가상현실에서 에이전트 지식구조의 스키마 통합 기법

김동훈° 박종희
경북대학교 전자공학과
{sunshinesky°, jhpark}@ee.knu.ac.kr

Schema Integration mechanism of Agent knowledge structure in Virtual Reality

Donghoon Kim° Jonghee Park
Department of Electronic Graduate School, Kyungpook National University

요 약

인간의 지식구조는 현실세계의 무수히 많은 정보와 지식들의 복잡한 상호 관계들로서 연결 되어 있다. 가상 현실에서 에이전트의 지식구조는 인간의 지식구조와 같이 다양하고 다량의 정보들의 표현이 이루어져야 한다. 이를 위하여 지식구조는 지식 표현 기술 분야로서 연구 되어지고 있는 온톨로지의 관계표현으로 구성하여 표현한다. 자율 에이전트의 행위나 다양한 상황의 표현을 위해 다수의 스키마를 이용하여 지식구조를 구성하게 되지만 중복적인 스키마의 구성이 많아지게 되어 정보의 갱신이나 삽입 시에 문제를 야기시키게 된다. 본 논문에서는 이러한 중복적인 스키마들의 구성을 최소화 시키고 스키마들의 체계적인 관리를 위한 스키마 통합의 방법들을 제시하고자 한다.

1. 서 론

온톨로지는 특정한 영역을 표현하는 데이터 모델로 객체의 종류 및 속성, 객체 사이의 관계에 대하여 표현한다[1]. 가상현실에서 온톨로지의 관계표현, 즉 스키마의 구성으로서 이벤트 및 에이전트의 행동과 관련된 여러 가지 현상이 일어나도록 하는 지식들이 구조화되어 표현 된다. 가상현실에서 에이전트의 행위와 다양한 상황의 표현을 위해서 에이전트의 지식구조는 다수의 스키마 들로 구성된다.

많은 수의 스키마들이 여러 개념 안에서 각기 다른 상황이나 관계들을 표현하기 위해서 복잡하고 중복적으로 구성되면서 구조의 체계적인 관리 및 정보의 갱신 이상이나 삽입 이상[2]과 같은 문제들을 야기 시킨다. 가상현실에서 에이전트의 지식구조 통합을 위한 방법과 유사한 database schema integration을 위한 방법에 관한 연구가 이루어져 왔다. 이 연구에서는 view integration과 database integration의 두 환경으로 정의되는데 사용자의 관점에서 여러 형태들로 구성된 모델을 전체적인 모델로 통합하거나 데이터베이스의 수집으로 전체 구조를 만들어 내는 연구를 수행하였다[3]. 기존의 구조 통합의 연구들은 사용자의 관점을 통해서 전체 구조를 구성하는 연구를 수행하였기에 단편적이고 정형화된 상황의 통합에 적합하지만 가상현실에서 에이전트의 많은 지식을 구성하고 있는 스키마들의 통합 시엔 부족한 면이 있다. 우리는 여러 객체들과 속성 및 행위들이 다양한 관계로서 구성되어 있는 가상현실에서의 에이전트의 지식구조 인 스키마 들의 중복적인 구성을 최소화하기 위해서 스키마 통합 방법을 제시하고자 한다.

본 논문은 다음과 같이 구성 되어진다. 2장에서는 본 논문에서 다루어지는 관련 연구들에 대해서 살펴보고 3장에서는 고유한 특성이나 속성들로 구성된 지식구조에 대해서 살펴보겠다. 4장에서는 효율적인 구조 통합을 위한 방법을 제시하고 5장에서는 결론 및 앞으로의 연구 방향에 대해서 제시하도록 하겠다.

2. 관련 연구

2.1 Schema integration

Schema integration을 위해서 View integration[4], Database Schema integration[3] model 등이 연구되어 왔다. View integration에서의 통합 방법은 속성 형태나

연결 형태 등으로 구성된 모델들을 통합하는데 중점을 두고 연구해 왔다. 하지만 본 논문에서 제시 되어지고 있는 스키마의 형태에서는 객체, 속성 및 연결과 같은 부분들이 하나의 스키마를 구성하고 통합하기 때문에 스키마 통합에 적용하기에는 제한적이다. Database schema integration 에서는 데이터베이스 구조를 통하여 구조들의 비교, 재구성 및 머지 등의 방법들을 이용하여 통합을 시도하였고 그 구조는 모든 개념이 일관성 있고 중복되지 않도록 처리하였으며 사용자나 구성자가 쉽게 이해할 수 있도록 구성하였다. 그러나 단지 E-R model의 구조를 가진 스키마의 통합 시 이름의 충돌에 의해 이름의 변경, 객체의 생성 등의 단순한 통합 방법에 대하여 제시하였기 때문에 본 논문에서 다루고자 하는 하나 이상의 중복적인 지식을 구성하는 스키마의 통합에 적용하기에는 부족함이 있다.

