

# 시간 웹 온톨로지 언어를 이용한 뉴스 동향 분석 서비스

## Trend Analysis Service using a Temporal Web Ontology Language in News Domains

김상균(Sang-Kyun Kim)\*, 이규철(Kyu-Chul Lee)\*\*

### 초 록

본 연구에서는 시맨틱 웹 기술을 이용해 뉴스의 동향을 분석하는 서비스를 제안한다. 뉴스 동향 분석 서비스는 뉴스들간의 시간 흐름 또는 관계들을 분석해 줄 수 있기 때문에 현재의 일반적인 뉴스 검색 엔진보다 지능적인 검색 결과를 보여줄 수 있다. 이러한 서비스를 제공하기 위해서는 시간 정보에 대한 추론 기능이 필요하지만 OWL과 같은 시맨틱 웹 언어는 이를 지원하지 못하기 때문에 시간 기반 추론을 지원하기 위해 본 연구에서는 OWL을 확장한 TL-OWL(TemporaL Web Ontology Language)을 제안한다.

### ABSTRACT

In this paper we investigate a trend analysis service using Semantic Web technology in a news domain. The trend analysis service can provide more intelligent answers rather than the answer given in current news search engines since it can analyze the passage of time and the relation among news. In order to provide the trend analysis service, the capability of temporal reasoning is required, but the Semantic Web language such as OWL does not support the reasoning capability. Therefore, we propose a language TL-OWL(TemporaL Web Ontology Language) extending OWL with the temporal reasoning.

키워드 : 시간 기술 로직, TL-OWL, 동향 분석, 시맨틱 웹  
Temporal Description Logic, TL-OWL, Trend Analysis, Semantic Web

---

\* 충남대학교 공과대학 컴퓨터공학과 박사과정

\*\* 충남대학교 공과대학 컴퓨터공학과 교수

## 1. 서론

시맨틱 웹은 웹 상에 존재하는 정보에 시맨틱을 부여하여 사람뿐만 아니라 기계가 이해하고 처리할 수 있도록 함으로써 자동화된 서비스가 가능하도록 한다. 이러한 시맨틱 웹을 실현하기 위해서 지금까지 많은 시맨틱 웹 관련 기술들이 연구되어 왔으며 다양한 분야에서 시맨틱 웹 기술을 실생활에 적용하기 위한 플랫폼 및 사례 연구들이 진행되어 왔다. 본 연구에서는 이러한 시맨틱 웹 연구들 중의 하나로써 새로운 뉴스 검색 서비스를 제안한다.

일반적인 뉴스 검색 엔진에서는 뉴스를 검색하면 검색의 결과로써 뉴스의 헤드라인이 보여진다. 하지만 이 질의는 뉴스의 메타데이터나 원문 안의 키워드를 기반으로 수행되기 때문에 단편적인 정보밖에 검색하지 못하며 뉴스의 시간 흐름이나 관계들을 제공하지 못한다. 이에 본 연구에서는 시맨틱 기술을 사용하여 보다 지능적인 뉴스 서비스를 제공할 수 있도록 뉴스에 대한 동향 분석 서비스를 구축한다.

예를 들어 “이라크 전쟁이 어떻게 진행되어 왔는가” 라는 질문에 대해서, 일반 뉴스 검색 엔진은 “이라크 전쟁” 또는 “진행”와 매칭되는 키워드를 가지는 뉴스들을 보여준다. 하지만 동향 분석 서비스는 이라크 전쟁이 언제 시작되고 끝났으며 전쟁 후 이라크 전쟁이 세계 정세에 어떤 영향을 미쳤는지 등 이라크 전쟁과 관련된 동향을 분석하고 보여줄 수 있다.

이러한 동향 분석 서비스는 시간 정보가

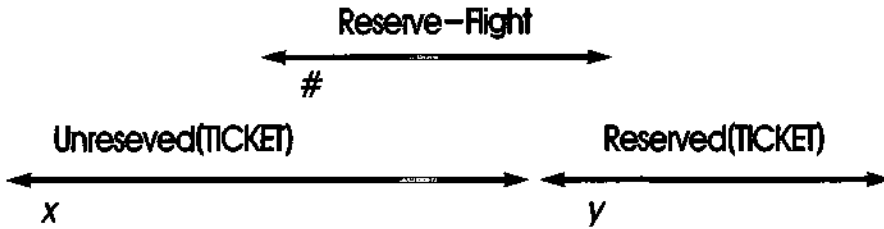
중요하기 때문에 시간 정보를 가지는 온톨로지가 필요하며 또한 시간 기반의 추론 기능을 수행할 수 있어야 한다. 하지만 현재 OWL[14]과 같은 시맨틱 웹 언어는 시간 기반 추론을 지원하지 않기 때문에 본 연구에서는 기존의 OWL에서 시간 추론을 지원할 수 있도록 확장한 TL-OWL을 제안한다.

TL-OWL의 이론적 바탕은 시간 간격 기반 기술 언어(Interval-based Temporal Description Logic)의 한 종류로써 Artale과 Franconi가 제안한 TL-ALCF[1]를 기반으로 한다. 특히 Artale과 Franconi는 TL-ALCF의 포함(subsumption) 문제를 해결하는 것이 결정가능(decidable)하다는 것을 보였는데, 본 연구에서는 TL-OWL에서도 포함 문제가 결정가능하다는 것을 보인다. 이것은 TL-ALCF와 같이 TL-OWL의 포함 문제가 비시간(non-temporal) 부분, 즉, OWL의 포함 문제로 환원될 수 있기 때문이다.

본 연구에서는 Artale의 연구를 참고하여 OWL 기반의 시간 온톨로지 언어를 만들기 위해 TL-OWL의 추상 구문과 시맨틱을 정의하고 이 추상 구문을 RDF 그래프로 매핑하는 방법을 정의한다. 이는 OWL 표준안의 방법과 비슷하며 본 연구에서도 OWL 표준안의 표기법에 따라 정의하였다. 또한 RDF/XML 구문형태로 뉴스 도메인의 온톨로지를 기술하였으며 동향 분석에서 어떻게 이 온톨로지를 사용하는지 보인다.

본 논문의 구성은 다음과 같다. 제 2장에서는 TL-ALCF를 간단히 소개하며, 제 3장에서는 OWL을 확장한 시간 언어인 TL-SHOIN(D)를 제안한다. 또한 제 4장에서는

$$\text{Reserve-Flight} \doteq \diamond(x y) (\# f x)(\# m y). \\ ((\star\text{TICKET} : \text{Unreserved})@x \sqcap (\star\text{TICKET} : \text{Reserved})@y)$$



〈그림 1〉 Reserve-Flight 개념이 가지고 있는 시간 간격들의 시간 의존 관계

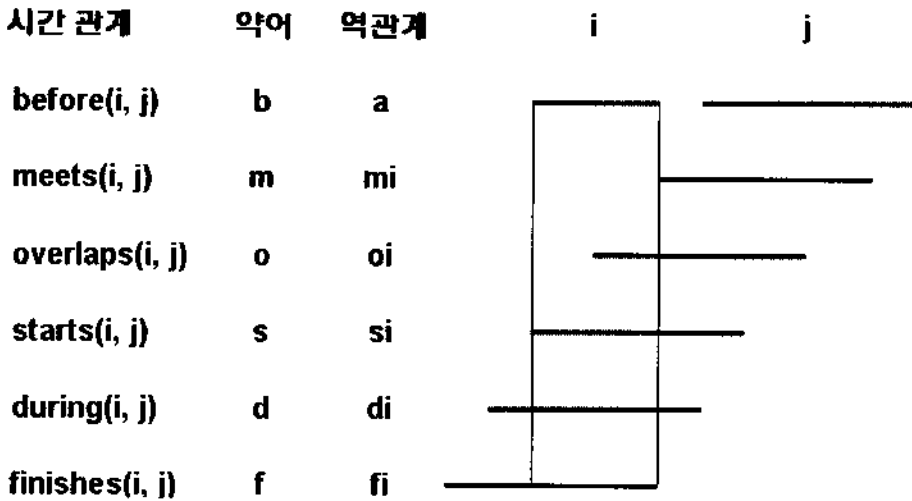
이 언어를 이용한 동향 분석 서비스를 기술하며, 제 5장에서 관련 연구들과 본 연구를 비교하고 마지막으로 제 6장에서 결론을 맺는다.

