

인 터 넷 진 화 의 열 쇠

온톨로지

웹 2.0에서 3.0으로

노상규·박진수 공저

god's toy business, 2007

SNU IDB laboratory

Ontology Contents

MODULE 1 온톨로지의 개념 및 응용

- Chapter 1 온톨로지 개요
 - 1. 온톨로지의 유래
 - 2. 분류와 개념화 과정
 - 3. 컴퓨터 온톨로지
- Chapter 2 온톨로지의 분류와 용도
 - 1. 온톨로지의 분류
 - 2. 온톨로지의 사용 목적과 중요성
 - 3. 온톨로지와 시맨틱 웹
- Chapter 3 온톨로지 구축 프로젝트
 - 1. 사이크(Cyc)
 - 2. 워드넷(WordNet)
 - 3. 전자거래문서
 - 4. 통합의학언어시스템
 - 5. 오픈 디렉터리 프로젝트
 - 6. 국제상품분류코드(UNSPSC)
- Chapter 4 온톨로지 적용 분야
 - 1. 전자상거래 분야
 - 2. 의료 분야
 - 3. 법률 분야
 - 4. 검색 서비스 분야
 - 5. 문화컨텐츠 분야

MODULE 2 온톨로지 언어와 구축도구

- Chapter 5 온톨로지 언어
 - 1. 온톨로지 언어의 발전 과정
 - 2. 인공지능 기반의 온톨로지 언어
 - 3. 온톨로지 마크업 언어
- Chapter 6 RDF(S): RDF와 RDF Schema
 - 1. XML과 RDF
 - 2. RDF
 - 3. RDF Schema
 - 4. RDF(S)의 한계점
- Chapter 7 OWL(Web Ontology Language)
 - 1. OWL의 기본 요소: 클래스와 속성
 - 2. OWL의 새로운 기능
 - 3. 세 종류의 OWL
 - 4. OWL 예제
- Chapter 8 토픽맵(Topic Maps)과 XTM(XML Topic Maps)
 - 1. 토픽맵(Topic Maps) 개념
 - 2. 토픽맵 구성요소
 - 3. XTM 예제
- Chapter 9 온톨로지 틀
 - 1. 온톨로지 틀의 분류
 - 2. 온톨로지 개발 틀
 - 3. 주요 온톨로지 틀 요약 정보

Chapter 5 온톨로지 언어

- 1. 온톨로지 언어의 발전 과정
- 2. 인공지능 기반의 온톨로지 언어
- 3. 온톨로지 마크업 언어

1. 온톨로지 언어의 발전 과정

■ 1.1 온톨로지 언어의 기초가 되는 지식표현 패러다임 이론

- 1.1.1 시맨틱 네트워크(Semantic Network)
- 1.1.2 프레임(Frame)
- 1.1.3 규칙(Rules)
- 1.1.4 일차논리(FOL : First Order Logic)
- 1.1.5 기술논리(DL : Description Logic)

■ 1.2 온톨로지 언어의 역사

- 1980년 이전 : 지식표현 철학적 이론
- 1980-1990년: 이론을 바탕으로 AI Knowledge Representation 언어
- 1990-1995년 : AI Knowledge Representation에 기반한 온톨로지 언어
- 1995년 이후 : 마크업(markup) 기반 온톨로지 언어

1.1 온톨로지 언어의 기초가 되는 지식표현 패러다임

■ 지식

- 정보를 보다 쉽게 이해하고 응용할 수 있도록 체계화하고 구조화한 것

■ 지식 표현

- 지식을 사람과 컴퓨터가 함께 이해할 수 있는 형태로 나타내는 것
- 지식표현 방법
 - ▶ 사람이 사용하는 언어와 컴퓨터 언어 사이의 타협점을 찾아 결정
 - ▶ 지식표현의 접근방법에 근본적 영향을 주는 지식표현 패러다임의 이해가 필요

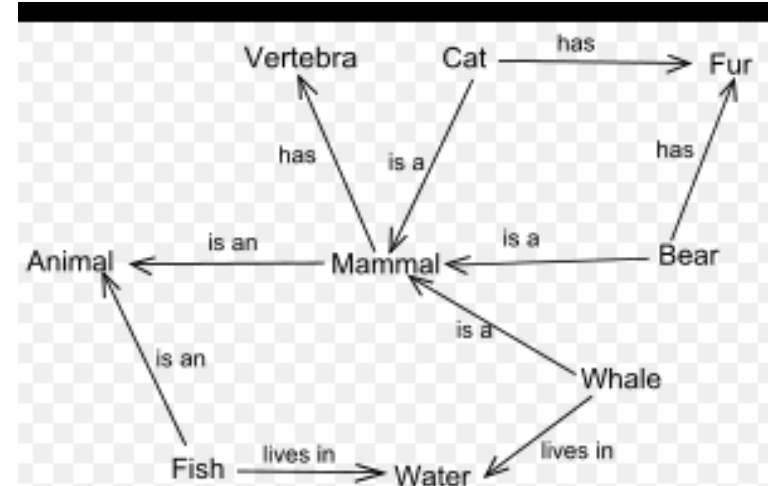
1.1.1 시맨틱 네트워크(Semantic Network)

■ 역사

- “Existential Graphs” [Charles S. Peirce 1905]
- Richard Richens (1956), Simmons & Quillian (1960년대 초)
- HyperText System의 가장 기본단위의 시맨틱 링크의 개념의 제안

■ 구조

- 객체(object)나 개념(concept)을 나타내는 노드
- 노드 사이의 관계를 보여주는 링크
 - ▶ 하위 개념에서 상위 개념으로 향하는 방향성
 - ▶ 객체에서 속성으로 향하는 방향성
 - ▶ 관계는 링크 옆에 라벨을 붙여 명시



Origin of Inheritance

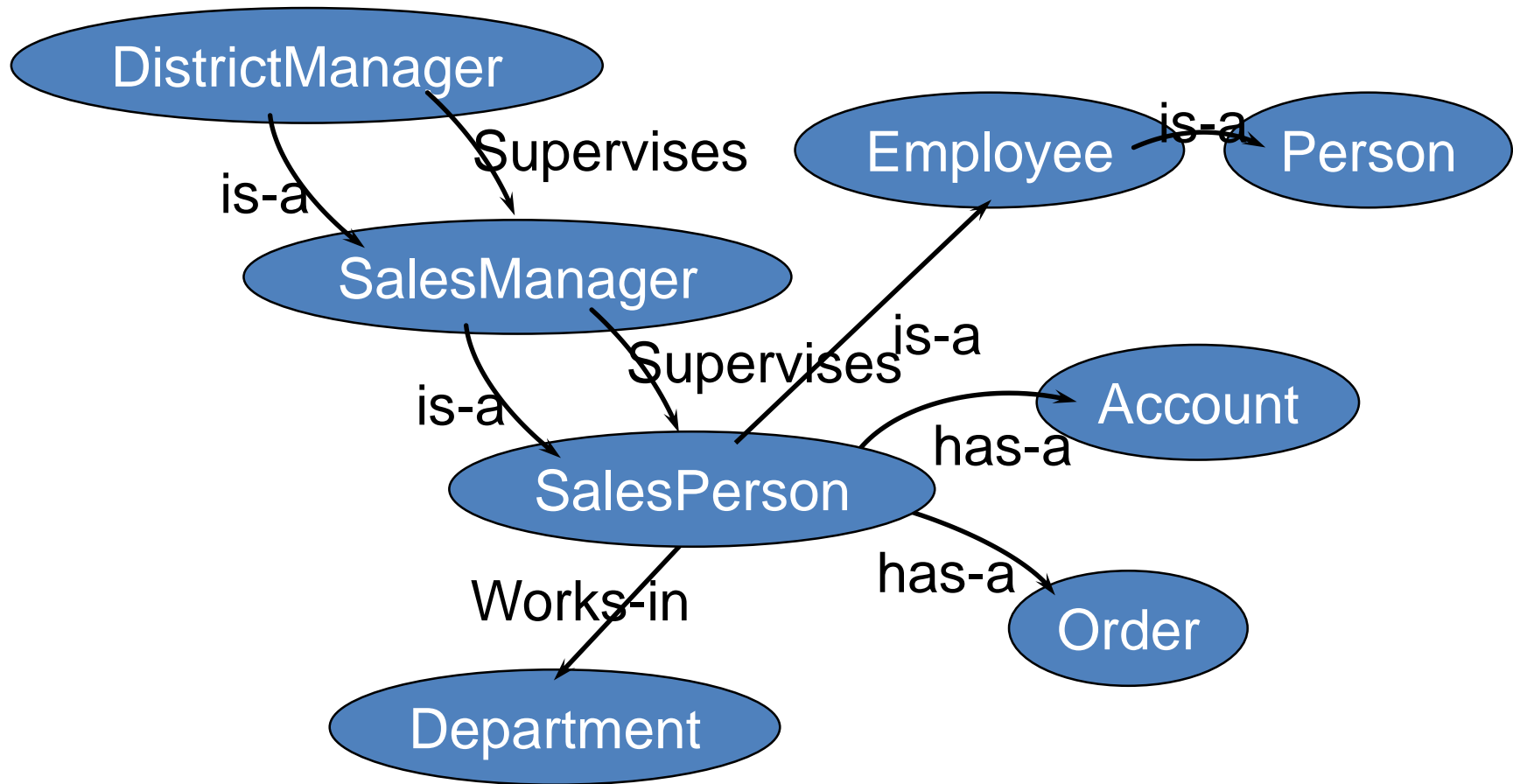
■ Semantic network – Quillian (1968)

- “Node-and-link” model
- *Node*: concepts(objects)
- *Link*: relationships among concepts
- *Label*
 - ▶ ‘IS-A’: inheritance relationships
 - ▶ ‘HAS-A’: attributes of concepts

■ Frames – Minsky (1975)

- Record-like structure: *Slot*
- New concepts from previously defined frames

AI Semantic Network for SalesPerson



1.1.1 시맨틱 네트워크(Semantic Network)

