ISIT315

Assignment 4 Report Anime & Manga Ontology

Group members

Daniel Barnes	5053511
Hoang Duong Nguyen	5177832
Jingwang Teh	5238699
Minh Tan Le	5142179

Methodological Approach on Developing Ontology

The domain that we chose to develop the ontology on is the Anime & Manga Industry. The ontology was based on data from myanimelist.net, the world's largest database and social community for Anime and Manga.

Planning

Purpose

The purpose of building our Media Ontology is to provide a knowledge model of Anime & Manga series' which can be utilised by users for searching relevant media, and which can be expanded easily to include data from other mediums (e.g. Visual Novels, Games, Blu-ray Drama, and etc.) in order to produce a more comprehensive ontology with a larger scope. The purpose of developing the ontology is also to allow the querying of data and to support the automated inferencing of relations between entities/individuals within the ontology.

Scope

The ontology will only focus on the anime & manga domain. The level of detail will be based on the entries described in myanimelist.net.

Implementation Language

The ontology is to be implemented using protégé in OWL language, Manchester syntax.

Intended Users

- This ontology is mainly intended for people who are interested in the Anime & Manga industry.
    - This could include both professionals and fans in the anime/manga industry
- It is also for people who wish to find out information about a particular series.
- The ontology is also useful for users who want to find the source material of a certain media adaptation in order to support the original author.

Intended Uses

- A user can find out information about the studio, staff, characters, age rating, genre and current status of a particular series.
- This ontology should allow users to look for all series belonging to a particular season (e.g. Fall 2017, Winter 2017, etc.)
- The user should be able to find out the score rating of a particular series.
- Users can also find the source material that the series adapted from.
- Users can find the prequel and sequels of the series.
- Users can view relations between entities/individuals within the ontology. For example, a user can query all the anime's a certain voice actor plays in or can query all the anime's developed by a certain studio.

Requirement Specification

Ontology Requirements

A. Functional Requirements

The ontology includes the following entities:

- MediaType
    - Anime
        - Movie
        - OVA
        - Special
        - TV
    - Comic
        - Doujinshi
        - Manga
        - Manhua
        - Manhwa
        - Novel
        - One-shot
- Character
    - MainCharacter
    - SupportingCharacter
- Staff
    - AnimationDirector
    - Director
    - EpisodeDirector
    - Producer
    - SoundDirector
    - VoiceActor
- Studio
- Magazine
- Author
- Genre
- Status
    - AnimeStatus
    - ComicStatus
- AgeRating
- Season (from Winter2010 to Fall2018)
- Series
- SeriesScore
    - Appalling

  - Average
  - Bad
  - Fine
  - Good
  - Great
  - Horrible
  - Masterpiece
  - VeryBad
  - VeryGood
- SourceType

There are also object properties, such as:

- characterIn
  - mainCharacterIn
  - supportingCharacterIn
- hasCharacter
  - hasMainCharacter
  - hasSupportingCharacter
- isVoicedBy
- voicesCharacter
  - voicesMainCharacter
  - voicesSupportingCharacter
- hasStaff
- isStaffOf
- hasPublisher
  - hasSerializaiton
  - hasStudio
- publisherOf
  - serializationOf
  - studioOf
- hasAuthor
- isAuthorOf
- hasGenre
- isGenreOf
- hasStatus
- isStatusOf
- hasAgeRating
- isAgeRatingOf
- hasPrequel
- hasSequel
- hasSourceType
- isSourceTypeOf

- isAdaptionOf
- isPartOfSeries

Data properties include:

- hasSynonymTitle
- hasSynopsis
- hasStartDate
- hasEndDate
- hasScore
- hasTotalEpisode
- hasDuration
- hasBirthday
- hasDescription

B. Non-Functional Requirements

1. The ontology should use real data.
2. The synonym title data property should be in romanized format, and not in katakana, hiragana, or kanji format.
3. The seasons should follow Japanese seasons.
4. Class, instance and property titles should be easy and straightforward to understand.
5. Individual/Instance names should be prefixed with the class they belong to.
6. Each anime and comic must have its start-airing date, however the finish-airing date is optional.
7. Classes about media should be populated with at least one instance.
8. There should be a sufficient (50-100) number of classes and properties.
9. Classes and subclasses should be organised rationally.
10. The ontology should be constructed logically to support easy maintenance at a later date
11. The structure of the ontology should support expansion at a later date.

Conceptualization

1. Determine scope

We decided that our ontology will only focus on the anime & manga domain. The ontology will be used to store information and relations of anime and manga series, which is a large amount of information. The ontology will provide information relating to these media types ranging from the name of series to characters and staff involved in the series. The main users of the ontology will include fans and professionals in the Anime & Manga industry.

