ONTOLOGY TO DEFINE DATA DOMAIN

By

Walaa Elgailani Eltayeb

A thesis submitted in partial fulfillment of the requirements for the degree of

B.Sc. in computer science

Approved by

Chairperson of Supervisory Committee:

Dr. Bognár Katalin

University of Debrecen

2008

UNIVERSITY OF DEBRECEN

ABSTRACT

ONTOLOGY TO DEFINE DATA DOMAIN

By

Walaa Elgailani Eltayeb

Chairperson of the Supervisory Committee/ Professor Dr. Bognár Katalin

Department of Computer Science

The management of large amounts of information and knowledge is perpetually increasing importance in today's large organizations. Finding the right information becomes an increasingly difficult task. The main goal of this thesis is to present methodological principles for building ontology based domain level.

Contents

List of Figures

1. Introduction

1.1. Acknowledgements

I would like to thank my family for supporting me during my study,
I would like to thank my advisor Dr. Bognár Katalin and for give me the opportunity to make this thesis. I thank all the people from the Faculty of Informatics for their efforts during the years without which this work would not have been possible.

1.2. Motivations

The so-called information society demands for complete access to available information, which is often heterogeneous and distributed. Most information systems use specific data models and databases for this purpose. This implies that making new data available to a system requires, that the data is either transferred into the system's specific data format or is even acquired again. This process is very time consuming and tedious. Data acquisition, automatically or semi-automatically, often makes large-scale investment in technical infrastructure and/or manpower inevitable. The idea of information sharing is to overcome these problems by providing common access to existing information sources.

However, in order to establish efficient information sharing, many technical problems have to be solved. Firstly, a suitable information source must be located that might contain data needed for a given task.
Once the information source has been found, access to the data therein has to be provided. This means that each of the information sources found in the first step have to work together with the system that is using the information.

It has been argued that ontologies are a key technology for this kind of applications. Successful approaches and applications are reported from the database area where ontologies have been used to enable Intelligent Information.

1.3. Paper structure

Chapter 1 Introduction

In this chapter1 I talk about some structure of the thesis, and I give general idea what the thesis all about.

Chapter 2 Ontology Theories

We talk about ontology theory what is ontology? What is it good for? And we show the steps of developing ontology.

Chapter 3 Knowledge representation

In this chapter we show some tools for knowledge representation like:

XML (Extensible Markup Language),

RDF (Resource Description Framework),

RDFS (RDF Schema).

Chapter 4 Movie ontology

In this chapter we present an example how to use ontology, based on ontology theory in chapter2. I create my own domain movie ontology.

Chapter 5 Implementation

In this chapter we introduce some programming tools to handle movie ontology.

Chapter 6 Conclusion

1.4. Introduction

In the Semantic Web data itself becomes part of the Web and is able to be processed independently of application, platform, or domain. This is in contrast to the World Wide Web as we know it today, which contains virtually boundless information in the form of documents. We can use computers to search for these documents, but they still have to be read and interpreted by humans before any useful information can be extrapolated. Computers can present you with information but can't understand what the information is well enough to display the data that is most relevant in a given circumstance. The Semantic Web, on the other hand, is about having data as well as documents on the Web so that machines can process, transform, assemble, and even act on the data in useful ways.

Implementing the Semantic Web requires adding semantic metadata, or data that describes data, to information resources. This will allow machines to effectively process the data based on the semantic information that describes it. When there is enough semantic information associated with data, computers can make inferences about the data, i.e., understand what a data resource is and how it relates to other data. Semantic Web architecture is described by W3C consortium as shown in figure 1.1, I will explain some of the layers in chapter3.

XML (eXtensible Markup Language) has paved the road by adding some metadata in the form of human-readable tags that describe data. In addition, XML documents can include information about the author of a Web page, relevant keywords for search engine optimization, and the software tools used to create the XML file, for example.

Before XML, data was stored in flat file and database formats, where most data was proprietary to an application. XML came along and made data interoperable within a single domain, i.e., within the domain defined by a schema or a set of related schemas. By itself, XML provides syntactic interoperability only when both parties

know and understand the element names used. If I label an element <price>12.00</price> and someone else labels it <cost>12.00</cost>, there's no way for a machine to know that those are the same thing without the aid of a separate, highly customized application to map between the elements. Semantic Web technologies help address this problem by making tags understandable not just to humans – but to machines as well.

The first step required for machines to understand data is to get that data into a uniform format, where, for instance, a field labeled "street" always has the same format and contains the same type of information, and so on. This type of functionality can be found today on Web sites that use forms that allow users to enter information and run a query, such as airline Web sites that allow visitors to search for and book flights based on a variety of criteria. However, considering the amount and variety of data available from different sources today, this method of data typing does not scale beyond very specific applications.

The next step towards the Semantic Web requires that data from multiple domains is classified based on its properties and its relationship with other data. This is where Semantic Web technologies such as RDF, RDFS. Show their operabilities.