## 2. 시간 기술 언어 TL-ALCF

이 절에서는 시간 간격 기반 기술 언어의 한 종류인 TL-ALCF를 소개한다. TL-ALCF는 Artale과 Franconi가 제안한 시간 기술 언어로써 TL-ALCF의 포함(subsumption) 문제에 대해 사운드(sound)하고 완전(complete)한 추론 절차를 소개하고 이 절차가 결정 가능(decidable)하다는 것을 보였다. TL-ALCF는 크게 시간을 표현할 수 있는 시간 로직(temporal logic)인 TL과, 기능적 롤(functional role)을 나타내는 특징 로직(feature logic)이 ALC에 추가된 비시간 기술 로직인 ALCF[3]로 구성된다. 이 연구에서는 행동(action)과 계획(plan)을 시간 간격에 따라 일어나는 것들로 정의하는데 여기에서 행동은 해당 도메인에서의 월드 상태(world state)들에 시간

제약(temporal constraint)을 부여한 것으로 표현하며, 각각의 월드 상태는 특정 시간에 가지고 있는 월드 프로퍼티들의 집합이 된다. 그리고 계획은 시간적으로 관련된 행동과 상태들로 표현한다. 이러한 TL-ALCF를 설명하기 위해 다음과 같이 비행기 표를 예약하는 간단한 행동을 보자.

〈그림 1〉은 Reserve-Flight 개념(concept)이 가지고 있는 시간 간격들의 시간 의존 관계를 보인 것으로, Reserve-Flight는 특정 시간 간격 동안 Unreserved상태에 있다가 Reserved상태에 있게 되는 행동을 표현한다. 이 개념에서 Unreserved와 Reserved는 비시간 개념(non-temporal concept)을 나타내고 \*TICKET은 파라미터 특징(parametric feature)을 나타내는데, 특히 \*TICKET은 시간에 상관 없이 Reserve-Flight 타입의 개체(Individual)들을 예약될 비행기 표에 매핑하는 파라미터의 역할을 한다. 시간 변수들은 정의된 행동이 발생하는 현재 시간을 나타내는 특별한 시간 변수인 # (now라고 부른다)를 제외하고는, 시간 존재 정량자(temporal existential



〈그림 2〉 Allen의 시간 관계

quantifier)인 “◇”에 함께 표현된다. (# f x) (# m y)은 시간 제약(temporal constraint)로써 시간 간격 #가 시간 간격 x와 같이 끝나고, 시간 간격 #는 시간 간격 y와 만난다는 것을 나타내는데, f와 m은 〈그림 2〉에서 보이는 바와 같이 Allen의 시간 관계 중의 하나이다.

이 외에도 어떤 개념이 특정 시간 간격에 일어난다는 것을 나타내기 위해서 bindable 이라고 부르는 @연산자를 이용하는데, 예를 들어 위 예제의 (\*TICKET: Unreserved)@x와 (\*TICKET: Reserved)@y에서는 \*TICKET: Unreserved가 시간 x에 한정되고 \*TICKET: Reserved가 시간 y에 한정된다는 것을 나타낸다. 또한 개념을 기술할 때 “:” (특정 선택) 연산자를 사용하는데 이 연산자는 존재 정량자(existential quantifier) ∃와 유사하지만 부분 함수(partial function)로 해석되는 의미를 가진다. 다음 테이블은 TL-

ALCF의 구문을 보인 것이다. 본 논문에서는 이에 대한 시맨틱을 설명하지는 않으며 자세한 내용은 [1]를 참고한다.

### 3. 시간 웹 온톨로지 언어

OWL은 최근 W3C 웹 온톨로지 워킹 그룹에서 만든 시맨틱 웹을 위한 온톨로지 언어이다. OWL은 RDF를 확장하여 정의되었기 때문에 OWL의 구문은 RDF 구문이며 OWL의 시맨틱은 RDF 시맨틱을 확장한 것이다. 또한 OWL의 이론적 기반은 기술 로직(Description Logics)을 바탕으로 하고 있어 기술 로직의 연구 결과들을 사용할 수 있다. 하지만 OWL에서는 기존의 인공지능 분야에서 많이 연구되어 왔던 시간 로직을 포함하고 있지 않기 때문에 시간에 기반한 추론을 수행할 수 없다. 따라서 본 연구에서는

〈표 1〉 TL-ALCF 의 구문

TL	$E, F \rightarrow C$	(비시간 개념)
	$E \cap F$	(연언)
	$E @ X$	(한정자)
	$E[Y] @ X$	(대체 한정자)
	$\diamond(\bar{X}) \bar{T} E . E$	(존재 정량자)
	$T_c \rightarrow (X (V) Y)   (X (V) \#)   (\# (V) Y)$	(시간 제약조건)
	$\bar{T} c \rightarrow T_c   T_c \bar{T} c$	
	$V, W \rightarrow V . W$	(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)
	$\bar{X} \rightarrow X   X \bar{X}$	
ALCF	$C, B \rightarrow A$	(원자 개념)
	$T   \perp   \neg C   C \cap B   C \sqcup B   \forall R.C   \exists R.C$	
	$p \downarrow q$	(동의)
	$p \uparrow q$	(비동의)
	$p \uparrow \uparrow$	(미정의)
	$p : C$	(선택)
	$p, q \rightarrow f$	(원자 특징)
	$*g$	(원자 파라미터 특징)
	$p \circ q$	(패스)

OWL DL에 시간 언어인 TL을 추가한 TL-OWL을 제안한다. TL은 제 2장에서 언급한 TL-ALCF의 시간 로직 부분이며 OWL DL은 기술 로직의 TL-SHOIN(D)와 유사한 언어이다.

### 3.1 TL-SHOIN(D)의 구문과 시맨틱

OWL은 RDF를 확장한 언어이기 때문에 RDF 트리플로 처리되지만 이는 사람이 읽기에는 너무 어렵다. 이 때문에 OWL 표준안에서는 OIL과 같은 프레임(Frame) 언어에서 사용하는 스타일의 구문을 이용하는데 프레임 스타일의 구문에서는 클래스와 프로

퍼티의 정보를 하나의 커다란 구문으로 표현한다. 하지만 일반적으로 기술 로직과 RDF에서는 원자 또는 트리플 단위로 조개어 표현하기 때문에 한 눈에 클래스와 프로퍼티의 정의를 보기가 어렵다. 이와 같은 이유로 OWL 표준안에서는 프레임 스타일의 추상 구문(abstract syntax)을 정의하고 이 구문을 RDF 그래프로 매핑하는 방법을 제공하고 있다. 특히 프레임 스타일의 추상 구문을 EBNF(Extended BNF)를 사용하여 기술하였다. 따라서 이 절에서는 TL-OWL언어를 기술하기 위해 OWL 표준안에서와 같이 TL-SHOIN(D)에 대한 추상 구문을 정의하고 이 구문을 RDF 그래프로 매핑하는 방