2. Consider reuse

Since the assignment requires us to develop a new ontology, therefore we built our ontology from scratch and no ontology was reused.

3. Enumerate terms

As our ontology covers the domain of anime and manga, relevant terms that can be expected to appear in our ontology will be all the characteristics associated with an anime/manga series. These terms were used to define the classes and properties in our ontology.

Classes:

- MediaType
- Anime
- Movie
- OVA
- Special
- TV
- Comic
- Doujinshi
- Manga
- Manhua
- Manhwa

- Novel
- One-shot
- Magazine
- Character
- MainCharacter
- SupportingCharacter
- Staff
- AnimationDirector
- Director
- EpisodeDirector
- Producer
- SoundDirector
- VoiceActor
- Studio
- Author
- Genre
- Status
- AnimeStatus
- ComicStatus
- AgeRating
- Season
- Series
- SeriesScore

Properties:

- characterIn
- mainCharacterIn
- supportingCharacterIn
- hasCharacter
- hasMainCharacter
- hasSupportingCharacter
- isVoicedBy
- voicesCharacter
- voicesMainCharacter
- voicesSupportingCharacter
- hasStaff
- isStaffOf
- hasPublisher
- hasSerializaiton
- hasStudio
- publisherOf

- serializationOf
- studioOf
- hasAuthor
- isAuthorOf
- hasGenre
- isGenreOf
- hasStatus
- isStatusOf
- hasAgeRating
- isAgeRatingOf
- hasPrequel
- hasSequel
- hasSourceType
- isSourceTypeOf
- isAdaptionOf
- isPartOfSeries
- hasSynonymTitle
- hasSynopsis
- hasStartDate
- hasEndDate
- hasScore
- hasTotalEpisode
- hasDuration
- hasBirthday
- hasDescription

4. Define taxonomy

We used the top-down method of defining taxonomy in order to populate our ontology with classes and subclasses. We started with general concepts and then gradually expanded them. For example, we first defined a series score class, and then later expanded this class to include more detailed subclasses of scores ranging from Horrible to Masterpiece.

5. Define properties

Properties were also defined during the insertion of new classes to make sure all the properties of each class was covered.

Object properties:

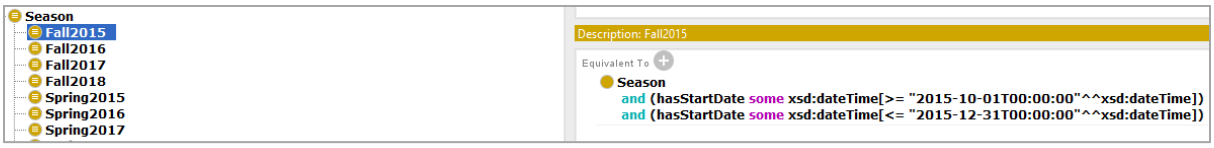
- characterIn → (domain: Character range: MediaType)
  - mainCharacterIn
  - supportingCharacterIn
- hasCharacter → (domain: MediaType range: Character)
  - hasMainCharacter
  - hasSupportingCharacter
- isVoicedBy → (domain: Character range: VoiceActor)
- voicesCharacter → (domain: VoiceActor range: Character)
  - voicesMainCharacter
  - voicesSupportingCharacter
- hasStaff → (domain: Anime range: Staff)
- isStaffOf → (domain: Staff range: Anime)
- hasPublisher
  - hasSerializaiton → (domain: Comic range: Magazine)
  - hasStudio → (domain: Anime range: Studio)
- publisherOf
  - serializationOf → (domain: Magazine range: Comic)
  - studioOf → (domain: Studio range: Anime)
- has Author → (domain: Comic range: Author)
- isAuthorOf → (domain: Author range: Comic)
- hasGenre → (domain: MediaType range: Genre)
- isGenreOf → (domain: Genre range: MediaType)
- hasStatus → (domain: MediaType range: Status)
- isStatusOf → (domain: Status range: MediaType)
- hasAgeRating → (domain: Anime range: AgeRating)
- isAgeRatingOf → (domain: AgeRating range: Anime)
- hasPrequel
- hasSequel
- hasSourceType → (domain: Anime range: SourceType)
- isSourceTypeOf → (domain: SourceType range: Anime)
- isAdaptionOf → (domain: Anime range: Comic)
- isPartOfSeries → (domain: MediaType range: Series)

Data properties include:

- hasSynonymTitle → (domain: MediaType range: xsd:string)
- hasSynopsis → (domain: MediaType or Series range: xsd:string)
- hasStartDate → (domain: MediaType range: xsd:dateTime)
- hasEndDate → (domain: MediaType range: xsd:dateTime)
- hasScore → (domain: MediaType range: xsd:double)
- hasTotalEpisode → (domain: Anime range: xsd:integer)
- hasDuration → (domain: Anime range: xsd:integer)
- hasBirthday → (domain: Staff range: xsd:dateTime)
- hasDescription → (domain: Staff range: xsd:string)

