DL 추론과 시간적 추론을 적용한 상황 정보 관리

김제민°, 박영택
숭실대학교 컴퓨터학과
kimjemins@hotmail.com° , park@computing.ssu.ac.kr

Semantic Context Management Using DL Reasoning and Temporal Reasoning

Je-Min Kim° Young-Tack Park
Dept of Computer Science, Soongsil University

요 약

상황 정보 관리 시스템은 외부에서 입력된 상황 정보의 숨겨진 의미를 파악하여 상황인지 에이전트 및 상황인지 브로커가 효과적으로 상황정보를 획득하도록 한다. 본 논문에서는 외부 환경으로부터 받은 상황 정보의 숨겨진 의미를 파악하기 위해 DL 추론과 시간적 추론을 적용한 상황 정보 관리 시스템을 제안한다. 이를 위해서 3가지 부분에 초점을 두었다. 첫 번째, 외부에서 입력된 상황 정보를 효율적으로 표현하고 여러 에이전트간의 상황 정보 공유가 가능하도록 온톨로지 모델을 적용한다. 온톨로지로 표현된 상황 정보는 정보의 속성을 제약함으로써 숨겨진 상황 정보를 추론할 수 있도록 해준다. 두 번째로 상황 정보의 관계를 추론할 수 있도록 서술 논리(Description Logic)를 적용한다. 마지막으로 상황 정보의 시간적 관계를 추론할 수 있도록 시간 논리(Temporal Logic)을 적용한다. 따라서 본 논문에서의 최종 목표는 상황 정보 관리 시스템 연구를 통해 상황인지 에이전트 및 상황인지 브로커에 활용이 가능한 온톨로지 기반 추론 기능을 보유하는 지능형 모듈의 기본 프레임워크를 구축하는 것이다.

1. 서 론

유비쿼터스 컴퓨팅(Ubiquitous Computing)은 수많은 지능형 컴퓨터들이 유기적으로 연결되어 언제 어디서나 사물(Object)을 인식하고 사람들에게 필요한 정보나 서비스를 제공을 목적으로 하며, 유비쿼터스 컴퓨팅 환경은 다양한 형태의 지능형 컴퓨터가 현실 세계와 효과적으로 결합되어 언제 어디서나 컴퓨팅 자원을 활용할 수 있는 환경을 제공한다. 외부에서 입력된 상황 정보를 정형화된 형태로 인지하고, 숨겨진 의미를 파악하여, 상황 인지 에이전트 및 상황인지 브로커에게 올바른 상황 정보를 전달하는 시멘틱 상황 정보 관리 시스템은 유비쿼터스 컴퓨팅에 있어 매우 중요한 역할을 담당한다.

시멘틱 상황 정보 관리 시스템이 효과적인 기능을 발휘하기 위해서는 외부에서 입력된 상황 정보들을 계층별로 정확하게 분류하고, 상황 정보들과 시간의 관계를 파악하는 추론 모듈이 필요하다. 본 논문에서는 이러한 사항을 고려하여 서술 논리 추론(Description Logic Reasoning)과 시간적 추론(Temporal Reasoning)을 적용한 상황 정보 관리 시스템을 제안한다.

적절한 상황 정보를 제공하기 위해서, 시멘틱 상황 정보 관리 시스템은 외부에서 입력되는 상황 정보(Context Information)간의 차이를 조절하고 상황 정보가 속하는 클래스의 계층 관계를 정확하게 분류하며, 실제 발생한 상황 정보의 시간적인 관계를 추론해야 한다. 이러한 기능을 가지는 상황 정보 관리 시스템을 제안하기 위해서 다음 3가지 부분에 중점을 두었다. 첫 번째, 외부에서 입력된 상황 정보를 효율적으로 표현하고 여러 에이전트간의 상황 정보 공유가 가능하도록 온톨로지 모델을 적용한다. 유비쿼터스 컴퓨팅 환경 내에서 상황 정보(Context Information)에 대한 중요성이 점점 증가하고 있는데, 상황(Context)은 사용자와 유비쿼터스 서비스 시스템간의 상호작용을 위해 필요한 사용자 주변의 환경, 객체 또는 사용자의 위치에 관한 정보를 의미한다[1]. 상황 정보는 유비쿼터스 컴퓨팅 환경 내에 존재하는 다양한 센서에 의해 다양한 형태로 획득 된다. 상황 정보 관리 시스템과 상황 정보 에이전트들 간의 일관성 있는 상황 정보의 공유를 위해, 모든 상황 정보들은 일관성 있는 개념(Concept)으로 정의되어야 한다. 온톨로지는 모든 상황 정보 에이전트들이 상황 정보들을 공유할 수 있도록, 상황 정보에 대한 개념을 형식적이고, 명백하게 명시한 스키마를 제공하기 때문에, 에이전트간의 상황 정보 공유 및 관리를 수월하게 하게 해준다. 또한 온톨로지로 표현된 상황 정보는 정보의 속성을 제약함으로써 숨겨진 상황 정보를 추론할 수 있도록 해준다. 두 번째로 상황 정보의 관계를 추론할 수 있도록 서술 논리(Description Logic)를 적용한다. 상황 정보 관리 시스템에 적용된 서술 논리의 추론은 Subsumption 추론이며, 이 과정에서 상황 정보들의 Class/SubClass의 관계를 추론하게 된다. 따라서 Subsumption 추론은 상황 정보가 속하는 클래스의 계층 관계를 정확하게 추론하는 역할을 한다. 마지막으로 상황 정보의 시간적 관계를 추론할 수 있도록 시간 논리(Temporal Logic)을 적용한다.

