW</i>	$\text{CEXT}_1(x)$ 는 현재 시구간
If E is	then $\langle x, y \rangle \in \text{EXT}_1(S_1(E))$ iff
<i>tl:before</i>	$\text{CEXT}_1(x) = [u_1, v_1] \wedge \text{CEXT}_1(y) = [u_2, v_2]$ implies $v_1 < u_2$
<i>tl:after</i>	$\text{CEXT}_1(x) = [u_1, v_1] \wedge \text{CEXT}_1(y) = [u_2, v_2]$ implies $u_1 > v_2$
다른 시간 관계들

태의 TL-OWL 은톨로지, 공리, 사실 집합이라고 하자. xsd:string, xsd:integer를 적당한 XML 스키마 데이터 타입으로 매핑하고 rdf:XMLLiteral에 대한 RDF 매핑을 포함하는 데이터타입 맵 D가 주어진다면, D와 관련해서 O가 O'를 함의하는 것은 D와 관련해서 O의 변환이 O'의 변환을 TL-OWL 함의하는 것에 필요충분조건이다. 정리 3.1에 증명은 [13]에서 소개한다.

5. 맺음말

OWL은 W3C에서 시맨틱웹을 위해 만든 은톨로지 언어로서 웹 상에 존재하는 정보 자원에 대한 개념을 표현하고 추론할 수 있도록 개발되었다. 하지만 OWL은 시간에 대한 의미체계를 표현할 수 없기 때문에 시간 정보에 기반한 추론을 수행할 수 없다. 따라서 본 연구에서는 시맨틱웹에서 시간 정보를 효율적으로 표현하고 추론할 수 있는 시구간 은톨로지 언어인 TL-OWL을 제안하였다. 이를 위해 OWL 표준안과 같은 방법으로 TL-OWL에 대한 추상 구문과 교환 구문을 정의하고 추상 구문의 직접 모델 의미체계와 교환 구문의 RDF

교환 모델 의미체계를 정의하였다. 또한 이 두 의미체계 간의 변환 방법을 제공하였으며 이러한 변환을 수행하더라도 TL-OWL 은톨로지에 대한 함의성이 보존된다는 것을 증명하였다.

본 논문에서는 TL-OWL의 의미체계를 제안하는 것이 목적이기 때문에 활용 예에 대해서는 구체적으로 기술하지 않았지만, TL-OWL은 시간 정보 추론이 필요한 여러 도메인에서 사용이 가능하다. 예를 들어 이벤트의 시간 정보가 중요한 뉴스의 동향 분석 서비스[11] 나 시맨틱 웹서비스 조합[12] 과 같이 웹서비스들간의 순서 정보가 필요한 어플리케이션에서 사용이 가능하다. 이 뿐만 아니라 시맨틱웹에서 시간을 표현하기 위한 다양한 어플리케이션에서 이용될 수 있다.

본 연구에서는 OWL을 시구간 기반으로 확장하였다. 시간 정보는 다양한 형태로 정형화될 수 있기 때문에 본 연구는 시간 기반 은톨로지 언어의 한 형태가 될 수 있는데, 특히 본 연구에서 언급한 시점 논리를 기반으로 OWL의 확장이 가능하며 향후에는 이에 대한 연구가 필요하다.

표 11 TL-OWL에 내포된 조건 (Comprehension principle)