〈표 2〉 TL-SHOIN(D)의 구성자

추상 구분	DL 구분	DL 시맨틱
NOW	$\mathbb{I}$	$\mu^x$ is a current interval
before( $t_1, t_2$ )	b	$(b)^{\mathcal{E}} = \{([u_1, v_1], [u_2, v_2]) \in \mathcal{T}_{<}^x \times \mathcal{T}_{<}^x \mid u_1 < u_2\}$
after( $t_1, t_2$ )	a	$(a)^{\mathcal{E}} = \{([u_1, v_1], [u_2, v_2]) \in \mathcal{T}_{<}^x \times \mathcal{T}_{<}^x \mid u_1 > u_2\}$
meets( $t_1, t_2$ )	m	$(m)^{\mathcal{E}} = \{([u_1, v_1], [u_2, v_2]) \in \mathcal{T}_{<}^x \times \mathcal{T}_{<}^x \mid v_1 = u_2\}$
metBy( $t_1, t_2$ )	mi	$(mi)^{\mathcal{E}} = \{([u_1, v_1], [u_2, v_2]) \in \mathcal{T}_{<}^x \times \mathcal{T}_{<}^x \mid u_1 = v_2\}$
overlaps( $t_1, t_2$ )	o	$(o)^{\mathcal{E}} = \{([u_1, v_1], [u_2, v_2]) \in \mathcal{T}_{<}^x \times \mathcal{T}_{<}^x \mid u_1 < u_2 \wedge v_1 < v_2 \wedge u_1 > u_2\}$
overlappedBy( $t_1, t_2$ )	oi	$(oi)^{\mathcal{E}} = \{([u_1, v_1], [u_2, v_2]) \in \mathcal{T}_{<}^x \times \mathcal{T}_{<}^x \mid u_1 > u_2 \wedge v_1 > v_2 \wedge u_1 < u_2\}$
starts( $t_1, t_2$ )	s	$(s)^{\mathcal{E}} = \{([u_1, v_1], [u_2, v_2]) \in \mathcal{T}_{<}^x \times \mathcal{T}_{<}^x \mid u_1 = u_2 \wedge v_1 < v_2\}$
startedBy( $t_1, t_2$ )	si	$(si)^{\mathcal{E}} = \{([u_1, v_1], [u_2, v_2]) \in \mathcal{T}_{<}^x \times \mathcal{T}_{<}^x \mid u_1 = u_2 \wedge v_1 > v_2\}$
during( $t_1, t_2$ )	d	$(d)^{\mathcal{E}} = \{([u_1, v_1], [u_2, v_2]) \in \mathcal{T}_{<}^x \times \mathcal{T}_{<}^x \mid u_1 > u_2 \wedge v_1 < v_2\}$
contains( $t_1, t_2$ )	di	$(di)^{\mathcal{E}} = \{([u_1, v_1], [u_2, v_2]) \in \mathcal{T}_{<}^x \times \mathcal{T}_{<}^x \mid u_1 < u_2 \wedge v_1 > v_2\}$
finish( $t_1, t_2$ )	f	$(f)^{\mathcal{E}} = \{([u_1, v_1], [u_2, v_2]) \in \mathcal{T}_{<}^x \times \mathcal{T}_{<}^x \mid u_1 > u_2 \wedge v_1 = v_2\}$
finishedBy( $t_1, t_2$ )	fi	$(fi)^{\mathcal{E}} = \{([u_1, v_1], [u_2, v_2]) \in \mathcal{T}_{<}^x \times \mathcal{T}_{<}^x \mid u_1 < u_2 \wedge v_1 = v_2\}$
equal( $t_1, t_2$ )	=	$(=)^{\mathcal{E}} = \{([u_1, v_1], [u_2, v_2]) \in \mathcal{T}_{<}^x \times \mathcal{T}_{<}^x \mid u_1 = u_2 \wedge v_1 = v_2\}$
qualification( $X$ bindVariable( $C$ ))	$\text{Qual } X$	$(C \text{ @ } X)_{V, I, N}^x = C_{V, V, X, I, N}^x$
qualification( $X$ Y bindSubstitutiveVariable( $C$ ))	$C[Y]@X$	$(C[Y]@X)_{V, I, N}^x = C_{V, I, N, U(Y \mapsto v(X))}^x$
restriction( $p$ selectValuesFrom( $E$ ))	$p : E$	$(p : E)_{V, I, N}^x = \{a \in \text{dom } p_{V, I, N}^x \mid p_{V, I, N}^x(a) \in E_{V, I, N}^x\}$
agreementOf( $p$ q)	$p \downarrow q$	$(p \downarrow q)_{V, I, N}^x = \{a \in \text{dom } p_{V, I, N}^x \cap \text{dom } q_{V, I, N}^x \mid p_{V, I, N}^x(a) = q_{V, I, N}^x(a)\}$
disagreementOf( $p$ q)	$p \uparrow q$	$(p \uparrow q)_{V, I, N}^x = \{a \in \text{dom } p_{V, I, N}^x \cap \text{dom } q_{V, I, N}^x \mid p_{V, I, N}^x(a) \neq q_{V, I, N}^x(a)\}$
undefinednessOf( $p$ )	$p \uparrow$	$(p \uparrow)_{V, I, N}^x = \Delta \setminus \text{dom } p_{V, I, N}^x$

법을 기술한다.

TL-SHOIN(D)의 시간 로직 부분(TL)과 비시간 로직 부분 SHOIN(D)의 구문과 시맨틱에 대한 정형화된 명세(formal specification)는 [1]와 [6]에서 이미 정의되어 있으며 특히 SHOIN(D)의 추상 구문에서 사용되는 구성자(constructor)와 공리(axiom)는 [5]에 정의되어 있다. 본 논문에서는 기존 연구와 비슷한 방법으로 우선 시간 로직 부분을 포함한 TL-SHOIN(D)의 구성자와 공리를 정의한다. 또한 OWL 표준안에서와 같이 TL-SHOIN(D)의 구성자와 공리를 사용해 추상 구문을 EBNF 형태로 기술하였으며 이를 부록 A에 소개하였다. 또한 이 추상 구문을 RDF 그래프로 매핑한 것은 부록 B에서 기술하였다.

TL-SHOIN(D)를 기술하기 위해 사용되는 구성자는 〈표 2〉와 같다. SHOIN(D)에 대한 구성자는 이미 [5]에서 정의되어 있기 때문

에 〈표 2〉에서는 SHOIN(D)의 구성자를 제외한 것만 기술하였다. 첫 번째 절림은 구성자를 프레임 기반의 추상 구문으로 표현한 것이며, 두 번째와 세 번째 절림은 구성자에 대한 기술 로직의 구문과 시맨틱을 표현한 것이다. 위 테이블에서 보인 구성자에 대한 의미는 bindSubstitutiveVariable을 제외하고 2절에서 이미 소개하였는데, bindSubstitutiveVariable은 시간 대체 한정자로서 [1]에서 정의한 것과 같이 변수 Y를 변수 X로 바꾸기 위해 사용하며 이 두 시간 변수간에 상호 참조를 만들어 준다. 이외에도 C는 시간 개념, E는 비시간 개념, p와 q는 퀘스, X, Y는 시간 변수, before, ..., equal은 Allen의 시간 관계이다. 기술 로직 구문과 시맨틱 부분에서  $\mathcal{I}$ 는  $\mathcal{I} = \langle \Delta, \mathcal{I}_{V, I, N} \rangle$ 와 같은 시간 해석(temporal interpretation)을 나타내는데, 여기에서 V는 시간 변수를 시간 간격값과 연관시키는 변수 할당 함수이며,  $\mathcal{I}$ 는 시간 간격,