본 논문에서의 최종 목표는 상황 정보 관리 시스템 연구를 통해 상황인지 에이전트 및 상황인지 브로커에 활용이 가능한 온톨로지 기반 추론 기능을 보유하는 지능형 모듈의 기본 프레임워크를 구축하는 것이다.

2. 시멘틱 상황 정보 관리 시스템 구조

시멘틱 상황 정보 관리 시스템은 외부에서 입력받은 상황 정보가 속하는 클래스의 계층 관계와 상황 정보들의 시간적 관계들을 추론하여, 상황 정보 에이전트가 정확한 상황 정보를 얻을 수 있도록 도와준다. 따라서 상황 정보 에이전트는 필요한 상황 정보를 획득하여, 적당한 서비스나 기능을 실행하는 클라이언트이며, 시멘틱 상황 정보 관리 시스템은 추론되어진 상황 정보를 각각의 클라이언트에게 전송하는 서버 형태의 구조를 갖는다. 시멘틱 상황 정보 관리 시스템은 상황 정보를 온톨로지를 기반의 인스턴스로 표현하여, 서술 논리 추론엔진과 시간적 추론 엔진을 이용하여 숨겨진 상황 정보를 추론하고, 이를 블랙 보드에 저장한다. [그림 1]은 시멘틱 상황 정보 관리 시스템의 전체적인 구조를 보여준다.

본 논문에서 제안하는 상황 정보 관리 시스템은 크게 여섯 파트로 구성된다.

· 온톨로지 - 크게 상황 정보를 표현할 온톨로지, 상황 정보의 시간적인 관계를 표현할 시간 온톨로지, 상황 정보가 사용되는 범위에 따라서 부분적으로 적용되는 규칙을 정의한 규칙 온톨로지로 나뉜다.

· 객체(Instance) 생성기 - 입력 받은 상황 정보를 OWL 기반의 온톨로지 객체로 변환하여 추론 엔진에 전달한다.

· 추상화 엔진(Abstraction Engine) - 서술 논리 추론엔진과 시간적 추론 엔진이 추론하지 못하는 상황 정보의 관계들을 추론한다. 따라서 추상화 엔진에 작용되는 규칙은 시스템을 사용하는 범위에 따라 다를 수 있다.

· 온톨로지 추론 엔진(Ontology Reasoner) - 서술 논리를 기반으로 상황 정보가 속하는 클래스의 계층을 추론한다.

· 시간적 추론 엔진(Temporal Reasoner) - 시간적 논리를 기반으로 상황 정보들의 시간적 관계를 추론한다.

· 블랙 보드 - 외부에서 입력된 상황 정보와 새롭게 추론된 상황 정보를 저장한다.

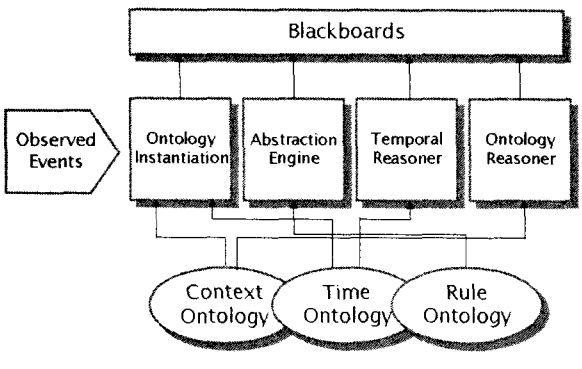

그림 1 시멘틱 상황 정보 관리 시스템의 구조

2.1 온톨로지

본 논문에서 제안하는 상황 정보 관리 시스템은 효율적으로 상황 정보를 추론하고, 공유할 수 있도록 시멘틱 웹 온톨로지를 사용한다. 온톨로지를 적용하면 지능형 공간에 존재하는 사람, 이벤트 및 시간 등의 관계에 대한 모델을 제공하기 때문에, 상황 정보의 일관성 있는 관리와 공유가 가능하다.

