An Introduction and UML Profile for the Web Ontology Language (OWL)

October 23, 2002

Elisa F. Kendall
CEO & Founder
ekendall@sandsoft.com

Mark E. Dutra
Chief Architect
mdutra@sandsoft.com

(408) 402-0727

Definitions

- **An ontology is a specification of a conceptualization.** – *Tom Gruber*
- **Knowledge engineering is the application of logic and ontology to the task of building computable models of some domain for some purpose.** – *John Sowa*
- **Knowledge representation means that knowledge is formalized in a symbolic form, that is, to find a symbolic expression that can be interpreted.** – *Klein and Methlie*
- **The Semantic Web is the abstract representation of data on the World Wide Web, based on the RDF standards and other standards to be defined. It is being developed by the W3C, in collaboration with a large number of researchers and industrial partners.** – *W3C*

What is an Ontology?

- An ontology specifies a rich description of the
  - Terminology, concepts, nomenclature
  - Properties explicitly defining the terms, concepts
  - Relations among concepts (hierarchical and lattice)
  - Rules distinguishing concepts, refining definitions and relations (constraints, restrictions, regular expressions)

relevant to a particular domain or area of interest.

History

- Early work in knowledge representation (KR) done by the artificial intelligence community for knowledge sharing
- Examples: DARPA’s SHOE, HPKB, RKF programs; work done in the Knowledge Interchange Format (KIF)-based languages, description logics, conceptual graphs, intelligent agent communities
- Mapping between KR concepts and software engineering concepts not always straight forward
- Granularity of ontologies and domain models varies greatly between organizations, applications

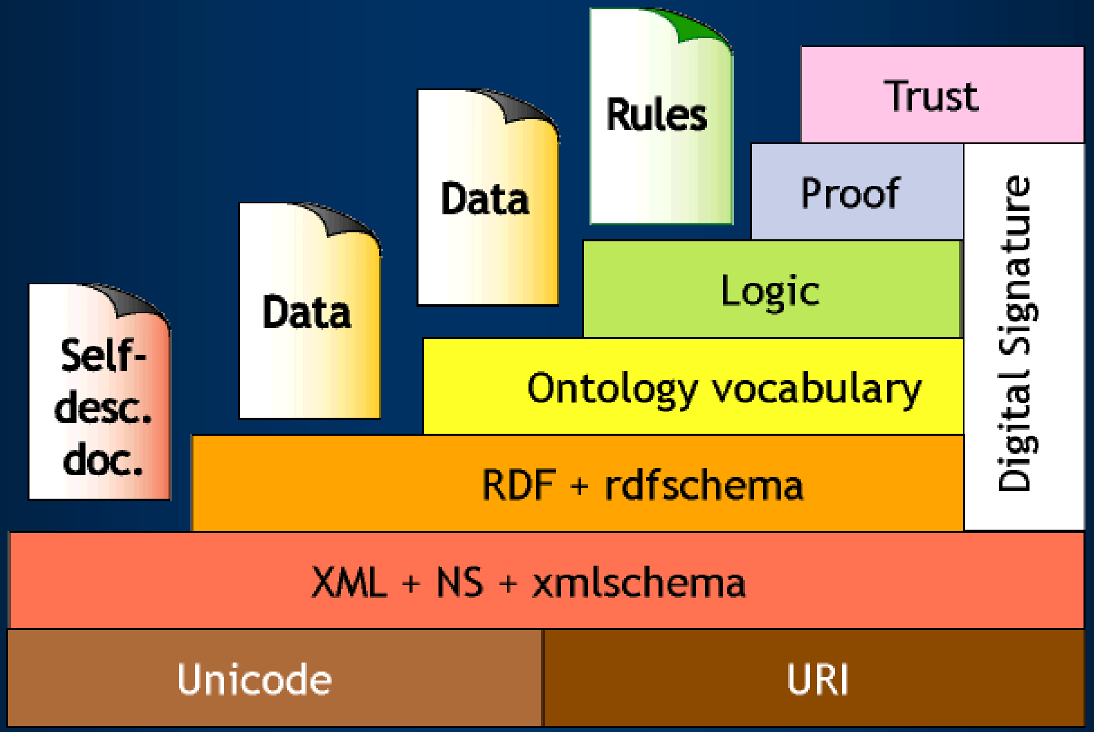
The Semantic Web

- **The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.** – *Tim Berners-Lee, James Hendler, Ora Lassila*
- **We expect the Semantic Web to be as big a revolution as the original web itself.** – *Rick Hayes-Roth, formerly HP CTO for software*
- **Ontologies facilitate greater machine readability of web content than XML, RDF, and RDF-S by providing additional vocabulary for term descriptions.** – *Deborah McGuinness, Frank van Harmelen (W3C Web Ontology Working Group)*
- References:
  - http://www.semanticweb.org/
  - http://www.w3.org/2001/sw/
  - http://www.daml.org/