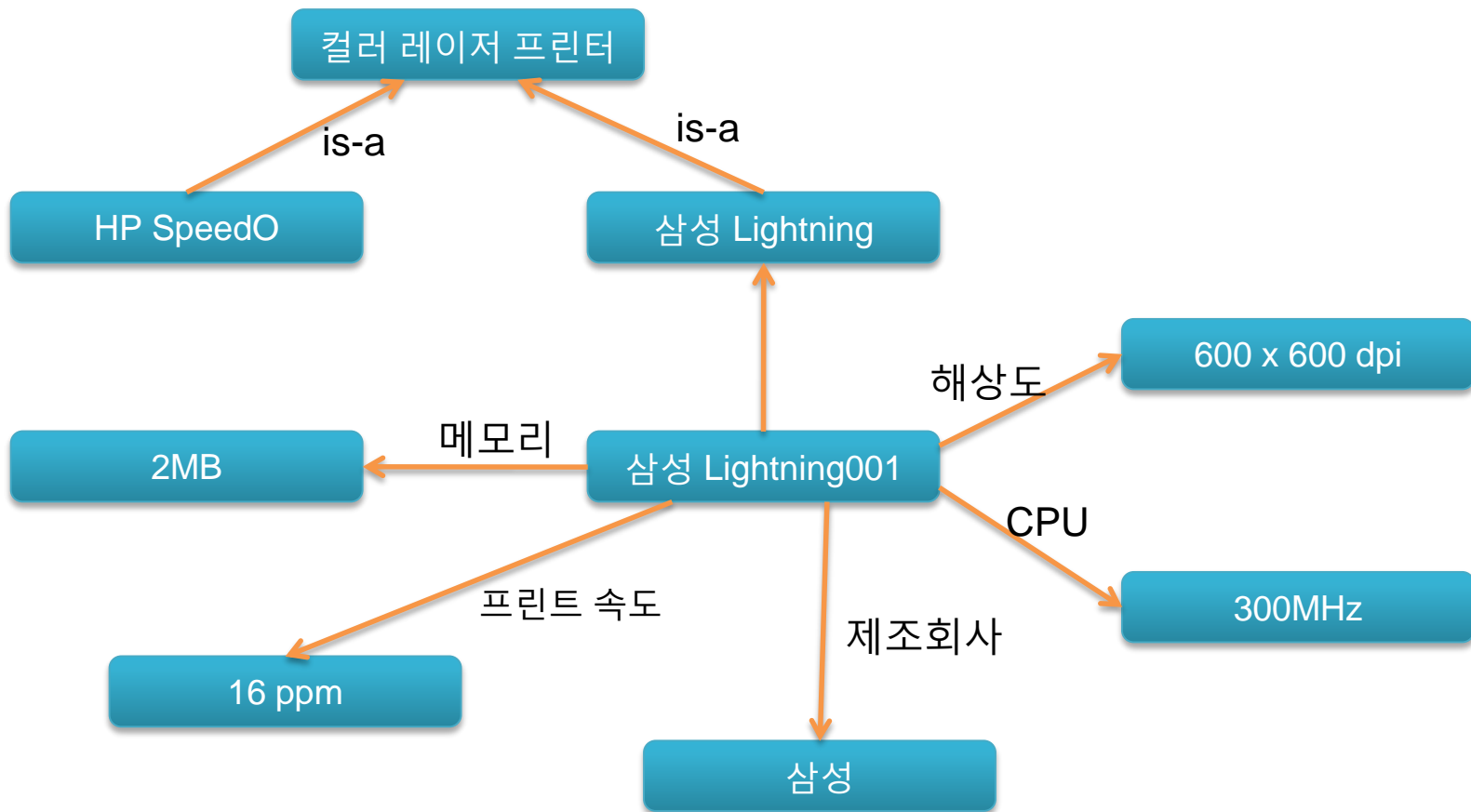
■ 장점 & 단점

- 장점 : 표현이 유연 (이해하기 쉬운 기본 방식으로 나타냄)
- 문제가 복잡해지면 노드와 링크의 수가 대폭 증가하여 검색시간이 길어짐

■ 시맨틱 관계 (Semantic relations)

- Meronym (A is part of B, i.e. B has A as a part of itself)
- Holonym (B is part of A, i.e. A has B as a part of itself)
- Hyponym (or troponym) (A is subordinate of B; A is kind of B)
- Hypernym (A is superordinate of B)
- Synonym (A denotes the same as B)
- Antonym (A denotes the opposite of B)

1.1.1 시맨틱 네트워크(Semantic Network)



1.1.2 프레임(Frame)

■ Marvin Minsky "A Framework for Representing Knowledge" (1974)

■ 구조

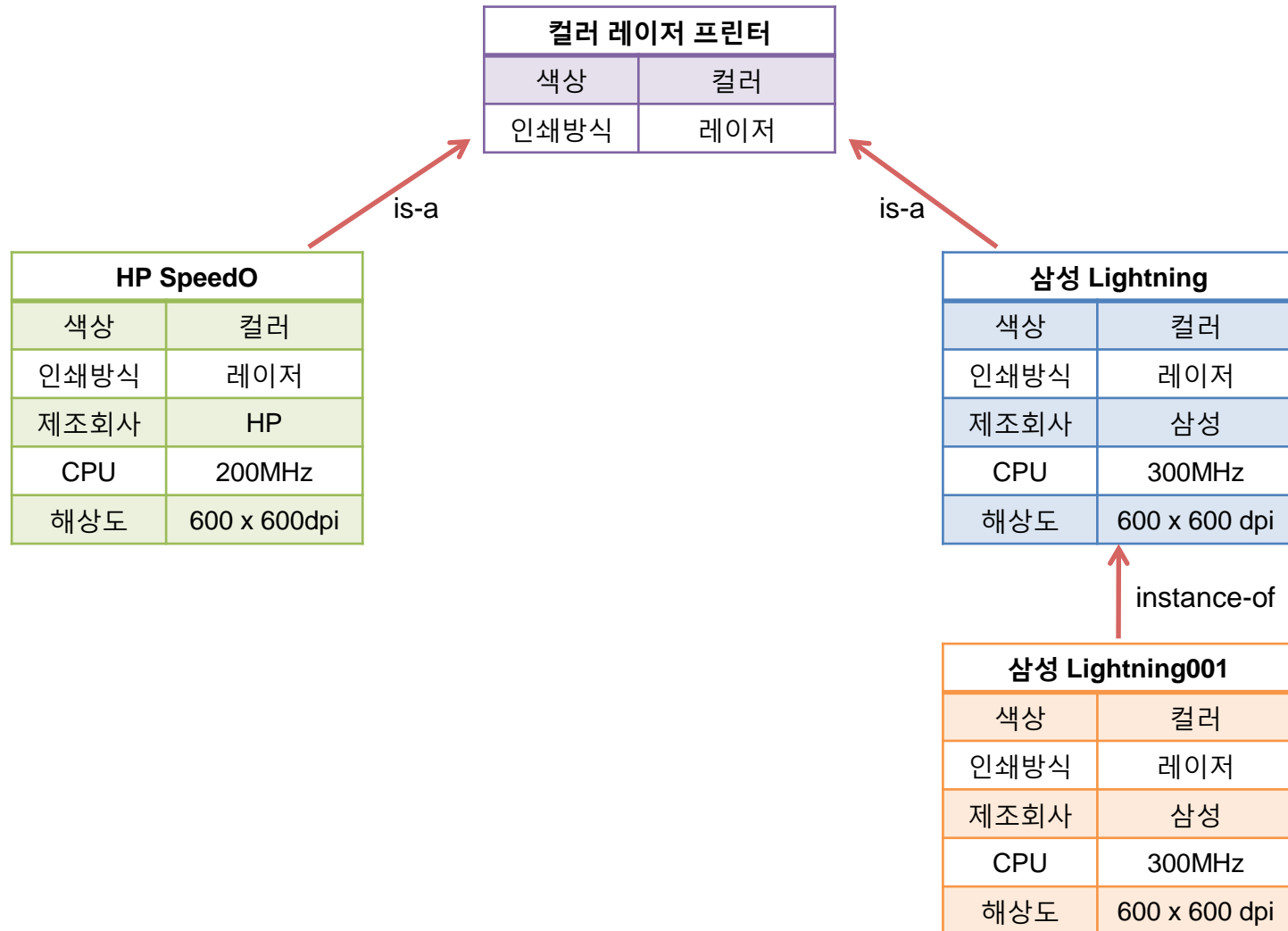
- 시맨틱 네트워크와 비슷한 구조
 - ▶ 시맨틱 네트워크로부터 유래
 - ▶ 상속관계를 형성(추론 메커니즘의 핵심)
- 노드 대신에 프레임을 사용
- 객체지향(object-oriented) 방법과 유사 (1 frame \approx 1 object)

1.1.2 프레임(Frame)

■ 프레임

- 시맨틱 네트워크의 단점을 보완하기 위해 노드의 구조 체계화
- 특정 개념에 대한 정보를 보다 밀도 높게 표현하는 방식
- 여러 개의 슬롯(slot)으로 구성
 - ▶ 각 슬롯에는 대상 속성을 나타내는 슬롯 값이 저장
 - ▶ 슬롯 값으로 다른 프레임 이름 사용 가능
 - ▶ 슬롯 값의 추가, 삭제, 변경 요구 등이 일어날 때 이루어져야 하는 절차(procedure)가 정의될 수 있음
- 모이면 상속을 수반하는 계층적 구조 및 기타 관계가 링크로 정의

1.1.2 프레임(Frame)



1.1.3 규칙(Rules)

■ 가장 널리 알려진 지식표현 방법 중의 하나

■ 생성규칙(Production Rule)

- IF ~ THEN ~

- ▶ IF 구문의 조건이 만족되면 THEN 구문이 수행 or 논리적으로 참

- 지식의 내용 분류

- ▶ 절차적 지식(상황-행동)

- IF 기온이 30도 이상이면, THEN 에어컨을 켜라.

- ▶ 가설의 설정(증거-가설)

- IF 성적이 우수한 학생이면, THEN 공부하는 것을 좋아할 것이다.

- ▶ 논리적인 암시(조건-결과)

- IF 그녀가 살아있으면, THEN 숨을 쉴 것이다.

- ▶ 인과관계(원인-결과)

- IF 자동차 수가 증가하면, THEN 공기 오염도가 높아진다.

