Design of Ontology for The Internet Movie Database (IMDb)

Sasikanth Avancha, Srikant Kallurkar, Tapan Kamdar
{savanc1,skallu1,kamdar}@cs.umbc.edu
Semester Project, CMSC 771, Spring 2001
Table of Contents

1. Introduction
2. Database Construction in IMDb
   2.1. Overview
   2.2. Adding Information to the Database
   2.3. Guides List
   2.4. Regular Expression Matching
3. IMDb Web Service Ontology
   3.1. Overview
   3.2. The Movie Component
   3.3. The Schedules and Location Component
   3.4. The Awards and Nominations Component
   3.5. The MoviePerson subclass
   3.6. The MovieCompany subclass
4. Conclusions
1. Introduction

The IMDb is a large database consisting of relevant and comprehensive information about movies - past, present and future. It began as a set of shell scripts and data files. The data files were a collection of email messages exchanged between users of the rec.arts.movies Usenet bulletin board. These movie fans exchanged information on actors, actresses, directors etc. and biographical information about various moviemakers. At some point, these data files were made searchable using commands built by the shell scripts. The current database uses Perl and its built-in support for the Berkeley DBM database structure and almost all the software used for searching the database is written in Perl. The fuzzy search routines are implemented in C. The website is served using the Apache web server using the mod_perl extensions for tighter integration of the server and database. The message boards are provided using the WWWThreads discussion forum system. The U.S based website is served using machines running Digital Unix, while the European website is served using Intel based PCs running FreeBSD.

It is interesting to note that the company makes available all data files and software required to construct a copy of the database on any PC for non-commercial use. Currently the database can be constructed and accessed locally through various interfaces like plain text files, Latex files and a non-graphical Windows interface.

2. Database Construction in IMDb

2.1. Overview

The IMDb's building blocks are lists of data covering various categories of the information that is cataloged. For example, “Actors” is a list, but it just contains their acting credits. Biographical information about actors is in the “Biographies” list. A database manager handles each list. When information on a title or person is submitted (explained in Section 2.2), it is checked by the additions processing robot to make sure it is correctly formatted, and then information that is properly formatted is broken up and distributed to the pending additions files for the appropriate lists.

2.2. Adding Information to the Database

The IMDb uses two methods of adding information to the database: Web forms or e-mail forms. Information we obtained from the submission procedure indicates that it is simpler to use the web forms rather than use the humungous e-mail format, if the only addition to the information is an update. If all together new information has to be submitted, a user has to request and obtain templates of format from IMDb via e-mail. The information to be submitted has to be formatted according to the templates and submitted for validation.

The information presented for a title at IMDb consists of five main sections: The Title, Production Status, Cast, Crew, and Miscellaneous. Miscellaneous entries would consist of the following categories: Alternative Titles, Alternate versions, Awards, Business information, Certification, Comments, Country of origin, Distributors, DVD details, Genre(s), Keywords, Languages, Movie literature, Filming location(s), Miscellaneous companies, Production company, Plot outline, Date of release, Running time, Sound mix, Soundtrack information, Special effects company, Taglines from posters, Technical information, Trivia, URL's to other
sites and Year of first release. We felt that it is important to list these categories because we have described these categories as subclasses of the Movie class in our ontology.

Artist related information consists of Name, Sex, Filmography, TV Guest Appearances, Biographical Information, URLs and Awards/Nominations.

The e-mail interface for adding data is the best solution when one wants to add information about multiple titles and/or multiple people in a single batch. For example, when one wants 10 reviews to be linked, one can either look up each movie or add the reviews individually, using the web additions process 10 separately times, or simply send one e-mail. This is NOT simpler than the web process. One will have to read and follow directions for every type of information she wants to add.