Enabling Technologies

- Initial work motivated by XML
  - Extensible Markup Language (XML) 1.0 (Second Edition), 10/00
  - XML Information Set, 10/01
  - XML Schema, 5/01
- Second level – Resource Description Framework (RDF)
  - RDF Model and Syntax Specification, 2/99
  - RDF Vocabulary Description Language 1.0: RDF Schema (RDF-S), 3/2000 (updated 4/02)
  - RDF Model Theory, 4/02
  - RDF / XML Syntax Specification (Revised), 3/02
  - Vocabulary layer supports resources, properties, motivated in part by the Dublin Core
- Next layer – Web Ontology Language (OWL)
  - W3C Semantic Web Activity launched, 2/01
  - Working Draft specification released, 7/02
  - Provides higher order logic – negation, conjunction, disjunction, disjoint relations, union, intersection

Architecture



- Ontologies provide semantics through constraints, restrictions, complex relations among terms, and rules that build on/extend the metadata created in XML/RDF.
- Ontologies improve accuracy, promote completeness, with rich descriptions of concepts, terminology, and context

* Tim Berners-Lee, Semantic Web on XML, XML 2000 Washington DC

Kinds of Ontologies

- An *upper* ontology defines the base concepts upon which other ontologies are created. (See IEEE Standard Upper Ontology effort, http://www.suo.ieee.org/)
- A *domain, or classic*, ontology defines the terminology and concepts relevant to a particular topic or area of interest.
- A *process* ontology defines the inputs, outputs, constraints, relations, terms, and sequencing information relevant to a particular business process or set of processes. (See NIST's Process Specification Language, http://www.mel.nist.gov/psl/pubs/PSL1.0/paper.doc)
- An *interface* ontology defines the structure, content, messaging, and other restrictions relevant for a particular interface (*e.*g., application programming interface (API), database, scripting language, etc.). We envision a CORBA/MDA based interface definition ontology for this purpose.
- A *service* ontology defines a core set of markup language constructs for describing the properties and capabilities of Web services in unambiguous, computer-interpretable form. (See http://www.daml.org/services/)
- A *role-based* ontology defines terminology and concepts relevant for a particular end-user (person or consumer application).

Applications

- Ontologies provide a common vocabulary and definition of rules for use by independently developed services
- Agreements among companies, organizations sharing common services can be made with regard to their usage and the meaning of relevant concepts can be expressed unambiguously
- By composing component ontologies, mapping ontologies to one another and brokering terminology among participating resources and services, independently developed systems, agents and services can work together to share information and processes consistently, accurately, and completely.
- Ontologies also facilitate conversations among agents to collect, process, fuse, and exchange information.
- Improves search accuracy by enabling contextual search using concept definitions and relations among them instead of/in addition to statistical relevance of keywords.

Open Issues (General)

- Ontology language fragmentation – KIF, LOOM, OIL, XOL, DAML+OIL/OWL have led to varying requirements
- Rule language has yet to be defined
- Lack of methodology or standards for ontology development, configuration management, registry support – for web services or other applications
- Poor tool support
- Lack of existing ontology base
- Scale of ontologies – need component-based approach
- Scope of ontologies – where do you stop?