1.1.3 규칙(Rules)

■ 패턴 매칭

- IF 문장의 만족여부를 판단하기 위해 단순히 패턴 매칭을 사용
- 같은 의미적 지식이라도 표현이 다르면 인식 불가
 - ▶ "밤을 세워 일하면" 을 "밤에 안자고 일하면" 으로 바꾸면 다른 규칙으로 인식

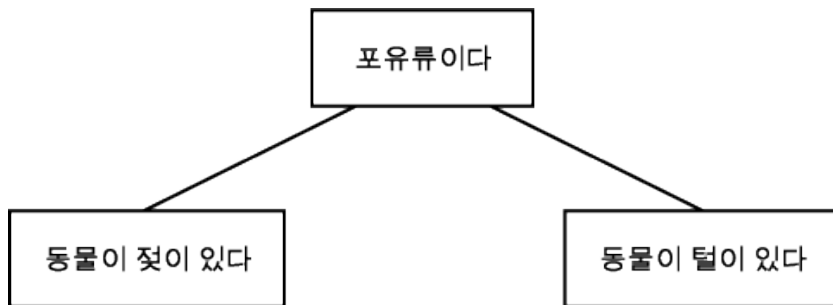
■ Boolean의 사용

- 문장을 보다 명확히 할 수 있게 함
- IF 송곳니가 있는 포유류이다 THEN 그것은 육식동물이다
→ IF 송곳니가 있다 AND 포유류이다 THEN 그것은 육식동물이다
- IF 동물이 깃털이 있거나 날면서 알을 낳으면 THEN 새이다
→ IF 동물이 깃털이 있다 OR (동물이 난다 AND 동물이 알을 낳는다)
THEN 새이다

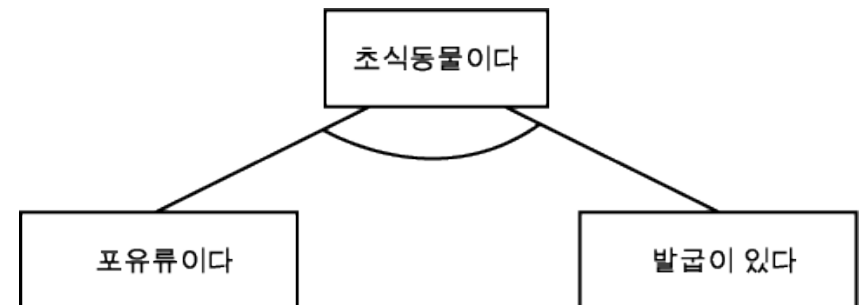
1.1.3 규칙(Rules)

■ 규칙의 AND/OR 그래프

- 규칙의 조건 및 결론 관계와 AND/OR 관계를 그래프 형태로 표현
- OR Graph
 - ▶ 같은 THEN 문을 갖는다는 것을 여러 규칙을 OR로 표현
 - ▶ IF 동물이 젖이 있다 THEN 포유류이다
IF 동물이 털이 있다 THEN 포유류이다
→ IF 동물이 젖이 있다 OR 동물이 털이 있다 THEN 포유류이다
- AND Graph
 - ▶ IF 포유류이다 AND 발굽이 있다 → 초식동물이다



< OR 그래프 표현 >

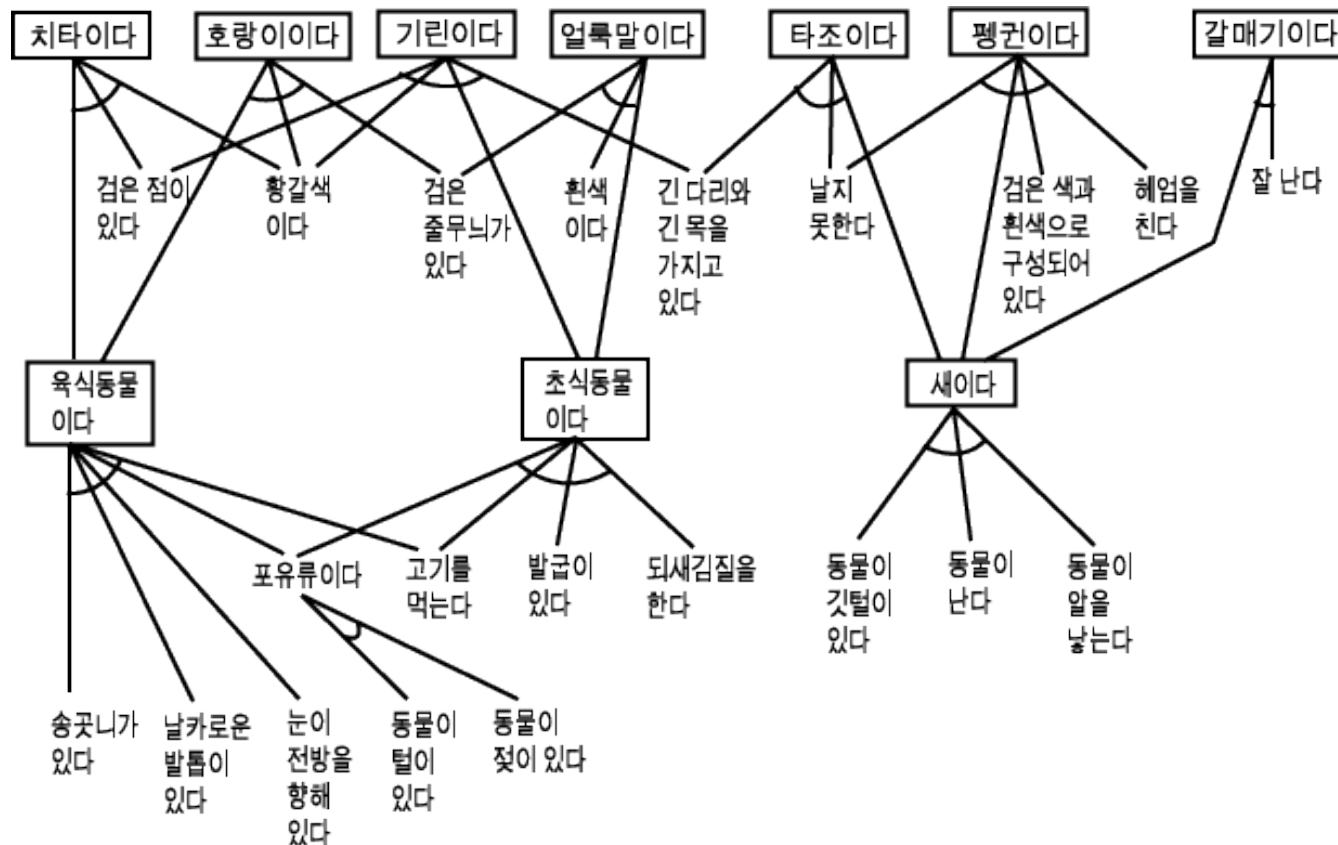


< AND 그래프 표현 >

1.1.3 규칙(Rules)

■ 규칙의 AND/OR 그래프

- 지식 전체의 논리적 구조를 파악하기 쉬움



1.1.4 일차논리(FOL : First Order Logic)

■ 일차논리(FOL) = 일차술어논리(First Order Predicate Logic)

- Quantifier (\forall or \exists)가 변수에만 붙고 '작곡하다'와 같은 술어에는 불허용
- '모든 과일은 건강에 좋다' & '사과는 과일이다'
 - ▶ $(\forall x)(\text{과일이다}(x) \rightarrow \text{건강에 좋다}(x))$
 - ▶ 과일이다(사과)
 - 첫 번째 판명문의 x 대신 사과를 대입하면 '과일이다(사과) \rightarrow 건강에 좋다(사과)' 도출
 - 두 번째 문장에 의해 전제가 만족
 - 긍정식(Modus Ponens) 추론방식에 의해 '건강에 좋다(사과)'는 참
- 장점: 표현력이 뛰어나 논리적인 내용의 대부분을 나타낼 수 있음
- 단점: 계산 속도가 느리고 결과를 예상할 수 없는 경우가 발생

1.1.4 일차논리(FOL : First Order Logic)

평상문	술어논리에 의한 표현
철수는 남자이다.	$\text{man}(\text{철수})$
철수는 대학생이다.	$\text{collegestudent}(\text{철수})$
모든 대학생은 학생이다.	$\forall x(\text{collegestudent}(x) \rightarrow \text{student}(x))$
민수는 배우이다.	$\text{actor}(\text{민수})$
모든 학생은 민수를 좋아하거나 싫어한다.	$\forall x(\text{student}(x) \rightarrow \text{like}(x, \text{민수}) \vee \text{dislike}(x, \text{민수}))$
모든 사람은 누군가를 좋아한다.	$\forall x \exists y(\text{like}(x, y))$
사람들은 그들이 좋아하지 않는 배우를 비난한다.	$\forall x \exists y(\text{person}(x) \wedge \text{actor}(y) \wedge \text{blame}(x, y) \rightarrow \sim \text{like}(x, y))$
철수는 민수를 비난한다.	$\text{blame}(\text{철수}, \text{민수})$
남자는 사람이다.	$\forall x(\text{man}(x) \rightarrow \text{person}(x))$

1.1.5 기술논리(DL : Description Logic)

- First Order Logic의 expression의 satisfiability는 uncomputable!
- Description Logic은 First Order Logic의 computable sublanguage
 - No functions
- Propositional Logic < Description Logic < First Order Logic
- Many AI Knowledge Representation Languages are based on DL
 - KL-One (1985), Krypton (1983), Loom (1987)

1.1.5 기술논리(DL : Description Logic)

- 용어적 지식(terminological knowledge) 을 주로 다루는 지식표현 체계
- 일차논리의 문제점 해결
 - 고급 계산 알고리즘으로 함축적인 관계를 미리 계산하여 빠른 응답 제공
 - 항상 답를 찾는 뛰어난 결정가능성 (computable)
 - ▶ 실용적인 비즈니스 솔루션에 최적의 표현력과 결정가능성 제공
- 기호(symbol) : 기본적인 개념을 표현
 - 기본 개념(atomic concept)
 - 개념들 간의 기본 역할(atomic role)
- 구성자(constructor) : 개념들을 연결
 - 논리학의 부울(Boolean) 연산자
 - 각종 제약을 표현하는 한정자

1.1.5 기술논리(DL : Description Logic)

■ 백만장자와 결혼한 여자로 2명 이상의 자식이 있으며 자식들은 모두 의사인 여자

- $\text{Human} \cap \text{Female} \cap \exists \text{married.Millionaire} \cap (\geq 2 \text{hasChild}) \cap \forall \text{hasChild.Doctor}$