온톨로지는 단어와 관계들로 구성된 사전으로서 어느 특정 도메인에 관련된 단어들을 계층적 구조로 표현하고, 이를 확장하여 표현할 수 있는 추론 규칙을 포함한다. 제안하고 있는 시스템에 적용하기 위해 구축된 온톨로지는 웹 온톨로지 언어인 OWL(Web Ontology Language)로 작성되었으며 실제 공간에서 발생하는 여러 상황 정보를 표현하기위해 여러 가지 상황 모델(이벤트, 사람, 시간)을 제공한다. 따라서 이전에 구축되어 공유 가능한 온톨로지(Friend of s friend, DAML Time, spatial ontology)들이 적용되어 있으며, 물리적인 공간, 시간, 사람, 행위에 대한 전형적인 개념과 관계를 정의한다. 본 연구를 위해 구축한 온톨로지는 다음과 같이 구성 되어있다.

· *Person 온톨로지* - Person 온톨로지는 사람의 정보를 상황 정보로 표현하기 위한 클래스 형태로써 이름, 성별, 나이, 식별자의 프로퍼티를 가진다. 따라서 사람이 가지고 있는 여러 가지 속성에 대한 전형적인 어휘들로 구성된다. 이 온톨로지는 FOAF(Friend of s friend) 온톨로지를 기반으로 구축됐다.

· *Temporal 온톨로지* - 실제 공간에 존재하는 모든 개체에 시간 개념을 적용시키기 위한 온톨로지다. Temporal 온톨로지는 Time이라는 클래스로 표현되는데, 이 클래스는 특정 시간과 시간 간격에 대한 정의 및 이들 간의 관계를 표현하는 전형적인 어휘들로 구성된다. 따라서 Temporal 온톨로지의 시간 관계를 이용하여 서로 다른 시간에 발생하는 이벤트의 시간적인 속성을 정의할 수 있다. 이 온톨로지는 DAML Time 온톨로지를 기반으로 구축됐다.

· *Spatial 온톨로지* - Spatial 온톨로지는 실제 공간(회의실, 강의실, 세미나실 등등)내에서 객체의 위치를 일정한 장소에 매치시키기 위한 전형적인 어휘 및 기호들로 구성된다. 시멘틱 정보 관리 시스템에서 장소에 대한 온톨로지는 매우 중요한 요소이다. 사용자의 행동으로 실시간 발생하는 상황 정보들이 위치에 대한 정의가 없다면 상황 인지 에이전트가 적합한 서비스를 수행하는 것은 불가능 하다고 볼 수 있다.

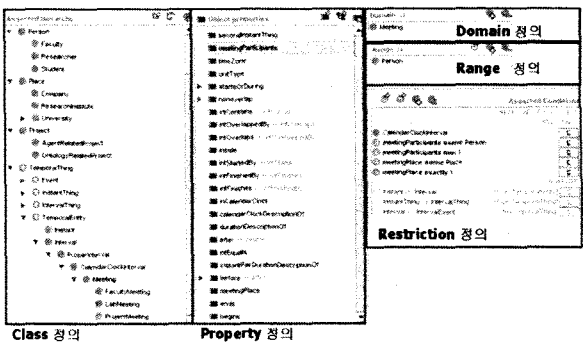

그림 2 정의된 온톨로지 구조

· *event 온톨로지* - 사람이 실행하고 있는 일(미팅, 강의, 세미나 등등)에 대한 상황(Context)을 정의한다. event 온톨로지는 일의 주체가 되는 사람, 일이 일어난 장소, 일이 일어난 시간 등을 속성(property)으로 정의 하고 있으며, 일의 주체가 되는 사람을 정의한 Person 온톨로지, 장소를 정의한 Spatial 온톨로지, 일이 일어난 시간을 정의한 Temporal 온톨로지의 개체(instance)들을 속성 값으로 가지고 있기 때문에 누가, 언제, 어디서, 어떤 일을 하는지 일관성 있게 표현할 수 있다.

다음 [그림 2]는 상황 정보 관리 시스템을 위해 온톨로지 저작 도구인 protege를 사용하여 정의한 온톨로지의 구조이다.