3. 에이전트의 지식구조

이 절에서는 가상현실에서의 에이전트, 즉 주위의 환경을 고려하면서 미리 주어진 지식을 활용해 자율적으로 행동하는 인간적인 기능을 갖추고 있는 행위자[5]의 지식구조의 구성에 대해 살펴본다.

3.1 Class hierarchy

전 우주의 구성요소로서 추상화 되어지는 개념들은 객체들과 그들의 상호 관계의 용어로서 온톨로지를 이용한다[6]. 온톨로지를 구성하는 클래스는 시간과 공간의 통계적인 분석으로서 형성되며 class hierarchy로서 구성되어 진다. 각 클래스는 그것의 대표적이거나 통계에 근거한 값들로서 나타내어 진다. 클래스에서 instance의 group들은 클래스의 sub클래스들을 한정할 수 있는 충분한 특성들을 나타낼 수 있다. 클래스가 만약 클래스 자체를 정의하는 특성들의 집합을 가진다면 많은 클래스들로 specialized 될 수 있다. 그림 1과 같이 가상현실에서 에이전트의 지식구조는 각 클래스 들이 상호 관계를 통해서 Class hierarchy 구조를 이용해서 구성하고 있다.

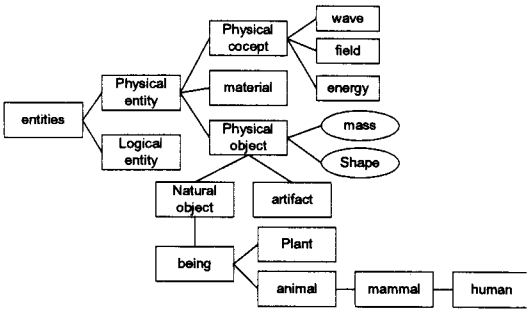

그림 1. ontology의 구조

3.2 Definitional attribute

Definitional attribute는 클래스의 계층구조에서 개체의 위치에 따라 그 위상이 바뀌는 경우를 표현하기 위해서 사용 되어진다[7]. 유사한 인스턴스 또는 발생(occurrence)의 추상화로써 클래스는 그것의 definitional attribute에 의해서 정의된다. 서로 다른 클래스들은 공통적인 속성값을 가질 수 있고 클래스 들의 속성값에 관점을 두고 의미론적으로 하나 또는 그 이상의 클래스의 계층구조를 조직화 할 수 있다. 하나의 클래스는 그 클래스에서부터 최상위 클래스까지 가진 모든 definitional attribute를 합함으로써 정의된다. definitional attribute는 specialization link상에 기술 되어지는데 예를 들어 그림 4.a에서와 같이 living thing 클래스는 mobility라는 속성에 따라서 plant 클래스와 animal 클래스로 연결되는 link로 구성된다.

3.3 Characteristic properties

우주를 구성하는 모든 개체 즉, 인스턴스나 클래스들은 그 자체로서 고유한 identity를 형성하고 있다. 모든 개체의 고유한 identity는 다른 개체들과 공유되지 않는 activity나 attribute와 같은 characteristic property들에 의해서 인식 되어진다. 예를 들어 지구 클래스에서 shape, material, size와 같은 특성들이나 이빨을 가진 animal 클래스에서는 tooth, bite()와 같은 characteristic property들로서 고유한 identity를 형성한다. 만약 앞의 예에서와 같이 객체만이 가진 characteristic property를 인식 할 수 있다면 이 객체가 포함 되어져 있는 클래스를 알 수 있을 것이다.

4. 스키마 통합 방법

스키마의 통합을 위해서는 먼저 스키마들의 비교가 우선시 되어야 한다.