구성자 (constructor)	선택스 (syntax)	시맨틱스 (semantics)	보충 설명
개념 이름 (concept name)	C	$\{x \mid C(x)\}$	'C(x)'는 'x가 C이다'라는 의미이다. 'C(x)'를 만족하는 x를 그냥 'C'로 표기한다. 예를 들어 'C'가 'Human' 이라면, 'Human인 x'를 뜻한다.
결합 (conjunction)	$C \cap D$	$\{x \mid C(x) \wedge D(x)\}$	' $C \cap D$ '는 'C이면서 동시에 D인 x'를 나타낸다.
값 제한 (value restriction)	$\forall R.C$	$\{x \mid \forall y, R(x,y) \rightarrow C(y)\}$	' $R(x,y)$ '는 '관계를 나타내는 R의 주어가 x이고 목적어가 y이다'를 나타낸다. 예를 들어 ' $\text{married}(x,y)$ '는 'x married y'라는 뜻이다. 또, \forall 는 '모든'을 ' \rightarrow '는 조건과 결과를 나타내는 기호이다. 그러므로 ' $\forall R.C$ '는 '모든 y에 대해 x가 y와 R의 관계에 있다면, 그 y는 C(y)를 만족하는 x'를 가리킨다. ' $\forall \text{hasChild.Doctor}$ '는 'x와 hasChild 관계에 있는 모든 y, 쉽게 말해 x의 모든 자식 y가 Doctor인 조건을 만족하는 x'를 뜻한다.
존재 정량 (existential quantification)	$\exists R.C$	$\{x \mid \exists y, R(x,y) \wedge C(y)\}$	\exists 는 '어떤'을 의미하는 기호이다. ' $\exists R.C$ '는 'x와 R의 관계에 있으면서 C(y)를 만족하는 어떤 y를 갖는 x'라는 뜻이다. ' $\exists \text{married.Millionaire}$ '는 x가 married한 어떤 y는 Millionaire인 x'를 가리킨다. 다시 말하면 'x가 결혼한 사람 중에 한 사람 이상은 백만장자인 조건을 만족하는 x'를 가리킨다.
수 제한 (number restriction)	$(\geq nR)$	$\{x \mid \{y \mid R(x,y)\} \geq n\}$	' $ \{y \mid R(x,y)\} $ '는 'x와 R의 관계에 있는 y의 개수'를 의미한다. 그러므로 ' $(\geq nR)$ '는 'x와 R의 관계에 있는 y의 개수가 n이상인 조건을 만족하는 x'라는 뜻이다. ' $(\geq 2 \text{hasChild})$ '는 'x와 hasChild 관계에 있는 y, 즉 x가 가진 자식 y의 수가 2 이상인 조건을 만족하는 x'를 나타낸다.

1.1.5 기술논리(DL : Description Logic)

■ 용어 공리(terminological axiom)

- 기본 개념을 이용하여 더욱 복잡한 개념을 만들고 이것에 이름을 부여
 - ▶ $\text{HappyWoman} \equiv \text{Human} \cap \text{Female} \cap \exists \text{married.Millionaire} \cap (\geq 2 \text{hasChild}) \cap \forall \text{hasChild.Doctor}$

■ 선언적 공리(assertional axiom)

- 특정 개체의 속성을 설명
 - ▶ $\text{HappyWoman}(\text{영희})$
 $\text{hasChild}(\text{영희}, \text{철수})$

■ Tbox (Terminology Box)

- 기본 개념, 기본 역할 및 구성자, 용어적 공리 형식으로 용어(terminology)에 관한 지식을 저장하는 지식베이스

■ Abox (Assertion Box)

- 특정 개념에 속하는 인스턴스들에 대한 속성을 설명하는 선언적(assertional) 지식을 저장하는 지식베이스

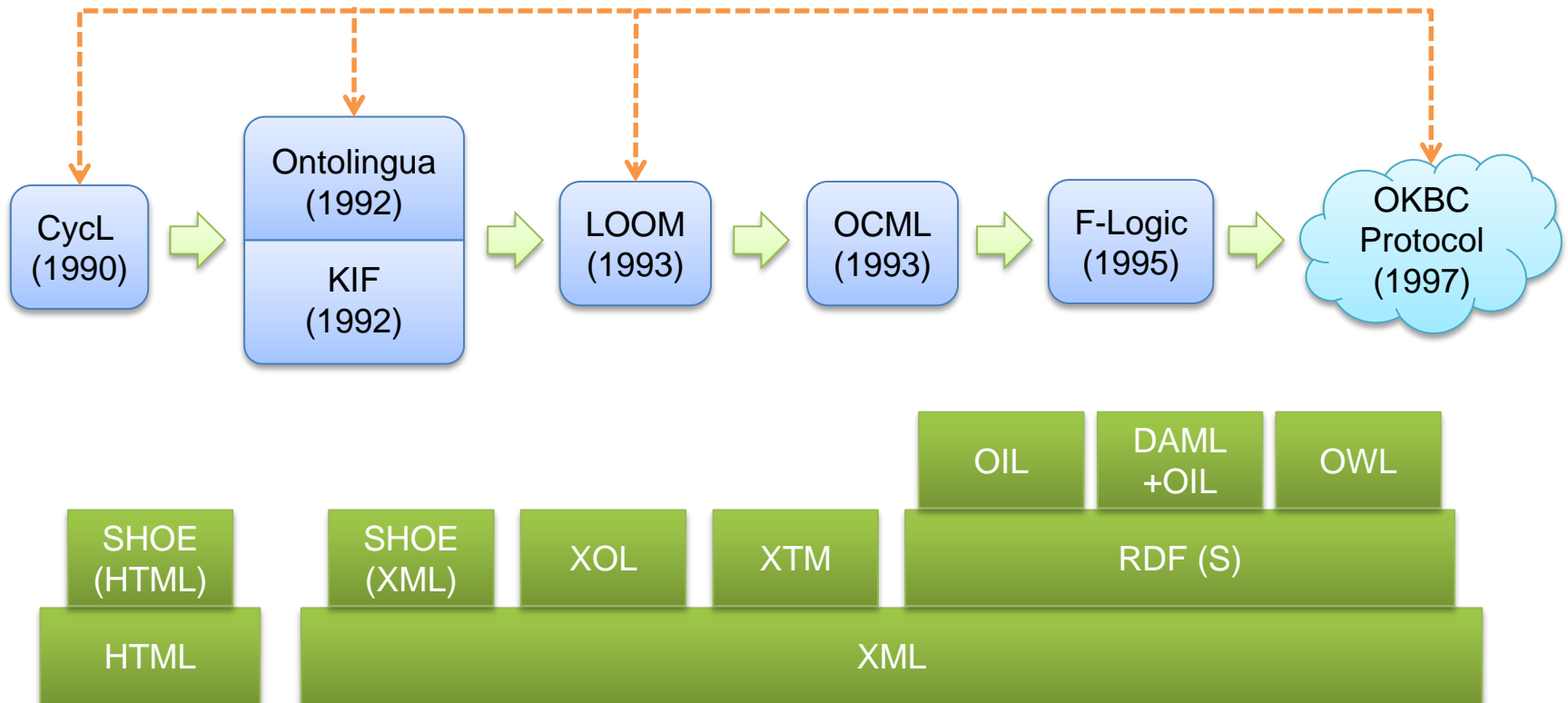
1.1.5 기술논리(DL : Description Logic)

■ 추론 기능 (inference)

- 지식베이스에 저장된 도메인 지식들을 활용할 수 있도록 지원
- 포함 확인(subsumption check) : 개념의 계층구조 결정
- 인스턴스 확인(instance check) : 임의 개체가 어떤 개념의 인스턴스인지 확인
- 일관성 확인(consistency check) : 지식베이스에 모순이 있는지 검사

1.2 온톨로지 언어의 역사 [1/7]

- 1990년대 전반 : AI에 기반한 온톨로지 언어
- 1990년대 후반 : WEB 마크업(markup) 언어에 기반한 온톨로지 언어



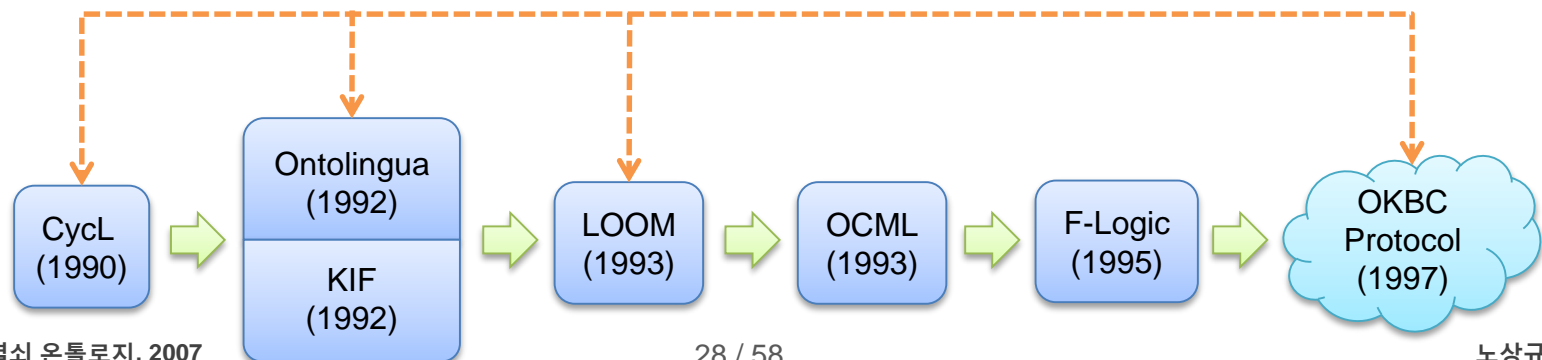
1.2 온톨로지 언어의 역사 [2/7]

- 1990년대 초반의 AI에 기반한 온톨로지 언어의 흐름
 - CycL 개발(1990)
 - ▶ 사이크(Cyc) 온톨로지 구축에 사용
 - KIF(knowledge Interchange Format) 개발(1992)
 - ▶ 다양한 정보시스템 간의 지식교환 목표
 - 프레임 온톨로지(Frame Ontology) 개발
 - ▶ 스탠포드 대학교의 KSL(knowledge Systems Laboratory)이 개발
 - ▶ KIF는 직접적으로 온톨로지를 구축하는 것이 어려움
 - Ontolingua 개발(1992)
 - ▶ KIF와 프레임 온톨로지 기반
 - ▶ Ontolingua 서버에 의해 지원
 - ▶ 1990년대 사실상의 표준(by 온톨로지 커뮤니티)
 - LOOM 개발(1993)
 - ▶ 온톨로지 구축보다는 일반적인 지식베이스를 위함

1.2 온톨로지 언어의 역사 [3/7]

