



UNIVERSITY  
OF TRENTO - Italy



DIPARTIMENTO DI INGEGNERIA E SCIENZA DELL'INFORMAZIONE

– KNOWDIVE GROUP –

# Entertainment media formal model

---

Document Data:

Reference Persons:

Davide Corradini (190996), Andrea Tupini (194578)

---

© 2018 University of Trento  
Trento, Italy

KnowDive (internal) reports are for internal only use within the KnowDive Group. They describe preliminary or instrumental work which should not be disclosed outside the group. KnowDive reports cannot be mentioned or cited by documents which are not KnowDive reports. KnowDive reports are the result of the collaborative work of members of the KnowDive group. The people whose names are in this page cannot be taken to be the authors of this report, but only the people who can better provide detailed information about its contents. Official, citable material produced by the KnowDive group may take any of the official Academic forms, for instance: Master and PhD theses, DISI technical reports, papers in conferences and journals, or books.



---

## **Index:**

<b>1 Scenario description</b>	<b>1</b>
1.1 Personas . . . . .	2
1.2 Storytelling . . . . .	2
<b>2 Model formalization description</b>	<b>4</b>
2.1 Process of creation . . . . .	4
2.1.1 Seeking for available data . . . . .	4
2.1.2 Relations . . . . .	4
2.1.3 Inspiration from other ontologies . . . . .	4
2.1.4 Improvements to support users' queries . . . . .	5
2.1.5 Visualization of the informal model . . . . .	5
2.1.6 Core, auxiliary and common entities . . . . .	5
2.1.7 A continuous development... . . . . .	7
2.2 Classes . . . . .	7
2.3 Object Properties . . . . .	9
2.4 Datatypes . . . . .	10
2.5 Data Properties . . . . .	11
<b>3 Lexical information upload</b>	<b>12</b>
3.1 Synsets used in our ontology . . . . .	12
<b>4 Top-level grounding</b>	<b>15</b>
4.1 Encountered issues . . . . .	17
<b>5 Model visualization</b>	<b>18</b>
<b>6 Generalized queries</b>	<b>20</b>
6.1 All supported queries . . . . .	20
6.2 Example Queries . . . . .	22
6.2.1 Example queries for: Media consumer . . . . .	23
6.2.2 Example queries for: Relation explorer . . . . .	23
6.2.3 Example queries for: Market Analyst . . . . .	24
6.2.4 Example queries for: Rights manager . . . . .	24
<b>7 Final considerations and open issues</b>	<b>25</b>
<b>8 Evaluation of the model</b>	<b>26</b>

---

## Revision History:

Revision	Date	Author	Description of Changes
0.1	16.01.2017	Subhashis Das	Document Created
0.2	16.01.2017	Subhashis Das	Document Modified
1.0	16.12.2018	Davide Corradini, Andrea Tupini	Document finalized

## References to existing KnowDive internal reports:

- Mind Product Etype Graph Description

---

# 1 Scenario description

A scenario description usually involves real people and real actions. The absence of these due to the theoretical nature of this project has driven us to write four use cases describing some possible usage of our application.

## Use case 1:

Sarah, 42 years old housewife, wants to know information about the movie *Titanic*: who directed it, in what year it was published and in which countries it was made. She opens our application and searches for the movie with name *Titanic*. The system will show her the list of movies with said name, she'll choose the correct one and open the media page. There, she can see all the information regarding the movie, among which she'll also find the information she's interested in. Later, she wants to know what other movies that were directed by the same director, so she clicks the director's name and is sent to the director's page. There she can see, among other things, the movies directed by this director.

## Use case 2:

John, 16 years old high-school student, just saw a movie called *The Hobbit* in the theater. He really liked it and wants to know if there is a book on which the movie is based because he would like to read it. John opens our application, searches for the movie and opens its media page. John will see that the movie is based on a book, and clicking on the book's name will go to the book page. John will be also able to see who wrote it, when it was first released, and a link that takes him directly to the Amazon page where he can buy the book.

## Use case 3:

Robert, 24 years old intern at a marketing company, was asked to create a report about the most successful genre of movies in the period 2016-2018. He knows that he can find this information in our application, so he heads there and searches for the average revenue of movies released in the 2016-2018 time period, by genre. The system will display to Robert the list of all movie genres, and the average revenue that movies published in 2016-2018, in said genre, obtained. Robert will then proceed to analyze this data to complete his report.

## Use case 4:

Elyse, 37 years old, is a rights manager. She helps people in organizing events. Next Saturday a Pink Floyd tribute band is performing in town, and they need a license in order to execute a copyrighted music. On Sunday, a movie will be shown at the local theatre: a license is needed as well. Elyse can search for the involved medias (the movie and the songs that will be performed) and discover the copyright holders. She will contact them in order to get a license.

As you probably understood by reading the uses cases, our application's goal is to provide a database of information concerning a specific category of mind products: the entertainment media products. The application is focused on movies, TV series, music and books. The purpose is to provide a repository of data about the above, allowing users to search for information and relations among them. These kind of products may sometimes be related: for example, some movies are based on books, or music is used as part of their soundtrack.

The application will not hold the actual media content (movies, music, etc...), instead it will only contain information about the product and provides the link to the content whenever possible.

To extend the capabilities of the application, some auxiliary entities such as actors, directors, publishers, authors, artists, etc... will be provided as well.

Along to the most common use cases, the application is able to perform more advanced tasks, for example statistical tasks.

## 1.1 Personas

The following is a formal description of the different types of user we expect for our application:

### Media consumer

We expect this to be the most common type of user. He/she will basically use the application to search for information about a specific media product he/she is interested in. He/She is a basic user and doesn't require technical knowledge. These users can be of any age, just old enough to be able to use a computer and being interested in this kind of information (we suppose age  $\geq 7$ ).

### Relation explorer

This is also a common type of user, but he/she wants more specific information about relations between media products. As with the media consumer user, this one doesn't require advanced technical knowledge and must be old enough to use a computer and be interested in this information (also age  $\geq 7$ ).

### Market analyst

Advanced user who will be able to see statistical data from the platform. This user does have moderate technical knowledge. Because of this, the estimated age for this user is  $\geq 16$ .

### Rights manager

This is a (legal professional) advanced user which searches for data about authors or publishers in order to verify whether a content is guilty of rights infringement. Does not require advanced technical knowledge but does require legal knowledge. We expect this kind of user to be  $\geq 23$  years old.

## 1.2 Storytelling

Following are table that formally represent the sequence of actions that happened in the cases exposed in the last section.

User	Real World Action by Persona	System action
Sarah	Submits a search, using the search functionality of the application, for a movie with name <i>Titanic</i> .	System fetches movies which have a similar name and displays the list of results to Sarah.
Sarah	Once the results are given, she clicks on the item which she thinks corresponds to the actual movie she wants to know information about.	System fetches specific movie information and shows it in the media page for said movie.
Sarah	On the media page she sees the movie information she was looking for. Now Sarah clicks on the name of the director with the objective of knowing more information on him/her.	System fetches information on that specific director and displays it to Sarah in a person page.

Table 1: Storytelling table for use case 1

User	Real World Action by Persona	System action
John	Submits a search, using the search functionality of the application, for a movie with name <i>The Hobbit</i> .	System fetches movies which have a similar name and displays the list of results.
John	Once given the results of his search he clicks on the item which he thinks corresponds to the actual movie he wants to know information about.	System fetches specific movie information and shows it in the media page for said movie.
John	In the media page John sees that the movie is based on a book. He clicks the name of the book to see more information on it.	System fetches book information and shows the media page for it.
John	Here he sees the information he was looking for. He clicks on the link to buy the book.	System redirects John to the appropriate Amazon page.

Table 2: Storytelling table for use case 2

User	Real World Action by Persona	System action
Robert	Asks the application for the possibility to do an advanced query.	System redirects Robert to the appropriate form.
Robert	Asks the system for the average revenue of movies published during 2016-2018, separated by genre.	System fetches the requested information and shows it to Robert, sorted by average average income.

Table 3: Storytelling table for use case 3

User	Real World Action by Persona	System action
Elyse	Wants to check who is the copyright holder of a specific media item. So she searches for said item by name.	System fetches items of the same time which have a similar name and displays the list of results.
Elyse	Once given the results of her search she clicks on the item which she thinks corresponds to the actual item she wants to know information about.	System fetches specific item information and shows its media page.

Table 4: Storytelling table for use case 4

---

## 2 Model formalization description

In this section we will analyze the process of formalization of our model, taking into consideration the input tech report provided by *KnowDive Group*, some other public ontologies (for example from *schema.org* and *Linked Open Vocabularies*) and the availability of data concerning our interests on some online platforms.