2.2 서술 논리와 서술 논리 추론

본 논문에서 적용하고자 하는 서술 논리(Description Logic)는 기본적으로 Concepts, Roles, Individuals로 사람이 가지고 있는 지식을 표현한다[2]. 따라서 본 논문에서 다루고자 하는 상황 정보를 서술 논리를 표현하기 위해 T-Box와 A-Box의 구조를 활용한다. T-Box에는 상황 정보의 Terminological 부분이 선언되는데, 이것은 온톨로지의 Class와 Property 및 Restriction 정의 부분이다. A-Box에는 실제 상황 정보의 Assertional 부분이 정의되는데, 이것은 온톨로지의 Instance 정의 부분이다. 이처럼 본 논문에 적용된 온톨로지 언어인 OWL과 서술 논리는 그 구조가 일치하기 때문에 쉽게 하나의 시스템으로 연결되어질 수 있다.

상황 정보 관리 시스템에 적용된 서술 논리의 추론은 크게 두 가지로 나뉜다. 첫 번째는 Subsumption Relation을 추론하는 것을 의미하는데, 이 과정에서 상황 정보들의 Class/SubClass의 관계를 추론하게 된다. 여기서 Subsumption은 하나의 Concept이 다른 Concept을 포함하는 것을 의미한다. 예를 들면 "세미나는 2명 이상이 참가하면서 발표자가 있는 이벤트"는 "학술회의는 최소한 발표자가 한명 이상 존재하는 이벤트"를 Subsumption한다. 두 번째는 Insanitation 관계를 추론하는 것으로써, 하나의 상황 정보가 어느 Concept에 속하는지 추론한다.

일반적으로 Subsumption 관계는 $C \sqsubset D$ 와 같은 형식으로 표현되는 데, 이를 증명하기 위해서는 $\neg C \cup D$가 satisfiable[3] 하다는 것을 보이면 된다. 이는 $C \sqsubset D$가 satisfiable 하다는 것을 증명하기 위해서는 $\neg C \cup D$의 negation인 $C \cap \neg D$가 unsatisfiable하다는 것을 보이면 된다는 것을 의미하기도 한다.

Tableaux[4] 알고리즘은 satisfiability 테스트를 실험적으로 Tractable하게 보여줄 수 있음을 증명하였고, 현재 많은 서술 논리 기반의 온톨로지 추론 기술은 Tableaux 알고리즘을 기반으로 하고 있다. 따라서 제안하고자 하는 상황 정보 관리 시스템에는 Tableaux 알고리즘 기반의 서술 논리 추론을 적용하였다.

Tableaux 알고리즘의 기본 아이디어는 증명하고자 하는 내용의 Negation에 대해서 다양한 변환 규칙을 적용하여 satisfiable하지 않다는 것을 보여주는 방식을 취하고 있다. 서술 논리는 $\cap$, $\cup$, $\neg$, $\exists$, $\forall$등으로 표현되기 때문에 Tableaux 알고리즘에서는 이에 대한 Expansion 규칙을 반복적으로 적용하면서 탐색공간을 확장하게 된다. 그리고 모든 탐색 공간의 말단 노드가 모순(Contradiction)이라는 것을 보여줌으로써 증명하고자 하는 Subsumption 관계의 Negation이 Unsatisfiable 하다는 것을 증명하게 된다. 다음 표 1은 Tableaux 알고리즘에서 활용하는 기본 확장 규칙을 보여준다.

$\cap$-rule	if 1. $(C_1 \cap C_2) \in \mathcal{L}(x)$ 2. $\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$ then $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{C_1, C_2\}$
$\cup$-rule	if 1. $(C_1 \cup C_2) \in \mathcal{L}(x)$ 2. $\{C_1, C_2\} \cap \mathcal{L}(x) = 0$ then a. save T b. try $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{C_1\}$ If that leads to a clash then restore T and c. try $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{C_2\}$
$\exists$-rule	if 1. $\exists R.C \in \mathcal{L}(x)$ 2. there is no y s.t $\mathcal{L}(<x,y>)=R$ and $C \in \mathcal{L}(y)$ then create a new node y and edge <x,y> with $\mathcal{L}(y)=\{C\}$ and $\mathcal{L}(<x,y>)=R$
$\forall$-rule	if 1. $\forall R.C \in \mathcal{L}(x)$ 2. there is some y s.t. $\mathcal{L}(<x,y>)=R$ and $C \notin \mathcal{L}(y)$ then $\mathcal{L}(y) \rightarrow \mathcal{L}(y) \cup \{C\}$