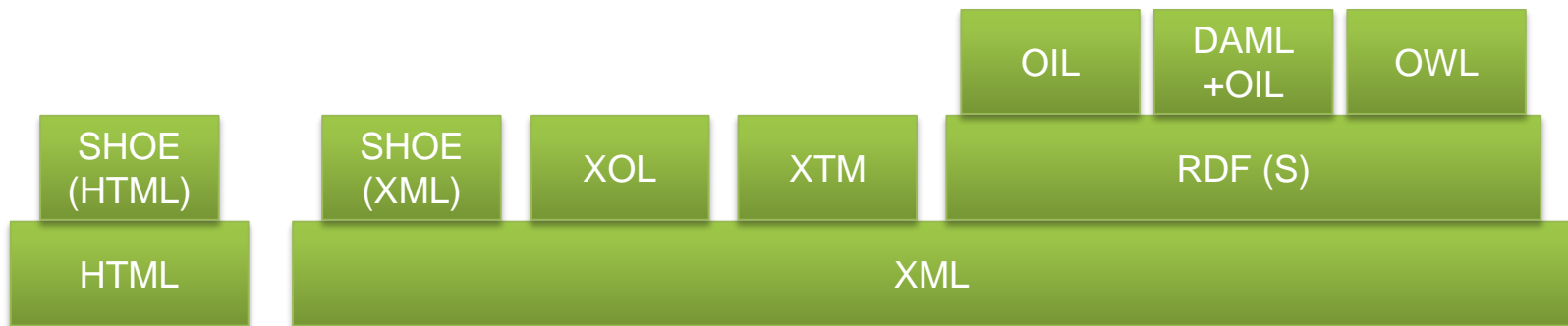
■ 1990년대 초반의 AI에 기반한 온톨로지 언어의 흐름

- OCML(Operational Conceptual Modeling Language) 개발(1993)
 - ▶ 실행 가능한 온톨로지와 문제해결을 위한 모델링 제공
 - ▶ 대부분의 정의는 Ontolingua와 유사
- F-Logic(Frame Logic) 개발(1995)
 - ▶ 프레임과 일차논리를 결합
- OKBC(Open Knowledge Base Connectivity) 프로토콜 개발(1997)
 - ▶ 대규모 지식베이스의 문제점 해결 목표
 - ▶ DARPA(Defense Advanced Research Projects Agency) 후원
 - ▶ 서로 다른 언어로 구현된 지식베이스에 액세스하는 프로토콜
 - ▶ CycL, Ontolingua, LOOM 지원



1.2 온톨로지 언어의 역사 [4/7]

- 1990년대 후반의 WEB기반 온톨로지 마크업 언어의 흐름
 - 웹의 특성을 최대한 이용하는 '웹 기반 온톨로지 언어'
 - 기존 마크업 언어(HTML, XML)을 기반



1.2 온톨로지 언어의 역사 [5/7]

■ 1990년대 후반의 WEB기반 온톨로지 마크업 언어의 흐름

- SHOE(Simple HTML Ontology Extension) (1996)
 - ▶ 최초의 온톨로지 마크업 언어
 - ▶ 프레임과 규칙을 결합한 언어
 - ▶ HTML을 확장: 온톨로지를 표현할 수 있는 특수 태그 제공
 - ▶ XML과의 호환을 위해 변형된 SHOE Syntax가 후에 개발
- XOL(XML-based Ontology exchange Language)
 - ▶ OKBC-Lite의 어휘 일부분을 XML화하여 개발
- RDF(Resource Description Framework)
 - ▶ 시맨틱 네트워크 기반 언어
 - ▶ 1999년 W3C 권고안

1.2 온톨로지 언어의 역사 [6/7]

■ 1990년대 후반의 WEB기반 온톨로지 마크업 언어의 흐름

● RDF Schema

- ▶ RDF를 프레임 기반 어휘들로 확장
- ▶ 2004년 W3C 권고안
- ▶ RDF(S) : RDF + RDF Schema
 - 확장으로 OIL, DAML+OIL, OWL 발표

● OIL(Ontology Inference Layer/Ontology Interchange Language) (2000)

- ▶ 유럽 OnTo-Knowledge 프로젝트에 의해 개발
- ▶ RDF(S)에 프레임 기반 지식표현 어휘 추가
- ▶ 시맨틱스는 기술논리에 기반

● DAML(DARPA Agent Markup Language)+OIL

- ▶ 미국과 유럽의 공동노력으로 개발
- ▶ DAML-ONT와 OIL을 기반
- ▶ RDF(S)에 기술논리기반 지식표현 어휘를 추가

1.2 온톨로지 언어의 역사 [7/7]

■ 1990년대 후반의 WEB기반 온톨로지 마크업 언어의 흐름

● OWL(Ontology Web Language)

- ▶ W3C의 Web Ontology Working Group 주도
- ▶ RDF의 부족한 표현력을 포함하려는 노력
- ▶ DAML+OIL의 특징을 대부분 포함
- ▶ 2004년 W3C 권고안

● 토픽맵 (Topic Map)

- ▶ 대용량 지식 정보를 의미적 연관성에 따라 토픽별 구조화
- ▶ 2000년 온톨로지 구축에 대한 국제 표준(ISO/IEC 13250)으로 채택
- ▶ 표준 스펙
 - HyTM(HyTime Topic Maps) : SGML(standard Generalized Markup Language) 기반
 - XTM(XML Topic Maps) : Topicmaps.org에서 발표된 실질적 토픽맵표준

1.2 온톨로지 역사

대분류	온톨로지 언어	개발기관	발표 시기	지식표현 패러다임	설명
인공지능기반 온톨로지언어	CycL	Cyc	1990	Frames, FOL	<ul style="list-style-type: none"> • Cyc 온톨로지 구축에 사용됨 • OKBC와 호환
	KIF	ARPA	1992	FOL	<ul style="list-style-type: none"> • 지식표현언어의 이질성 문제를 해결하고 다양한 정보 시스템 간에 지식교환을 가능하게 하는 것을 목표로 개발 • 직접적으로 온톨로지를 표현하기는 어려움
	Ontolingua	스탠포드대 KSL	1992	Frames, FOL	<ul style="list-style-type: none"> • 온톨로지 표현을 위해 KIF 위에 올려짐 • Ontolingua 서버에 의해 지원되는 온톨로지 언어 • LISP과 비슷한 문법을 가지고 있음 • 1990suseo 사실상의 온톨로지 표준으로 간주됨 • OKBC와 호환
	LOOM	USC ISI	1993	DL	<ul style="list-style-type: none"> • 온톨로지 구축이 아닌 일반적인 지식베이스를 위해 개발 • 생성규칙 및 자동 개념분류 기능 등을 지원 • OKBC와 호환
	OCML	영국 KMI	1993	Frames, FOL	<ul style="list-style-type: none"> • 실행가능한 온톨로지와 문제해결을 위한 모델링을 제공하는 '운영(operational) Ontolingua'로 개발 • 대부분의 정의가 Ontolingua와 비슷
	F-Logic	뉴욕주립대 컴퓨터학과	1995	Frames, FOL	<ul style="list-style-type: none"> • 일차논리에 대한 객체지향 접근법으로 개발 • 객체지향 연역 데이터베이스를 위해 사용됨 • 온톨로지 구현을 위해 수정된 후 이용됨 • LISP과 비슷한 문법을 아님

1.2 온톨로지 역사

대분류	온톨로지 언어	개발기관	발표 시기	지식표현 패러다임	설명
온톨로지 마크업언어	SHOE	메릴랜드대	1996	Frames, Rules	<ul style="list-style-type: none"> 최초의 온톨로지 마크업 언어 HTML의 확장으로 개발 HTML과 다른 태그를 사용하여 HTML 문서에 온톨로지를 삽입할 수 있도록 했으며 SHOE문법은 후에 XML에 맞춰짐 프레임과 규칙이 통합된 상태
	XOL	Pangea Systems Inc. & SRI International	1999	Frames	<ul style="list-style-type: none"> 온톨로지 구축이 아닌, 서로 다른 시스템 사이에서 온톨로지를 교환하기 위한 매개(intermediate) 언어로 개발됨 OKBC-Lite(OKBC 프로토콜의 어휘 일부분)를 XML화 하여 개발 XOL 온톨로지를 위한 전문 툴이 없음
	RDF	W3C	1999 (권고안)	Semantic Nets	<ul style="list-style-type: none"> 주로 웹 자원을 기술하는 외부 메타데이터를 트리플 구조로 표현 이미지, 오디오, 비디오 파일 같은 자원을 기술하는데 특히 유용
	RDF Schema	W3C	2004 (권고안)	Semantic Nets, Frames	<ul style="list-style-type: none"> 객체지향 프로그래밍 언어의 데이터모델과 유사 객체지향과의 차이점 : 메소드(method)가 속성과 합쳐진 형태, 상향식의 클래스 모델링, 속성의 계층구조를 지원
	OIL	유럽 On-To-Knowledge	2000	Semantic Nets, Frames, DL	<ul style="list-style-type: none"> 웹 자원의 의미를 표현하는 웹기반 지식표현 언어 XML 문법, 프레임 기반 모델링 어휘, DL의 형식적 시맨틱스와 추론지원을 결합