### 2.1 Process of creation

#### 2.1.1 Seeking for available data

The first step when creating our ontology was referencing to already existing services which we knew provided the kind of information we were interested in. We identified 3 main sources to use as references:

**IMDb** [2] was chosen as the main reference when it comes to movies and TV series items. We also took into consideration how they stored information about actors, directors and producers.

**Goodreads** [1] was our main references for books and, more in general, for all literary works. We saw how different kinds of work were stored and also how these were related with their writers. *Goodreads* deals mainly with entertainment works, so it is particularly useful for our application (where we deal with Novels only).

**Spotify and TIDAL** [3, 4] gave us information about what kind of data is stored for musical items. We also saw how the different pages of authors and music groups were represented and related to their music.

#### 2.1.2 Relations

Keeping in mind the available data we found, we started to think about how we could join media items together.

Our first insight was that the items are commonly related among themselves via an *inspired-by* relation. This means that an item (be it a movie, novel, or song) can inspire other items. We identified the following possible *inspire relations*: books can inspire movies, and viceversa; and both books and movies can inspire music.

Then we found that these kind of media can be part of one another: a movie script (which is a literary work) is part of a movie, a song can be part of a movie soundtrack, the lyrics (another literary work) are part of songs.

Finally, we saw that these relations also extend to the creators and participants of said media items. More formally, a person can have different roles and it can be involved with different items of different types. We proceeded to identify the main roles of: writer, director, author, musician. We also saw that a person can be any of these at a given time, without them being mutually exclusive. This lead us to a new way to relate items (ie, by means of their creator). Having this information properly modelled would let us know if an actor has written any books, or if a director has released any musical album.

#### 2.1.3 Inspiration from other ontologies

With the relations from the previous section in mind, we started to think how model in our ontology. However, instead of starting from scratch, we decided first to take a look at pre-existing ontologies which already cover the subjects we're interested in.

First, we considered the "*Mind Product Etype Graph Description*" [6] paper by *KnowDive Group* at UNITN, as suggested by the teachers. Inside this paper, that deals with the more general concept of mind product, we found some interesting relations between entities that helped us while building our ontology. Furthermore, since these

---

relations were created by the same group that created the CSK, using this reference has probably helped us in the top-level grounding task (section 4) providing many equivalent classes.

Later, we moved to some public domain ontologies.

We first considered The Music Ontology [8], which is a popular ontology used in the music domain. We found it a bit too specific for our purposes, but regardless of this we took a lot of inspiration from it, especially for the way in which they model musicians, musical albums and musical groups.

Then we considered the Bibliographic Ontology [5]. This is another popular ontology, specifically used for the modelling of bibliographical items. However, as with the music ontology, we found it to be too specific for our purposes. But as for the music ontology, we took a lot of inspiration from it, especially in how it models the different kind of literary work items and how they're related to authors and to producers.

Finally we considered how schema.org [9] was modelling the Movie entity. We just saw the movie entity since schema.org is quite a big ontology, and we found that the level of abstraction provided by schema.org was exactly what we needed. In fact, we used it also to refine the ideas we had obtained from Music Ontology and from Bibliographic Ontology. This was done by referencing to how schema.org was modelling the different entities (songs, music groups, music albums, writers, directors, etc ...) and by adapting our model if necessary.

We tried to keep our ontology's structure as similar as possible to schema.org, with the idea that it should be compatible. However, we didn't include many properties and/or relations in our ontology since we felt they were not necessary for the purposes of our application.

#### 2.1.4 Improvements to support users' queries

Having a clearer idea of possible model of entities and relations, we started thinking about the users (*the personas*) that we expect to use our application. We made a list of possible queries/actions each of them was going to perform, and how they expected the system to behave. These queries have been formally defined in subsection 6.1.

Then, we reasoned about new entities or relations necessary to support the queries/actions listed before in order to comply with the users' needs.

#### 2.1.5 Visualization of the informal model

After all this process was done, we had a basic, informal, schema of the model, with entities, relations, and some of the attributes. An image of said schema can be seen in the next page (Figure 1).

Note that this schema does not exactly reflect the entities and relations used in the final ontology. We did some changes while creating the final ontology because we noticed that some entities/properties were missing and some other could be joined. The final schema of the ontology will be shown later in section 5. Regardless of this, the image has been included anyway as part of the documentation of our *ontology building* process.

#### 2.1.6 Core, auxiliary and common entities

Based on the model (informal schema) defined above, we were able to identify the following Core, Auxiliary and Common entities that we would need to use for our project. These are:

##### Core entities

CinematographicWork, LiteraryWork, MusicalWork, Movie, TVShowEpisode, CinematographicWorkScript, MusicLyrics, and Novel.

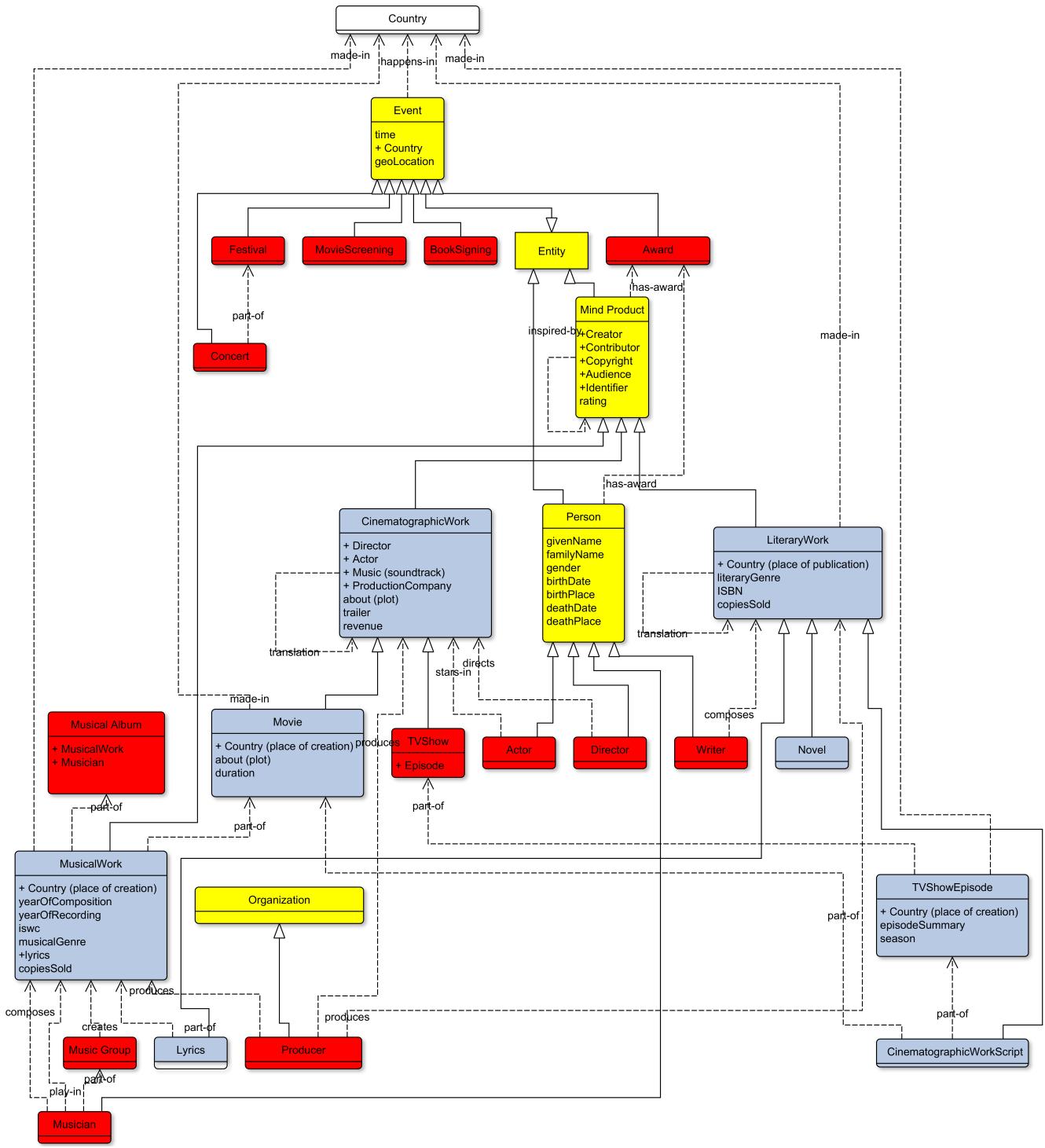


Figure 1: Informal schema generated in yED when thinking about which entities and relations we would need for the ontology.