4.1 속성 및 특성 비교

지식구조를 구성하고 있는 엔티티들은 그들의 속성들이나 특징적인 값들로서 표현 되어 질 수 있다[8]. 특정적인 값들이나 그들의 구체적인 값들은 그들의 상호 의존 관계를 나타내며 연결 되어 있다. 기본적인 엔티티에서 제외된 각 기술적인(descriptive) 속성들은 그것의 구성하는 모든 면들을 나타내는 값들로 할당되어 사용된다. 그러한 값들은 전체나 통계적인 성질을 나타낼 수 있다. 또한 어떤 다른 속성들은 엔티티를 대표하는 값을 가질 수도 있다. 클래스의 내부적 특성들은 보통의 속성들로서 그들의 정의를 포함한다. 그것의 속성들의 값의 통계적인 특징들은 클래스 특성의 key element 이다. 이렇게 노드들은 그 클래스 특징을 가진 attribute와 activity들과의 연결을 통해서 구성된다. 구성된 노드들의 특성, 즉 앞 절에서 설명한 definitional attribute 나 characteristic property들의 비교를 통해서 유사 노드를 찾게 된다. 그림 2에서처럼 우측 woman 클래스의 스키마에서 속성 즉, attribute = {size, shape} 와 characteristic property = {bear()} 로서 구성되어 있을 때 좌측 being 클래스 스키마에서 human 클래스의 속성 비교 시 woman 클래스와 유사성을 가지고 있지만 human을 상속하고 있는 woman 클래스에서 연결되어 있는 characteristic property인 bear()에 의해서 human보다 더 유사한 노드로 인식할 수 있고 두 스키마의 통합 시에 두 노드를 선택 할 수 있다.

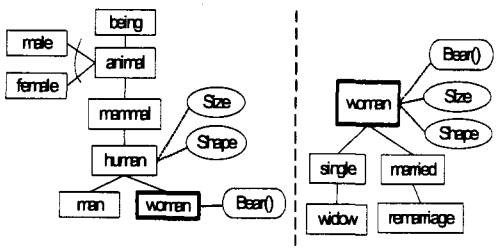

그림 2. 속성 및 특성을 통한 비교

4.2 이름 비교

통합하고자 하는 스키마의 노드들을 이름의 비교를 통해서 같은 이름 이나 이름이 완전히 일치하지 않을지라도 이름의 유사성 즉 의미론적인 유사성의 비교로서 같은 노드를 찾아내는 방법이다. wordnet에서 명사, 동사, 형용사, 부사들의 동의어 집합[9]을 구성한 것과 같이 유사성을 찾기 위해 synset이라는 동의어 집합을 사용한다. 그림 3에서처럼 female이라는 노드의 synset을 구성함으로써 woman과 비교 시 유사 노드로서 인식할 수 있다.

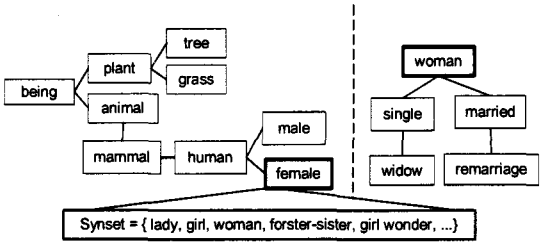

그림 3. 이름 비교

4.3 구조적 유사성 및 계층적 비교

스키마의 노드의 속성 이나 특성의 비교 뿐만 아니라 두 스키마의 구조적 유사성이나 클래스 계층 구조의 비교를 통해서 구조의 통합을 위한 노드를 선정할 수 있다. 그림 4의 두 구조는 generalization hierarchy[6]구조로 구성되어 있고 객체들은 상호 관계를 이루면서 연결되어 있으며 각 객체들의 링크는 a kind of 로서 연결되어 구조적인 유사성을 찾을 수 있다. 그림 4.b의 구조에서 object와 animal 두 객체는 AKO 링크로서 연결되어 있는데 그림 4.a와 비교 했을 때 living thing이라는 객체가 추가적으로 연결되어 있음을 알 수 있다. 두 스키마는 구조적 차이를 보이긴 하지만 living thing이라는 객체에서 상위 노드 및 하위 노드의 level을 1단계씩 비교 함으로써 링크의 연결 및 구조의 계층적 유사성을 찾을 수 있다. 그러므로 그림 4의 구조에서 통합 시 living thing은 object와 animal과 계층적 유사성을 통해 연결 되어 있다. 그림 4.b에서 animal 객체가 좀더 구체적인 계층구조로 구성되어 있기 때문에 두 구조의 통합을 위해 animal 객체를 통합의 노드로서 선정 할 수 있다.