추상 구문	DL 구문	DL 시맨틱
TemporalClass(A)	$A = \diamond(x_1 \dots x_k)$	$A^{\mathcal{I}} = \diamond(x_1 \dots x_k)$
Qualification(TemporalVariable( $x_1 V_1(x_1), \dots, V_m(x_1)$ ))	$(x_1 V_1, \dots, V_m x_1) \dots$	$(x_1 V_1^{\mathcal{I}} \cup \dots \cup V_m^{\mathcal{I}} x_1) \dots$
bindVariable( $C_1$ ) $\sqcap \dots \sqcap$	$(x_k V_1, \dots, V_n x_j)$	$(x_k V_1^{\mathcal{I}} \cup \dots \cup V_n^{\mathcal{I}} x_j)$
Qualification(TemporalVariable( $x_k V_1(x_j), \dots, V_n(x_j)$ ))	$(C_1 @ x_1 \sqcap \dots \sqcap C_k @ x_k)$	$(C_1 @ x_1 \sqcap \dots \sqcap C_k @ x_k)^{\mathcal{I}}$
bindVariable( $C_k$ ))		
DatatypeProperty(T ... [ParametricFunctional] ...)	$T \sqsubseteq \mathbb{1} * T$	$T^{\mathcal{I}}$ is parametric functional
ObjectProperty(R ... [ParametricFunctional] ...)	$T \sqsubseteq \mathbb{1} * R$	$R^{\mathcal{I}}$ is parametric functional
[InverseParametricFunctional] ...	$T \sqsubseteq \mathbb{1} * R^{-}$	$(R^{\mathcal{I}})^{-}$ is parametric functional
pathOf(pq) ...)	$p \circ q$	$(p \circ q)^{\mathcal{I}} = p^{\mathcal{I}} \circ q^{\mathcal{I}}$

〈표 3〉 TL-SHOIN(D)의 공리

H는 내부 변수를 할당하는 조건들의 집합으로써 대체 한정자(substitutive variable)를 사용할 때 외부 변수와의 연결을 나타내기 위해 사용된다. 또한 [u,v]는 시간 u와 v사이의 시간 간격을 나타내며 T는 T=(P, <)와 같은 시간 구조(temporal structure)를 나타낸다. 여기에서 P는 시간 포인트(time point)의 집합, <는 P에 대한 부분 순서(partial order)이다. 그리고 T'는 이 시간 구조 T에 대한 시간 간격 집합을 나타내며 시간 해석 함수 S는 시간 구조 T에 대한 시간 해석을 제공한다. 위 테이블에서 마지막 네 개의 구성자는 [3]에서 소개된 특징들에 대한 것으로서 원자 특징은 부분 함수로 해석되며 이들간의 연결 또는 부정을 표현하기 위해 사용된다.

〈표 3〉은 TL-SHOIN(D)의 공리를 보인 것으로 하나의 클래스 정의와 두 개의 프로퍼티 정의 정의를 나타낸다. TemporalClass는 rdfs:Class를 확장하여 정의한 것으로 제 2장에서 소개한 예제와 같이 시간 개념을 정의하기 위해 사용하는 이 클래스에 대한 의

미는 다음 절에서 더 자세히 기술한다. TemporalClass의 추상 구문에서 하나의 temporalConstraint는 하나의 시간 제약을 표현하는데, A는 클래스 이름, U는 Allen의 시간 관계 중 하나의 관계, x는 (1 <= i, j <= k)와 같은 시간 변수를 나타낸다. 두 프로퍼티 공리는 파라미터 특징에 대한 것으로서 OWL의 객체 프로퍼티와 데이터타입 프로퍼티 공리에 대한 정의의 일부로 기술되며 특히 이는 OWL의 함수(Functional) 프로퍼티에 대한 기술과 비슷하다. 또한 여기에서는 객체 프로퍼티와 데이터타입 프로퍼티가 가지고 있는 도메인(domain), 범위(range) 등은 이미 [5]에 기술되어 있기 때문에 생략한다.

우선 본 연구에서는 다음과 같이 두 가지 OWL 확장을 고려한다. 첫째, OWL에서 owl:Class는 위에서 언급한 완전 존재 형태를 가지지 않기 때문에 rdfs:Class를 완전 존재 형태를 가지도록 확장한 TemporalClass라는 클래스를 새로 정의한다. 이 클래스는 비

시간 부분인 OWL의 owl:Class와 구별됨으로써 시간 기반 추론이 오직 TemporalClass에서만 수행되도록 하기 위한 것이다. 둘째, OWL의 구문에 <표 1>에서 소개한 파라미터 특징들을 추가한다. 파라미터 특징은 제2장에서 소개한 것과 같이 시간에 영향을 받지 않는 파라미터 정보를 표현하기 위해 필요하며, 또한 이는 부분 함수로 해석되기 때문에 포함 문제의 복잡도에는 영향을 미치지 않는다. 이와 같이 본 연구에서 두 가지 기능을 OWL에 추가한 TL-OWL에 대한 추론 알고리즘은 다음 절에서 설명한다.

### 3.2 TL-SHOIN(D)에서의 추론

Artale과 Franconi는 시간 간격 기반 기술 언어의 하나로써 ALCF를 제안하였다. 또한 이 언어에서 시간 개념간의 포함 관계가 결정 가능하다는 것을 보였는데 이는 개념에 대한 일반화(normalization) 과정을 통해 시간 부분(TL)과 비시간 부분 TL-ALCF에 대한 추론을 분리하여 수행할 수 있도록 함으로써 가능하게 하였다. 이 절에서는 이러한 [1]의 추론 방법을 TL-SHOIN(D)에 따라 수정한 것으로 여기에서 언급하지 않은 내용은 [1]의 것이 우선한다.

시간 개념간의 포함 문제를 계산하기 위해서 우선 시간 개념의 일반화를 수행해야 한다. 일반화 과정이란 시간 개념 안에 있는 시간 노드들간의 내재된 모든 가능한 연관

관계를 명시적으로 찾는 과정이다. 제 2장에서 보였듯이 모든 시간 개념은 구문적으로는 존재 형태(Existential Form)인  $\diamond(\{X\}\{Tc\}(Q_0 \sqcap Q_1 @ X_1 \sqcap \dots \sqcap Q_n @ X_n))$ 로 표현이 되며, 이를 그래프로 표현하면 개념적 시간 제약 네트워크(Conceptual Temporal Constraint Network)인  $(\{X\}, \{Tc\}, \{Q @ X\})$ 로 볼 수 있다. 일반화 과정은 이러한 존재 형태에 닫힘(closure), =-접기(=collapsing), 덮기(covering), 파라미터 삽입(parameter introduction), 단순 형태(simple form)<sup>3</sup>등을 수행하여 개념에 대한 완전 존재 형태(Completed Existential Form)를 구하고 완전 존재 형태에서 시간과 상관없는 노드를 삭제한 후 최종의 일반 형태, 즉 본질 그래프(Essential Graph)를 얻는 과정으로 구성된다.

#### 정의 1 (완전 존재 형태)

존재 형태의 개념에 대한 완전 존재 형태(CEF: Completed Existential Form)는 다음과 같은 과정을 통해 얻어진다.

- 닫힘, = 접기, 덮기: [1]의 정의 7.6에서 정의한 것과 같이 개념 TL-SHOIN(D) Qy에 대해 닫힘, = 접기, 덮기 변환을 수행한다.
- 파라미터 소개: 파라미터 소개는 다음과 같은 두 과정을 통해 수행한다.
  1. 각 개념 Q를 선언 일반 형태(disjunctive normal form)으로 변환한다. 이를 위해

1 Q는 비시간 개념, {X}는 시간 변수들의 집합, {Tc}는 시간 제약 조건의 집합이다.

2 시간 제약 네트워크는 레이블된 방향성 그래프(labeled directed graph)로써 아크는 임의의 시간 관계들의 집합이며 노드는 비시간 개념들을 가진다.