---

## Auxiliary entities

Actor, Director, Musician, Writer, MusicalGroup, MusicalAlbum, TVShow, Award, BookSigning, Concert, Festival, MovieScreening, and Producer.

## Common entities

Entity, Event, Organization, Person, Collection, and MindProduct.

*Note: The actual definition of the entities can be found in subsection 2.2*

Since our project deals mainly with how MindProduct items are related, it makes sense to choose the different types of these as our core entities. Then, since we're also interested in knowing how these MindProduct items are indirectly related, through their creators and/or participants, we included these as part of the auxiliary entities. We're also including events as an auxiliary entity because we're interested in showing information about events related to mind products as well.

Finally, general concepts like Entity, Event, etc, have been selected as the common entities, since we're using them in our application but they're very general and can also be considered as part of other entities. In fact, all our common entities are directly mapped to entities in our top-level ontology; in section 4 we'll talk more about this.

### 2.1.7 A continuous development...

Once we had this informal model, we worked on it continuously so as to improve it and make sure that we were capturing all the relations we were interested in. This was quite a lengthy process since we always saw something that was not 100% correct, or that we could add some entity/relation so as to make the model more precise. During this part we experienced what was said during the course many times: that an ontology is in constant evolution.

In the next page (Figure 2) is the current, final, shape of our ontology. We can easily see that the final version is much more complex than the one presented above.

The visualization can be a bit confusing, but the same structure of entities can be seen in subsection 2.2, and the object properties that link said entities can be seen in subsection 2.3. In section 5 we will present a more detailed visualization of the final model.

## 2.2 Classes

In this subsection we'll present a listing of all the entities used in our ontology and a brief description of their meaning. Note that these descriptions don't come from *WordNet*, those will be exposed later, in section 3. The indentation in the list below corresponds to the hierarchy of entities: more indented entities are sub-classes of less indented ones.

**Collection** This describes entities that are composed of other entities.

**MusicalAlbum** A collection of MusicalWork items.

**MusicGroup** A group composed of musicians which create and perform MusicalWork.

**TVShow** A collection of TVShowEpisode items.

**Entity** Basic entity. Stands for a general *thing*.

**Country** A country, in the geopolitical sense. It is used with the objective of identifying a location.

**Event** Basic event, something that occurs in a specific date and time.

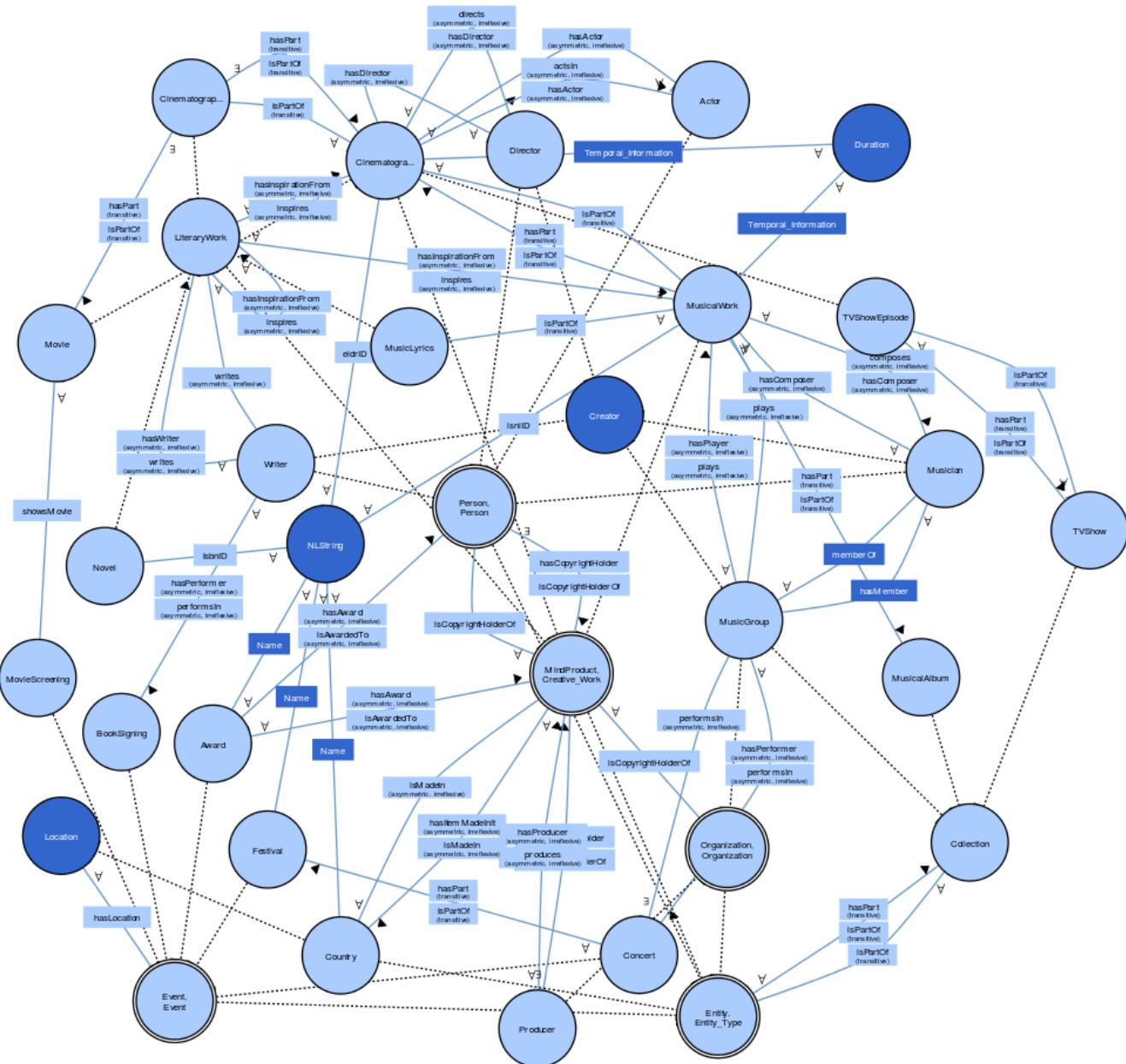


Figure 2: Visualization of the final ontology.

---

**Award** An award or recognition that can be given to either a *Person* or a *MindProduct*.

**BookSigning** A book signing event, where a writer signs books he/she has written.

**Concert** A concert, a performance done by a specific musical band.

**Festival** Collection of concerts, which happens in a specific location and time.

**MovieScreening** A show, where a specific movie is played.

**MindProduct** Everything that is the output of a creative process.

**CinematographicWork** Creative items composed of video and audio.

**Movie** A movie, is shown in cinemas, and is normally of greater length than *TVShowEpisode*.

**TVShowEpisode** A single episode from a TV show, is part of a *TVShow*.

**LiteraryWork** Creative item that is composed exclusively of text.

**CinematographicWorkScript** The script for a *CinematographicWork* (movie or tv show) that dictates the actor's speech and behaviour.

**MusicLyrics** The text/words/lyrics of a song.

**Novel** Contains a story, and is written/read with the purpose of entertaining.

**MusicalWork** Creative item that is composed exclusively of audio.

**Organization** Unit constituted by members with common goals.

**Producer** Company that is in charge of publishing *MindProduct* items, making them available to other persons besides the creator.

**Person** A human being.

**Actor** Appears in a *CinematographicWork*, pretends to be someone else, follows a *CinematographicWorkScript*.

**Director** Directs a *CinematographicWork*.

**Musician** Is the composer and player of *MusicalWork* items.

**Writer** Is the creator of *LiteraryWork* items.

## 2.3 Object Properties

Following is a list of all the object properties used in our ontology. As in the case of the list of entities, the indentation denotes the structure of the hierarchy.