2.3. Guides List

For every type of information in the database, the database manager responsible for it has created a formatting guide for it. The database is broken up into lists and a list is not comprehensive. The Actors list contains the names of actors, the titles they were in and the names of characters they played, but not biographical information about them. Biographical information about the actors is in the Biographies list. One has to review the list of guides and try to pick the guide that looks most appropriate to the type of information one wants to add. Each guide is broken up into four sections.

- Section manager – People who handle this section of the database and an e-mail address for contacting them.
- Additions – These sections mention how to add the different types of information that the guide covers.
- Deletions – If one is trying to remove an erroneous piece of data, this guide mentions the steps to be followed to do so.
- Corrections – The way to make a correction is to add the right information and delete the wrong information. If the information one is trying to change is too subtle or complicated to be handled in that manner, this guide instructs one on how to e-mail in a free-form description of the correction.

We now describe a small example to show the update process in the IMDb.

Example update to send:
Let us assume we have a movie called “All the King's Eggs” that is due out in 2003. Assume we already have a director and two of the stars listed, but Don Ribblefinger, the actor is unlisted, and he has just signed on to play a character named Ollie Garchy. The email entry would look as follows:

```
NAME
Ribblefinger, Don|

ACTOR
Ribblefinger, Don|All the King's Eggs (2003)||Ollie Garchy|

END
```

If Don was already in the database, the update sent should be as follows:

```
ACTOR
Ribblefinger, Don|All the King's Eggs (2003)||Ollie Garchy|
```
2.4. Regular Expression Matching

IMDb provides a regular expression match facility, in addition to fixed pattern matching, to speed up the information lookup process. It provides Perl-like matching capabilities, but in a limited fashion. Anchors supported are ^ (beginning of title), $ (end of title) and \b (word boundary). It also supports special characters such as + (one or more instances of the characters right before it), * (zero or more instances of the characters right before it), ? (Zero or one instances of the letter right before it), [] (specify a variety of different characters that must match, or must not match), . (zero or one instances of any character) and ^ (negation when placed in brackets). The matching process also supports Ranges and Sequences using \d for digits and \w for matching any digit or letter.

3. IMDb Semantic Web Service Ontology

3.1. Overview

We have described, in brief, the current format and structure of the IMDb. We have also described the information sharing and updating procedures in brief. We believe that there is enough scope for vastly improving the knowledge representation on the IMDb and ensuring that it described using a single ontology that everybody can use to view and update the information. In this project we have attempted to design a detailed ontology for the IMDb. In addition to providing the standard information about people like Actors, an attempt has been made to describe the information in such a way that programs can use it to reason and infer certain facts.

Information on people is easy to describe as we experienced. We believe that movie related information cannot be restricted to description of people, although they are a central part. In addition, this information is made up of processes and events that are associated with a movie as it goes through the phases of being made and being screened. We intended the term “processes” to encompass the description of everything related to every aspect of movie making and selling. However, we found that most of the “processes” were described chiefly by the people that were engaged in them. In fact, the information on the IMDb reflects this fact. We feel that the ontology we have described is quite comprehensive as version 1.0. The entire ontology was described in RDFS using the Protégé-2000 RDF editor. The tool did not support the rdf:Bag resource. We, therefore, described as much information possible using the tool and hand coded the rest.

The entire ontology and the movie instance created have been placed on the web. The URLs are as follows.

IMDb ontology: www.cs.umbc.edu/~kamdar/IMdb2.rdf

Instance of Movie “Gladiator” described using this ontology: www.cs.umbc.edu/~kamdar/IMdb2.rdf

We have divided all relevant IMDb information into three components. The root of our ontology is the “IMdb” object. This object has no properties of its own. The three components – Movie, AwardsandNominations and SchedulesandLocation – are subclasses of the “IMdb “ object. We describe each component in detail.
3.2. The Movie Component