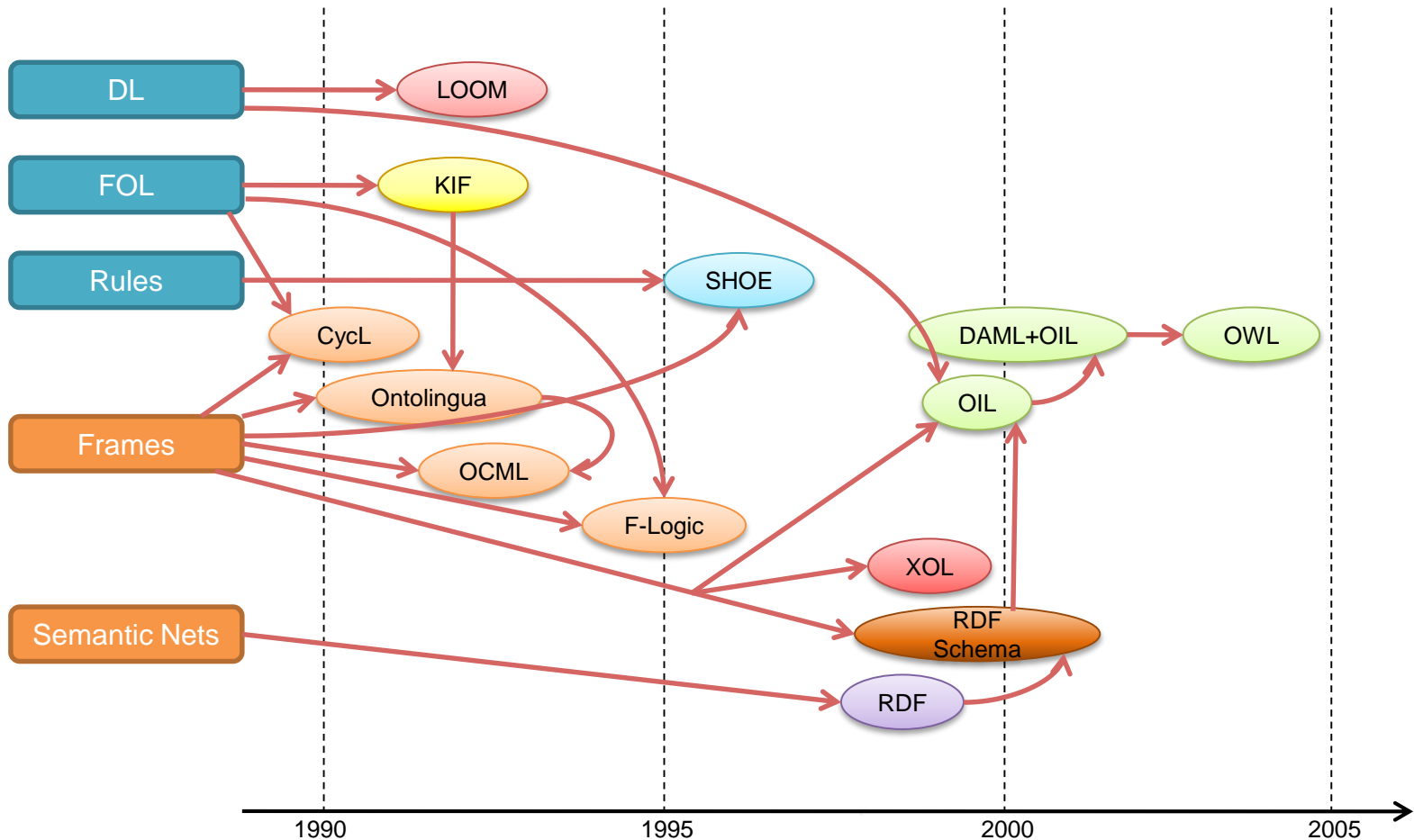
1.2 온톨로지 역사

대분류	온톨로지 언어	개발기관	발표 시기	지식표현 패러다임	설명
온톨로지 마크업언어	DAML+OIL	US & EU Joint Committee	2000	Semantic Nets, Frames, DL	<ul style="list-style-type: none"> • DAML+OIL 온톨로지는 XML이나 RDF 트리플 표기로 작성될 수 있음 • RDF(S)에 프레임과 기술논리기반 지식표현 어휘가 추가됨
	OWL	W3C	2004 (권고안)	Semantic Nets, Frames, DL	<ul style="list-style-type: none"> • 표현력이 가장 뛰어난 시맨틱 웹 온톨로지 언어 • OWL Lite, OWL DL, OWL Full 순으로 표현력이 증가함
	XTM	Topicmaps. Org	2001	Semantic Nets	<ul style="list-style-type: none"> • 서로 연결된 토픽의 형태로 웹 문서의 내용에 따른 인덱스를 제공 • 토픽, 관계, 어커런스, 주제 지시자, 범위 등의 개념으로 이루어짐

1.2 온톨로지 역사

■ 지식표현 패러다임의 영향과 시간에 따른 온톨로지 언어의 발전

- 타원형의 바탕이 같은 것은 기초가 되는 지식표현 패러다임이 같은 언어들



Chapter 5 온톨로지 언어

- 1. 온톨로지 언어의 발전 과정
- 2. 인공지능 기반의 온톨로지 언어
- 3. 온톨로지 마크업 언어

2. 인공지능 기반의 온톨로지 언어

- 2.1 KIF(Knowledge Interchange Format)와 Ontolingua
- 2.2 LOOM
- 2.3 OCML(Operational Conceptual Modeling Language)
- 2.4 F-Logic(Frame Logic)
- 2.5 OKBC(Open Knowledge Base Connectivity)

2.1 KIF(Knowledge Interchange Format)와 Ontolingua

■ KIF(Knowledge Interchange Format) (1992)

- ARPA(Advanced Research Projects Agency)가 후원하는 지식공유노력 (Knowledge Sharing Effort) 프로젝트의 일환
- 목표
 - ▶ 지식표현 언어의 이질성 문제를 해결
 - ▶ 다양한 정보시스템 간의 지식 교환 가능
- 특징
 - ▶ 교환 형식이므로 온톨로지를 쉽게 다룰 수 있음
 - ▶ KIF 위에 구축된 프레임 온톨로지를 사용하면 작업이 더욱 쉬워짐

■ 프레임 온톨로지

- 프레임 기반 접근법으로 온톨로지 모델링
- 클래스, 이진 관계, 명명된 공리 등에 관한 어휘 제공
- KIF와 KIF의 확장 시리즈를 기반으로 개발
 - ▶ Ontolingua 서버의 번역기에 의해 KIF로 완벽하게 변환 가능
- OKBC 온톨로지가 KIF와 프레임 온톨로지 사이에 추가되면서 수정됨

2.1 KIF(Knowledge Interchange Format)와 Ontolingua

■ Ontolingua (1992)

- Stanford의 KSL(Knowledge Systems Laboratory)에서 발표
- 구축 방법
 - ▶ 프레임 온톨로지 어휘 사용
 - ▶ OKBC 온톨로지 어휘 사용
 - ▶ KIF 표현을 사용
 - ▶ 프레임 온톨로지 어휘, OKBC 온톨로지 어휘, KIF 표현을 동시에 복합적으로 사용
- Ontolingua 서버에 보관
 - ▶ 제공되는 온톨로지 에디터를 이용해서 구현 가능
 - ▶ 이미 만들어진 다른 Ontolingua 기반 온톨로지 불러들여 변환 가능
 - KIF
 - OKBC
 - LOOM
 - Prolog(Programming in Logic)
 - CORBA(Common Object Request Broker Architecture)
 - IDL(interface Definition Language)

2.2 LOOM

- 남가주대학교(University of Southern California)에 의해 개발
- 기술논리 패러다임에 기반
- 목적
 - 온톨로지의 구현보다는 일반적인 용도의 전문가 시스템이나 다른 지능 어플리케이션의 구축 환경 제공
- 구성
 - 기술언어(description language)
 - ▶ 객체와 관계로 도메인 모델을 설명
 - 선언언어(assertion language)
 - ▶ 개체에 대한 사실을 명시
- 규칙과 술어논리, 패턴지향법에 의한 절차적 프로그래밍 지원

2.3 OCML(Operational Conceptual Modeling Language)

■ 영국 KMI(Knowledge Media Institute)에 의해 개발

■ 특징

- Ontolingua와의 호환성을 개발단계에서부터 고려
- 정리증명과 함수평가 기능을 제공하는 '운영 Ontolingua'
- LISP과 비슷한 문법을 가진 프레임 기반 언어
- 후방/전방 연결(backward/forward chaining)을 포함하는 규칙과 절차를 정의하는 어휘 제공

■ 사용 분야

- 지식 경영, 온톨로지 구축, 전자상거래, 지식기반 시스템의 개발

2.4 F-Logic(Frame Logic)

- 뉴욕주립 대학교(State University of New York) 스톤 브룩(Stony Brook) 캠퍼스의 컴퓨터과학과에서 개발
- 객체지향 접근법
 - 초기에 객체지향 연역 데이터베이스를 위해 사용
 - ▶ 이후에 온톨로지 구현을 위해 수정되어 이용
- 응용분야
 - 객체지향 데이터베이스
 - 인공지능을 위한 프레임 기반 언어 제공
 - ▶ 인공지능 언어는 복잡한 객체, 상속, 연역법 등의 개념을 중심으로 구축
 - ▶ 프레임 대신 객체, 슬롯 대신 애트리뷰트(attribute)라는 용어 사용

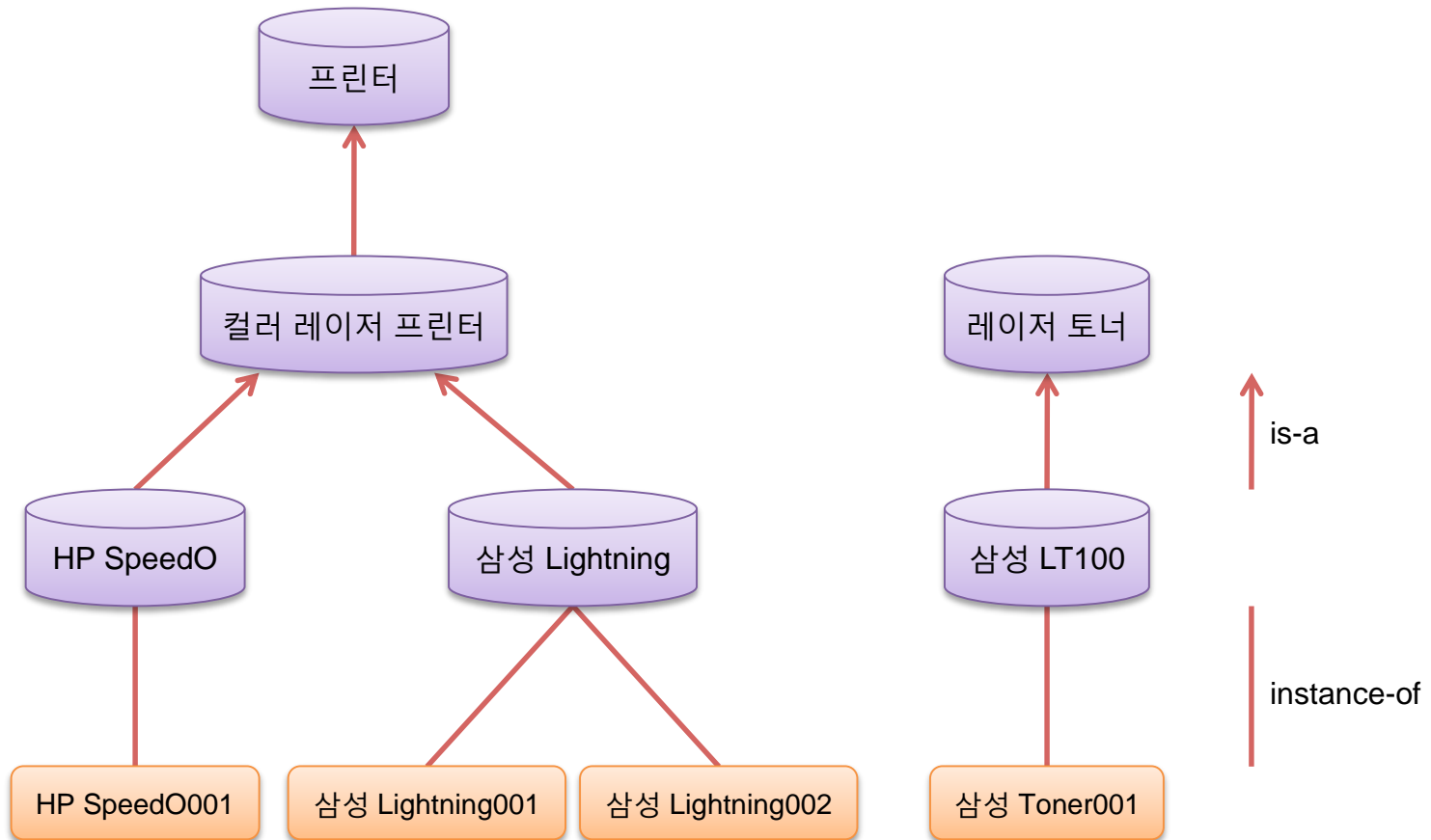
2.4 F-Logic(Frame Logic)