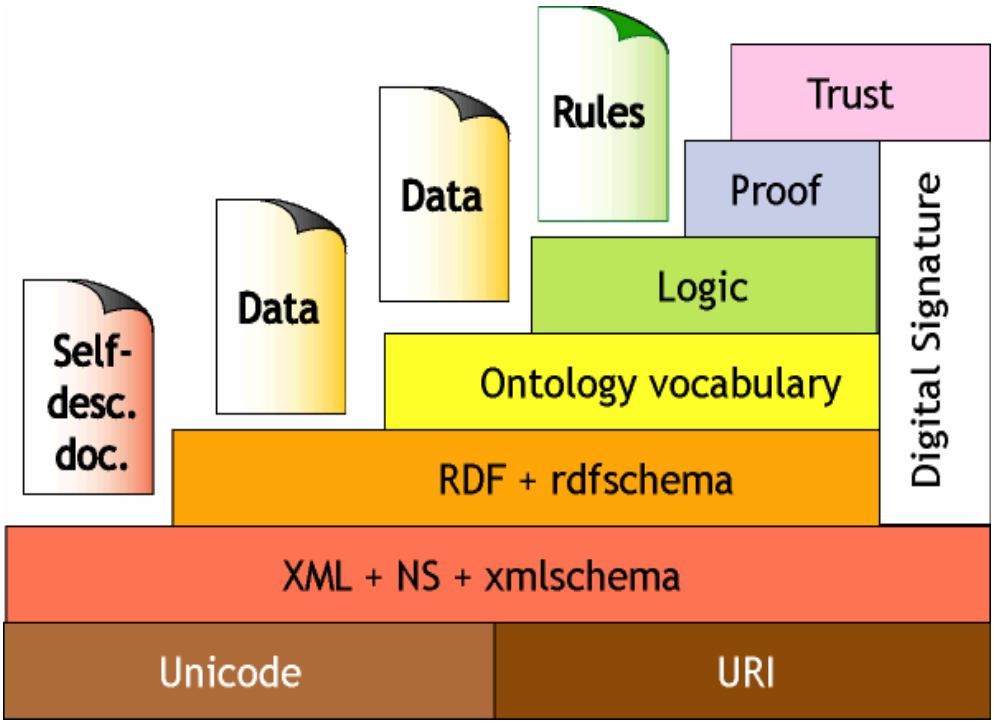

Figure 1.1: Semantic Web Architecture

2. Theory of ontology

2.1 What is Ontology?

Knowledge in ontologies can be captured and made available to both machines and humans. Traditional ontology definition is the specification of conceptualizations, used to help programs and humans share knowledge.

The conceptualization is the capturing of knowledge about the world in terms of entities (things, the relationships they hold and the constraints between them).

The specification is the representation of this conceptualization in a concrete form. One step in this specification is the encoding of the conceptualization in a knowledge representation language. Ontology is used in a wide range of application scenarios:

- A community reference: Its benefits include reusing knowledge, improving maintainability and long term knowledge retention.
- Either defining database schema or defining a common vocabulary for database annotation. Benefits include documentation, maintenance, reliability and sharing.
- Providing common access to information.
- Ontology-based search by forming queries over databases.
- Understanding database annotation and technical literature.

Some ontology's are designed to support natural language processing (NLP) that not only link domain knowledge but also how knowledge is related to linguistic structures such as grammar and lexicons.

The main components of ontology's are concepts, relations, instances and axioms. Concepts represent a set or class of entities within a domain. Relations describe the interactions between concepts or a concept's properties. Instances are the 'things' represented by a concept.

It is not necessary for ontology to have instances, because ontology is supposed to be a conceptualization of the domain. The combination of ontology with associated instances is what is known as a knowledgebase. Axioms are being used to constrain values for classes or instances. A common ideal for ontology is that it should be reusable. This ambition distinguishes ontology from a database schema, even though both are conceptualizations. Usually a database schema is intended to satisfy only one application, but ontology could be reused in many applications. However, ontology is only reusable when it is to be used for the same purpose for which it was developed.

Not all ontologies have the same intended purpose and may have parts that are reusable and other parts that are not. They will also vary in their coverage and level of detail.

2.2 Building Ontologies

The process of building ontologies is a high-cost process. Some people believe that the construction of ontologies is an art rather than a science. The goal for building ontologies is to create an agreed-upon vocabulary and a semantic structure for exchanging information about that domain.

There are no standard methodologies for building ontologies. Such a methodology would include a set of stages that occur when building ontologies, guidelines and principles to assist in the different stages, and an ontology life-cycle which indicates the relationships among stages.

Scientists usually distinguish between formal and informal techniques for ontology development.

They vary between informal methods, where the ontology is sketched out using either natural language descriptions or some diagram techniques like UML, and formal methods where the ontology is encoded in a formal

knowledge representation language such as RDFS, which is machine computable.

As one can see in Figure 2.1 there are common phases in most ontology life cycles:

- Specification: Identify purpose, scope and granularities. This phase is important for design, evaluation and reuse of ontologies.
- Knowledge Acquisition: the process of acquiring domain knowledge from experts; database metadata; standard text books; and other ontologies.
- Conceptualization: identifying the key concepts that exist in the domain, their properties and the relationships that hold between them; identifying natural language terms to refer to such concepts, relations and attributes; and structuring domain knowledge into explicit conceptual models.
- Integration: use or combine available data from existing databases and ontologies to obtain a consistent ontology.
- Encoding: representing the conceptualization in some formal language, for example frames, object models or logic.
- Evaluation: by assessing the competency of the ontology to satisfy the requirements of its application, including determining the consistency, completeness and conciseness of an ontology. We evaluate ontologies for completeness, consistence and avoidance of redundancy.
- Documentation: an ontology that can not be understood can not be reused. Informal and formal complete definitions, assumptions and examples are essential to promote the appropriate use and reuse of ontology.