**Relational\_Property** Category of properties that denote relation between entities.

**hasActor** *CinematographicWork* has an *Actor*.

**actsIn** An *Actor* performs the act of *acting* in a *CinematographicWork*.

**hasComposer** *MusicalWork* has a *Musician* as its composer.

**composes** *Musician* composes a *MusicalWork*.

**hasCopyrightHolder** The rights of a *MindProduct* item are held by a *Person* or *Organization*.

**isCopythingHolderOf** A *Person* or an *Organization* have some rights over a *MindProduct* item.

**hasDirector** *CinematographicWork* has a *Director*.

---

**directs** A Director *directs* a CinematographicWork item.

**hasInspirationFrom** Denotes that a MindProduct was inspired by another MindProduct.

**inspires** Denotes that a MindProduct inspired the creation of another MindProduct.

**hasPerformer** A Concert is performed by a MusicGroup.

**performsIn** A MusicGroup performs in a Concert.

**hasPlayer** A MusicalWork item has a player which is a Musician.

**plays** A Musician is the player of a MusicalWork item.

**hasProducer** A MindProduct is produced by a Producer.

**produces** A Producer produces a MindProduct.

**translantionOf** A LiteraryWork item is the translation of another LiteraryWork item. This is a symmetric property, so if a literary work is a translation of another we also know that the inverse relation is true.

**hasLocation** Specifies a location for a specific Entity.

**hasItemMadeInIt** A Country has a MindProduct made in it.

**isMadeIn** A MindProduct is made in a specific country.

**hasWriter** A LiteraryWork item is written by a Writer.

**writes** A Writer writes a LiteraryWork item.

**Parthood\_Property** Category of properties which denote a belonging or part composition.

**showsMovie** A MovieScreening event shows a specific Movie (because of this it is *part of* the event).

**hasPart** Entity is composed of another entity.

**isPartOf** Entity is part of another Entity.

**Context\_Property** Category of properties which describe a context of an entity.

**hasAward** A Person or MindProduct have an Award.

**isAwardedTo** Award is given to a MindProduct or Person.

## 2.4 Datatypes

Below is a listing and a brief description of the datatypes we defined for our ontology.

**rating** in range [0, 10], defines the average rating for any MindProduct. (During the data integration phase all the ratings from different sources should be normalized to this range)

**cinematographicGenre** defines the possible genres for any CinematographicWork item. The possible genres are the following:

*Action, Adventure, Animation, Biography, Comedy, Crime, Documentary, Drama, Family, Fantasy, Film Noir, History, Horror, Music, Musical, Mystery, Romance, Sci-Fi, Short, Sport, Superhero, Thriller, War, Western*

---

**literaryGenre** defines the possible genres for any LiteraryWork item. The possible genres are the following:

*Action, Adventure, Anthology, Art, Autobiographies, Biographies, Children's, Comics, Cookbooks, Diaries, Dictionaries, Drama, Encyclopedias, Fantasy, Guide, Health, History, Horror, Journals, Math, Mystery, Poetry, Prayer books, Religion, Spirituality & New Age, Romance, Satire, Science, Science fiction, Self help, Series, Travel, Trilogy*

**musicGenre** defines the possible genres for any MusicalWork item. The possible genres are the following:

*Blues, Classical, Country, Electronic, Folk, Hip-hop, Jazz, Reggae, Religious, Rock, Traditional*

## 2.5 Data Properties

In this section we present the data properties that we've used in our ontology. All of the datatypes exposed above also have a corresponding data property of the same name, to save space these have not been re-included below.

**eidfID** Universal ID to identify CinematographicWork items.

**isbnID** Universal ID with which to identify Novels.

**isniID** Universal ID to identify MusicalWork items.

**releaseDate** the date and time at which a specific MindProduct item was released to the public.

**wasGivenOn** the date in which an Award was given.

**numberOfSoldCopies** defines the number of copies that have been sold of a specific MindProduct item (if applicable).

**storyPlot** a summary of the plot of a story.

**revenue** is used to indicate the revenue that a specific MindProduct item has generated. It is a *float* value.

**hyperlink** a link is a hyperlink to an external resource regarding a specific MindProduct item. For example, a link to buy a Novel in an online store, or a link to the trailer of a Movie.

### 3 Lexical information upload

The goal of the lexical information upload task is to provide an accurate description of the concepts used in our ontology. Delivering such information inside our ontology allows an easy understanding of the exact meanings of the terms and concepts we used and makes life easier to people that may reuse the ontology or improve it in the future.

The lexical information upload procedure has been conducted thanks to *WordNet* [7], a lexical database that provides a short definition of concepts and synsets, basically sets of words with the same meaning.

*WordNet* provides an unique synset ID for each concept. For example, in *WordNet* the director of a movie is defined as "*the person who directs the making of a film*" with synset ID 10107969 and "*film director*" and "*director*" as synonyms. Providing this information explicitly, we can differentiate the movie director from any other kind of director, such as the one concerning with an orchestra.

We related the synset IDs both to the classes and the object properties in our ontology. We used the annotation `synset_ID` to store the synset IDs.

Since, unfortunately, *WordNet* does not contain all the existing and known concepts, it has happened that we had to create some custom synsets. According to the procedure we learned in class, we set the `synset_ID` to -1 and added some other annotations to explain the concept. In particular, `gloss` containing a short description, `parent_synset_ID` containing the synset ID of the parent concept, `POS` containing the part of speech of the term (noun, verb, adjective, etc...) and `synonyms` containing, obviously, a list of synonyms, words with the same meaning.

#### 3.1 Synsets used in our ontology

This sections is made of three tables, containing respectively the new synset we created for Entities (Table 5), the existing synsets used for the Entities (Table 6) and the existing ones used for the Object Properties (Table 7).

Entity name	Parent synset ID	Definition
CinematographicWork	04014270	Creative items composed of video and audio.
BookSigning	07461861	A book signing event, were a writer signs books he has written.

Table 5: Synsets we created for our Entities.

We had the necessity to create these two synsets because:

**CinematographicWork** the difference between a CinematographicWork and a Movie is subtle, the CinematographicWork includes all the outputs of the creative process that are composed of a sequence of pictures which create the illusion of motion. A Movie on the other hand is a CinematographicWork which has been made by a director and producer, with the explicit objective of having it shown in cinemas or in the "viewers" personal media reproducer.

**BookSigning** we could not find any term in *WordNet* referred to this kind of event.

<b>Entity name</b>	<b>Synset ID</b>	<b>WordNet definition</b>
Actor	09784701	A theatrical performer.
Award	06709228	A tangible symbol signifying approval or distinction.
CinematographicWorkScript	07023062	A written version of a play or other dramatic composition; used in preparing for a performance.
Collection	07968050	Several things grouped together or considered as a whole.
Concert	06905814	A performance of music by players or singers.
Country	08185877	A politically organized body of people under a single government.
Director	10107969	The person who directs the making of a film.
Entity	00001740	That which is perceived or known or inferred to have its own distinct existence (living or nonliving).
Event	00029677	Something that happens at a given place and time.
Festival	00518784	An organized series of acts and performances (usually in one place).
LiteraryWork	06375736	Imaginative or creative writing.
MindProduct	03133774	An artifact that has been brought into existence by someone.
Movie	06626039	A form of entertainment that enacts a story by sound and a sequence of images giving the illusion of continuous movement.
MovieScreening	06901395	The display of a motion picture.
MusicGroup	08263534	An organization of musicians who perform together.
MusicLyrics	07066015	The text of a popular song or musical-comedy number.
MusicalAlbum	06604096	One or more recordings issued together.
MusicalWork	07034009	An artistic form of auditory communication incorporating instrumental or vocal tones in a structured and continuous manner.
Musician	10360025	Someone who plays a musical instrument (as a profession).
Novel	06379324	An extended fictional work in prose.
Organization	08024893	A group of people who work together.
Person	00007846	A human being.
Producer	10499603	Someone who finds financing for and supervises the making and presentation of a show.
TVShow	06633086	A program broadcast by television.
TVShowEpisode	06634278	A part of a broadcast serial.
Writer	10813654	Writes (books or stories or articles or the like) professionally.

Table 6: Synsets used in the ontology's Entities.

Object Property	Synset ID	WordNet definition
actsIn	01723161	Play a role or part.
composes	01709353	Write music.
directs	01714176	Guide the actors in plays and films.
hasActor	09784701	A theatrical performer.
hasAward	13289169	Something given for victory or superiority in a contest or competition or for winning a lottery.
hasComposer	09966711	Someone who composes music as a profession.
hasCopyrightHolder	06484839	A document granting exclusive right to publish and sell literary or musical or artistic work.
hasDirector	10107969	The person who directs the making of a film.
hasInspirationFrom	05842891	Arousal of the mind to special unusual activity or creativity.
hasItemMadeInIt	01643749	Create by artistic means.
hasLocation	00027365	A point or extent in space.
hasPart	13831419	Something determined in relation to something that includes it.
hasPerformer	10435383	An entertainer who performs a dramatic or musical work for an audience.
hasPlayer	10360025	Someone who plays a musical instrument (as a profession).
hasProducer	10499603	Someone who finds financing for and supervises the making and presentation of a show (play or film or program or similar work).
hasWriter	10813654	Writes (books or stories or articles or the like) professionally.
inspires	01650224	Supply the inspiration for.
isAwardedTo	02266695	Give, especially as an honor or reward.
isCopythingHolderOf	02244315	Secure a copyright on a written work.
isMadeIn	01643749	Create by artistic means.
isPartOf	13831419	Something determined in relation to something that includes it.
linksTo	00715072	Make a logical or causal connection.
performsIn	01718067	Give a performance (of something).
plays	01728928	Play on an instrument.
produces	02161530	Bring onto the market or release.
showsMovie	02153218	Give an exhibition of to an interested audience.
translationOf	06548380	A written communication in a second language having the same meaning as the written communication in a first language.
writes	01702148	Produce a literary work.