6. Define facets

Based on the properties of the class, we have also identified proper facets for classes. For example, each subclass of class Season has certain minimum and maximum cardinality (based on the start date value). This can be used later by the inference to infer which season a certain anime belongs to, depending on the anime's start date.



```
Season
 and (hasStartDate some xsd:dateTime[>= "2015-10-01T00:00:00"^^xsd:dateTime])
 and (hasStartDate some xsd:dateTime[<= "2015-12-31T00:00:00"^^xsd:dateTime])
```

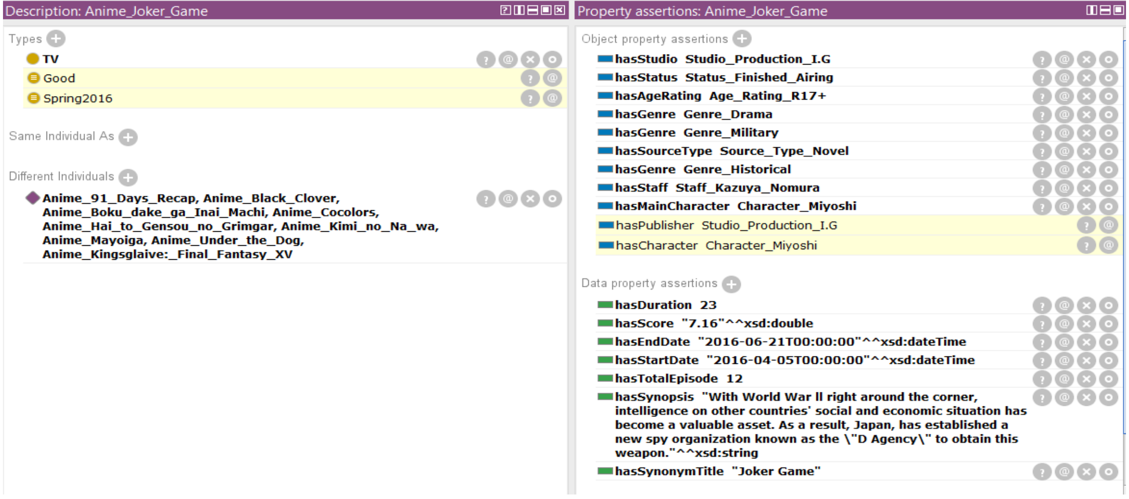
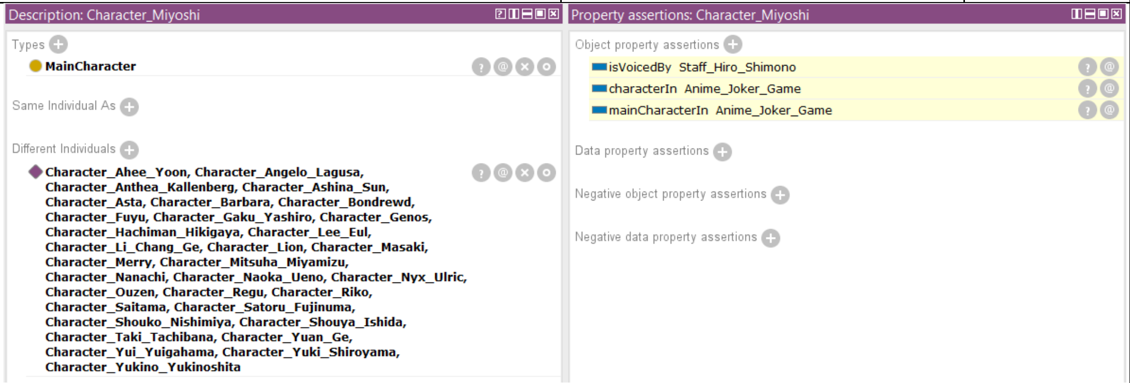
7. Define instances