3 Artale과 Franconi는 일차 로직의 부정 일반 형태(Negation Normal Form)를 단순 형태라고 명명하였다.



$$\begin{array}{ll} \neg \leq nR. \rightarrow \geq (n+1)R & \neg \exists T.d \rightarrow \forall T.\neg d \\ \neg \geq (n+1)R. \rightarrow \leq nR & \neg \forall T.d \rightarrow \exists T.\neg d \end{array}$$

〈그림 3〉 TL-ALCF의 것을 제외한 TL-SHOIN(D)의 단순 형태

우선 개념 Q를 [1]의 〈그림 14〉과 〈그림 3〉에서 보인 규칙에 따라 단순 형태로 변환한다. 다음에 다음과 같은 규칙을 사용해서 선언 일반 형태를 만든다. 이 규칙들은 일차로직의 선언 일반 형태를 만드는 방법과 유사하다.

$$(C_1 \sqcap C_2) \sqcap D \rightarrow (C_1 \sqcap D) \sqcup (C_2 \sqcap D)$$

$$p : (C \sqcup D) \rightarrow p : C \sqcup p : D$$

2 각  $Q_i = E_{i1} \sqcup \dots \sqcup E_{in}$ 에 대해서 시간에 의존적이지 않는 부분  $Q_i'$ 을 계산한다. 이는  $Q_i$ 의 각  $E_{ij}$ 에 대한 시간 비의존적인 부분  $E_{ij}'$ 을 계산함으로써 얻을 수 있는데, 만약  $E_{ij} \sqsubseteq_{SHOIN} \perp$ 이면  $E_{ij} = \perp$  아니면 [1]의 〈그림 15〉에서 보인 규칙을 각  $E_{ij}$ 를 적용하여 얻는다. 다음에 만약  $E_{ij}' = \top$ 인 것이 없다면 모든 다른 노드에  $Q_i' = E_{i1}' \sqcup \dots \sqcup E_{in}'$ 를 연언(conjunction)적으로 추가한다.

정의 1에서 알 수 있듯이 완전 존재 형태를 구하는 과정은 파라미터 특징과 시간 제약 네트워크, 즉 시간 부분과 관련되어 있기 때문에 TL-ALCF에서 TL-SHOIN(D)로 확장할 때 단지 단순 형태(simple form)를 정의하는 부분만 수정하면 TL-SHOIN(D)에 대한 완전 존재 형태를 얻을 수 있다. 다음은 이와 같이 TL-ALCF에서 TL-SHOIN(D)로 확장할 때 추가되는 단순 형

태들이다. 이 단순 형태는 [6]에서의 단순 형태와 유사하며 TL-ALCF의 단순 형태는 [1]의 〈그림 14〉에 정의되어 있다.

### 명제 1(CEF의 동치성)

모든 TL-SHOIN(D)개념의 존재 형태는 동일한 완전 존재 형태로 환원될 수 있다.

증명 : TL-ALCF의 경우와 같이 덧기와 파라미터 소개는 =-접기를 수행한 후에 각각 독립적으로 계산될 수 있으며 이후에 각 결과들을 합쳐 완전 존재 형태를 얻을 수 있다. 이러한 일반화 과정에서 시간 관계의 이행적 닫힘과 덧기 과정을 제외한 =-접기와 파라미터 소개는 선형적 시간(linear time)을 필요로 한다[1]. 특히 TL-SHOIN(D)개념에 대한 일반화 과정에서 TL-ALCF의 과정에 추가되는 부분은 단순 형태뿐인데 이 또한 선형적 시간에 쉽게 구할 수 있다. 시간 관계의 이행적 닫힘(transitive closure)과 덧기 과정은 NP-complete한 문제이지만 [11, 1], [1]에서 소개한 것과 같이 이 문제들은 다항식 시간에 해결 가능한 부분집합을 만드는 방법이 [8]에서 제안되었다.

### 명제 2(CEF의 노드 독립성)

({X}, {Tc}, {Q@X})를 완전 존재 형태의

개념적 시간 제약 네트워크라고 하자. 그러면 모든  $Q \in (Q)$ 와 모든 SHOIN(D)개념 C ( $C \notin Q$ )에 대해서, 어떤 시간 간격 t 에 대해  $(\{X\}, \{Tc\}, \{Q @ X\})_t \neq (\{X\}, \{Tc\}, \{Q @ X\})_t$ 가 성립하는 해석 I가 존재한다.

증명 : 이 명제는 CBF의 각 노드에 있는 정보가 다른 노드에 있는 정보와 독립적이라는 것을 의미한다. 실제 만약 어떤 노드의 개념 표현이 어떤 다른 노드에서 새로운 정보를 의미한다면  $(\{X\}, \{Tc\}, \{Q \cap C\} @ X)_t \neq (\{X\}, \{Tc\}, \{Q @ X\})_t$ 가 성립이 된다. 이 명제에 대한 증명은 [1]의 명제 6.7과 동일하기 때문에 자세한 내용은 생략한다. 증명의 내용은 다음 두 가지로 구성된다.

1. 덜힌 노드: 덜힘을 통해 모든 시간 관계가 명시적으로 표현될 수 있으며 추가적으로 =-접기와 덜기를 통해 덜힌 노드가 가장 구체적인 개념 표현을 가지도록 한다.
2. 동시에 일어나는 노드가 없음: 모든 시간과 상관없는 정보가 다른 모든 노드에 포함되도록 하기 때문에 새로운 정보가 생기지 않으며 따라서 동시에 일어나는 노드가 없다.

위에서 언급했듯이 개념에 대한 완전 존재 형태가 구해지면 일반화 과정의 마지막 과정으로써, 완전 존재 형태에서 시간과 상관없는 노드를 삭제하여 최종 일반 형태,

즉 본질 그래프(Essential Graph)를 얻는다.

### 정의 2(본질 그래프)

개념적 시간 제약 네트워크  $T = (\{X_1\}, \{Tc_1\}, \{Q @ X_1\})$ 의 완전 존재 형태에서 시간에 의존적이지 않은 개념 표현만 가지고 있는 노드를 삭제한 (#노드를 제외) 그래프를 본질 그래프라고 한다.

### 명제 3(본질 그래프의 동치성)

완전 존재 형태의 모든 개념은 동일한 본질 그래프 형태로 선형 시간 내에 환원될 수 있다.