Table 7: Synsets used in the ontology's Object Properties.

For some object properties, the mapping to a synset is quite straightforward. For example, the property *directs* maps correctly to the *verb direct* in *wordnet*. However, for the inverse of these properties (in this example *hasDirector*) the mapping is not so easy to find so, in most of these cases, all the *has\** properties are instead mapped to the *noun* representing the target of the property.

## 4 Top-level grounding

As explained at class, the top-level ontology provides a high level structure which can be shared by applications dealing with different domains. One of the main benefits of having many applications using the same top-level ontology is that these applications will be compatible with each other, since the concepts which are not application specific are shared. Another benefit is that the development of our ontology becomes much easier, since when we link our ontology with a top level ontology we're automatically getting a lot of information already encoded in the top-level ontology. The provided top-level ontology is called *Common Sense Knowledge*, or shortly CSK.

As stated in subsection 2.1, we started building our ontology by identifying which entities and properties were the most important ones, and then we built on top of them to extend our ontology and support the queries we were interested in. So, at the end, we obtained a structure that goes from the very specific to the high level. When we added the CSK to our project we were very lucky since the CSK encoding matches very well with the top level of our ontology. Therefore, we did not have to apply many changes and we had many entities and properties which had their equivalent in the CSK (although there were some discrepancies and some extensions that we added, which will be discussed later in this section).

The following is an image that shows how we mapped the top-level entities from our ontology to the bottom level (leafs) of the CSK.

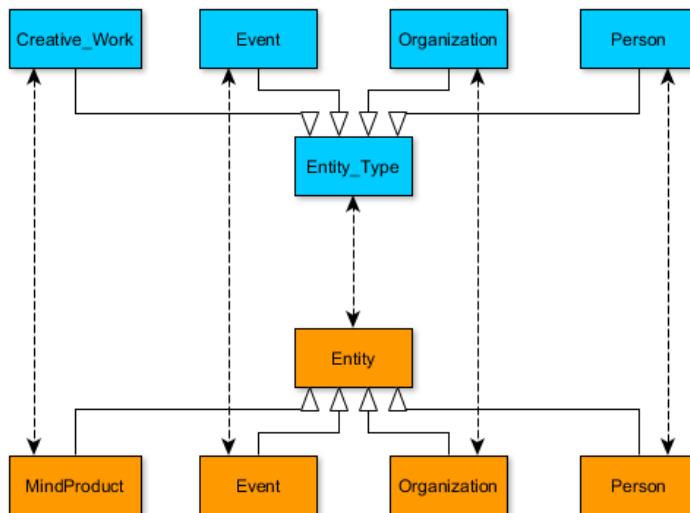


Figure 3: Mapping from the top level of our ontology (orange nodes) to the bottom leafs of the CSK (blue nodes). The bi-directional arrows indicate that these entities are equivalent. Note how very similar is our top level to this bottom level of the CSK (it's basically the same).

Concerning the properties, we only had one pair of properties which matched directly with the pair provided in CSK (the hasPart/isPart properties), but none of the other properties had an equivalent in CSK. However, as part of the integration process we structured them according to categories exposed in the CSK. The separation is shown in Figure 4.

Since we were able to add properties to the already existing one in CSK, we can effectively say that these are an extension of the CSK. There are no object properties in our ontology which have not been mapped in this way.

We did something similar for some of the entities: we placed them as sub-classes of other entities already provided in the CSK. In Figure 5 we can see a figure showing these.

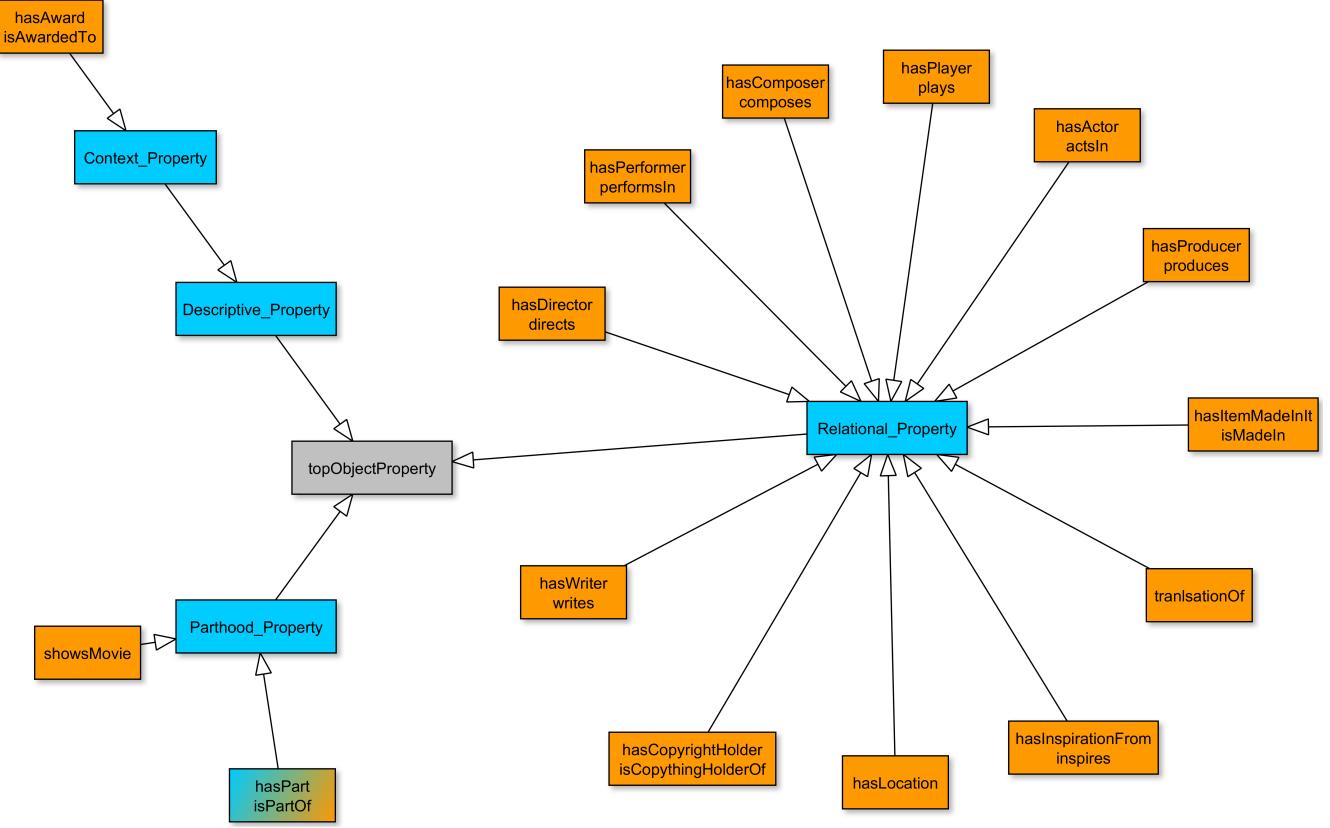


Figure 4: This is how we structured our properties to correspond with the structure provided by CSK. Orange nodes are properties defined in our ontology, blue nodes are properties defined in CSK, half-blue-half-orange are properties which are equivalent in the CSK and in our ontology, the grey node is just the root of all properties. Inverse properties appear in the same node, to make the resulting graph easier to understand.