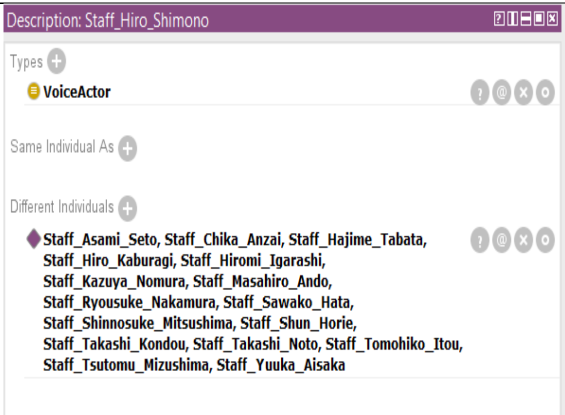
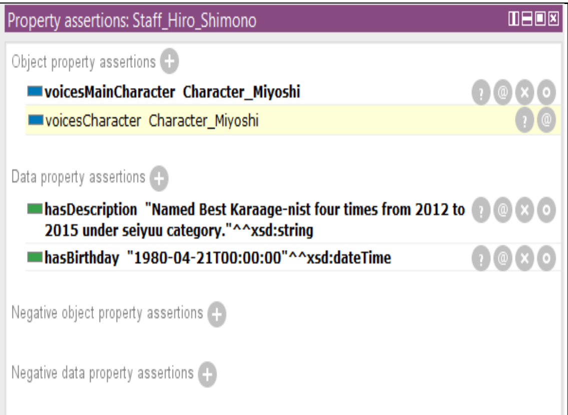
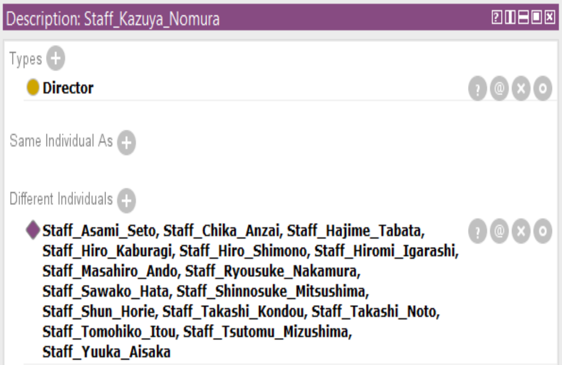
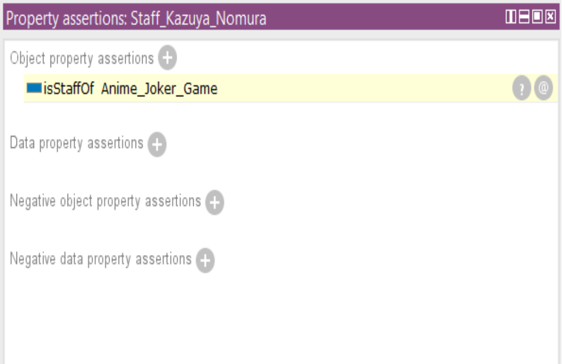
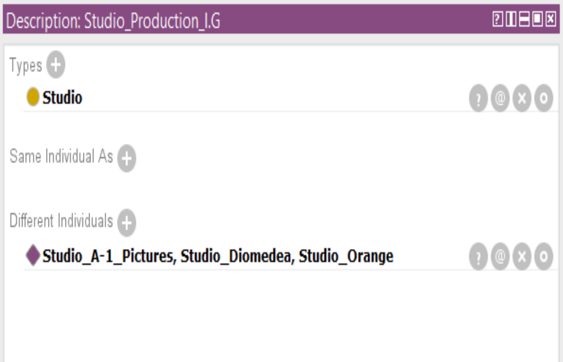
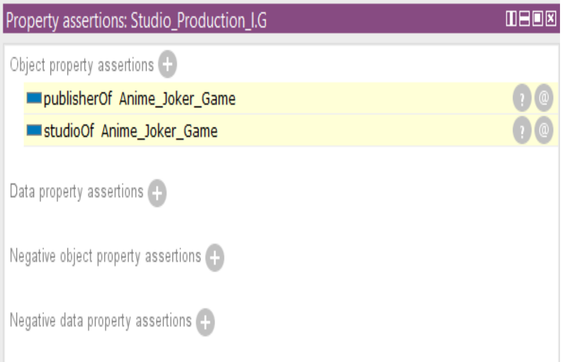
Instances were defined based on real data collected from myanimelist.com. This was done by choosing a class, creating an individual instance of the class and then filling in its values with the data found on myanimelist.com.