This component contains 20 main processes described as subclasses of the Movie class. These processes are: Acting, Advertising, Art, Business, Casting, Cinematography, Composition, Costume design, Direction, Distribution, Editing, Makeup, Music, Production, Review, Sound, Special effects, Stunts and Visual effects. Two generic subclasses – MoviePerson and MovieCompany – are used to describe people and organizations (profit and non-profit) related to movies. The list of 20 processes was created after studying many web pages describing movies on the IMDb website. The Movie class itself has a set of basic properties. These are described in a subclass called MainMovieInfo. For the ontology instance that we created, the MainMovieInfo subclass looks like this:

```xml
<IMdb:MainMovieInfo rdf:about="&IMdb;IMdb2_00241"
    IMdb:Color="Technicolor"
    IMdb:Language="English"
    IMdb:OriginCountry="UK/USA"
    IMdb:Title="Gladiator">
    <IMdb:DateOfRelease rdf:resource="&IMdb;IMdb2_00423"/>
    <IMdb:GenreList rdf:resource="&IMdb;IMdb2_00440"/>
    <IMdb:CertificationList rdf:resource="&IMdb;IMdb2_00441"/>
    <IMdb:LengthList rdf:resource="&IMdb;IMdb2_00442"/>
</IMdb:MainMovieInfo>

We described the “DateOfRelease” property as a Date, which we described separately in the ontology. The movie belongs to different Genre like Action and Drama. Therefore, we created a bag of the GenreInstance class. We applied similar rules to Certification and Length properties. As an example of the description of a particular process we show the Music class, which brings together information in many other subclasses:

```xml
<IMdb:Music rdf:about="&IMdb;IMdb2_00365"
    IMdb:backgroundMusicType="regal, royal, war"
    IMdb:musicMood="sombre">
    <IMdb:musicDirector rdf:resource="&IMdb;IMdb2_00366"/>
</IMdb:Music>

<IMdb:MoviePerson rdf:about="&IMdb;IMdb2_00366"
    IMdb:countryOfBirth="Germany"
    IMdb:movieSkills="Music direction"
    IMdb:nameOfPerson="Hans Florian Zimmer">
    <IMdb:dateOfBirth rdf:resource="&IMdb;IMdb2_00367"/>
    <IMdb:awardsList rdf:resource="&IMdb;IMdb2_00427"/>
</IMdb:MoviePerson>

<IMdb:AwardCollection rdf:about="&IMdb;IMdb2_00427">
    <rdf:li><IMdb:awardsWon rdf:resource="&IMdb;IMdb2_00368"/></rdf:li>
</IMdb:AwardCollection>

<IMdb:Date rdf:about="&IMdb;IMdb2_00367"
    IMdb:Day="12"
    IMdb:DayOfWeek="Thursday"
    IMdb:Month="9"
    IMdb:Year="1957"/>