Motivation for UML-based Ontology Development

- Importance of knowledge representation (ontologies) increasing
- Limited commercial tools for ontology development
- Pool of experienced ontologists small
- Population of UML experienced engineers is growing
- Need to make ontology modeling accessible to domain experts
- Interest growing in development of a UML-based presentation syntax for OWL by W3C
- Working group within OMG to develop a UML profile for OWL

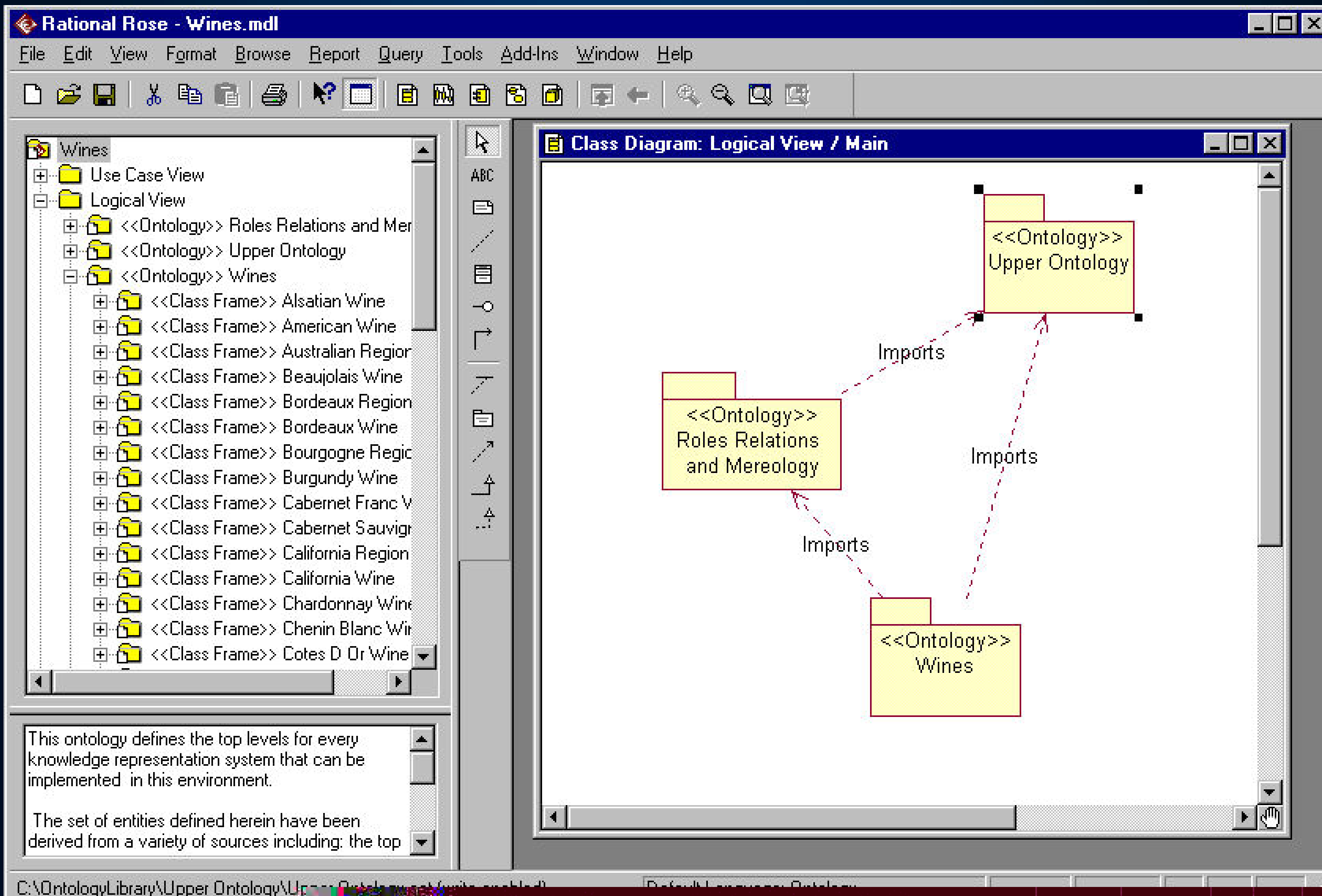
Basic Profile Constructs

OWL Ontology Item	UML Construct	UML Stereotype
Ontology	package	ontology
Class Frame	package	classFrame
Class	class	ontologyClass
Relation Frame	package	relationFrame
Relation (property restriction)	class	relation
Individual Frame	package	individualFrame
Individual	class	individual
	association	individualOf
	association	typeOf
Import Ontology	dependency	<none used>
Axiom (rule)	operation	axiom
	external file	