표 1 Tableaux 알고리즘

2.3 시간 논리와 시간적 추론

본 절에서는 상황 정보 관리 시스템이 시간을 갖는 상황 정보들의 관계를 효과적으로 추론하기 위해 적용된 시간 논리(Temporal Logic)와 시간적 추론엔진(Temporal Reasoner)에 관하여 설명한다. Allen, J.F 은 행위와 시간 간의 관계를 시간 논리(Temporal logic)로 표현함으로써 시간적 추론(Temporal Reasoning)이 가능하도록 하였다[5][6][7]. Allen, J.F가 제안한 시간 논리는 현재 시맨틱 웹 분야에 적용되어 웹 페이지를 구성하는 여러 메타 데이터들 간의 시간 관계를 표현한다[8]. 다음 [그림 3]은 Allen이 정의한 시간 논리를 통하여 시간과 시간 간격간의 관계를 보여준다.

시간 논리에서 Instance는 하나의 시간 포인트를 나타내고, Interval은 시간 간격을 나타낸다. 시간 간격들의 관계를 표현하기 위해서는 Proper-Interval이라는 개념을 사용되는데, Proper-Interval은 시작점(Start-Point)과 끝점(End-Point)이 항상 존재하며, 이 두 점이 같지 않아야 성립된다. 시간 간격간의 관계는 크게 7가지로 표현된다. Before는 하나의 Proper-Interval의 끝점이 다른Proper-Interval의 시작점보다 앞서 있을 때를, Interval-Equals는 하나의 Proper-Interval의 시작점과 끝점이 다른 Proper-Interval의 시작점과 끝점에 일치할 때를, Meet는 하나의

Proper-Interval의 끝점과 다른 Proper-Interval의 시작점

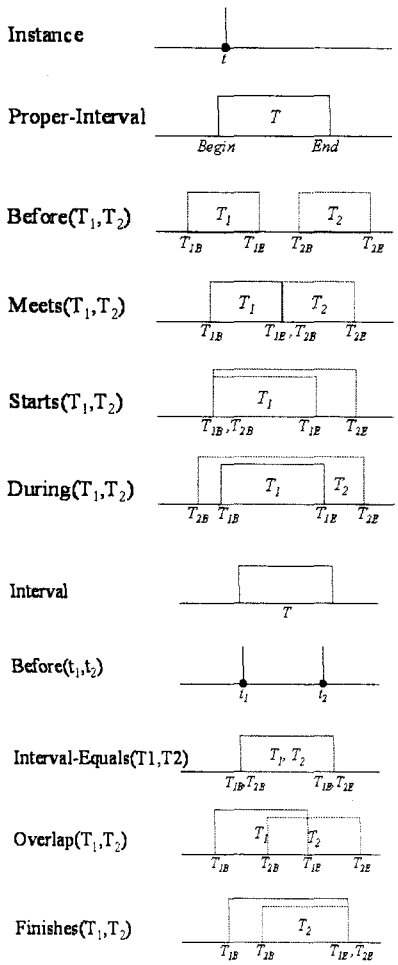

그림 3 시간과 시간 간격의 관계

이 일치할 때를, Overlap은 하나의 Proper-Interval의 끝점이 다른 Proper-Interval의 시작점과 끝점의 사이에 존재하고 하나의 Proper-Interval의 시작점과 끝점 사이의 다른 Proper-Interval의 시작점이 존재할 때를, Start는 두 Proper-Interval의 시작점은 같고 끝점이 서로 다를 때를, Finish는 두 Proper-Interval의 시작점은 서로 다르고 끝점이 같은 때를, During은 하나의 Proper-Interval의 시작점과 끝점 사이에 다른Proper-Interval이 존재할 때를 표현한다. 이러한 시간 간격들의 관계는 시간규칙(Time-Axiom)으로 적용되어 시간적 추론을 가능하게 한다.

2.3.2 시간적 추론 엔진

본 논문에서의 시간적 추론을 위한 기본적인 추론 전략은 단일화(Unification)을 기반으로 Rete 알고리즘을 사용하는 전방향 추론(Forward Chaining) 기법 이다. 전방향 추론(Forward Chaining)은 A → B 즉 조건 A 가 Working Memory 에 있으면 B를 수행한다. 따라서 당연히 연역법(Deduction) 에 속한다.