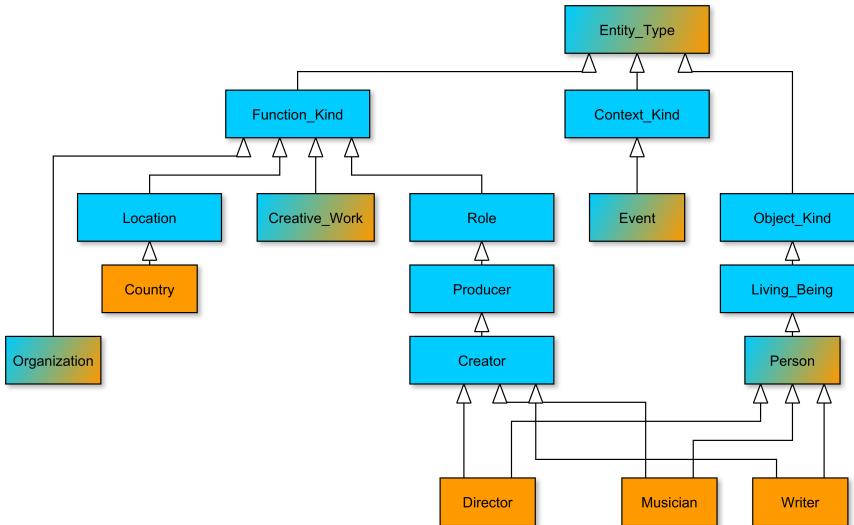


Figure 5: This is how we extended CSK with our own entities. The blue nodes are from CSK, the orange nodes are from our ontology, the half-blue-half-orange nodes are the nodes which are equivalent between CSK and our ontology's top level (as shown in Figure 3). Note that in this figure we are only showing entities which make direct use of the CSK entities (as sub-classes), so this is not a complete representation of our ontology.

---

When mapping the CSK to our ontology, we found a bunch of relations that we did not consider before but that are actually interesting. To include them in our model, we used the entities and the properties directly provided by CSK. For instance, we widely used *CSK:memberOf* and *CSK:hasMember* properties. These is the way we modeled the membership to Organization and to MusicGroup. We also leveraged the inheritance of properties in CSK: for instance in the case of our Person entity (and sub-classes) we rely on the CSK to provide the basic functional information (ie, has a name).

#### 4.1 Encountered issues

Even though our ontology and CSK were very compatible we encountered a number of problems that, in some cases, prevented us from using some of the classes provided by CSK.

One of the first problems we encountered was how to model collections of entities. We needed to model MusicalAlbum (which is a collection of MusicalWork items), MusicGroup (a collection of Musicians), and TVShow (a collection of TVShowEpisode). By default there is no *collection* in CSK, so we ended up extending CSK with our own "Collection" entity, which is on the same hierarchical level as the *Entity* entity.

We also noted an inconsistency in CSK (which might be on purpose of not), and we solved it by extending it. The inconsistency was on the *Event* entity. According to CSK, an Event is "*an occurrence that has both place and time*", however, as far as we could see the Event entity of CSK, it only contained *time* information. Therefore, we extended our own Event entity with a new property *hasLocation*, whose domain is the Event entity and range is the *Location* entity provided by CSK.

The following is more likely a difficulty rather than an issue, but nonetheless it was something that needed quite a bit of thinking before we could decide how to solve it. Initially, we intended the *partOf* and *hasPart* properties to be transitive. For example, if there is a LiteraryWork element *Lyrics* which is part of a MusicalWork item (a song), and if this song then becomes part of a CinematographicWork Movie, then we want to know that the Lyrics are part of the Movie. This *transitive* parthood property is incompatible with the one provided by CSK, which is not transitive. We ignored the CSK encoding and instead used our own version. But as we went on in the refinement of the ontology, we noticed that actually it was better for the *partOf* properties to be intransitive, since we wanted (referring to the example above) the *MusicLyrics* to be part of a *MusicalWork*, and the latter to be part of the *Movie*. So we ended up marking our *isPartOf/hasPart* as equivalent with the ones provided in CSK.

## 5 Model visualization

In this section we will present a visualization of the final ontology model. We will continue using the notation we have been using in previous visualizations (blue nodes are from CSK, orange nodes are from our ontology, and half-blue-half-orange nodes are the nodes which are equivalent between CSK and our ontology's top level). We will show two main visualizations: one including only the hierarchy of entities and another one including both the entities and the relations between them (Object Properties). This choice has the objective of making both visualizations more understandable and less cluttered. Note that in both visualizations we will include only the part of CSK which is relevant to our ontology (so we will not be visualizing the complete CSK).

The following image shows our ontology's structure.

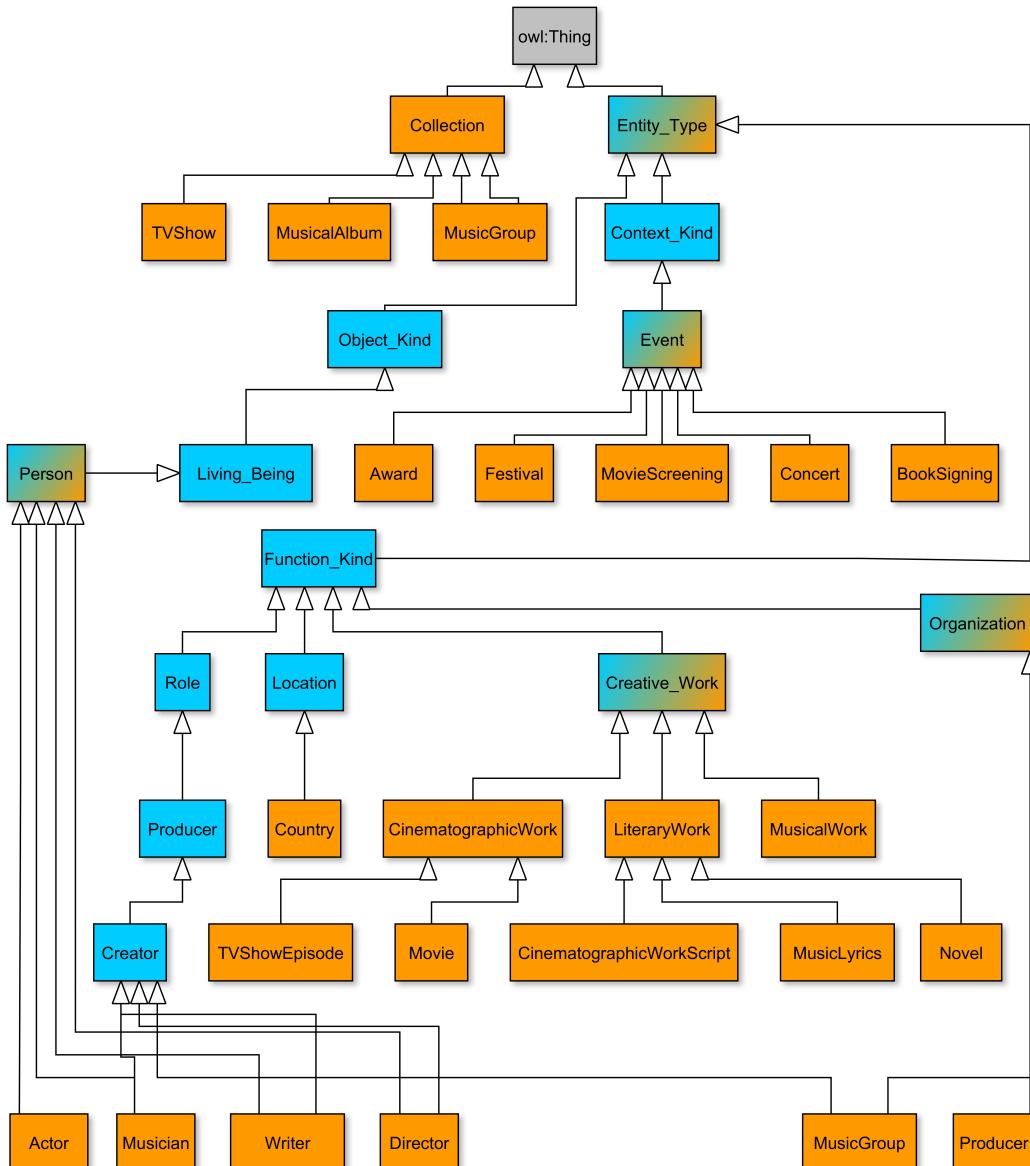


Figure 6: The structure of our ontology. It includes all the entities we defined and the entities of CSK that we used. The grey node (*owl:Thing*) is just the root of all entities.

This is another visualization of the model, which includes all the relations we used in our ontology.