이와 같이 시간 개념에 대한 일반화 과정을 거치면 개념들간의 포함 문제를 계산할 수 있다. 우선 비시간 부분을 제외한 시간 제약 네트워크의 포함 문제부터 보자. 두 개의 시간 제약 네트워크  $TCN_1 = (\{X_1\}, \{Tc_1\})$ 와 시간 제약 네트워크  $TCN_2 = (\{X_2\}, \{Tc_2\})$ 에서 만약  $M : \{X_1\} \mapsto \{X_2\}$ 와 같은 변수 매핑이 존재하고 시간 제약  $Tc_1$ 이  $Tc_2$ 를 포함하면  $TCN_1$ 이  $TCN_2$ 를 포함한다고 한다. 또한  $S : \{X_2\} \mapsto \{X_1\}$ 와 같은 변수 매핑에 대해서 시간 제약 네트워크  $TCN_1$ 이 시간 제약 네트워크  $TCN_2$ 를 포함하고, 즉  $(\{X_1\}, \{Tc_1\}) \supseteq_S (\{X_2\}, \{Tc_2\})$ 이고  $\{X_1\}$ 에서 각 노드의 비시간 개념이  $S(\{X_1\})$ 에 해당하는 노드의 비시간 개념을 포함한다면 개념적 시간 제약 네트워크  $CTCN_1 = (\{X_1\}, \{Tc_1\}, \{Q @ X_1\})$ 에서 개념적 시간 제약 네트워크  $CTCN_2 = (\{X_2\}, \{Tc_2\}, \{Q @ X_2\})$ 로의

s-매핑(s-mapping)이 존재한다고 한다. 또한 모든 존재 형태의 개념은 일반화 과정을 통해서 동일한 의미의 본질 그래프로 변환될 수 있다고 했으므로, 개념  $C_1$ 의 본질 그래프에서  $C_2$ 의 본질 그래프간의 s-매핑이 존재하면 개념  $C_1$ 은 개념  $C_2$ 를 포함한다고 정의할 수 있다.

이와 같은 추론 방법에서 알 수 있듯이 TL-SHOIN(D)에서 시간 개념간의 포함 문제는 일반화 과정을 통해 만들어진 개념적 시간 제약 네트워크에 해당하는 본질 그래프에서 s-매핑이 존재하는지의 문제로 볼 수 있으며, s-매핑이 존재하는지의 문제는 만약 시간 제약 네트워크간의 포함 문제가 만족 가능(satisfiable)하다면 비시간 개념(TL-SHOIN(D))들간의 포함 문제로 환원될 수 있음을 알 수 있다.

#### 명제 4(개념 만족 가능성)

완전 존재 형태의 TL-SHOIN(D)개념, 즉  $C = (\{X\}, \{T\}, \{Q@X\})$ 에서  $\{X\}$ 의 각 노드에 있는 비시간 개념이 만족 가능한 것은 (시간 제약이 만족 가능한 경우) TL-SHOIN(D)개념 C가 만족 가능하다는 것에 필요충분 조건이 된다.

증명 : 이 명제에 대한 증명은 [1]의 명제 7.8에서와 같은 방법을 이용한다. 즉, 명제 2에 따라 완전 존재 형태의 TL-SHOIN(D)개념의 노드 독립성이 보장되는 경우 비시간 개념이 만족 가능하면 C가 만족 가능하며, 또한 반대의 경우에도 마찬가지로 C가 만족 가능하면 비시간 개념이 만족 가능하다는 것을 쉽게 알 수 있다.

#### 정리 1

개념  $C_1$ 의 본질 그래프에서 개념  $C_2$ 의 본질 그래프로의 s-매핑이 존재한다는 것은 개념  $C_1$ 은 개념  $C_2$ 를 포함하는 것에 필요충분조건이 된다.

증명 : 정리 1의 증명은 정리 7.11의 것을 따른다.

정리 1은 TL-SHOIN(D) 개념들간에 포함 문제를 계산하기 위한 사운드하고 완전한 알고리즘이 있음을 나타낸다. 다시 말해서 TL-SHOIN(D)의 일반 형태 덕분에 우리는 TL-SHOIN(D)의 포함 문제가 SHOIN(D)개념(즉, OWL DL)들간의 포함 문제로 환원될 수 있음을 알 수 있으며 비시간 개념(SHOIN(D)개념)의 포함 문제가 결정 가능하기 때문에 TL-SHOIN(D)의 포함 문제 또한 결정 가능하다는 것을 알 수 있다.

#### 3.3 TL-SHOIN(D)온톨로지 예제

이 절에서는 TL-SHOIN(D)를 이용해서 시간 기반의 온톨로지를 어떻게 기술하는지를 보인다. 이를 위해 하나의 지식 베이스(Knowledge base)에 다음과 같은 원유 가격 하락에 대한 이벤트를 나타내는 하나의 시간 개념이 있다는 것을 가정한다. 이 이벤트는 이벤트의 파라미터로써 OILPRICE를 가지는 두 개의 하위 이벤트를 가지며, 시간 간격 x에 일어난 이벤트 때문에 시간 간격 y에 원유 가격이 하락하는 상황을 기술한다.

CrudeOilPriceFall 개념은 다음과 같이 RDF/XML 형태의 교환 구문으로 변환될

$$\text{CrudeOilPriceFall} \doteq \diamond(x y) (x \text{ b } \#)(y \text{ b, m } \#)(x \text{ b } y). \\ ((\star \text{OILPRICE} : \text{Cause})@x \sqcap (\star \text{OILPRICE} : (\text{Effect} \sqcap \\ \exists \text{hasCause.Cause} \sqcap \exists \text{hasEvent.Fall}))@y)$$

수 있으며 또한 반대로 XML을 파싱하여 기술 로직 시맨틱을 얻을 수 있다. 이는 <표 2>와 <표 3>의 추상 구문 매핑과 부록 B의 RDF 매핑을 통해 가능하다. 아래의 XML 표현에서 TL-SHOIN(D) 개념에 해당하는 모든 엘리먼트는 "tl" 네임스페이스를 가진다.

```
<tl:TemporalClass rdf:about = "# CrudeOilPriceFall">
  <owl:intersectionOf rdf:parseType = "Collection">
    <tl:Qualification>
      <tl:onVariable rdf:resource = "#x"/>
      <tl:bindVariable>
        <owl:Restriction>
          <owl:onProperty rdf:resource = "#OILPRICE"/>
          <tl:selectValuesFrom rdf:resource = "#Cause"/>
        </owl:Restriction>
      </tl:bindVariable>
    </tl:Qualification>
    <tl:Qualification>
      <tl:onVariable rdf:resource = "#y"/>
      <tl:bindVariable>
        <owl:Restriction>
          <owl:onProperty rdf:resource = "#OILPRICE"/>
          <tl:selectValuesFrom>
            <owl:Class>
              <owl:intersectionOf rdf:parseType = "Collection">
                <owl:Class rdf:about = "#Effect"/>
                <owl:Restriction>
                  <owl:onProperty rdf:resource = "#hasCause"/>
                  <owl:someValuesFrom rdf:resource = "#Cause"/>
                </owl:Restriction>
                <owl:Restriction>
                  <owl:onProperty rdf:resource = "#hasEvent"/>
                  <owl:someValuesFrom rdf:resource = "#Fall"/>
                </owl:Restriction>
              </owl:intersectionOf>
            </owl:Class>
          </tl:selectValuesFrom>
        </owl:Restriction>
      </tl:bindVariable>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
</tl:selectValuesFrom>
```

```

</owl:Restriction>
  </tl:bindVariable>
</tl:Qualification>
</owl:intersectionOf>
</tl:TemporalClass>
<tl:TemporalVariable rdf:about = "#x">
  <tl:before rdf:resource = "#NOW"/>
  <tl:before rdf:resource = "#y"/>
</tl:TemporalVariable>
<tl:TemporalVariable rdf:about = "#y">
  <tl:before rdf:resource = "#NOW"/>
  <tl:meet rdf:resource = "#NOW"/>
</tl:TemporalVariable>

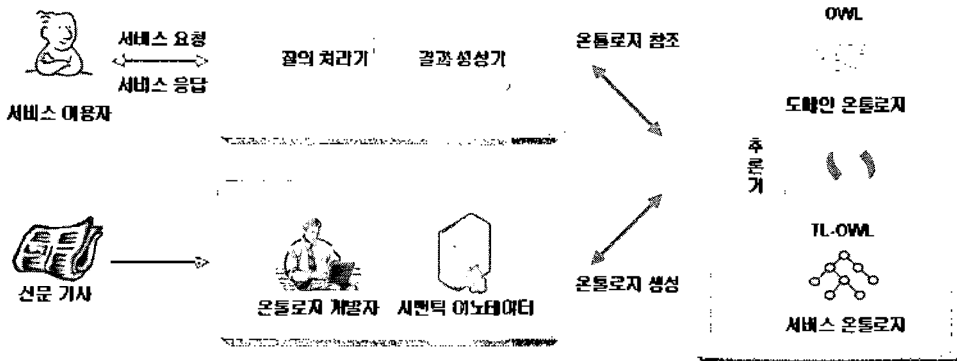
```

