

My ontology is about one of the best television series to hit the airwaves. Despite critical acclaim, *Freaks and Geeks* had difficulty finding an audience when it was first released in 1999. It only lasted one season but has since made it to cult status and the show now has a loyal and growing fanbase. Many of its stars, such as James Franco, Jason Segel, Seth Rogan and Busy Philipps launched their careers by starring in *Freaks and Geeks*.

The series is about a group of teenagers attending William McKinley high school in 1980. The show outlines how the characters navigate the complicated high school environment while trying to figure out who they are and where they fit. It is a heartfelt and honest show about growing up, friends, family, hardships and young love. It is a show that many people can both relate to and reminisce through. Whether a person's high school experience was good or bad, there is something for everyone to enjoy watching *Freaks and Geeks*.

The areas/domains I chose to focus on for my ontology were the following:

Actors. The real-life people who brought the characters to life on screen. Casting is incredibly important in creating a successful dynamic. I just provided basic information about the actors who played both the main and frequently recurring characters. I used IMDb to assist me with defining each actor. I also created an object property between each actor and their character.

Awards and Nominations. As the show was cancelled after only one season, I thought it was important to showcase the many awards that the show was nominated for and the one that it won. I used multiple sources to determine what the show was nominated for and I used the exact descriptions noted on the awards' websites to define them. Where I could, I created an object property between the nominee/winner and the award (and vice-versa).

Characters.

- **Frequently Recurring Characters.** I used my own thoughts on the characters to define which characters I thought belonged in this class. While there are more recurring characters than I included, I opted to add the ones that I thought had a major impact on the story lines. I defined the characters based on my own experience with the show. As noted above, I created an object property between the actors and their characters as well as any applicable story arcs.
- **Main Characters.** There are a few characters I included in this class that perhaps some would not have, such as Kim Kelly and Ken Miller. I selected the characters that immediately came to mind when I thought of the series. As with the recurring characters, I defined the characters based on my own experience with the show. Again, as noted above, I created an object property between the actors and their characters as well as any applicable story arcs.

I also created male and female subclasses of each in an attempt to create inverse relationships – not sure it worked well. I also used this to create existential property restrictions as some of the male and female characters are considered boyfriends and girlfriends respectively.

Episodes. There would not be a television series without the episodes. To keep the synopses short and sweet, I used the descriptions provided by NETFLIX. There were 18 episodes in total. Unfortunately, I was unable to get Protégé put them in the correct order. I tried to number them, but that did not help. At least the reader will be able to determine the order by the associated episode number. I was able to use the episodes to build object properties between the episodes, characters and story arcs.

Executive Producer. I thought it was important to add this as Judd Apatow, the Executive Producer, loved working on this series. He has cast many of the actors from the show in his successive projects. I have read quite a few articles where Mr. Apatow begrudges and laments the cancellation of the show. I quoted the definition of an executive producer's job to properly define the job itself and researched information about Mr. Apatow using IMDb.

Genre. I thought it important to include the genre of the television series for readers to gain a sense of the type of series it is.

Main Writer. Without the great writing, the show would not be what it is. I noted the one writer, Paul Feig, who was involved in writing all 18 episodes. This writer was also nominated for some awards. I used my own ideas to define the writer position and researched information about the writer himself using IMDb. As noted, I created object properties between him and the awards for which he was nominated.

Story Arcs. To interest viewers, television series need to have strong and intriguing story arcs. I took the ones that I found to be the most prominent. While story arcs typically run over more than one episode, I did select some that were just encapsulated within a single episode. Perhaps if the show had been picked up for more seasons, the episodes would not seem as important, but since the series covered only a single season, I feel as though some arcs presented themselves on a micro level. I defined the story arcs based on my own thoughts of what I thought were important elements of the show's story line. As noted above, I created object properties between characters, episodes and story arcs.

I attempted to create universal relationships – again not sure it worked well – as certain story arcs related only to specific episodes.

Television Network. Without a network, there would not be a television series, so I thought it was important to include NBC, as it was the network that brought us *Freaks and Geeks*. I used NBC's website to help me define it.