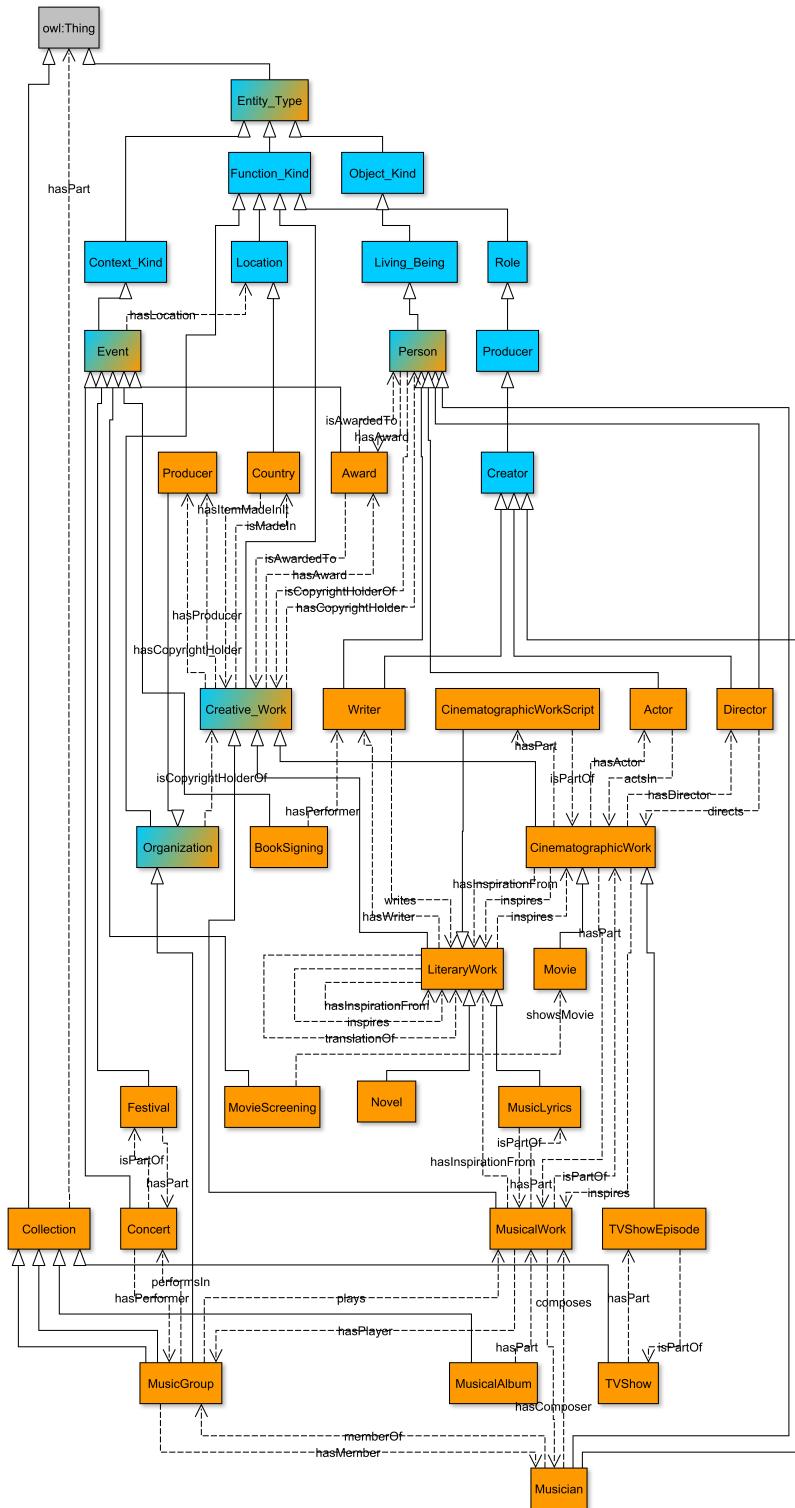


Figure 7: Visualization of the ontology, including all the relations. Note that basic relations, specifically those used to identify entities (*name*, *eidrID*, *isbnID*, and *isniID*), have not been included to simplify the diagram.

## 6 Generalized queries

In this section we'll see all the queries that our system will support, as well as some example queries.

### 6.1 All supported queries

Persona	Generalized Query	Expected Answer
Media consumer	Show me items of a specific media type released during a specific date range.	Page with items released during the specified date interval.
Media consumer	Show me items of a specific media type with a name similar to the one I specify.	Page with items that have a name similar to the one specified by the user.
Media consumer	Show me items of a specific media type which were produced in a specific country.	Page with items that were produced in the country specified by the user.
Media consumer	Show me the country in which a specific item of a specific media type was produced.	Page with the country in which the item was produced.
Media consumer	Show me CinematographicWork items in which a specific actor appears.	Page with CinematographicWork items that star the actor specified by the user.
Media consumer	Show me CinematographicWork items that were directed by a specific director.	Page with CinematographicWork items directed by the director specified by the user.
Media consumer	Show me the plot related with a specific CinematographicWork/LiteraryWork item.	Page with the plot of said item.
Media consumer	Show me MusicalWork items in which a specific musician performs.	Page with MusicalWork items in which the musician specified by the user performs.
Media consumer	Show me LiteraryWork items that have been written by a specific writer.	Page with LiteraryWork items written/composed by the writer specified by the user.
Media consumer	Show me items of a specific media type that correspond to a specific genre.	Page with items of the specified media type corresponding the genre specified by the user.
Media consumer	Show me the events related to a specific item of a specific media type.	A page of events related to the item, sorted by date.
Media consumer	Show me the events related to a specific writer.	A page of events related to the writer, sorted by date.
Media consumer	Show me the events related to a specific musician.	A page of events related to the musician, sorted by date.
Media consumer	Show me the events related to a specific actor.	A page of events related to the actor, by date.
Media consumer	Show me the events related to a specific director.	A page of events related to the director, sorted by date.
Media consumer	Show me events in a location, in a date range.	A page with events occurring at the specified location on the date range, sorted by date.
Media consumer	Show me the awards that have been given to a specific item of a specific media type.	A page of awards given to the item, sorted by date.
Media consumer	Show me the awards that have been given to a person.	A page of awards given to the person, sorted by date.

<b>Persona</b>	<b>Generalized Query</b>	<b>Expected Answer</b>
Relation explorer	Show me the actors associated with a specific CinematographicWork item.	Page with the actors in said CinematographicWork item.
Relation explorer	Show me the directors associated with a specific CinematographicWork item.	Page with the directors of said CinematographicWork item.
Relation explorer	Show me the soundtrack associated with a specific CinematographicWork item.	Page with the soundtrack in said CinematographicWork item.
Relation explorer	Show me the plot writer associated with a specific CinematographicWork item.	Page with the plot writer of said CinematographicWork item.
Relation explorer	Show me the producer associated with a specific CinematographicWork item.	Page with the producer of said CinematographicWork item.
Relation explorer	Show me the lyrics associated with a specific MusicalWork item.	Page with the lyrics in said MusicalWork item.
Relation explorer	Show me the musical group associated with a specific MusicalWork item.	Page with the musical band of said MusicalWork item.
Relation explorer	Show me the composer associated with a specific MusicalWork item.	Page with the composer of said MusicalWork item.
Relation explorer	Show me the writer associated with a specific MusicalWork item.	Page with the writer of said MusicalWork item.
Relation explorer	Show me the production company associated with a specific MusicalWork item.	Page with the production company of said MusicalWork item.
Relation explorer	Show me the writer associated with a specific LiteraryWork item.	Page with the writer of said LiteraryWork item.
Relation explorer	Show me the production company associated with a specific LiteraryWork item.	Page with the production company of said LiteraryWork item.
Relation explorer	Show me the LiteraryWork items that have inspired a specific CinematographicWork item.	Page with said LiteraryWork items.
Relation explorer	Show me the CinematographicWork that have inspired a specific LiteraryWork item.	Page with said CinematographicWork items.
Market analyst	Show me the revenue made by a specific CinematographicWork item.	Page containing information on the revenue made by the specified item.
Market analyst	Show me the amount of copies sold of a specific MusicalWork items.	Page containing information on the number of copies sold of the specified item.
Market analyst	Show me the amount of copies sold of a specific LiteraryWork items.	Page containing information on the number of copies sold of the specified item.
Market analyst	Show me the revenue generated in a specific country, by CinematographicWork items released during a certain period of time.	Page containing information of the total revenue made in a country, by CinematographicWork items released during the specified period of time.