단일화는 두 개의 문장이 동일한 표현이 될 수 있도록 변수부호에 항을 대치하는 것을 말한다. 예를 들어서 (∀x) W(x) 문장에서 W(A) 라는 문장을 생성하기 위해서는 변수 x 에 문제의 대상 영역에 있는 상수부호 A 가 할당되어야 한다. 즉, W(x) 라는 문장과 W(A) 문장이 동일하게 되기 위해서는 x 가 A로 대치되어야 한다. 이러한 단일화 과정에서 여러 개의 항이 x 에 대치될 수도 있다. 변수를 대응하는 항으로 대체해서 얻어진 표현을 대치문 이라고 하는데, W(A)는 W(x) 문장에서 얻어진 대치문이 된다.

Rete 알고리즘(Forgy, 1982)은 전방향 추론 규칙 기반 시스템의 속도를 향상시키기 위한 만들어졌다. Rete 알고리즘은 순환이 없는 방향성 그래프(Acyclic Directed Graph)상에 규칙에 관한 정보를 저장함으로써 속도를 향상시키는 패턴 매치 알고리즘 이다. 즉 그래프의 모든 노드상에서 모든 규칙에 대한 사실(Facts)들을 매치 시키는 것이 아니고, 변화된 사실들에 대해서만 매치 시킨다. 따라서 각 노드에서 변화가 없었던 정적인 데이터는 무시되기 때문에 사실들의 매치 속도는 크게 향상된다. Rete 알고리즘이 적용된 시간적 추론 엔진은 시간 규칙들의 조건들을 하나씩 분리하여 루트 노드를 제외한 노드들에 표현한다. 즉, 루트 노드로부터 최하위 노드까지의 경로를 통해서 각 시간 규칙의 조건들이 표현되며 루트 노드로부터 현 위치의 노드까지 각 경로에 저장된 조건을 만족하는 사실(Fact)에 대한 정보를 각각의 노드에 저장한다. 이것은 추론엔진을 구동했을 때의 시간 중복성(Temporal Redundancy)을 줄여주는 장점이 있다.

2.4 추상화 엔진

분야 별로 상황 정보를 사용하는 쓰임새는 틀릴 것이다. 또한 추론하고자 하는 정보 역시 사용하는 범위에 따라 틀리다. 예를 들어서, 회의에 관련 상황 정보를 다루는 상황 정보 관리 시스템에는 "세미나의 발표자는 그 세미나의 참여자도 된다."라는 도메인 규칙이 적용 가능하다. 따라서 세미나의 관련된 상황 정보에 발표자들이 명시되어 있다면 추상화 엔진을 통하여 참여자 정보가 추론된다.

본 논문에서는 상황 정보 관리 시스템이 외부로부터 상황 정보를 받아들여 도메인 추론을 실행하는 엔진을 구축하기 위해 JESSTab[9] API를 사용하였다. JESS[10]는 썬 마이크로 시스템에서 제작한 규칙-기반 전문가 시스템 저작 도구이며, 자바 환경에서 구동된다. 전문가 시스템(Expert System) 은 전문가가 가지고 있는 지식을 인위적으로 컴퓨터에게 부여하여 그 방면에 비전문가라 할지라도 그러한 전문가의 지식을 이용하여 상호 대화를 통하여 원하는 결과를 얻는 일종의 자문형 컴퓨터 시스템이다. JESS는 규칙을 바탕으로 한 추론 엔진을 제공하고, 추론 엔진에서 사용하는 사실(Fact)과 규칙(Rule)들을 묘사해 준다. 이러한 규칙 기반 프로그램은 수백 혹은 수천 개의 규칙을 가지고 있으며 규칙 베이스에 내장된 데이터들을 규칙에 계

속적으로 적용한다. JESSTab은 이러한 JESS와 OWL간의 연결을 해주는 역할을 한다. 따라서 OWL로 표현된 상황 정보에 JESS로 구축한 엔진을 연결할 수 있으며, 온톨로지를 조작하는 명령어가 추가적으로 만들어져 있기 때문에 쉽게 OWL문서에 도매인 규칙을 적용할 수 있다. 다음 표 2는 제안하고자 하는 시스템에서 사용하고 있는 도매인 규칙을 JESS 규칙으로 표현한 일부를 보여준다.

```
(defrule part01
    (object (is-a Seminar)(OBJECT ?x) (speaker ?a))
    (not(object (is-a Seminar)(OBJECT ?x)
        (speaker ?a)))
    =>
    (slot-set ?x hasParticipant  ?a))
(defrule beforeDes
    (object (is-a CalendarClockInterval)
        (OBJECT ?t1) (begine ?x))
    (object (is-a CalendarClockInterval)
        (OBJECT ?t2) (end ?y))
    (object (is-a CalendarClockDescription)
        (OBJECT ?x) (year ?a))
    (object (is-a CalendarClockDescription)
        (OBJECT ?y) (year ?b))
    (> ?a ?b)
    =>
    (slot-set ?t2 before  ?t2))
)
```