If there exists	then there exists y with
부분함수 p,q에 대해 IOOP ∪ IODP에 속하는 p,q의 시퀀스인 I	$y \in \text{IOC}, \langle y, I \rangle \in \text{EXT}_1(S_i(\text{tl:agreementOf}))$
부분함수 p,q에 대해 IOOP ∪ IODP에 속하는 p,q의 시퀀스인 I	$y \in \text{IOC}, \langle y, I \rangle \in \text{EXT}_1(S_i(\text{tl:disagreementOf}))$
$p \in \text{IOOP} \cup \text{IODP}$ 인 부분함수 p	$y \in \text{IOC}, \langle y, p \rangle \in \text{EXT}_1(S_i(\text{tl:undefinednessOf}))$
If there exists	then there exists y ∈ IOR with
$x \in \text{IOOP} \cup \text{IODP} \wedge w \in \text{IOC} \cup \text{IDC}$	$\langle y, x \rangle \in \text{EXT}_1(S_i(\text{owl:onProperty})) \wedge \langle y, w \rangle \in \text{EXT}_1(S_i(\text{tl:selectValuesFrom}))$
If there exists	then there exists z ∈ ITQ with
$x \in \text{ITV} \wedge w \in \text{IOC} \cup \text{IDC}$	$\langle z, x \rangle \in \text{EXT}_1(S_i(\text{tl:onVariable})) \wedge \langle z, w \rangle \in \text{EXT}_1(S_i(\text{tl:bindVariable}))$
$x, y \in \text{ITV} \wedge w \in \text{IOC} \cup \text{IDC}$	$\langle z, x \rangle \in \text{EXT}_1(S_i(\text{tl:onVariable})) \wedge \langle z, y \rangle \in \text{EXT}_1(S_i(\text{tl:onSubstitutiveVariable})) \wedge \langle z, w \rangle \in \text{EXT}_1(S_i(\text{tl:bindSubstitutiveVariable}))$
If there exists	then there exists y ∈ ITV with
$x \in \text{ITV}$	$\langle y, x \rangle \in \text{EXT}_1(S_i(\text{tl:before}))$
$x \in \text{ITV}$	$\langle y, x \rangle \in \text{EXT}_1(S_i(\text{tl:after}))$
다른 시간 관계들...	...
If there exists	then there exists $y_1 \dots y_k \in \text{ITV}$ with
$x \in \text{ITV}$	$\langle x, y_1 \rangle \in \text{EXT}_1(S_i(\text{before})) \cup \text{EXT}_1(S_i(\text{after})) \cup \text{EXT}_1(S_i(\text{meets})) \cup \text{EXT}_1(S_i(\text{metBy})) \cup \text{EXT}_1(S_i(\text{overlaps})) \cup \text{EXT}_1(S_i(\text{overlappedBy})) \cup \text{EXT}_1(S_i(\text{starts})) \cup \text{EXT}_1(S_i(\text{startedBy})) \cup \text{EXT}_1(S_i(\text{during})) \cup \text{EXT}_1(S_i(\text{contains})) \cup \text{EXT}_1(S_i(\text{finish})) \cup \text{EXT}_1(S_i(\text{finishedBy})) \cup \text{EXT}_1(S_i(\text{equal})) \cap \dots \cap \langle x, y_k \rangle \in \text{EXT}_1(S_i(\text{before})) \cup \text{EXT}_1(S_i(\text{after})) \cup \text{EXT}_1(S_i(\text{meets})) \cup \text{EXT}_1(S_i(\text{metBy})) \cup \text{EXT}_1(S_i(\text{overlaps})) \cup \text{EXT}_1(S_i(\text{overlappedBy})) \cup \text{EXT}_1(S_i(\text{starts})) \cup \text{EXT}_1(S_i(\text{startedBy})) \cup \text{EXT}_1(S_i(\text{during})) \cup \text{EXT}_1(S_i(\text{contains})) \cup \text{EXT}_1(S_i(\text{finish})) \cup \text{EXT}_1(S_i(\text{finishedBy})) \cup \text{EXT}_1(S_i(\text{equal}))$

참 고 문 헌

[1] W3C Recommendation, "OWL Web Ontology Language Semantics and Abstract Syntax," <http://www.w3.org/TR/owl-semantics>, 2004.

[2] Artale, A., and Franconi, E., "A Survey of Temporal Extensions of Description Logics," *Annals of Mathematics and Artificial Intelligence*, Kluwer Academic Press, vol.30, issue 1-4, pp.171-210, 2001.