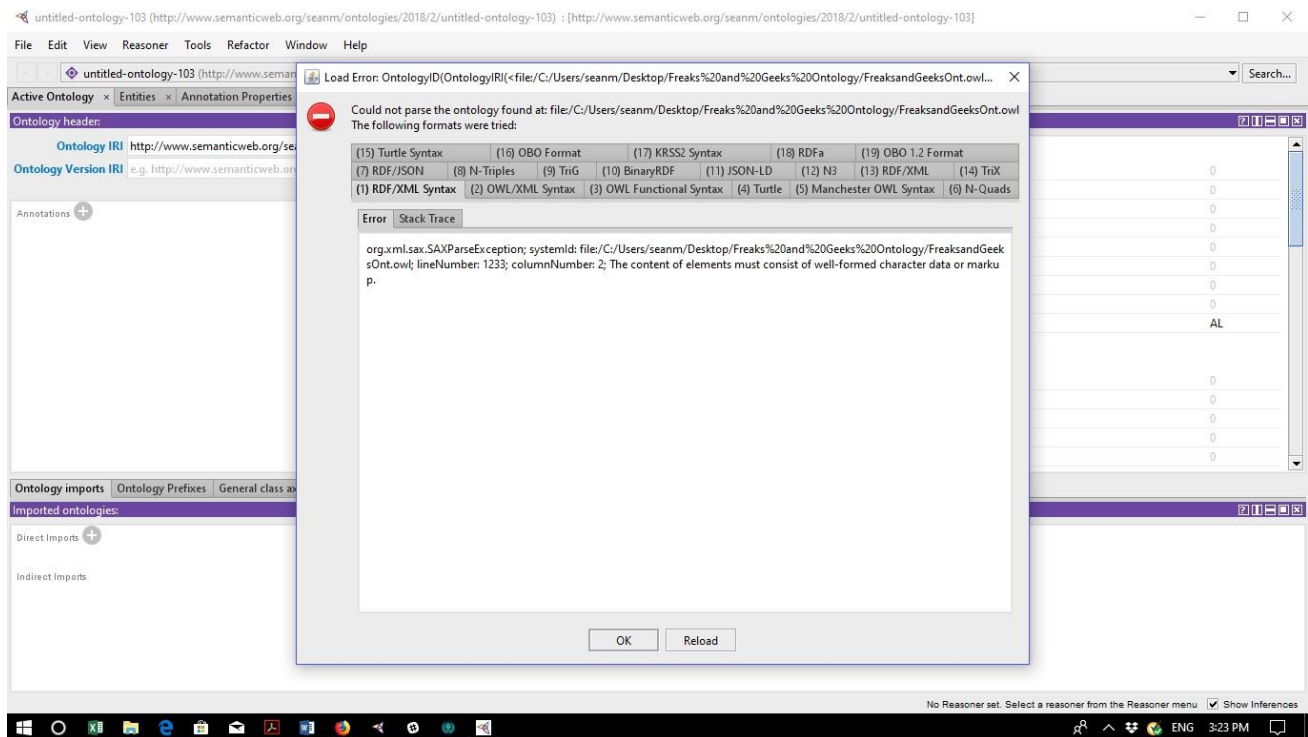
Object Properties. I had a hard time creating inverse functional properties. I followed the tutorial several times and Googled options for further assistance, but I just could not get my head around how to make it work properly. I found that using simple functional/symmetric object properties enabled me to view the relationships in the OntoGraph (which was my favourite feature). Using inverse relationships, I could not figure out how to make the actual relationships visible in the OntoGraph – perhaps that is not possible. My attempts at inverse functional properties can be viewed in the *Object Properties* tab in the subclasses entitled *characters* and *storyArcs*.

I also am aware that the reasoner is still picking up issues. I battled with its findings quite a bit. When I changed one thing to fix an issue, it simply created issues somewhere else.

In addition: My unfortunate experience with GitKraken and GitHub

I must note that I struggled immensely using GitKraken. I was able to get it up and running and it worked well for a few of my commits, but when I tried to change a commit name and thought I had hit cancel when I received a warning, things went off the rails. Apparently, a secondary branch was created. I was able to make a few more commits and then I got in a situation where they were merged and then

Protégé would no longer open the OWL file on my desktop. I received the below error message when I went to do so:



I was able to recover a previous version via my GitHub account from a few hours earlier, but that resulted in quite a bit of lost hours of work that had been done between commits.

For my final submission, it was determined that it would be best for me to create a new repository, which I tried to do via GitKraken, but I could not get GitKraken to work with my desktop, but my GitHub account did respond to it. I made multiple attempts to set up a repository using GitKraken, and somehow ended up my original repository and somehow made an additional merge commit when I thought I was making a cancellation in my new repository. When I went to open my OWL document in Protégé, I once again got the above error message. Fortunately, I had made a new copy of OWL file and renamed it for my new repository, so I still had a version of my current document that worked in Protégé. You can see the history in my original repository by checking out the various commits in the *Freaks-and-Geeks-Ontology*.

In the end, what saved me was essentially backing up my work on my own PC versus using GitKraken and GitHub. This made me incredibly nervous and while I was considering making additional annotation properties, I opted to just submit my ontology immediately (as it was) so that I could no longer corrupt my files. In the end, I found it ironic that the version control system is what, in the end, made it more difficult for me to complete my assignment. While I understand the value of version control and know that this was user error, being a novice user with little formal version control experience, this has made me somewhat leery of using version control platforms in the future.