■ 특징

- 객체 정체성(object identity)
- 복잡한 객체(complex objects)
- 상속(inheritance)
- 다형적 유형(polymorphic types)
- 질의 방법(query methods)
- 캡슐화(encapsulation)
- 하나의 메소드에 대한 값으로 여러 개의 값을 집합 형태로 취할 수 있음
- 확장성이 뛰어남
 - ▶ 지식 추론 형식을 통합하는 중심적인 역할
- 온톨로지 개발 툴에 의해 구축 가능
 - ▶ Protégé, OntoEdit, WebODE 등

2.4 F-Logic(Frame Logic)

■ F-Logic 코딩 예를 위한 클래스 구성



2.4 F-Logic(Frame Logic)

■ F-Logic 코딩 예

온톨로지(Ontology)

```
1  ColorLaserPrinter :: Printer[substitute ==> ColorLaserPrinter].
2  HPspeedO :: ColorLaserPrinter
3  SamsungLightning :: ColorLaserPrinter[complement ==> SamsungLPT100].
4  SamsungLPT100 :: LaserToner.
5  FORALL X, Y  Y[substitute ->> X]  <-  X-substitute ->> Y].
```

지식베이스(Knowledge Base)

```
6  samsungToner001 : SamsungLPT100.
7  hpSpeedO001 : HPspeedO[substitute ->> {SamsungLightning001, SamsungLightning002}].
8  samsungLightning001 : SamsungLightning[complement -> samsungToner001.]
9  samsungLightning002 : SamsungLightning[complement -> samsungToner001.]
```

질의(Query)

```
10 FORALL X <- hpSpeedO001[substitute ->> X].
11 FORALL Y, Z <- hpSpeedO001[substitute ->> Y] ^ Y : SamsungLightning [complement -> z].
```

2.4 F-Logic(Frame Logic)

■ 전체 코드 이해

● 라인 1

- ▶ 컬러 레이저 프린터는 프린터의 하위 클래스
- ▶ 대체하다(substitute)라는 메소드를 가지며 이것은 여러 개의 컬러 레이저 프린터 인스턴스를 취할 수 있음

● 라인 3

- ▶ 삼성 Lightning이 컬러 레이저 프린터의 하위 클래스
- ▶ 보완하다(complement)라는 메소드를 가지며 이것은 하나의 삼성 LPT100 인스턴스를 취할 수 있음

● 라인 5

- ▶ 대체하다라는 메소드가 대칭적(symmetric)이라는 것을 표현

● 라인 7

- ▶ hpSpeedO001이 HPSpeedO의 인스턴스
- ▶ samsungLightning001과 samsungLightning002를 대체

● 라인 8

- ▶ Samsunglightning001이 SamsungLightning의 인스턴스
- ▶ samsungToner001을 보완

2.4 F-Logic(Frame Logic)

■ 전체 코드 이해

● 라인 10

- ▶ hpSpeedO001이 대체하는(substitute) 모든 인스턴스를 조회 질의
 - samsungLightning001, samsungLightning002

● 라인 11

- ▶ hpSpeedO001을 대체하는(substitute) 삼성 프린터인 삼성 Lightning 인스턴스와 이 프린터에 맞는, 즉 이 프린터와 보완관계(complement)에 있는 토너를 쌍으로 검색하는 질의
 - (samsungLightning001, samsungToner001),
(samsungLightning002, samsungToner001)

2.5 OKBC(Open Knowledge Base Connectivity)

■ SRI International의 AI센터와 Stanford의 KSL의 공동노력으로 개발

■ 특징

- 서로 다른 지식표현 시스템에 접근가능한 프레임 기반 프로토콜
- LISP와 비슷한 문법을 사용
 - ▶ 복잡한 오퍼레이션을 기술하는 절차 정의

■ 호환되는 언어와 시스템

- | | |
|--------------|------------|
| ● Ontolingua | ● Ocelot |
| ● LOOM | ● JTP |
| ● CycL | ● CLOS |
| ● Protégé | ● Tuple-KB |
| ● Theo | ● WebODE |
| ● SIPE-2 | |

Chapter 5 온톨로지 언어

- 1. 온톨로지 언어의 발전 과정
- 2. 인공지능 기반의 온톨로지 언어
- 3. 온톨로지 마크업 언어

3. 온톨로지 마크업 언어

- 3.1 SHOE(Simple HTML Ontology Extension)
- 3.2 XOL(XML-based Ontology exchange Language)
- 3.3 RDF(Resource Description Framework)와 RDF Schema
- 3.4 OIL(Ontology Inference Layer)
- 3.5 DAML+OIL
(DARPA Agent Markup Language+Ontology Inference Layer)
- 3.6 OWL(Ontology Web Language)
- 3.7 XTM(XML Topic Maps)

3.1 SHOE(Simple HTML Ontology Extension)

■ 메릴랜드 대학교(University of Maryland)에서 개발

■ 목표

- HTML을 확장하여 웹 문서에 컴퓨터가 이해할 수 있는 시맨틱 지식을 삽입
 - ▶ HTML에서는 정의되어 있지 않은 특별 태그를 제공
 - ▶ 표준 웹 브라우저에서는 보이지 않음
 - ▶ 이후에 XML과 호환되도록 수정
- 웹 검색 메커니즘의 향상을 위해 웹 페이지와 문서에 대한 의미 정보를 수집

■ 용도


- 개념과 이들 간의 관계를 기술하는 온톨로지를 정의
- HTML 페이지에 주석 달기
- 에이전트가 SHOE 주석이 있는 웹 페이지를 찾아 정보를 업데이트하고 시맨틱 정보를 검색하도록 지원

Simple HTML Ontology Extension (SHOE)

Department of Computer Science - Microsoft Internet Explorer

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(I) 도움말(H)

주소(D) http://www.cs.umd.edu/ 이동 연결



DEPARTMENT OF COMPUTER SCIENCE

[Public home page](#) [Local home page](#) [How to contact us](#) [Search](#)

Accomplishments

- [Alumni hall of fame](#)
- [Major alumni awards](#)
- [Major faculty awards](#)

Education

- [Graduate program](#)
- [Undergraduate program](#)
- [Undergrad. research](#)
- [Undergrad. honors prog.](#)
- [Class web pages](#)
- [Class discussion forums](#)
- [University schedule](#)

Location

- [A.V. Williams Building](#)
- [CS Instructional Center](#)
- [Directions](#)
- [How to contact us](#)

People

- [Faculty](#)
- [Faculty/staff phone book](#)
- [Users' web pages](#)

Information:

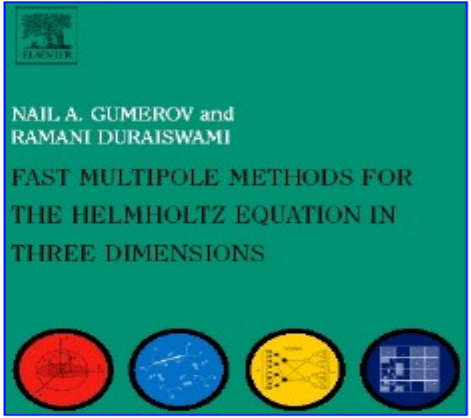
- ▶ [for prospective undergraduates](#)
- ▶ [for prospective graduate students](#)
- ▶ [for prospective faculty](#)
- ▶ [for prospective industrial partners](#)

All scheduled events

- ▶ [Summary](#) ▶ [Details](#)

CS/UMIACS distinguished lecture series

- ▶ [Schedule and abstracts](#)
- ▶ [Webcasts](#)



Announcements:

In the 2006 *US News and World Report* rankings of colleges and universities, here's where we placed:

	Among all schools	Among public schools
Entire department	13	6
Artificial intelligence	9	5

SHOE Ontology

■ CS-DEPT-Ontology

Categories

```
Person*
  Worker*
    Faculty*
      Professor*
        AssistantProfessor*
        AssociateProfessor*
        FullProfessor*
        VisitingProfessor*

      Lecturer*
      PostDoc*
      Assistant*
        ResearchAssistant*
        TeachingAssistant*
      AdministrativeStaff*
        Director*
        Chair* {Professor*}
        Dean* {Professor*}
        ClericalStaff*
        SystemsStaff*
      Student*
        UndergraduateStudent*
        GraduateStudent*
  Organization*
    EducationOrganization*
      Department*
      School*
      University*
        Program*
        ResearchGroup*
        Institute*
```

Relationships

```
advisor(Student, Professor)*
affiliateOf(Organization, Person)*
affiliatedOrganization(Organization, Organization)*
alumnus(Organization, Person)*
containedIn(Document, Document)*
doctoralDegreeFrom(Person, University)*
emailAddress(Person, .STRING)*
head(Organization, Person)*
listedCourse(Schedule, Course)*
mastersDegreeFrom(Person, University)*
member(SocialGroup, Person)*
name(base.SHOENity, .STRING)*
offers(University, Course)*
```

SHOE Semantic Search:

Find papers by assistant professors?