표 2 JESS로 표현된 도매인 규칙

3. 실험 및 고찰

본 논문에서 제안하는 서술 논리 추론과 시간적 추론이 적용된 상황 정보 관리 시스템은 명시적인 의미만을 갖는 상황 정보에 숨겨진 여러 가지 의미(계층 구조, 시간 관계, 도매인 지식)를 알아내어, 상황 정보 에이전트에 제공할 수 있다. 그래서 본 논문이 제안하는 상황 정보 관리 시스템의 실행 가능성을 평가하기 위해서 온톨로지 추론 엔진인 RacerPro를 상황 정보 관리 시스템에 적용하여 서술 논리 추론을 실험해보고, 이 외에 시스템을 구성하는 시간적 추론 엔진, 도매인 추론 엔진을 구현하여, 상황 정보가 적절한 상황 정보가 추론 되는지 실험해 보았다.

3.1 시나리오

본 절에서는 서술 논리 추론과 시간적 추론이 적용된 상황 정보 관리 시스템의 실험에 적용할 간단한 시나리오를 설명한다.

Park은 2006년 8월 13일 아침 9시부터 11시 50분까지 IT연구소에서 주최하는 학술회의에 참석했다. 그리고 오후 1시부터 4시까지 자신의 사무실에서 외부인과 프로젝트 미팅을 가졌다.

Park의 이러한 상황 정보는 온톨로지 인스턴스로 생성되며 시멘틱 상황 정보 관리 시스템은 숨겨진 정보를 추론한다. 일단 온톨로지에서 세미나는 "2명 이상이 참가하면서 발표자가 있는 이벤트"로 정의되고, 학술회의는 "최소한 발표자가 한명 이상 존재하는 이벤트"라고 정의 되어있기 때문에 서술 논리 추론에 의해 학술회의는 세미나에 SubClass가 된다. 따라서 만약 상황정보 에이전트가 "Park이 8월에 참가한 세미나"라고 질의를 주면, 비록 초기에는 학술회의 정보밖에 없더라도, 서술 논리 추론을 통해 Park이 참가한 학술회의 정보를 줄 수 있다.

또한 프로젝트 미팅 시작 시간보다 학술회의 종료시간이 앞서 있으므로 학술회의는 프로젝트와 Before 관계를 갖게 된다. 따라서 상황 정보 에이전트가 "Park이 프로젝트 미팅 이전에 참가했던 회의"라는 질의를 주면, 시간적 추론을 통해 Park이 참가한 학술회의 정보를 줄 수 있다.

3.2 실험

본 실험을 결과를 보기 위하여 온톨로지 구축 도구인 Protege를 사용하였으며, 서술 논리 추론 엔진으로는 RacerPro[11]를 적용하였다. 또한 시간적 추론 엔진과 도매인 추론 엔진은 JESS로 구축한 후, JESSTab을 이용하여 Protege와 연결하였다.

그림 4는 서술 논리가 적용되기 전의 상황 정보가 속한 클래스의 계층 구조와 서술 논리 추론을 통해 학술회의가 세미나에 SubClass로 분류되는 결과를 보여준다.

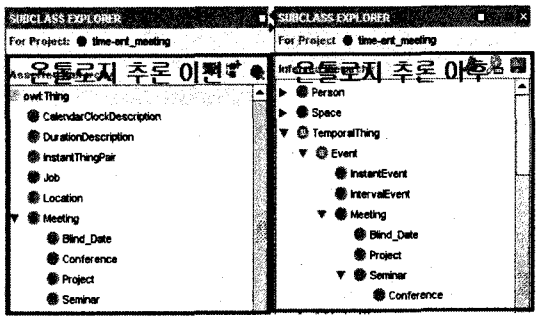

그림 4 서술 논리 적용 결과

그림 5와 6은 시간적 추론 과정을 거쳐 학술회의는 프로젝트와 Before 관계를 갖는다는 결과를 보여준다.