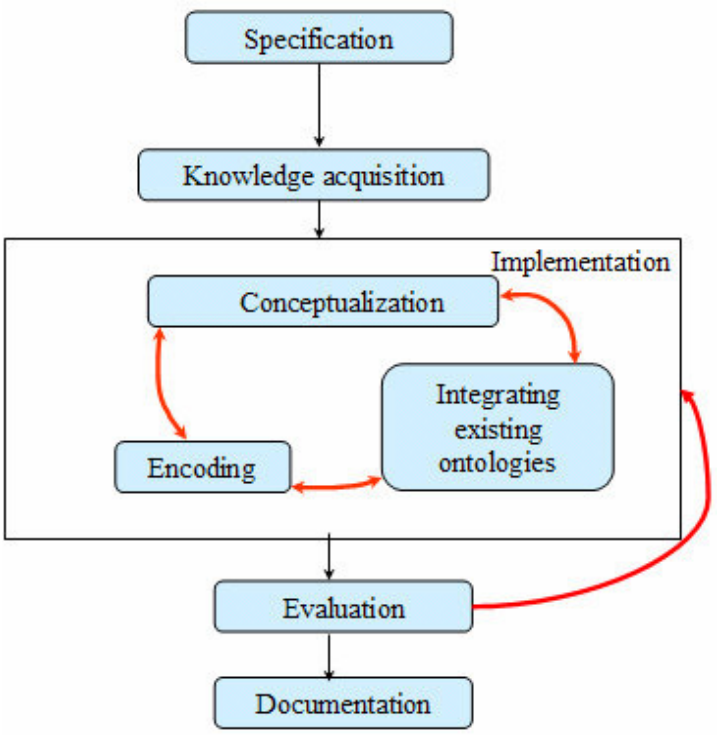

Figure 2.1.Ontology development life cycle

3. Knowledge Representation

In addition to describing an area of knowledge, we also need to represent that description. What does representation mean?

Representation means that we encode the description in the way that enables someone to use the description. A description consists of words and phrases in a natural language (such as English or Chinese), that is vocabulary and sentences that combine terminologies to express relationship among the terms.

In information technology, however, we use slightly more complicated notion of representing. We represent in order to use the description in information technology; in the other word, we create a model that software will able to utilize. We represent classes, instance, relationship, properties, and rules for area of knowledge.

If we want to represent our description as clear, precisely, and unambiguously as possible and natural language can be very ambiguous we also want to make its meaning available for information technology use. Representation thus has to do with defining the terms (the vocabulary that acts as label for the concepts) and that means also defining the concepts and their relationships that are behind the labels and that constitute the meaning of the model of the knowledge area we are interested in.

3.1. XML

XML (Extensible Markup Language) is a subset of SGML (Standard General Markup Language). In its original definition a markup language is a language which is intended for adding information ("markup" information) to an existing document. This information must stay separate from the original hence the presence of separation characters. In SGML and XML 'tags' are used.

There are two kinds of tags: opening and closing tags. The opening tags are keywords enclosed between the signs "<" and ">". An example: <author>.

A closing tag is practically the same, only the sign "/" is added e.g. </author>.

With these elements alone very interesting data structures can be built.

An example of a book description - see Figure 3.1

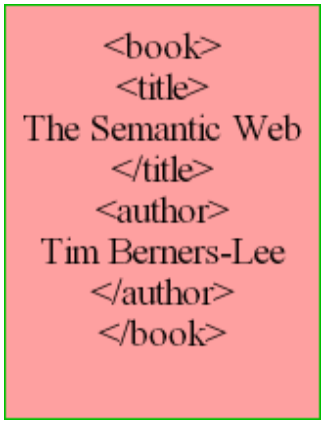

Figure 3.1 Book description

As can be seen it is quite easy to build hierarchical data structures with these elements alone. A tag can have content too: in the example the strings "The Semantic Web" and "Tim Berners-Lee" are content. One of the good characteristics of XML is its simplicity and the ease with which parsers and other

tools can be built. The keywords in the tags can have attributes too. The previous example could be written:

<book title="The Semantic Web" author="Tim Berners-Lee"></book>

Where attributes are used instead of tags, XML is not a language but a meta-language i.e. a language with as goal to make other languages ("markup" languages).

Everybody can make his own language using XML. A person doing this only has to use the syntax's of XML i.e. produce well formed XML. However more constraints can be added to an XML language by using a DTD or an XML schema. A valid XML document is one that uses XML syntax and respects the constraints laid down in a DTD or XML schema.

If everybody creates his own language then the "tower-of-Babylon" -syndrome is looming. How is such diversity in languages handled?

This is done by using namespaces. A namespace is a reference to the definition of an XML language. Suppose someone has made an XML language about birds. Then he could make the following namespace declaration in XML:

<birds:wing xmlns:birds=" http://birdSite.com/birds/" >

This statement is referring to the tag "wing" whose description is to be found on the site that is indicated by the namespace declaration xmlns (= XML namespace). Now our hypothetical biologist might want to use an aspect of the physiology of birds described however in another namespace:

<physiology:temperature xmlns:physiology=" http://physiology.com/xml/" >

By the semantic definition of XML these two namespaces may be used within the same XML-object.

```
<?xml version=" 1.0" ?>
<birds:wing xmlns:birds=" http://birdSite.com/birds/" > large </birds:wing>
<physiology:temperature xmlns:physiology=" http://physiology.com/xml/" >
43
</physiology:temperature>
```

Figure 3.2 XML name space

3.2. RDF

Resource Description Framework, The RDF model is often called a "triple" because it has three parts, described in terms of resource properties in the preceding text, in the knowledge representation community, those three parts are described in terms of the grammatical parts of a sentence: subject, predicate, and object.

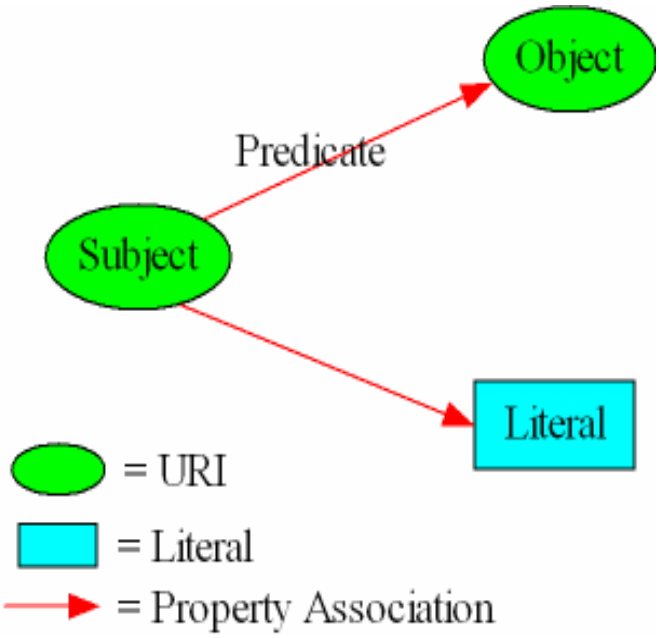

Figure 3.3 RDF model

Statement is objects attribute value triple consisting of resource, a property, and value, can either be resource or literals. Literals are atomic value (string).