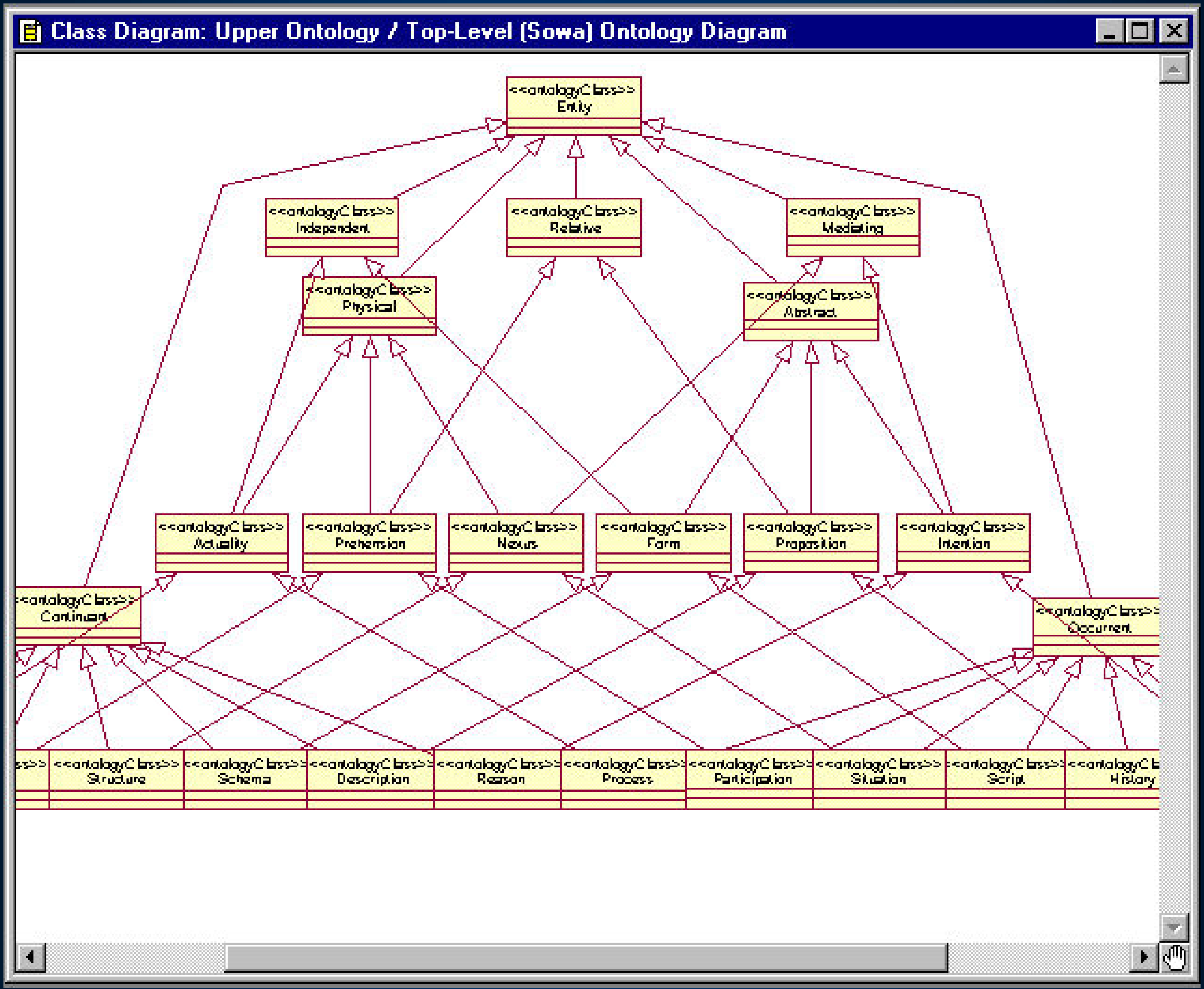
Select Class and Property Elements

OWL Ontology Item	UML Construct	UML Stereotype
complementOf	association	complementOf
disjointWith	class	disjointWith
	association	disjunct
disjointUnionOf Frame	package	disjointUnionOfFrame
disjointUnionOf Relation	class	disjointUnionOf
domain	association	domain
hasValue	association	hasValue
intersectionOf Frame	package	intersectionOfFrame
intersectionOf Relation	class	intersectionOf
inverseOf	association	inverseOf
range	association	range
sameClassAs	association	sameClassAs
sameIndividualAs	association	sameIndividualAs
samePropertyAs	association	samePropertyAs
subClassOf	inheritance relation	<none used>
subPropertyOf	inheritance relation	<none used>
unionOf Frame	package	unionOfFrame
unionOf Relation	class	unionOf