<그림 4> CrudeOilPriceFall 개념의 RDF/XML 표현

#### 4. 동향 분석

시간 추론은 인공지능 분야에서 보통 계획 시스템 또는 스케줄링 시스템에서 많이 사용되어 왔지만 본 연구에서는 시간 추론을 이용해 뉴스의 동향을 분석해 줄 수 있

는 동향 분석 서비스를 소개한다. 이러한 동향 분석 서비스는 뉴스의 메타데이터나 원문의 키워드를 기반으로 검색을 수행하고 해당 결과를 보여주는 일반적인 뉴스 검색 엔진과 달리 뉴스간의 시간 흐름이나 관계를 분석하고 추출하는데 목적이 있다. 이러



<그림 5> 동향 분석 시스템 구조

한 기능을 제공하기 위해 본 논문에서는 온톨로지 기반의 추론과 질의를 통해 동향 분석을 수행한다.

동향 분석 서비스를 제공하기 위해서는 우선 뉴스 도메인에 대한 일반적인 개념과 프로퍼티들을 표현할 수 있는 온톨로지를 구축해야 한다. 이 때 이 온톨로지는 본 연구에서 제안한 TL-OWL이 아니라 OWL을 이용하여 도메인 온톨로지를 만든다. 이러한 뉴스 도메인에 대한 온톨로지가 구축이 되면 이 도메인 온톨로지를 기반으로 뉴스에 나타나는 조직, 사람, 이벤트와 같은 주요 엔터티들간의 시간 관계 또는 인과 관계를 분석하고 추출해 새로운 서비스 온톨로지를 만든다. 이 때 엔터티들은 기존에 OWL로 구축된 온톨로지에 대한 참조가 되며 엔터티들간의 관계들은 TL-OWL을 사용해 만들어진다. 이러한 구축 과정은 시맨틱 어노테이터와 같은 자동 처리 엔진을 이용할 수 있으며 또한 사람(온톨로지 개발자)의 리뷰를 통해 반자동 또는 수동으로 구축될 수도 있다. 이 과정을 통해서 <그림 4>에서 소개한 원유 가격 하락에 대한 온톨로지가 구축이 된다.

TL-OWL은 Allen의 시간 관계를 사용하여 뉴스 기사 내 엔터티들간의 시간 흐름을 표현할 수 있을 뿐만 아니라 시간 관계를 기반으로 원인과 결과를 위한 프로퍼티를 정의하여 인과 관계도 표현할 수 있도록 하기 때문에 동향 분석 서비스에서 중요한 역할을 한다. 또한 뉴스 도메인에 대해 구축된 지식 베이스에서 함의성(entailment), 포함관계(subsumption), 만족가능성(satisfiability), 일관성(consistency)과 같은 기술 언어의 추론

기능들을 추론기를 통해 이용할 수 있도록 함으로써 보다 지능적인 서비스가 가능하다. 본 연구에서는 이러한 온톨로지 구축과 시간 추론을 구현하기 위해서 OWL에 대한 추론을 완벽하게 지원하고 있는 Pellet OWL 추론기[12]에 본 연구에서 제안한 내용을 확장 구현하여 OWL뿐만 아니라 TL-OWL을 지원할 수 있도록 하였다.

또한 본 시스템에서는 위와 같이 TL-OWL을 이용해 구축된 온톨로지를 기반으로 뉴스 기사에 대한 동향을 분석해주는 동향 분석 서비스를 구현하였다. 본 연구에서 구현한 시스템은 관계 데이터베이스를 위한 질의 방법으로 많이 사용되어 왔던 QBE(Query by Example) 스타일의 질의 인터페이스를 가진다. 이 인터페이스는 정보 검색이나 다른 시맨틱 엔진에서 사용하고 있는 자연어 질의나 단순 키워드 질의보다는 사용자가 질의를 입력할 때 직관성이 떨어지지만 사용자로부터 보다 정확한 질의를 얻을 수 있도록 정확한 결과를 보여줄 수 있도록 하기 위해서 사용하였다. 질의 인터페이스의 입력 필드는 조직, 사람, 위치, 이벤트, 시간, 조건등을 가진다. 여기에서 처음 5개 필드에는 검색하고자 하는 뉴스 엔터티들을 입력하게 되며 조건 필드에서는 "이전" 또는 "이후"와 같은 시간 조건과 "원인" 또는 "결과"와 같은 인과 조건을 선택하거나 검색결과가 포함해야 하는 키워드를 입력할 수 있다. 예를 들어 사용자가 "어제 두바이 원유 가격이 하락한 이유는?" 라는 질의를 한다고 가정하자. 위 질의 인터페이스에서는 시간에 "어제", 이벤트에 "두바이 원유 가격 하락" 그리고 조건에는 "원인"을 입력

할 수 있다. 이와 같이 사용자가 질의를 하게 되면 질의처리기에서 질의를 SPARQL [15]로 변환하여 온톨로지에 대한 질의를 수행하는데 질의 인터페이스가 QBE 방식을 사용하여 질의 엔터티가 정해져 있기 때문에 쉽게 SPARQL로 변환이 가능하다.

결과생성기에서는 질의가 수행된 후 검색 결과는 사용자에게 동향 검색 결과를 한눈에 쉽게 알아볼 수 있도록 하기 위해서 SIMILE 프로젝트[13]에서 개발한 타임라인 라이브러리를 이용한다. 이 라이브러리는 스크롤 가능한 타임라인 박스를 제공하는데 가로축은 시간을 표현하며 뉴스 이벤트는 박스 안에서 해당 시간 위에 하나의 점으로 표현된다. 이 각각의 점마다 뉴스의 제목이 나오며 제목을 클릭하면 해당 뉴스에 대한 메타데이터와 원문의 일부가 나오게 되어 있다. 특히 위의 두바이 원유 가격 하락에 대한 질의 결과는 두바이 원유가 하락에 대한 원인(예를 들어 원인은 미국의 원유 소비 감소 또는 나이지리아의 원유 생산 재개가 될 수 있다). 뿐만 아니라 유가 하락과 관련된 결과 뉴스도 보여지도록 하였다. 이를 통해 질의의 원인과 결과에 대한 뉴스 동향을 타임라인 상에서 시간대 별로 쉽게 분석이 가능할 수 있도록 하였다.

## 5. 관련 연구

그 동안 기술 로직 분야에서는 시간 관련 개념을 표현하고 추론하기 위한 여러 연구들이 제안되었다[2]. 이러한 연구들은 크게 포인트 기반(point-based) [4,8] 또는 시간 간

격 기반(interval-based) 언어로 나뉘어 지는데, 시간 간격 기반 연구는 본 연구에서 소개한 것과 같이 Allen의 시간 간격 개념을 기반으로 개념을 표현하며, 포인트 기반 연구는 시간 간격이 아니라 U(until)과 S(Since) 연산자를 이용해서 특정 시간 포인트간의 관계(예를 들어 오늘까지 또는 오늘부터)를 표현하는 방법을 사용한다. 이와 같이 두 방법은 시간을 표현하는 방법에서 근본적인 차이가 있으나 두 방법 모두 좋은 계산 성능을 보이고 있다.