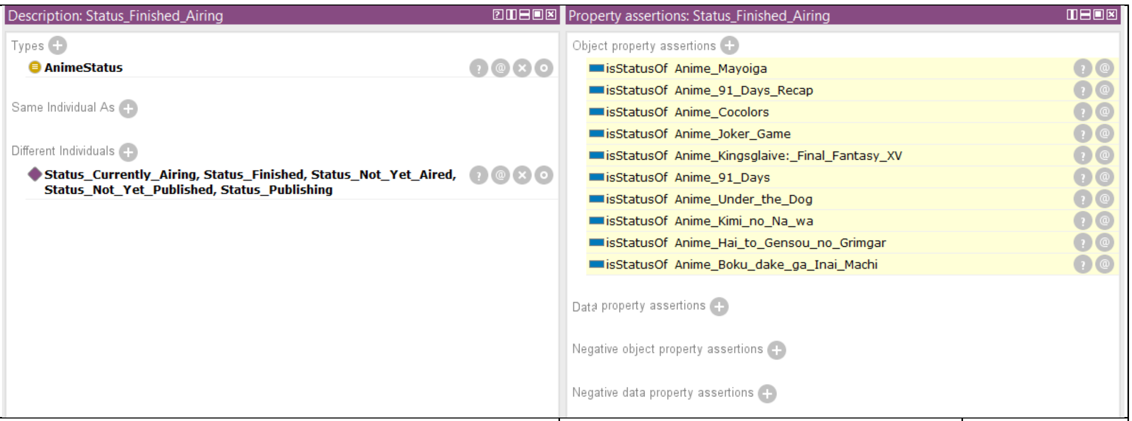
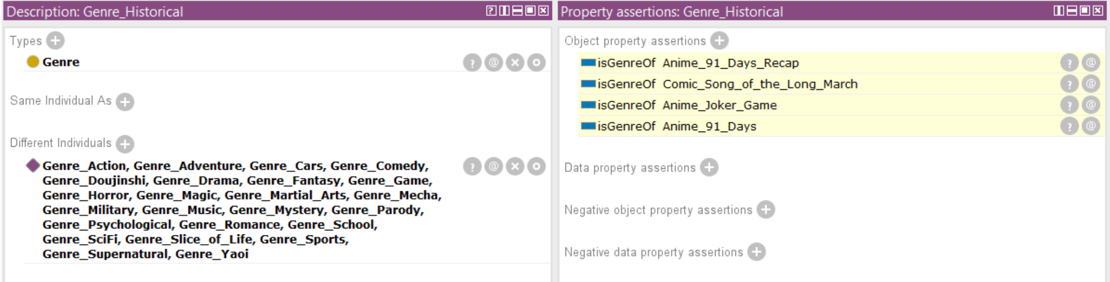
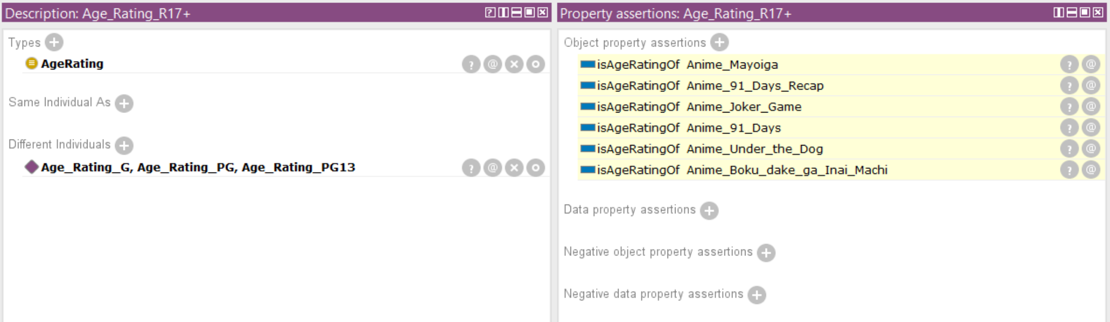
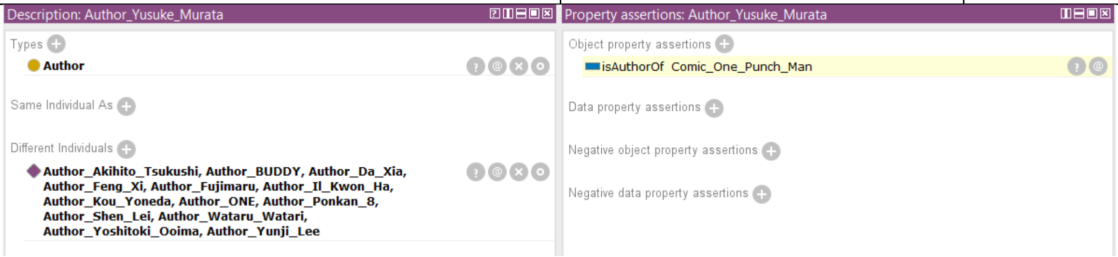
8. Check for anomalies

Domain/range incompatibility, cardinality inconsistency and property values were all checked for any anomalies. Overall, the ontology is consistent with the initial ideas about the anime/manga domain.