[3] Milea, V., Frasinca, F., Kaymak, U., Noia, T., "An OWL-based Approach Towards Representing Time in Web Information Systems," *Proc. of the 4th International Workshop of Web Information Systems Modeling Workshop*, pp.791-802, 2007.

[4] Allen, J. F., "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, no.11, pp.832-843, 1983.

[5] TimeML Working Group, "TimeML 1.2.1 A Formal Specification Language for Events and Temporal Expressions," http://timeml.org/site/publications/timeMLdocs/timeml_1.2.1.html, 2005.

[6] ISO, "Language Resource Management - Semantic Annotation Framework (SemAF) - Part 1: Time and Events," ISO Report ISO/TC37/SC4 N269 version 19 (ISO/WD 24617-1), 2007.

[7] Hobbs, J. R., and Pan, F., "An Ontology of Time for the Semantic Web," *ACM Transactions on Asian Language Processing (TALIP): Special issue on Temporal Information Processing*, vol.3, no.1, pp.66-85, 2004.

[8] Welty, C., Fikes, R., Makarios, S., "A Reusable Ontology for Fluents in OWL," *Proc. of the International Conference on Formal Ontology in Information Systems*, pp.226-236, 2006.

[9] Artale, A., and Franconi, E., "A Temporal Description Logic for Reasoning about Actions and Plans," *Journal of Artificial Intelligence Research*, vol.9, pp.463-506, 1998.

[10] Hollunder, B., and Nutt, W., "Subsumption Algorithms for Concept Languages," Technical Research Report RR-90-04, DFKI, Germany, 1990.

[11] 김상균, 이규철, "시간 웹 온톨로지 언어를 이용한 뉴스 동향 분석 서비스", *한국전자거래학회지*, vol.12, no.3, pp.133-150, 2007.

[12] 김상균, 이규철, 시맨틱 웹서비스 조합을 위한 지식 전제조건, *정보과학회논문지: 소프트웨어 및 응용*, vol.32, no.9, pp.888-900, 2005.

[13] "TL-OWL의 합성성 보존 증명(Supplementary Proof)",

http://dmlab.cnu.ac.kr/~skkim/papers/TL-OWL_SP.pdf, 2008.



김 상 균

1999년 충남대학교 공과대학 정보통신공학과(공학사). 2001년 충남대학교 공과대학 컴퓨터공학과(공학석사). 2008년 충남대학교 공과대학 컴퓨터공학과(공학박사) 2003년~2007년 (주)K4M 연구원. 2007년~현재 한국한의학연구원 정보연구센터 선임연구원. 관심분야는 시맨틱웹, 온톨로지, 소셜 네트워크



이 규 철

1984년 서울대학교 공과대학 컴퓨터공학과(공학사). 1986년 서울대학교 공과대학 컴퓨터공학과(공학석사). 1990년 서울대학교 공과대학 컴퓨터공학과(공학박사) 1994년 미국 IBM Almaden Research Center 초빙 연구원. 1995년~1996년 미국 Syracuse University 초빙 교수. 2001년~현재 전자상거래 표준화 통합 포럼 전자거래 기반 기술위원회 위원장 2003년~현재 한국전자거래학회 편집이사. 2003년~현재 웹코리아 포럼 부위원장. 현재 충남대학교 공과대학 컴퓨터공학과 교수. 관심분야는 데이터베이스, XML, 웹서비스, 시맨틱 웹서비스, 유비쿼터스 컴퓨팅



송 미 영

1984년 2월 숙명여자대학교 화학과(이학사). 1987년 2월 한국과학기술원 화학과(이학석사). 1991년 2월 한국과학기술원 화학과(이학박사). 2004년 10월~현재 한국한의학연구원 정보연구센터 책임연구원. 관심분야는 한의학 콘텐츠, 바이오정보 표준