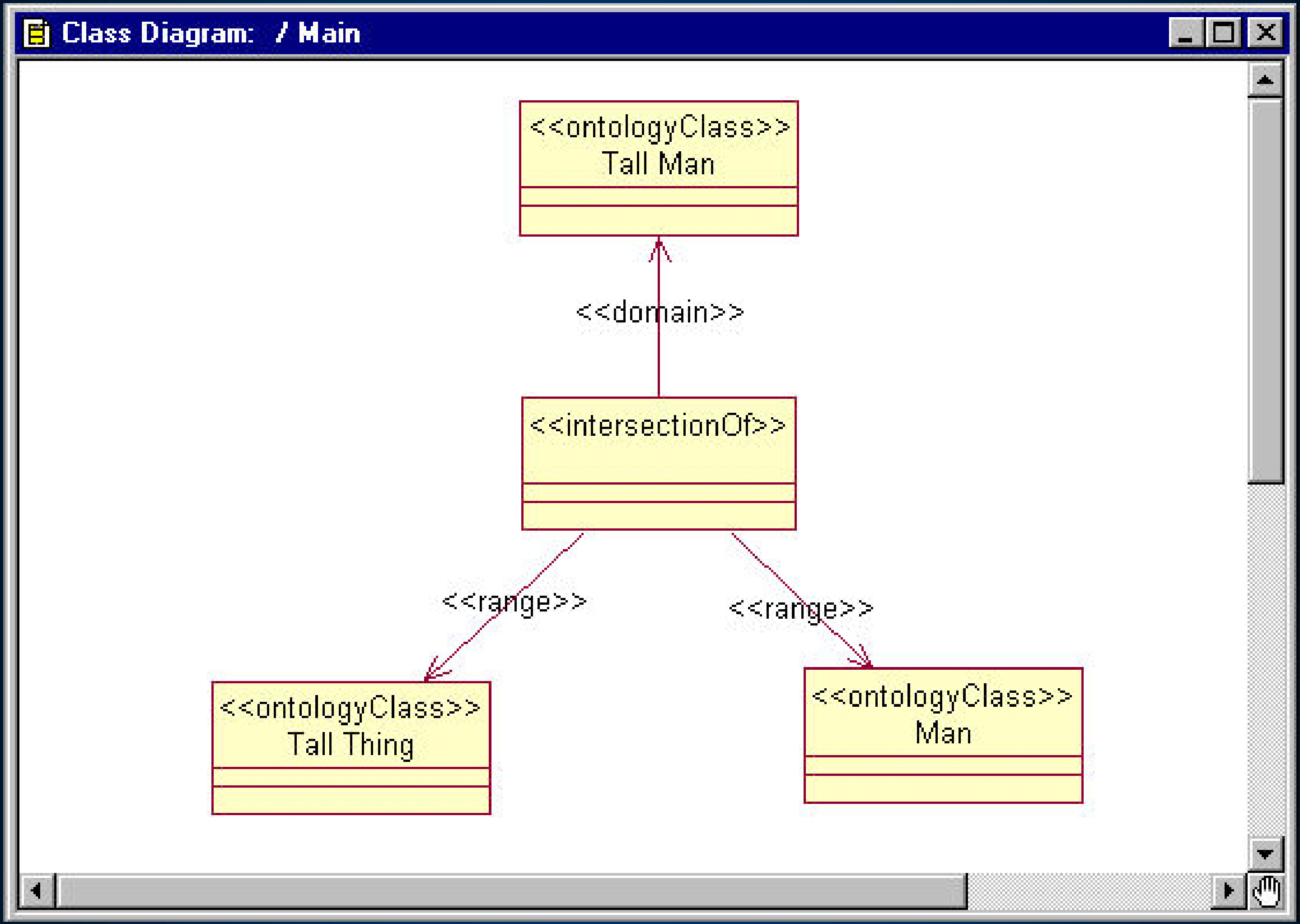
Example: Import Dependencies



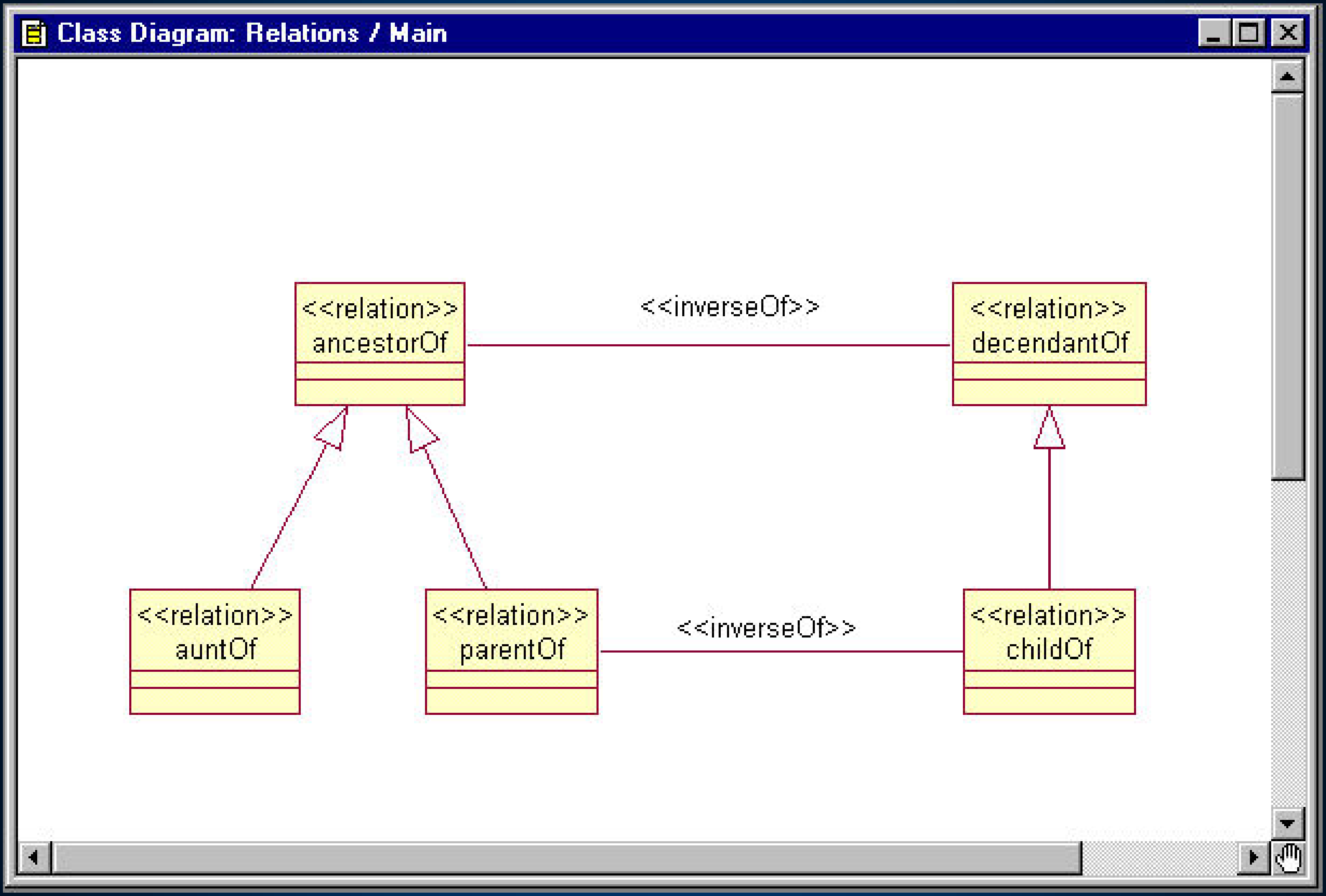
Example: Upper Ontology



Example: intersectionOf



Example: subPropertyOf



Implementation Issues

- Conceptual differences between KR and UML domains
  - UML Associations are not first class citizens; AssociationClasses must have defined endpoints
  - Need to keep additional information (slots, facets, rules) with classes
  - UML Attributes are not first class citizens
  - OCL is not sufficiently expressive for rule implementation (*e.g.*, variable representation)
- Limitations in tool support impacted implementation of ontology modeler add-in, and therefore impacted the profile itself
  - Representation of objects (individuals), classes on the same drawing