하지만 본 연구에서는 시간 간격 기반 방법이 포인트 기반 방법보다 다양한 시간 관계를 제공할 수 있기 때문에 동향 분석 시스템에서는 다양한 관계를 표현할 수 있어서 시간 간격 기반 방법을 채택하였다. 본 연구에서는 기존의 시간 간격 기반 언어인 TL-ALCF를 확장하여 TL-SHOIN(D)를 만들고 이 언어의 포함 문제가 결정 가능하다는 것을 보였듯이 만약 포인트 기반 방법으로 OWL을 지원하는 언어를 만들고 이 언어가 결정 가능하다는 것을 보일 수 있다면 본 연구와 같이 동향 분석에 적용 가능할 것이다.

동향 분석 관련 연구로써는 최근에 NEWS (News Engine Web Services) 플랫폼[10]이 제안되었는데 이 플랫폼은 시맨틱 웹 기반으로 뉴스 콘텐츠를 만들고 제공하는 시스템이다. 특히 NEWS에서는 뉴스 기사들을 자동으로 분류하고 어노테이션할뿐만 아니라 뉴스의 동향을 분석하고 헤드라인을 생성해 제공하는 서비스를 제공한다. 이와 같이 NEWS에서도 본 연구와 같이 뉴스 기사들의 동향 분석 서비스를 제공하지만 이 서비

스는 정보 검색 기술에 기반하고 있으며 시간 기반 추론을 수행하지 않는 점에서 본 연구와 차이가 있다.

## 5. 결 론

OWL과 같은 시맨틱 웹을 위한 온톨로지 언어는 웹 상에 존재하는 정보 자원을 기술할 수 있도록 개발되었다. 특히 OWL은 기술로직의 한 언어인 SHOIN(D)를 기반으로 정형화하였지만 인공지능 분야에서 지금까지 많이 연구되어 왔던 시간로직을 포함하지 않기 때문에 시간 정보에 기반한 추론을 수행할 수 없다.

따라서 본 연구에서는 OWL에서 시간 추론을 수행할 수 있도록 시간 간격 기반 기술로직의 하나인 TL-ALCF를 확장하여 TL-SHOIN(D) 즉, TL-OWL을 제안하였다. 이를 위해 TL-SHOIN(D)에 대한 추상 구문과 시맨틱을 정의하였으며 이 추상구문을 RDF 그래프로 매핑하는 방법을 제공하였다. 특히 TL-SHOIN(D)의 시간 부분과 비시간 부분에 대한 추론은 서로 분리되어 수행 가능하기 때문에 TL-SHOIN(D)의 포함 문제는 SHOIN(D) 개념들간의 포함 문제로 환원이 가능하다. 또한 SHOIN(D)의 포함 문제가 결정가능하기 때문에 TL-SHOIN(D)의 포함 문제 또한 결정 가능하다는 것을 보였다.

본 연구에서는 이러한 TL-OWL 언어를 사용한 시맨틱 웹 어플리케이션의 하나로 뉴스에 대한 동향 분석 서비스를 구현하

였다. 기존 뉴스 검색에서는 단순히 메타 데이터나 본문 내의 키워드를 기반으로 질의를 처리하기 때문에 정확한 검색 결과를 주기에는 한계가 있었다. 하지만 본 연구에서 구현한 동향 분석 서비스는 기존의 뉴스 검색 엔진과 달리 뉴스 간의 시간 흐름과 관계를 분석하여 온톨로지화하고 추론을 통해 암시적인 정보를 추출할 수 있도록 함으로써 보다 지능적인 검색 결과를 보여줄 수 있다.

---

## 참 고 문 헌

---

- [1] A. Artale and E. Franconi. A Temporal Description Logic for Reasoning about Actions and Plans, *Journal of Artificial Intelligence Research*, Vol. 9, pp. 463-506, 1998.
- [2] A. Artale and E. Franconi. A Survey of Temporal Extensions of Description Logics. *Annals of Mathematics and Artificial Intelligence*, Kluwer Academic Press, Vol. 30, No. 1-4, pp. 171-210, 2001.
- [3] B. Hollunder and W. Nutt. Subsumption Algorithms for Concept Languages. Technical Research Report RR-90-04, DFKI, Germany, 1990.
- [4] F. Wolter and M. Zakharyashev. Temporalizing description logics, *Frontiers of Combining Systems*. Studies Press-Wiley (1999) Also in the Proceedings of FroCoS' 98, 1998.



- [5] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, Vol. 1, No. 1, pp. 7-26, 2003.
- [6] Ian Horrocks and Ulrike Sattler. A tableaux decision procedure for SHOIQ. In *Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*, pp. 448-453, 2005.
- [7] J. F. Allen. Temporal Reasoning and Planning. In J. F. Allen, H. A. Kautz, R. N. Pelavin and J. D. Tenenber, editors, *Reasoning about Plans*, Chap. 1, pp. 2-68, Morgan Kaufmann, 1991.
- [8] J. Renz and B. Nebel. On the complexity of qualitative spatial reasoning: a maximal tractable fragment of the region connection calculus. In *Proc. Of the 14th IJCAI*, pp. 522-527, 1997.
- [9] K.D. Schild. Combining terminological logics with tense logic. In *Proceedings of the 6th Portuguese Conference on Artificial Intelligence (EPIA '93)*, 1993.
- [10] NEWS (News Engine Web Services). In <http://www.dfki.uni-kl.de/~bernardi/News/index.html>.
- [11] P. van Beek and R. Cohen. Exact and approximate reasoning about temporal relations. *Computational Intelligence*, Vol. 6, pp. 132-144, 1990.
- [12] Pellet: An OWL DL Reasoner. <http://pellet.owldl.com/>.
- [13] SIMILE project. In <http://simile.mit.edu/timeline>.
- [14] W3C Recommendation. OWL Web Ontology Language Semantics and Abstract Syntax. In <http://www.w3.org/TR/owl-semantics> (2004).
- [15] W3C Working Draft. SPARQL Query Language for RDF. In <http://www.w3.org/TR/rdf-sparql-query> (2006).
- [16] TL-OWL 추상 구문과 RDF 그래프 매핑 <http://dlablab.cnu.ac.kr/~skkim/papers/TL-OWL/Supplementary-One.pdf>.

## 저 자 소 개



김상군

1999.

2001.

2001 ~ 현재

2003 ~ 현재

관심분야

(E-mail : yskkimy@yahoo.co.kr)

충남대학교 공과대학 정보통신공학과 (공학사)

충남대학교 공과대학 컴퓨터공학과 (공학석사)

충남대학교 공과대학 컴퓨터공학과 박사과정

(주)K4M 연구원

사맨텍웹, e-비즈니스



이규철

1984.

1986.

1990.

1994.

1995 ~ 1996.

2001 ~ 현재

2003 ~ 현재

2003 ~ 현재

현 재

관심분야

(E-mail : klee@cnu.ac.kr)

2월 서울대학교 공과대학 컴퓨터공학과 (공학사)

2월 서울대학교 공과대학 컴퓨터공학과 (공학석사)

8월 서울대학교 공과대학 컴퓨터공학과 (공학박사)

미국 IBM Almaden Research Center 초빙 연구원

미국 Syracuse University 초빙 교수

전자상거래 표준화 통합 포럼 전자거래 기반

기술위원회 위원장

한국전자거래학회 편집이사

웹 코리아 포럼 부위원장

충남대학교 공과대학 컴퓨터공학과 교수

XML, 웹 서비스 시맨틱 웹 서비스, 유비쿼터스

웹 서비스