<IMdb:AwardsAndNominations rdf:about="&IMdb;IMdb2_00368"
    IMdb:awardCategory="Best Music, Original Score"
```
IMdb:awardName="Oscar"
IMdb:awardResult="Nominated"
IMdb:awardYear="2001">
   <IMdb:awardOrganization rdf:resource="&IMdb;IMdb2_00309"/>
</IMdb:AwardsAndNominations>

The class represented by &IMdb;IMdb2_00309 was reused many times. This is because it represented the Academy of Motion Picture Arts and Sciences (MPAA), that presents the Oscar awards every year.

<IMdb:MovieCompany rdf:about="&IMdb;IMdb2_00309"
                     IMdb:companyBusiness="Professional Non-Profit Honorary Organization"
                     IMdb:companyCreatedYear="May, 1927"
                     IMdb:movieCompanyName="Academy of Motion Picture Arts and Sciences">
   <IMdb:movieCompanyAddress rdf:resource="&IMdb;IMdb2_00310"/>
</IMdb:MovieCompany>

<IMdb:AddressOfPlace rdf:about="&IMdb;IMdb2_00310"
                      IMdb:BuildingNumber="8949"
                      IMdb:CityName="Beverly Hills"
                      IMdb:CountryName="USA"
                      IMdb:StateName="California"
                      IMdb:StreetName1="Wilshire Boulevard"
                      IMdb:ZIP="90211-1972"/>

In order to describe the address of the MPAA, we defined an ontology for Address, that has the properties shown above. Although we could have used the ontology for Place described in DAML, we felt that at least half of the properties in that description would remain empty in this instance.

3.3. The Schedules and Location Component

We had envisaged that this component would consist of detailed information on show timings and locations (either countrywide or zip code based). Thus, a query such as "Show cinemas in which Gladiator is screening" would return the address of the cinema along with show times, if the movie were currently being screened. We also assumed that if the movie were not being screened anywhere, it could potentially return information about other media (DVD, video) on which this movie is now available. However, we found that this linking was not logical and that we would need to change the design to accommodate this feature. In the case of the instance we have created, the movie Gladiator is no longer being screened, so we had no information to fill in. We show the ontology of the SchedulesAndLocation component.

<rdfs:Class rdf:about="&IMdb;SchedulesAndLocation"
           rdfs:comment="Component for describing show timings and locations">
   <rdfs:subClassOf rdf:resource="&IMdb;IMDb"/>
</rdfs:Class>

<rdfs:Class rdf:about="&IMdb;LocationDetail"
           rdfs:comment="detailed info of theatres screening movies all over the US">
   <rdfs:subClassOf rdf:resource="&IMdb;SchedulesAndLocation"/>
</rdfs:Class>

<rdfs:Class rdf:about="&IMdb;LocationInfo"
Project Report  
CMSC 771 Spring 2001

3.4. The Awards and Nominations Component

This component deals with information related to awards and nominations received by people associated with a movie. It describes each award in detail. As shown in the example in Section 3.3, an award has both a category and a name associated with it, along with the year it was awarded and the organization that awarded it. In a manner similar to the Genre and Length information, each person is associated with a bag of awards. Thus, the AwardsandNominations class can have multiple instantiations, and each instance is placed into the bag associated with a particular person.

3.5. The MoviePerson subclass

We note this is one of the most important subclasses described in the entire ontology. Every person related to movies in some way must be described using this subclass. In this ontology, for simplicity, we created the MoviePerson subclass as a nearly complete component by itself. That is, no properties are inherited from, say, the Person class described elsewhere. It would be a good idea to create it under such a hierarchy, but the Person class itself needs to be completely defined for it to be reusable. In this project, we felt that the only things that could be inherited from the Person class would be Name and Date of Birth. The other properties we described are specific and non-specific to people related to movies. For example, a MoviePerson has the movieSkills property that describes the movie related skills that this person has. However, this class also has the moneyEarned property, which could be applied to any person.

3.6. The MovieCompany subclass

This is another important subclass. There are many organizations, profit and non-profit, related to movies. There is a generic and complete way to describe organizations in general. We believe that we have described a movie related company in sufficient detail for this project. However, it may be possible to further improve upon it by describing the properties with finer granularity. For example, the companyBusiness property (currently a Literal) can be further described as profit or non-profit, qualified with the earnings if any for the current quarter, etc. An instance of the MovieCompany subclass, MPAA, was shown in Section 3.2.
4. Conclusions

We have designed an ontology using RDFS for the Internet Movie Database and described an instance of a movie using this ontology. We believe that this method of describing the information on the IMDb will revolutionize the way information is shared and searched. We have described the current cumbersome process used to update information in the IMDb. We can easily imagine the ease with which the same information can be updated using the ontology. A user who wants to submit information about a particular person or title simply describes an instance in a text file using RDF and submits it. A Prolog engine can easily validate the information and determine if a previous instance with the same values exists. If it does, it simply replaces it with the new information.

We note that we have not created instances of every possible property, because some properties didn’t have values or had insufficient information. It is also quite possible that the ontology itself is not complete. It is possible to improve it further, given time and resources.