Persona	Generalized Query	Expected Answer
Market analyst	Show me the total amount of sold copies of LiteraryWork items, in a specific country, released during a certain period of time.	Page containing information of the total number of copies sold in a country of LiteraryWork items that were released during a specific period of time.
Market analyst	Show me the total amount of sold copies of MusicalWork items, in a specific country, released during a certain period of time.	Page containing information on the number of copies sold in a country of MusicalWork items that were released during a specific period.
Market analyst	Show me the revenue generated by CinematographicWork items of a specific genre released during a certain period of time.	Page containing information of the total revenue made by CinematographicWork items of the specified genre released during the specified period of time.
Market analyst	Show me the total amount of sold copies of LiteraryWork items, of a specific genre, released during a certain period of time.	Page containing information of the total number of copies sold of LiteraryWork items of a specific genre that were released during a specific period of time.
Market analyst	Show me the total amount of sold copies of MusicalWork items, of a specific genre, released during a certain period of time.	Page containing information of the total number of copies sold of MusicalWork items of a specific genre that were released during a specific period of time.
Market analyst	Show me the rating of a specific CinematographicWork item.	Page containing information on the rating of the specified item.
Market analyst	Show me the rating of a specific LiteraryWork item.	Page containing information on the rating of the specified item.
Rights manager	Show copyright holder information for a specific media item.	Page with the copyright information for specified media item.
Rights manager	Show media items whose copyright is held by a specific copyright holder.	Page with list of media items.

Table 8: Generalized Queries

## 6.2 Example Queries

In this section we show 2 example requests for each of the types of Persona we expect for our application. All queries are done in SPARQL, and for all we define the following prefixes:

- rdf = <http://www.w3.org/1999/02/22-rdf-syntax-ns>
- owl = <http://www.w3.org/2002/07/owl>
- rdfs = <http://www.w3.org/2000/01/rdf-schema>
- xsd = <http://www.w3.org/2001/XMLSchema>
- mlo = <http://www.semanticweb.org/tupin/ontologies/2018/10/MLO>

Note: `mlo` is the prefix used for our ontology.

### 6.2.1 Example queries for: Media consumer

Show me LiteraryWork items that have been written by a specific writer

```
SELECT ?novel
WHERE {
?novel mlo:hasWriter mlo:Tolkien .
?novel rdf:type mlo:Novel
}
```

This query will return MLO:The\_Lord\_of\_the\_Rings. This query makes use of the following entities and properties:

- Entities: *Writer, Novel*
- Properties: *hasWriter*

Show me the events related to a specific director

```
SELECT ?movie ?movieScreening
WHERE {
?movie mlo:hasDirector mlo:Peter_Jackson .
?movieScreening mlo:showsMovie ?movie
}
```

This query will return the following:

movie	movieScreening
MLO:The_Fellowship_of_the_Ring	MLO:London_LOTR_Screening

- Entities: *Movie, Director, MovieScreening*
- Properties: *hasDirector, showsMovie*

### 6.2.2 Example queries for: Relation explorer

Show me the LiteraryWork items that have inspired a specific CinematographicWork item

```
SELECT ?novel
WHERE {
?novel mlo:inspires mlo:The_Fellowship_of_the_Ring .
```

This query will return MLO:The\_Lord\_of\_the\_Rings.

- Entities: *Movie, Novel*
- Properties: *inspires*

Show me the production company associated with a specific LiteraryWork item

```
SELECT ?producer
WHERE {
?producer mlo:produces mlo:The_Lord_of_the_Rings .
```

---

This query will return MLO:Allen\_&\_Unwin.

- Entities: *Producer, Novel*
- Properties: *produces*

#### 6.2.3 Example queries for: Market Analyst

Show me the amount of copies sold of a specific LiteraryWork items

```
SELECT ?novel ?copiesSold
WHERE {
VALUES ?novel { mlo:The_Lord_of_the_Rings }
?novel mlo:numberOfSoldCopies ?copiesSold
}
```

novel	copiesSold
MLO:The_Lord_of_the_Rings	150000000

- Entities: *Novel*
- Data Properties: *numberOfSoldCopies*

Show me the rating of a specific CinematographicWork item

```
SELECT ?movie ?rating
WHERE {
VALUES ?movie { mlo:The_Fellowship_of_the_Ring }
?movie mlo:rating ?rating
}
```

movie	rating
MLO:The_Fellowship_of_the_Ring	9.5

- Entities: *Movie*
- Data Properties: *rating*

#### 6.2.4 Example queries for: Rights manager

Show copyright holder information for a specific media item

```
SELECT ?movie ?copyrightHolder
WHERE {
VALUES ?movie { mlo:The_Fellowship_of_the_Ring }
?movie mlo:hasCopyrightHolder ?copyrightHolder
}
```

movie	copyrightHolder
MLO:The_Fellowship_of.the.Ring	MLO:New_Line_Cinema
MLO:The_Fellowship_of.the.Ring	MLO:Tolkien

- Entities: *Movie, Writer, Producer*
- Properties: *hasCopyrightHolder*

Show media items whose copyright is held by a specific copyright holder

```
SELECT ?copyrightHolder ?item
WHERE {
VALUES ?copyrightHolder { mlo:Tolkien }
?copyrightHolder mlo:isCopyrightHolderOf ?item
}
```

copyrightHolder	item
MLO:Tolkien	MLO:The_Lord_of_the_Rings
MLO:Tolkien	MLO:The_Fellowship_of.the_Ring

- Entities: *Writer, Movie, Novel*
- Properties: *isCopyrightHolderOf*

## 7 Final considerations and open issues

The process of building a formal model, and more in general performing data integration, is not an easy task. Thanks to the procedure we learned at class we were able to build our formal model step by step, sure that the processes we followed would have lead to a good result.

There is still some work which can be done in the ontology. For instance, the different MindProducts can be specified more in depth. The current separation works well, but it can be refined a bit more so that the ontology has more representation power. For example, we could have more separation in LiteraryWork, that at the moment just represents a Novel as a *book*. It is enough for the purposes of our project, but more types could be added to represent other types of LiteraryWork. Another possible extension is to the MusicalWork entity, which currently does not have any sub-entities, but a specification can be made to represent different types of MusicalWork.

One other possible extension would be to add more *data* to the entities, since currently we're only supporting the data needed for our project. However, there is more room for improvement here. For example, we could represent more personal data about each person. Currently we only capture the name and the role.

## 8 Evaluation of the model

	Class Coverage	Class flexibility	Attribute Coverage	Attribute Flexibility
CQ-Model	$\simeq 0.8$ (Ideal) 0.53	$\simeq 0.2$ (Ideal) 0.32	$\simeq 0.8$ (Ideal) 0.77	$\simeq 0.5$ (Ideal) 0.19
DS-Model	$\simeq 0.8$ (Ideal) -	$\simeq 0.2$ (Ideal) -	$\simeq 0.8$ (Ideal) -	$\simeq 0.5$ (Ideal) -
CQ-DE	$\simeq 0.8$ (Ideal) -	$\simeq 0.2$ (Ideal) -	$\simeq 0.8$ (Ideal) -	$\simeq 0.5$ (Ideal) -

### Schema Level

- Does the model including cycles in the class hierarchy? **No**
- Does the Model uses any polysemous terms for its class or property name? **No**
- Is Multiple Domain/Range defined for any property? **Yes**
- Does any class have more than one direct parent class? **No**
- Does the Model include multiple classes which have same meaning? **No**
- Is the class Hierarchy over specified? **No**
- Does the model use *isA* as a object Property or relation? **No**
- Does the model have any leaf class for which there is no relation with the rest of the model? **No**
- Did you use miscellaneous or others as one of the class name? **No**
- Does the model have any chain of Inheritance in class hierarchy? **No**
- Do all properties have explicit domain and range declarations? **Yes**
- Does the model have any classes or properties which are not used? **No**
- Are a collection of elements included as a group in a number of class/attribute? **No**

### Linguistic Level

- Do all elements of the model (i.e. class and property) have human readable annotations? **Yes**
- Do all elements of the model follow the same naming convention? **Yes**

### Metadata Level

- Is provenance information (Creator, Version, Date) available for the final protege model? **Yes**
- Is provenance information available for any property or class which is taken from some reference standard or ontology? **Yes**

---

## Bibliography

### References

Goodreads. <https://www.goodreads.com/>.

Imdb. <https://www.imdb.com/>.

Spotify. <https://www.spotify.com/>.

Tidal. <https://www.tidal.com/>.

Frdrick Giasson Bruce D'Arcus. Bibliographic ontology. <http://bibliontology.com>, 2013.

Vincenzo Maltese Enrico Bignotti, Amarsanaa Ganbold. Mind product etype graph description, 2013.

Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

motools. musicontology. <https://github.com/motoools/musicontology>, 2013.

schemaorg. schemaorg. <https://github.com/schemaorg/schemaorg>, 2018.