Subject in grammar, this is the noun or noun phrase that is the doer of the action. In the sentence "The Company sells batteries," the subject is "the company".
The subject of the sentence tells us what the sentence is about. In logic, this is the term about which something is asserted. In RDF, this is the resource that is being described by the ensuing predicate and object. Therefore, in RDF, we want a URI to stand for the unique concept "company" Predicate. In grammar, this is the part of a sentence that modifies the subject and includes the verb phrase. Returning to our sentence "The Company sells batteries," the predicate is the phrase "sells batteries." In other words, the predicate tells us something about the subject. In logic, a predicate is a function from individuals (a particular type of subject) to truth-values with an arity based on the number of arguments it has. In RDF,
a predicate is a relation between the subject and the object. Thus, in RDF, we would define a unique URI for the concept "sells" Object.
In grammar this is a noun that is acted upon by the verb. Returning to our sentence "The Company sells batteries," the object is the noun "batteries."
In logic, an object is acted upon by the predicate. In RDF, an object is either a resource referred to by the predicate or a literal value. In our example, we would define a unique URI for "batteries"

Capturing Knowledge with RDF

There is wide consensus that the triple-based model of RDF is simpler than the RDF/XML format, which is called the "serialization format." Because of this, a variety of simpler formats have been created to quickly capture knowledge expressed as a list of triples. Let's walk through a simple scenario where we express concepts in four different ways: as natural language sentences, Following the linguistic model of subject, predicate, and object, we start with three English statements:

-Buddy Belden owns a business.
-The business has a Web site accessible at http://www.c2i2.com/~budstv.
-Buddy is the father of Lynne.

In your business, you could imagine extracting sentences like these from daily routines and processes in your business. There are even products that can scan email and documents for common nouns and verbs. In other words, capturing statements in a formal way allows the slow aggregation of a corporate knowledge base in which you capture processes and best practices, as well as spot trends. This is knowledge management via a bottom-up approach instead of a top-down approach. Now let's examine how we capture the preceding sentences in N3 notation see Figure 3.4

```
<#Buddy> <#owns> <#business>.
<#business> <#has-website>
<http://www.c2i2.com/~budstv>.
<#Buddy> <#father-of> <#Lynne>.
```

Figure 3.4 N3 notation

From each sentence we have extracted the relevant subject, predicate, and object. The # sign means the URI of the concepts would be the current document.

This is a shortcut done for brevity; it is more accurate to replace the # sign with an absolute URI like *"http://www.c2i2.com/buddy/ontology"* as a formal namespace (see Figure 3.2)

```
<rdf:RDF
  xmlns:RDFNsId1='#'
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
  <rdf:Description rdf:about='#Buddy'>
    <RDFNsId1:owns>
      <rdf:Description rdf:about='#business'>
        <RDFNsId1:has-websiterdf:resource='http://www.c2i2.com/~budstv' />
      </rdf:Description>
    </RDFNsId1:owns>
  <RDFNsId1:father-of rdf:resource='#Lynne'/>
  </rdf:Description>
</rdf:RDF>
```

Figure 3.5 RDF/XML

The first thing you should notice is that in the RDF/XML syntax, one RDF statement is nested within the other. It is this sometimes no intuitive translation of a list of statements into a hierarchical XML syntax that makes the direct authoring of RDF/XML syntax difficult; however, since there are tools to generate correct syntax for you, you can just focus on the knowledge engineering and not author the RDF/XML syntax. Second, note how predicates are represented by custom elements (like RDFNsId1: owns or RDFNsId1: father-of). The objects are represented by either the RDF: resource attribute or a literal value.

3.3. RDF Schema (RDFS)

RDFS is used to create vocabularies that describe groups of related RDF resources and the relationships between those resources. An RDFS vocabulary defines the allowable properties that can be assigned to RDF resources within a given domain. RDFS also allows you to create classes of resources that share common properties.

Using the same triples paradigm defined by RDF, RDFS triples consist of classes, class properties, and values that define the classes and relationships between the resources within a particular domain.

In an RDFS vocabulary, resources are defined as instances of classes. A class is a resource too, and any class can be a subclass of another. This hierarchical semantic information is what allows machines to determine the meanings of resources based on their properties and classes. Overall, RDFS is a simple vocabulary language for expressing the relationships between resources.