Evaluation

No.	Question	Answer	Correct?
1.	Anime_Joker_Game has a score of 7.16, does it have a good (score of 7 inclusive to 8 exclusive) score?	“Good” Score	Yes
2.	Anime_Joker_Game has a start date of 2016 April 05, which is Spring of 2016 (according to Japan’s season)	“Spring 2016” Season	Yes
	 Description: Anime_Joker_Game Types TV Good Spring2016 Same Individual As Different Individuals Anime_91_Days_Recap, Anime_Black_Clover, Anime_Boku_dake_ga_Inai_Machi, Anime_Cocolors, Anime_Hai_to_Gensou_no_Grimgar, Anime_Kimi_no_Na_wa, Anime_Mayoiga, Anime_Under_the_Dog, Anime_Kingsglaive:_Final_Fantasy_XV Property assertions: Anime_Joker_Game Object property assertions hasStudio Studio_Production_I.G hasStatus Status_Finished_Airing hasAgeRating Age_Rating_R17+ hasGenre Genre_Drama hasGenre Genre_Military hasSourceType Source_Type_Novel hasGenre Genre_Historical hasStaff Staff_Kazuya_Nomura hasMainCharacter Character_Miyoshi hasPublisher Studio_Production_I.G hasCharacter Character_Miyoshi Data property assertions hasDuration 23 hasScore "7.16"^^xsd:double hasEndDate "2016-06-21T00:00:00"^^xsd:dateTime hasStartDate "2016-04-05T00:00:00"^^xsd:dateTime hasTotalEpisode 12 hasSynopsis "With World War II right around the corner, intelligence on other countries' social and economic situation has become a valuable asset. As a result, Japan, has established a new spy organization known as the \"D Agency\" to obtain this weapon."^^xsd:string hasSynonymTitle "Joker Game"		
3.	Is Character_Miyoshi a character in Anime_Joker_Game?	Character_Miyoshi is characterIn Anime_Joker_Game	Yes
4.	Does Staff_Hiro_Shimono voice this character?	Character_Miyoshi is voicedBy Staff_Hiro_Shimono	Yes
	 Description: Character_Miyoshi Types MainCharacter Same Individual As Different Individuals Character_Ahee_Yoon, Character_Angelo_Lagusa, Character_Anthea_Kallenberg, Character_Ashina_Sun, Character_Asta, Character_Barbara, Character_Bondrewd, Character_Fuyu, Character_Gaku_Yashiro, Character_Genos, Character_Hachiman_Hikigaya, Character_Lee_Eul, Character_Li_Chang_Ge, Character_Lion, Character_Masaki, Character_Merry, Character_Mitsuha_Miyamizu, Character_Nanachi, Character_Naoka_Ueno, Character_Nyx_Ulric, Character_Ouzen, Character_Regu, Character_Riko, Character_Saitama, Character_Satoru_Fujinuma, Character_Shouko_Nishimiya, Character_Shouya_Ishida, Character_Taki_Tachibana, Character_Yuan_Ge, Character_Yui_Yuigahama, Character_Yuki_Shiroyama, Character_Yukino_Yukinoshita Property assertions: Character_Miyoshi Object property assertions isVoicedBy Staff_Hiro_Shimono characterIn Anime_Joker_Game mainCharacterIn Anime_Joker_Game Data property assertions Negative object property assertions Negative data property assertions		

	 Description: Staff_Hiro_Shimono Types VoiceActor Same Individual As Different Individuals Staff_Asami_Seto, Staff_Chika_Anzai, Staff_Hajime_Tabata, Staff_Hiro_Kaburagi, Staff_Hiromi_Igarashi, Staff_Kazuya_Nomura, Staff_Masahiro_Ando, Staff_Ryousuke_Nakamura, Staff_Sawako_Hata, Staff_Shinnosuke_Mitsushima, Staff_Shun_Horie, Staff_Takashi_Kondou, Staff_Takashi_Noto, Staff_Tomohiko_Itou, Staff_Tsutomu_Mizushima, Staff_Yuuka_Aisaka	 Property assertions: Staff_Hiro_Shimono Object property assertions voicesMainCharacter Character_Miyoshi voicesCharacter Character_Miyoshi Data property assertions hasDescription "Named Best Karaage-nist four times from 2012 to 2015 under seiyuu category."^^xsd:string hasBirthday "1980-04-21T00:00:00"^^xsd:dateTime Negative object property assertions Negative data property assertions	
5.	is Staff_Kazuya_Nomura a staff for the Anime_Joker_Game?	Staff_Kazuya_Nomura is a staff of Anime_Joker_Game	Yes
	 Description: Staff_Kazuya_Nomura Types Director Same Individual As Different Individuals Staff_Asami_Seto, Staff_Chika_Anzai, Staff_Hajime_Tabata, Staff_Hiro_Kaburagi, Staff_Hiro_Shimono, Staff_Hiromi_Igarashi, Staff_Masahiro_Ando, Staff_Ryousuke_Nakamura, Staff_Sawako_Hata, Staff_Shinnosuke_Mitsushima, Staff_Shun_Horie, Staff_Takashi_Kondou, Staff_Takashi_Noto, Staff_Tomohiko_Itou, Staff_Tsutomu_Mizushima, Staff_Yuuka_Aisaka	 Property assertions: Staff_Kazuya_Nomura Object property assertions isStaffOf Anime_Joker_Game Data property assertions Negative object property assertions Negative data property assertions	
6.	is Studio_Production_I.G the studio that published Anime_Joker_Game?	Studio_Production_I is the studio of and publisher of Anime_Joker_Game	Yes
	 Description: Studio_Production_I.G Types Studio Same Individual As Different Individuals Studio_A-1_Pictures, Studio_Diomedea, Studio_Orange	 Property assertions: Studio_Production_I.G Object property assertions publisherOf Anime_Joker_Game studioOf Anime_Joker_Game Data property assertions Negative object property assertions Negative data property assertions	
7.	has Anime_Joker_Game finished Airing?	Yes, Anime_Joker_Game has a status of Status_Finished_Airing.	Yes