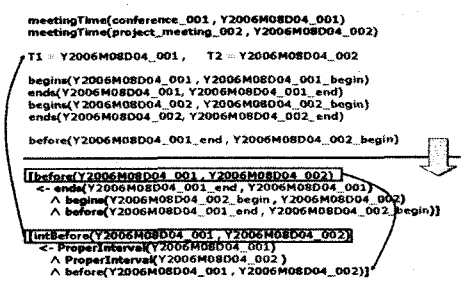

그림 5 시간적 추론 과정

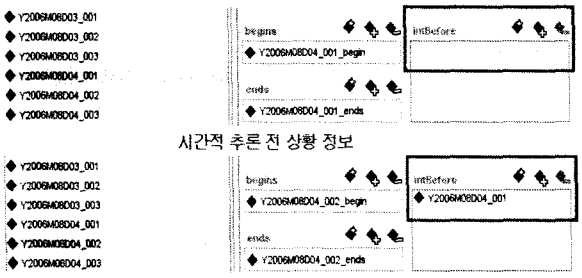

그림 6 시간적 추론 적용 결과

4. 결론 및 향후 연구

본 논문에서는 외부 환경으로부터 받은 상황 정보의 숨겨진 의미를 파악하기 위해 DL 추론과 시간적 추론을 적용한 상황 정보 관리 시스템을 제안하였다. 효과적인 상황 정보 제공을 위해 시멘틱 상황 정보 관리 시스템은 외부에서 입력되는 상황 정보(Context Information)간의 차이를 조절하고 상황 정보가 속하는 클래스의 계층 관계를 정확하게 분류하며, 실제 발생한 상황 정보의 시간적인 관계를 추론해야 한다. 따라서 다음과 같은 연구를 진행하였다. 첫 번째, 외부에서 입력된 상황 정보를 효율적으로 표현하고 여러 에이전트간의 상황 정보 공유가 가능하도록 온톨로지 모델을 적용하였다. 두 번째로 상황 정보의 관계를 추론할 수 있도록 서술 논리(Description Logic)를 적용하였다. 마지막으로 상황 정보의 시간적 관계를 추론할 수 있도록 시간 논리(Temporal Logic)을 적용한 시간적 추론 엔진에 대해 연구했다.

본 논문은 외부 환경에서 에이전트에게 적절한 상황 정보를 제공하기 위해 서술 논리 추론과 시간 추론(Temporal Reasoning)을 통해서 얻어진 새로운 상황 정보를 블랙 보드 저장한다. 일반적으로 시스템에서 제공하는 블랙 보드는 빠른 접근 속도를 가지고 있지만, 용량의 한계 때문에 대규모의 상황 정보를 한꺼번에 처리하는 데는 무리가 따른다. 또한 본 논문에서 사용하고 있는 Tableaux 알고리즘과 이를 기반으로 동작하는 추론 엔진 역시 대용량의 상황 정보를 처리하는데 있어 느린 속도를 갖는다. 이러한 문제점을 해결하기 위해서 효과적으로 DB를 적용하여 추론 횟수를 최소화 하고 대용량 상황 정보를 처리 할 수 있는 추론엔진의 최적화 기법이 연구될 것이다.

참고문헌

[1] Dey A.K., et al. "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications", anchor article of a special issue on Context-Aware Computing, Human-Computer Interaction(HCI) Journal, Vol.16, 2001.

[2] Baader F., et al,The Description Logic Handbook, Cambridge, 2003.

[3] Genesereth, M., Nilsson, N. Logical Foundations of Artificial Intelligence, Morgan Kaufman, 1987

[4] Ian Horrocks, Optimizing Tableaux Decision Procedures for Description Logics, PhD thesis, University of Manchester, 1997

[5] Jerry Hobbs, James Pustejovsky, "Annotating and Reasoning about Time and Events", http://www.cs.rochester.edu/~ferguson/daml/, DAML-Time Homepage

[6] Frank Schilder, Christopher Habel, "From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages", in Proceedings of the ACL-2001 Workshop on Temporal and Spatial Information Processing, ACL-2001. Toulose, France, pp.65 ~ 72, 2001.7

[7] James F Allen, "Planning as Temporal Reasoning", reprinted from the Proc. of the Second International Conference on Principles of Knowledge Representation and Reasoning, Cambridge, MA April, 1991

[8]Jerry R. Hobbs, "A DAML Ontology of Time", http://www.cs.rochester.edu/~ferguson/daml/, DAML-Time Homepage

[9] Henrik Eriksson, JessTab Manual, January 8, 2004

[10] Ernest F.H, "Jess In Action", Manning Publications Co, 2003

[11] RacerPro Reference Manual Version 1.9, http://www.racer-systems.com

[12] Abowd G, Mynatt E, "Charting past, present, and future research in ubiquitous computing", ACM Transactions on Computer Human Interaction, Vol.7, No.1, pp.29 ~ 58, March 2000