Who are in “Future Web” research group?

File Help

Ontology: cs-dept-ontology, v.1.0 Query

Select a category:

AssociateProfessor
Chair
Dean
FullProfessor
VisitingProfessor
Publication
Article
ConferencePaper
JournalArticle
TechnicalReport

Select

Subject Category: Article

publicationResearch Simple HTML Ontology Extensions
publicationAuthor Heflin
publicationDate
name
softwareDocumentation OF
orgPublication OF

First Item: 1 Last Item: 3

Get Last Group Get Next Group

Article	publicationResearch(name)	publicationAuthor(name)	name
http://www	Simple HTML Ontology Extensions	Jeff Heflin	Coping with Changing Ontologies in a
http://www	Simple HTML Ontology Extensions	Jeff Heflin	Applying Ontology to the Web: A Case
http://www	Simple HTML Ontology Extensions	Jeff Heflin	Reading Between the Lines: Using SH

Web Search Get Page Add To Query 3 answers.

Ontology Example – Human Resource

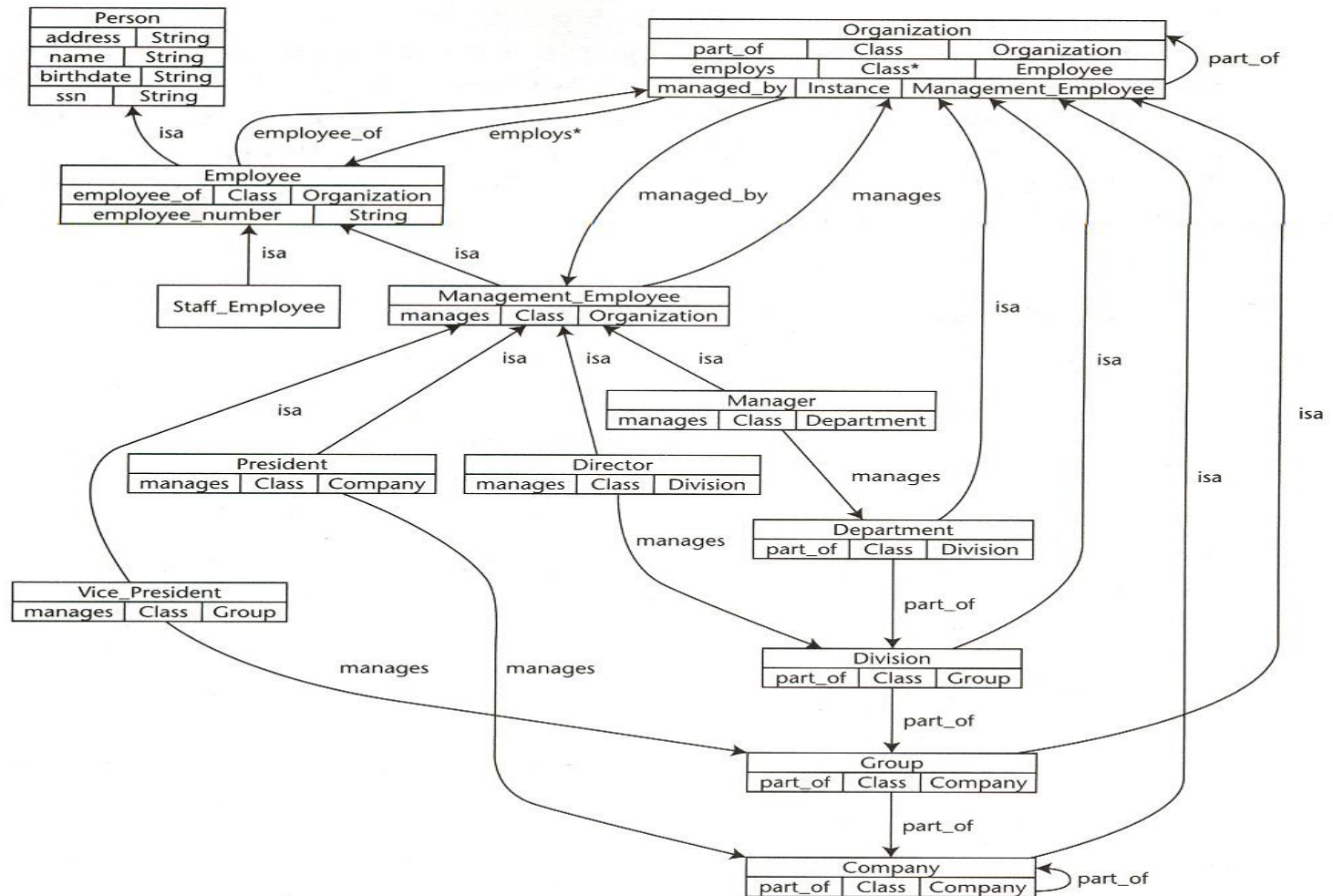


Figure 8.1 Graphical ontology example: Human resources.

3.2 XOL(XML-based Ontology exchange Language)

■ Pangea Systems Inc.와 SRI International의 AI센터에 의해 개발

■ 목적

- 이질적인 소프트웨어 시스템 사이에 온톨로지를 교환하기 위한 형식 제공
 - ▶ 온톨로지 개발보다는 서로 다른 데이터베이스 시스템, 온톨로지 개발 툴, 응용 프로그램 사이에 온톨로지의 이동을 위한 중간언어로서 개발

■ XML로 작성

■ 작성

- 전문 툴이 없음
- 일반 텍스트 에디터 / XML 에디터를 사용

3.3 RDF(Resource Description Framework)와 RDF Schema

■ RDF

- W3C에 의해 개발 (1999년 W3C 권고안)
- 웹 자원을 기술하는 XML 기반 언어
- 목적
 - ▶ 메타데이터들을 공통적인 규칙에 따라 컴퓨터가 이해할 수 있는 정보로 표현
- 표현 : 서술문(statements) 기본 단위
 - ▶ 주어부(subject)
 - ▶ 서술부(predicate or property)
 - ▶ 목적부(object)
- 이용분야
 - ▶ 자원 검색, 분류
 - ▶ 지능형 소프트웨어 에이전트
 - ▶ 내용 순위 부여 및 평가
 - ▶ 디지털 서명
 - ▶ 지적 재산권 보호

3.3 RDF(Resource Description Framework)와 RDF Schema

■ RDF Schema

- RDF 표현의 세부적인 의미와 속성에 대한 정의역(domain)과 공역(range) 및 제약조건(constraints) 등을 기술
- 클래스와 속성에 대한 기본적인 어휘로 표현
- 클래스의 계층구조 지원 (객체지향 모델링과 유사)
- 객체지향 모델링은 하향식(top-down), RDF Schema는 상향식(bottom-up)
- 속성(property)은 객체지향 모델링에서의 애트리뷰트와 메소드를 포함
- 속성도 계층구조를 가짐
 - ▶ ' ~의 아버지이다 ' 라는 속성은 ' ~의 부모이다 ' 라는 속성의 하위 속성
- 한계: 정교하고 복잡한 지식 표현 어려움
 - Transitivity, Cardinality restriction, Disjoint classes 등등 표현이 불가능

3.4 OIL(Ontology Inference Layer)

- 유럽 IST(Information Society Technologies) 프로젝트인 On-To-Knowledge의 일환으로 개발
 - RDF(S)에 프레임 기반 지식표현 어휘를 추가
 - RDF의 구체화 메커니즘을 제거하여 계층 접근법으로 개발
- 정의
 - XML 문법, 프레임 기반 지식표현 패러다임의 모델링 어휘, 기술논리의 형식적 시맨틱스(formal semantics)와 추론 지원 기능을 결합해서 만든 웹 기반 지식 표현 언어
- 온톨로지 툴로 구축 가능

3.5 DAML(DARPA Agent Markup Language)+OIL

■ 개발

- DARPA DAML(DARPA Agent Markup Language) 프로젝트의 일부로 개발
- DAML+OIL
 - ▶ DAML 초기 미국 버전이 나오면서 유럽 연합의 OIL 연구자들과 서로 노력
 - ▶ 두 언어가 통합
 - ▶ W3C 웹 표준 컨소시엄에 의해 후원
 - DAML-ONT : DAML+OIL의 초기버전
 - RDF(S)에 프레임 기반 지식표현 어휘를 추가 확장
 - 여기에 지식표현 패러다임이 프레임에서 기술논리로 확장 → DAML+OIL 최종버전
- DAML-S (DAML-Service)
 - ▶ DAML+OIL로 표현된 온톨로지 집합
 - ▶ 프로세스, 리소스, 서비스 프로파일, 서비스 모델 등 웹서비스의 시맨틱스 설명

■ XML이나 RDF 트리플(triple)로 작성 가능

3.6 OWL(Ontology Web Language)

■ W3C의 웹 온톨로지 워킹 그룹이 개발

■ 특징

- 표현력이 가장 뛰어난 시맨틱 웹 온톨로지 언어
- 클래스와 하위 클래스, 속성과 하위 속성, 속성 제한, 클래스와 속성 개체
 - ▶ 대부분의 DAML+OIL 어휘를 이해하기 쉽게 수정
- 3개 계층 언어로 구성
 - ▶ OWL Lite, OWL DL(Description Logic), OWL Full
 - ▶ 더 높은 계층 언어는 더 낮은 계층 언어를 포함

3.7 XTM(XML Topic Maps)

■ ISO/IEC 13250 국제 표준으로 채택된 온톨로지 구축 모델

■ 목적

- 지식정보를 의미적 상호 연관성에 따라 연결하고 체계화한 지식정보 구조로 표현
- 대용량의 지식정보를 효율적으로 검색하고 관리해 줄 수 있는 해결책

■ 표준 명세

- 처음 SGML(Standard Generalized Markup Language) 구조와 HyTM
- 현재는 XTM 1.0과 HyTM

■ 특징

- 웹 문서와 문서의 섹션을 내용(토픽/주제)에 따라 의미적으로 특징화하고 분류
- 책의 인덱스처럼 내용에 근거한 인덱스를 제공
 - ▶ 차이점 : 상호 연결된 토픽의 형태로 수많은 문서들에 대한 인덱스를 제공
- 문서나 정보자원에 대한 정보 계층으로 형식에 무관하게 내용에 따라 탐색 가능