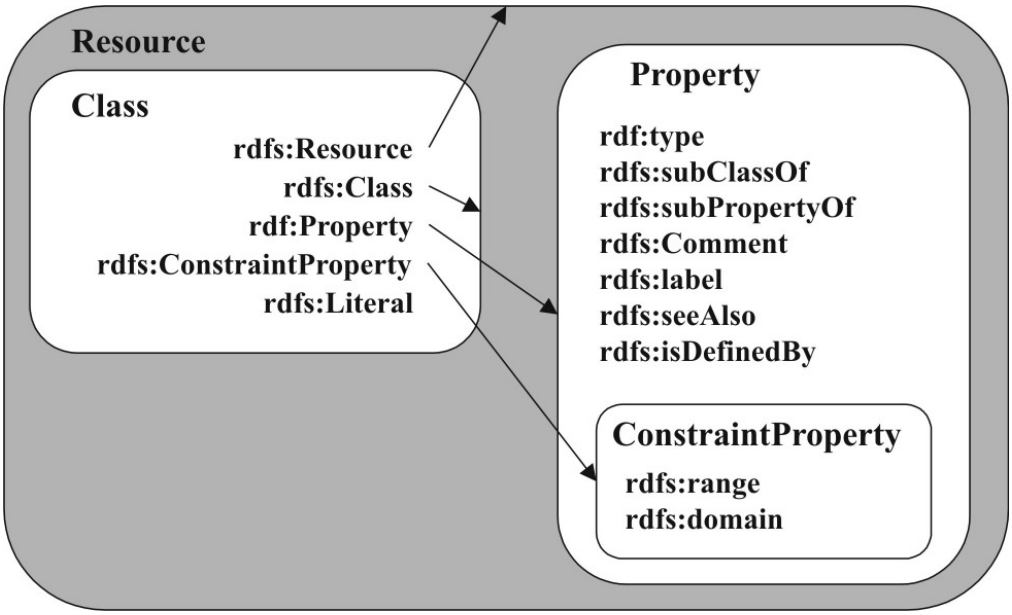

Figure 3.6: RDFS

4. Movie ontology

Accordant to theory ontology we mention earlier, I would like to define my own data domain.

4.1. Introduction to movie ontology

When we start to develop ontology first thing we specify the domain and the scope, what our ontology will cover?

In this term our ontology about Movie it may answer several questions like: which movie is action?

When specific movie released?

Movie ontology we can use it for many purposes like:

Online DVD shop, storing data.

4.2. Define Movie ontology

To define movie ontology we have to write all terms about the movie that we want to talk about and I think its good idea to write statement like this:

- Movies has a name
- Movies has Director
- Movies has genre
- Director has a name

4.3. Representation of Movie Ontology

To represent movie ontology we have to evaluate all statement we make About our ontology and underline concept to be visible, a collection of ontology describing specific movies might include information such as the movie title, genre, actors, director, etc. An actor or director's information may include name, nationality, etc.

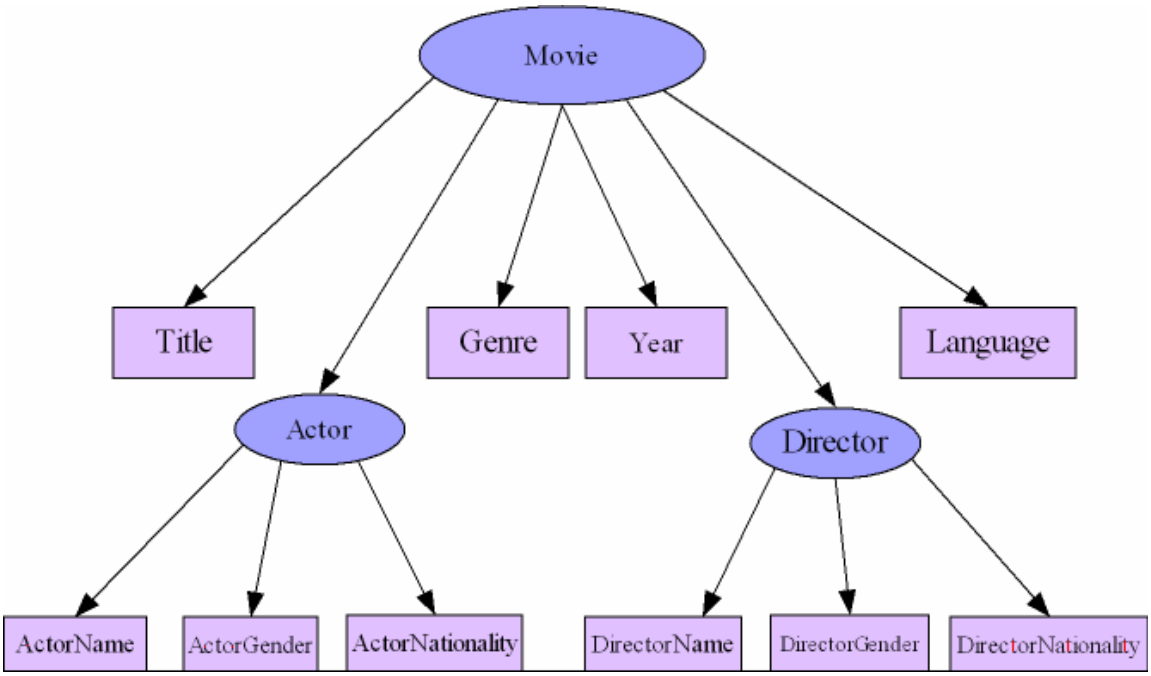

Figure 4.1 Movie tree representations

The portion of domain described, contains the classes Movie, Actor and Director. The collection of movies ontology represents a group of embedded objects that are the instances of these classes.

The class Movie has attributes Year, Actor (representing the role "acted by"), Director, etc. The Actor and Director Attributes have values that are other objects in the ontology, specifically, object instances of classes Actor and Director, respectively. The attribute Year is an example of an attribute whose data type is positive integers with the usual ordering.