	 Description: Status_Finished_Airing Types AnimeStatus Same Individual As Different Individuals Status_Currently_Airing, Status_Finished, Status_Not_Yet_Aired, Status_Not_Yet_Published, Status_Publishing Property assertions: Status_Finished_Airing Object property assertions isStatusOf Anime_Mayoiga isStatusOf Anime_91_Days_Recap isStatusOf Anime_Cocolors isStatusOf Anime_Joker_Game isStatusOf Anime_Kingsglaive:_Final_Fantasy_XV isStatusOf Anime_91_Days isStatusOf Anime_Under_the_Dog isStatusOf Anime_Kimi_no_Na_wa isStatusOf Anime_Hai_to_Gensou_no_Grimgar isStatusOf Anime_Boku_dake_ga_Inai_Machi Data property assertions Negative object property assertions Negative data property assertions		
8.	Is Anime_Joker_Game part of the Genre_Historical?	Genre_Historical is the genre of Anime_Joker_Game.	Yes
	 Description: Genre_Historical Types Genre Same Individual As Different Individuals Genre_Action, Genre_Adventure, Genre_Cars, Genre_Comedy, Genre_Doujinshi, Genre_Drama, Genre_Fantasy, Genre_Game, Genre_Horror, Genre_Magic, Genre_Martial_Arts, Genre_Mecha, Genre_Military, Genre_Music, Genre_Mystery, Genre_Parody, Genre_Psychological, Genre_Romance, Genre_School, Genre_SciFi, Genre_Slice_of_Life, Genre_Sports, Genre_Supernatural, Genre_Yaoi Property assertions: Genre_Historical Object property assertions isGenreOf Anime_91_Days_Recap isGenreOf Comic_Song_of_the_Long_March isGenreOf Anime_Joker_Game isGenreOf Anime_91_Days Data property assertions Negative object property assertions Negative data property assertions		
9.	Is Anime_Joker_Game of the age rating of R17+?	Age_Rating_R17+ is the age rating of Anime_Joker_Game.	Yes
	 Description: Age_Rating_R17+ Types AgeRating Same Individual As Different Individuals Age_Rating_G, Age_Rating_PG, Age_Rating_PG13 Property assertions: Age_Rating_R17+ Object property assertions isAgeRatingOf Anime_Mayoiga isAgeRatingOf Anime_91_Days_Recap isAgeRatingOf Anime_Joker_Game isAgeRatingOf Anime_91_Days isAgeRatingOf Anime_Under_the_Dog isAgeRatingOf Anime_Boku_dake_ga_Inai_Machi Data property assertions Negative object property assertions Negative data property assertions		
10.	Is Author_Yuusuke_Murata the author of the Comic_One_Punch_Man?	Yes, Author_Yuusuke_Murata is the author of Comic_One_Punch_Man	Yes
	 Description: Author_Yusuke_Murata Types Author Same Individual As Different Individuals Author_Akihito_Tsukushi, Author_BUDDY, Author_Da_Xia, Author_Feng_Xi, Author_Fujimaru, Author_Il_Kwon_Ha, Author_Kou_Yoneda, Author_ONE, Author_Ponkan_8, Author_Shen_Lei, Author_Wataru_Watari, Author_Yoshitoki_Ooima, Author_Yunji_Lee Property assertions: Author_Yusuke_Murata Object property assertions isAuthorOf Comic_One_Punch_Man Data property assertions Negative object property assertions Negative data property assertions		

Discussion of Issues/Problems

Issue: Not all series have its entries to fully satisfy the non-functional requirement
Solution (if possible): Extending the ontology to cover more database resource sites.

Issue: Large ontology may create overlapping classes/subclasses, especially when inferred.
Solution (if possible): May rename classes to avoid overlapping, or change relations between those classes.