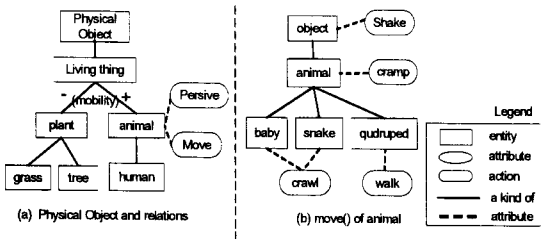

그림 4. 구조적 유사성 및 계층 비교

4.4 통합을 위한 순서도

비교를 통한 통합방법을 토대로 한 전체 프로세서의 순서도는 그림 5와 같다.

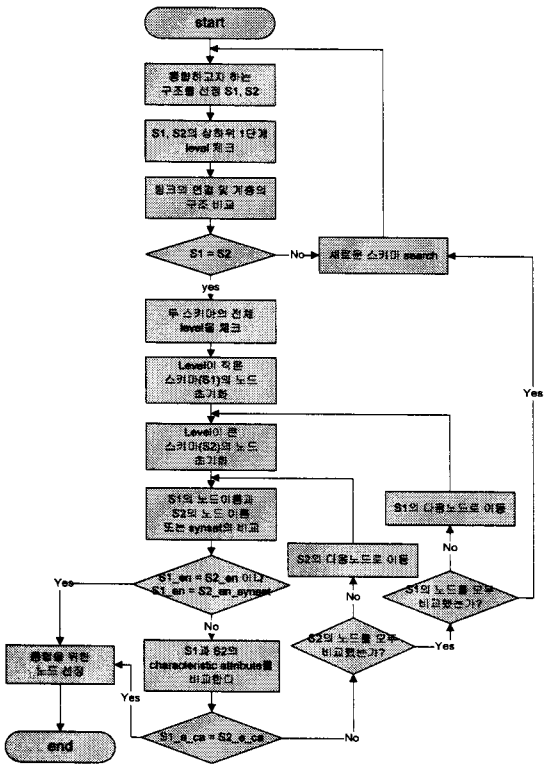

그림 5. 전체 프로세서에 관련된 순서도

5. 결론 및 향후 연구방향

본 논문에서는 인간의 지식구조와 같이 가상현실에서 많은 정보가 복잡한 상호관계를 맺고 있는 에이전트의 지식구조에 대해 살펴보았고 그 스키마들의 통합 방법으로 속성이나 특성들의 비교 뿐만 아니라 구조적인 관계나 계층적인 비교 및 이름의 비교를 통해서 통합을 위한 적합한 객체를 선정하는 방법들을 제시하였다. 통합 방법을 통해서 중복적으로 사용하고 있는 가상현실에서의 에이전트의 지식 구조들의 최소화 및 구조의 정보 갱신이나 삽입시의 이상문제를 해결 할 수 있다.

앞으로 구조 통합을 위한 각 노드들의 다양한 링크에 대한 규약을 부여하며 추가적으로 스키마 통합을 위한 tool 개발 연구를 진행할 예정이다.

[참고 문헌]

[1] B.Chandrasekaran and John R.Josephson, What are ontologies, and why do we need them?, IEEE INTELLGENT SYSTEMS

[2] Ramakrishnan, Gehrke, "Database Management Systems", McGRAW-HILL Co., 3rd ed., pp606-607 2003

[3] Carlo Batini, Maurizio Lenzerini, Shamkant B. Navathe: A Comparative Analysis of Methodologies for Database Schema Integration. ACM Comput. Surv. 18(4): 323-364(1986) BibTeX

[4] Navathe,s.b., Gadgil,s.g : A methodology for view integration in logical data base design. In Proceedings of the 8th International Conference on very Large Data Bases. VLDB Endowment, Saratoga, Calif, 1982

[5] 배경표., "사이버인간: 동적 환경에서 능동 에이전트간 상호작용",Proceedings of KISS Fall Conf, 96-98(1998)

[6] Park, J., "Ontology about the microcosm", Tech. report #9, AIMM Lab., Kyungpook Nat'l Univ.,Feb.,2004

[7] 노선미, "효율적인 표현을 위한 Cyber-Microcosm Ontology의 지식구조", 경북대학교 석사 학위 논문, 2005

[8] Park, J., "modeling physical entities and their associated relationships in the cyber world", Tech. report #10, AIMM Lab., Kyungpook Nat'l Univ.,Jan.,2004

[9] Miller, George A., Richard Beckwith, Christiane Fellbaum, Derek Gross and Katherine J. Miller. "Introduction to WordNet : an on-line lexical database", In : International Journal of Lexicography 3 (4), 1990.