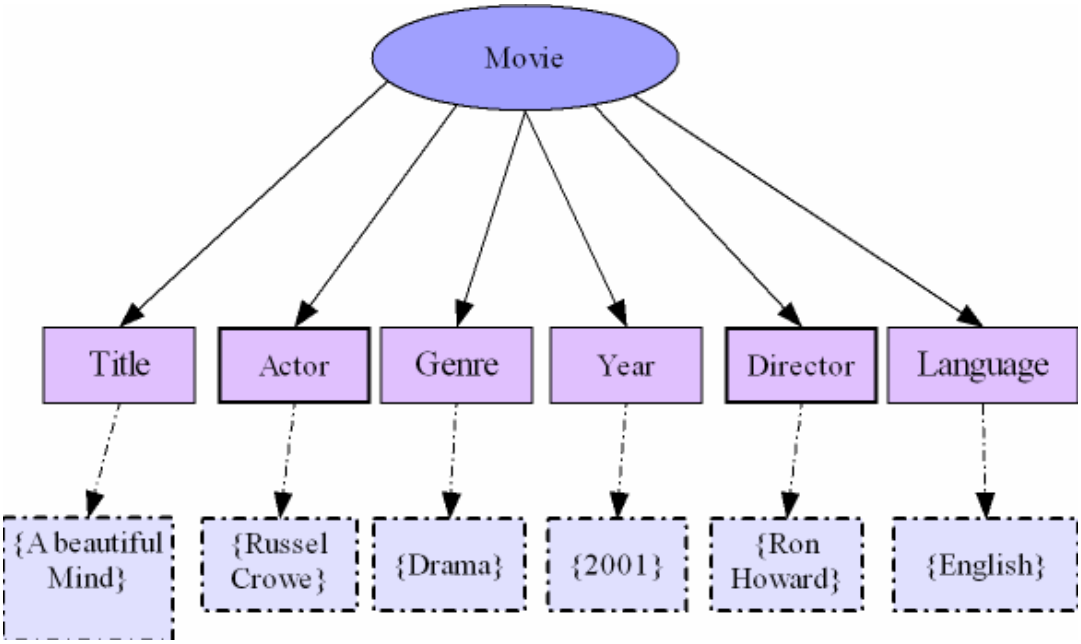

Figure 4.2 Movie instance

An attribute a of an object o has a domain Da. In cases when the attribute has unique value for an object, Da is a singleton. For example, consider an object instance of class Movie; "A beautiful mind" (see Figure 4.2).

The attribute Actor contain object instances (Russell Crowe) its instance of the class Actor (for the sake of presentation we use the Actor's name to stand for the object of Actor). Therefore, DActor = {Russell Crowe}. Also, A real object may have values for only some of the attributes.

We need to represent movie ontology in representation language to guarantee that we can store or use it or share with other domain. Representing movie ontology in RDF it will look like Figure 4.3

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
     xmlns:rdf
="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xsd
="http://www.w3.org/2001/XLMSchema#"
     xmlns:movie ="http://mydomain/movie-ns#"

<rdf:Description rdf:about="movie"
     <movie:Title ="A beautiful mind"/>
     <movie:Year="2001"/>
     <movie:Genre="drama"/>
    <movie:Language="English"
     <movie:Actor rdf:nodeID="A0"/ >
     < movie:Director rdf:nodeID="A1"/>
</rdf:Description>
<rdf:Description rdf:nodeID="A0">
     <movie:actorName="Russel Crowe"/>
     <movie:actorGender="male"/>
     <movie:actorNationality="New Zealand"/>
</rdf:Description>
<rdf:Description rdf:nodeID="A1">
     <movie:DirectorName="Ron Howard"/>
     <movie:DirectorGender="male"/>
     <movie:DirectorNationality="American"/>
</rdf:Description>
</rdf>
```

Figure 4.3 Movie RDF/XML representations

5. Implementation

To search our movie data domain that we define in chapter 4 we use some programming tools.

5.1 Jena

Jena is an API for the Java programming language, to create Semantic Web applications and manipulate RDF graphs. The Jena Framework is open source software and was mainly developed by the HP Labs Semantic Web Research team.

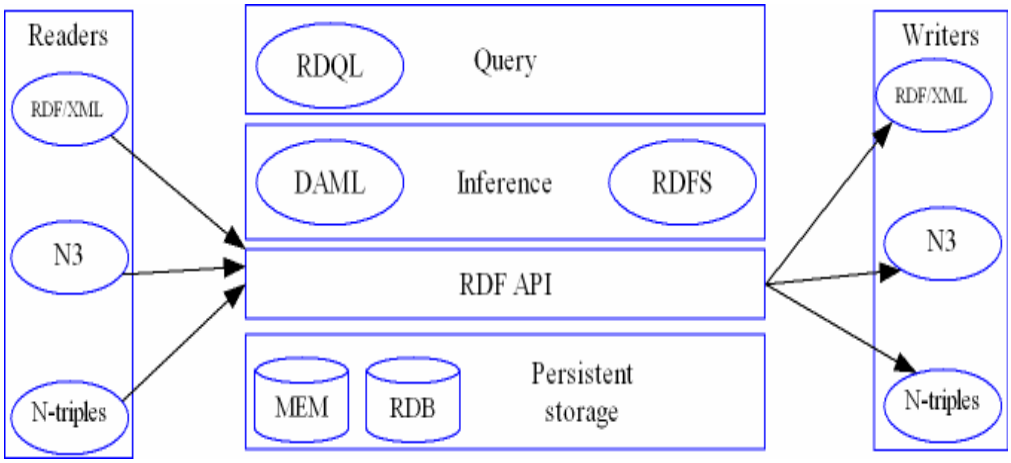

Figure 5.1 Jane architecture

Figure 5.1 gives an overview of the Jena architecture. The main parts are:

- RDQL - A query language for RDF.
- The Jena Ontology API - Supports RDFS and use of Inference Engines.
- The RDF API - conforms to the RDF specification.
- Persistent Storage-Store and retrieve RDF statements using a database.

Jena uses "Another RDF Parser" (ARP) as RDF/XML parser component. This parser can also be used standalone to convert RDF/XML files to different notations or to check RDF/XML files for syntax errors.

5.2 The RDF API

The package com.hp.hpl.jena.rdf.model contains the main classes for working with RDF graphs. The most important classes and interfaces are:

- **Model**

  An RDF model is a set of Statements. This interface defines a set of primitive methods for creating resources, properties and literals and the statements which link them, for adding statements to and removing them from a model, for querying a model and set operations for combining models. Internally the information is stored as an RDF graph.

- **ModelFactory**

  Provides methods for creating instances of standard kinds of models as all models are interfaces and cannot be instantiated directly.

- **Resource, Statement, Property, RDFNode,**

  Give access to the different parts of an RDF model.

- **ResourceFactory**

  This class creates resources and properties and things of that manner.

5.3 Searching

We like to know all title in the movies database. The fast way is to query database. Jena has methods for reading and writing RDF as XML. These can be used to save an RDF model to a file and later read it back in again. We represent movies ontology as RDF (see Appendix 1). To query movie database we have to create empty model then we read RDF in to our model.