Issue: Various classes may cause conflicts when inferencing
Solution (if possible): Go through each class carefully when inserting as well as check inferenced result to make sure the ontology is working as desired.

Issue: Protégé is not a platform that supports collaboration for a group to work on the same ontology. Towards the beginning of the project, the ontology file was uploaded to google drive and when a group member wanted to edit the file, they had to check if anybody else was working on the file, then they downloaded the latest version. This was tedious, especially when everyone wants to contribute to the ontology but don't want to overwrite others work.
Solution (if possible): We uploaded the file on GitHub, which makes collaboration so much easier than uploading the file every time someone makes new updates.

Issue: When developing the ontology, the larger the ontology got, the more computational power it required. Running the inferencer towards the end took a very long time. Some members could not run the final version on their own machines.

Solution (if possible): Members of our group takes turn run the ontology on machines provided at the lab. In addition, some unnecessary classes were removed to increase the speed of the inferencer (e.g. we limited the number of seasons to 2015-2018)

Contribution

Member	Contribution	Evaluation
Daniel Barnes	● Subclasses expansion ● Individuals/Instances gathering ● Presentation documentation ● Non-functional requirements ● Conceptualisation	Fully contributed
Hoang Duong Nguyen	● Subclasses expansion ● Individuals/Instances gathering ● Non-functional requirements ● Conceptualisation ● Discussion of Issues	Fully contributed
Jingwang Teh	● Domain contribution ● Initial ontology structure ● Scope ● Functional requirements ● Further class expansion ● Inference evaluation	Fully contributed
Minh Tan Le	● Subclasses expansion ● Individuals/Instances gathering ● Non-functional/functional requirements ● Conceptualisation ● Discussion of Issues	Fully contributed

Reflective Learning

Reflective Learning from Jingwang Teh

From the ontology that was created, I understood the need to efficiently plan the structure of the ontology and its relations so that there are fewer unnecessary details that would have otherwise lead to an overly complex ontology that can be slow to process.

I have also learnt to structurally expand the ontology in terms of functionality and complexity in order to accommodate an expanded scope.

Besides that, it is also important to communicate well with the team, which lets us establish a standardized method of implementation, such as a standardized naming convention.

Lastly, I have understood the different methods that parts of the ontology can be implemented, and it is important to have a bigger perspective even when looking at the details of the ontology.

Reflective Learning from Duong Nguyen

Throughout the duration of the assignment, I have learned so many things that were useful. We did an ontology about anime based on the anime database. I have improved my understanding about OWL and got a good chance to practice writing the ontology with my group. The ontology itself is also large and it was a good challenge to gather and handle all the information for the ontology.

Developing ontology also helps me understand the relations and properties of a certain area of information, which in our case is the anime database. This also helps me to have a logical and structural overview of the information. Moreover, working on this assignment also gives me a chance to collaborate with other group member and improve my teamwork skills. Overall, the assignment is a very beneficial experience.

Reflective Learning from Le Minh Tan

The first thing that I have discovered in this assignment was the advantage of working as a part of a group. I have been enlightened that good teamwork is the only key to success. And with every member has their own ideas for the assignment, I have learnt how to express mine and how to work collaboratively with the team.

Besides, I have also realised that a proper planning could be extremely beneficial when the time is limited. A good plan will definitely offset future changes, help increase the complexity of the ontology we are building, and provide an efficient and effective control of the whole process.

And most importantly, the assignment has offered us a great opportunity to learn to build an entire ontology from scratch. I have also discovered that working on a topic you are interested in could turn your assignment into tons of joys. For us, that topic is anime and manga. And I have indeed found a lot of fun just by creating the classes and properties, and adding the instances to demonstrate the use of our ontology.

Reflective Learning from Daniel Barnes

As the ontology was based on anime and manga, a subject which I was not too familiar with before, designing and implementing an ontology based on it greatly increased my knowledge of the domain.

Through completing this assessment, I gained first-hand experience designing and implementing a larger scale ontology. I now have a greater understanding of implementing defined classes and its importance for inferencing purposes. I also have a better understanding of how inferencing works and how/when to use it. Using protege to implement the ontology, allowed me to improve my proficiency with protege.

The assignment gave me experience in using a methodological approach to build an ontology from scratch, in addition to having to make design decisions such as what entities are classes, properties or individuals.

Having to test the ontology for errors/inconsistencies, I learnt how to evaluate an ontology to determine its correctness.

After completing the assignment, I realise the importance of having a plan before you begin implementation. This can save a lot of time and confusion, especially if you have to redo poorly designed elements of the project. It is also extremely important to maintain open and frequent communication between team members, as the quality of the project can reflect any problems between team members. Communication is also important to avoid any confusion/misunderstanding and to keep the project on schedule.