```
Model model = ModelFactory.createDefaultModel();
model.read( movie.rdf, "");
```

Model.listSubjectsWithProperty(Property p, RDFNode o) will return an iterator over all the resources which have property p with value o. If we assume that only movies resources will have MOVIE.Title property, and that in our data, all such resources have such a property, then we can find all the titles like this:

```
ResIterator iter = model.listSubjectsWithProperty(MOVIE.Title);
while (iter.hasNext()) {    Resource r = iter.nextResource();    ...}
```
[1]

The output of our program looks like:

The movie titles in the database are:

Gladiator
American Gangster
A beautiful Mind
Szabadság szerelem
Volver
Se jie
The Godfather
Philadelphia
Sahara
2LDK
The Terminator

[1] I have the complete java code in a disk.

The Matrix

Hack

Platoon

Romeo Must Die

The Master

Ali

Mission: Impossible II

Magyar szépség

Big Momma

6. Conclusion

In this thesis we have introduced and followed ontology design and development case study in the domain of movie using state-of-the-art semantic technologies.

After, reviewing the basic concepts, other similar systems and current available tools and technologies and uncovering their problems, we turned to a description of the context of our ontology design and development by focusing on the following key areas:

- Knowledge gathering: at this stage we studied which resources to use in our tasks and how to use each of them, how to discover instances from those resources how to link their content and understand the content of the resources and interpret the results.
- Ontology implementation: includes conceptualization, encoding and searching our data.

Through this entire feature I did my goal and, I defined movie data domain. In my opinion ontology is a good and fast way to define relationships among concepts inorder to use these concepts in several applications

References

1. Grigoris & Frank Van Harmeleu. Published: January 1, 2004. A semantic Web Prime

2. N. F. Noy & D. L. McGuinness (2001). Ontology Development 101: A Guide to Creating Your First Ontology'. http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html

3. Chris Welty: Knowledge Representation Recommendation, 17 SEP 1996 http://www.cs.vassar.edu/faculty/welty/papers/phd/HTML/dissertation-14.html

4. Tim Bray & Jean Paoli : Extensible Markup Language (XML) 1.0 Recommendation 16 August 2006, http://www.w3.org/TR/REC-xml/

5. Dave Beckett: RDF/XML Syntax Specification (Revised). W3C Recommendation, February 2004, http://www.w3.org/TR/rdf-syntax-grammar/

6. Dan Brickley, R.V. Guha, Epinions : Resource Description Framework (RDF) Schema Specification 1.0 Recommendation, 27 March 2000 http://www.w3.org/TR/2000/CR-rdf-schema-20000327/

7. Tim Berners-Lee: Uniform Resource Identifiers (URI): Generic Syntax. RFC 2396, August 1998, http://www.ietf.org/rfc/rfc2396.txt

8. Dave Reynolds: Jena 2 Inference support Recommendation, September 2007
http://jena.sourceforge.net/inference/

9. Andy Seaborne: A Programmer's Introduction to RDQL
Recommendation, April 2002, Updated February 2004
http://jena.sourceforge.net/tutorial/RDQL/index.html

10. Ashok Malhotra & Neel Sundaresan: RDF Query Specification
Recommendation, December 3, 1998
http://www.w3.org/TandS/QL/QL98/pp/rdfquery.html

11. Tim Bray, Textuality: Namespaces in XML 1.0 (Second Edition)
Recommendation, 16 August 2006
http://www.w3.org/TR/REC-xml-names/

12. http://jena.sourceforge.net/javadoc/index.html

13. Amazon Company: The Internet Movie Database
Recommendation, 2007
http://www.imdb.com/search

Appendix

Appendix 1. Movie RDF/XML

```
<rdf:RDF
 xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
 xmlns:vcard='http://www.w3.org/2001/movie-rdf/3.0#'
 >
<rdf:Description rdf:about="http://somewhere/Gladiator">
  <movie:Title>Gladiator</movie:Title>
  <movie:Genre>Action</movie:Genre>
  <movie:Year>2000</movie:Year>
  <move:Language>English</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Russell crow</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>New Zealand</movie:actorNationality>
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Ridley scott</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>English</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/American Gangster">
  <movie:Title> American Gangster</movie:Title>
  <movie:Genre>Drama</movie:Genre>
  <movie:Year>2007</movie:Year>
  <move:Language>English</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Denzel Washington</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>American</movie:actorNationality>
```

```
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Ridley scott</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>English</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/A beautiful Mind">
  <movie:Title>A beautiful Mind</movie:Title>
  <movie:Genre>Drama</movie:Genre>
  <movie:Year>2001</movie:Year>
  <move:Language>English</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Russell crow</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>New Zealand</movie:actorNationality>
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Ron Howard</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>English</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/Szabadság szerelem">
  <movie:Title>Szabadság szerelem</movie:Title>
  <movie:Genre>Romance</movie:Genre>
  <movie:Year>2006</movie:Year>
  <movie:Language>Magyar</movie:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Kata Dobó</movie:ActorName>
    <movie:ActorGender>Female</movie:ActorGender>
    <movie:ActorNationality>Hungarian</movie:actorNationality>
```

```
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Krisztina Goda</movie:DirectorName>
    <movie:DirectorGender>Female</movie:DirectorGender>
    <movie:Director Nationality>Hungarian</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/Volver">
  <movie:Title>Volver</movie:Title>
  <movie:Genre>Comedy</movie:Genre>
  <movie:Year>2006</movie:Year>
  <move:Language>Spanish</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Penélope Cruz</movie:ActorName>
    <movie:ActorGender>Female</movie:ActorGender>
    <movie:ActorNationality>Spanish</movie:actorNationality>
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Pedro Almodóvar</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>Spanish</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/Se, jie">
  <movie:Title>Se, jie</movie:Title>
  <movie:Genre>Drama</movie:Genre>
  <movie:Year>2007</movie:Year>
  <move:Language>Chinese</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Tony Leung Chiu Wai</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>Chinese</movie:actorNationality>
```

```
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Ang Lee</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>Chinese</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/The Godfather">
  <movie:Title>The Godfather</movie:Title>
  <movie:Genre>Drama</movie:Genre>
  <movie:Year>1972</movie:Year>
  <move:Language>English</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Marlon Brando</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>American</movie:actorNationality>
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Francis Ford Coppola</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>American</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/Philadelphia">
  <movie:Title>Philadelphia</movie:Title>
  <movie:Genre>Drama</movie:Genre>
  <movie:Year>1993</movie:Year>
  <move:Language>English</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Tom Hanks</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>American</movie:actorNationality>
```

```
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Jonathan Demme</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>American</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/Sahara">
  <movie:Title>Sahara</movie:Title>
  <movie:Genre>Action</movie:Genre>
  <movie:Year>2005</movie:Year>
  <move:Language>English</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Matthew David McConaughey</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>American</movie:actorNationality>
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Breck Eisner</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>American</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/2LDK">
  <movie:Title>2LDK</movie:Title>
  <movie:Genre>Thriller</movie:Genre>
  <movie:Year>2003</movie:Year>
  <move:Language>Japanese</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Maho Nonami</movie:ActorName>
    <movie:ActorGender>Female</movie:ActorGender>
    <movie:ActorNationality>Japanese</movie:actorNationality>
```

```
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Yukihiko Tsutsumi</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>Japanese</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/The Terminator">
  <movie:Title>The Terminator</movie:Title>
  <movie:Genre>Action</movie:Genre>
  <movie:Year>1984</movie:Year>
  <move:Language>English</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Arnold Schwarzenegger</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>Australian</movie:actorNationality>
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>James Cameron</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>Canadian</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/The Matrix">
  <movie:Title>The Matrix</movie:Title>
  <movie:Genre>Sci-Fi</movie:Genre>
  <movie:Year>1999</movie:Year>
  <move:Language>English</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Keanu Reeves</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>Lebanon</movie:actorNationality>
```

```
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Andy Wachowski</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>American</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/Hack">
  <movie:Title>Hack</movie:Title>
  <movie:Genre>Horror</movie:Genre>
  <movie:Year>2006</movie:Year>
  <move:Language>English</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Tony Burton</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>American</movie:actorNationality>
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Matt Flynn</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>American</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/Platoon">
  <movie:Title>Platoon</movie:Title>
  <movie:Genre>War</movie:Genre>
  <movie:Year>1986</movie:Year>
  <move:Language>English</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Tom Berenger</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>American</movie:actorNationality>
```

```
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Oliver Stone</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>American</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/Romeo Must Die">
  <movie:Title>Romeo Must Die</movie:Title>
  <movie:Genre>Action</movie:Genre>
  <movie:Year>2000</movie:Year>
  <move:Language>English</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Jet Li</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>Chinses</movie:actorNationality>
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Andrzej Bartkowiak</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>Polish</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/The Master">
  <movie:Title>The Master</movie:Title>
  <movie:Genre>Action</movie:Genre>
  <movie:Year>1989</movie:Year>
  <move:Language>Chinese</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Jet Li</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>Chinese</movie:actorNationality>
```

```
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Tsui Hark</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>Chinese</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/Ali">
  <movie:Title>Ali</movie:Title>
  <movie:Genre>Sport</movie:Genre>
  <movie:Year>2001</movie:Year>
  <move:Language>English</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Will Smith</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>American</movie:actorNationality>
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Michael Mann</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>American</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/Mission: Impossible II">
  <movie:Title>Mission: Impossible II</movie:Title>
  <movie:Genre>Adventure</movie:Genre>
  <movie:Year>2000</movie:Year>
  <move:Language>English</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Tom Cruise</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>American</movie:actorNationality>
```

```
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>John Woo</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>Chinese</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/Magyar szépség">
  <movie:Title>Magyar szépség</movie:Title>
  <movie:Genre>Comedy</movie:Genre>
  <movie:Year>2003</movie:Year>
  <move:Language>Hungarian</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Dorottya Udvaros</movie:ActorName>
    <movie:ActorGender>Female</movie:ActorGender>
    <movie:ActorNationality>Hungarian</movie:actorNationality>
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Péter Gothár</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>Hungarian</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/Lara Croft: Tomb Raider">
  <movie:Title>Lara Croft: Tomb Raider</movie:Title>
  <movie:Genre>Action</movie:Genre>
  <movie:Year>2001</movie:Year>
  <move:Language>English</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Angelina Jolie</movie:ActorName>
    <movie:ActorGender>Female</movie:ActorGender>
    <movie:ActorNationality>American</movie:actorNationality>
```

```
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Simon West</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>English</movie:DirectorNationality>
  </Director>
</rdf:Description>

<rdf:Description rdf:about="http://somewhere/Big Momma">
  <movie:Title>Big Momma</movie:Title>
  <movie:Genre>Comedy</movie:Genre>
  <movie:Year>2000</movie:Year>
  <move:Language>English</move:Language>
  <movie:Actor rdf:parseType="Resource">
    <movie:ActorName>Martin Lawrence</movie:ActorName>
    <movie:ActorGender>Male</movie:ActorGender>
    <movie:ActorNationality>American</movie:actorNationality>
  </Actor>
  <movie:Director rdf:parseType="Resource">
    <movie:DirectorName>Raja Gosnell</movie:DirectorName>
    <movie:DirectorGender>Male</movie:DirectorGender>
    <movie:Director Nationality>American</movie:DirectorNationality>
  </Director>
</rdf:Description>
</rdf:RDF>
```