Federation of Ontologies - Books

Anjali Krishna Prasad, Priya Ananthasankar, Nitish Mohan Shandilya and Pooja Bhandari
Viterbi School of Engineering
University of Southern California
Los Angeles, California 90007, USA
Email: akprasad@usc.edu, panantha@usc.edu, nshandil@usc.edu, poojab@usc.edu

***Abstract*—Our project aims at providing unified information about books. We have created a federated ontology that includes data from Goodreads Ontology, Amazon Ontology and Comics Ontology. We then query the federated ontology using SPARQL and show that federated ontology is more efficient than collection of individual ontologies.**

***Keywords*—RDF; Federatedontology**

I. Introduction

Ontology is a formal representation of knowledge by set of concepts within a domain and relationships between those concepts. We need ontologies for a number of reasons. Ontological analysis clarifies the structure of the knowledge. Given a domain, its ontology forms the heart of any system of knowledge representation from that domain. Without ontologies, or conceptualization that underlie knowledge, there cannot be a vocabulary for representing knowledge. Thus, the first step in devising an effective knowledge representation system and vocabulary is to perform an effective ontological analysis of the field, or domain. Weak analysis of the ontology leads to incoherent knowledge of bases. A federated ontology is formed when we merge two or more ontologies sematically to form one meta ontological structure.

A number of proposals are available from the database community for developing federated ontology systems and that resemble the decentralized structures required in the Semantic Web. We consider an architecture for federated ontologies on the Semantic Web as motivation and starting point of our work. A bottleneck for federated ontologies in the Semantic Web is the process of integrating or merging specific ontologies. The process of *ontology merging* takes as input two (or more) source ontologies and returns a merged ontology based on the given source ontologies.

In this paper we prove that querying the federated books ontology gives better results as compared to the individual ontologies. Moreover, we show that the federated ontology results are much more efficient.

The remainder of the paper is as follows. We start our paper introducing the problem statement in section II. Then we will look at the way we generated individual ontologies in section III , followed by problems in querying mutliple ontologies in section IV. Then we will discuss creation of federated ontology in section V. Section VI shows SPARQL queries fired on the two types of ontologies. Section VII talks about richness in meta ontology. Section VIII gives the results of the project. Section IX lists the conclusions, section X sets goals for future work and section XI lists the references.

II. Problem Formulation

We can come across various websites that provide information about books. Usually the information provided by the websites is not the same. They can look at books from many different perspectives and provide information based on that. The information provided from different websites can sometimes be completely disjoint or a part of it may even overlap with others. For example, Goodreads is a website that looks at books from social networking perspective. It provides various attributes related to books such as title, genre, reviews, story etc. organized across multiple web pages. Amazon is another website that looks at books from a buyer/seller marketplace and product perspective. It has attributes like hardcover pages, price, weight of shipment, etc. An idea developed by us is the comics website. A comic website looks at books from a character and cartoon perspective. A comic can be adaptation of a book, so it has a team of authors – letterer, inker, illustrator, publisher, etc.

Due to this distribution of diverse information about books on multiple websites, users cannot get a unified view of all the relevant information pertaining to a books at one place. For example, if a user wants to check out the price of a comic based on a book which has good reviews, he has to navigate to three different websites. Firstly, he has to check the name of the comic which is based on the book he likes, then navigate to the goodreads website to check if that book has good reviews and finally he has to go to amazon to check the price of the book. This is rather cumbersome. It would be better if the user can see the all the information about the book he wants at one place. Viewing unified information on a website works out better for the user.

Solving this problem of unifying the diverse pieces of information is the main goal of our project. We gather information about books from three different sources which looks at books from very different perspectives. We then create a unified meta ontology that merges these three ontologies in a very unique way. This basically makes sure that the redundant information, like title of the book, is merged. At the same time, it makes sure that there is still good level of distinction between the merged ontologies. For example, the goodreads ontology has list of genres and the comics ontology also has a single genre. We make sure that we retain both these genres and not pick one at random for our meta ontology. We then use SPARQL to query our meta/federated ontology. We will prove that the queries fired on meta ontologies gives much better results and efficiency as compared to the queries fired on individual ontologies. We will use edit distance measure to compare the efficiency of the queries fired on meta ontology and the individual ontologies. Here, we want to clarify that the goal of the project is not to create a mashup application. A mashup, in web development, is a web page, or web application, that uses content from more than one source to create a single new service displayed in a single graphical interface, but they cannot be used to run integrated queries. Our goal is to make an ontology that can give correct and efficient results to integrated queries.

III. Individual ontologies

In this section we will talk about the individual ontology design of goodreads, amazon and comics ontologies. In our project, we are focusing on data available from the following websites. The attributes crawled from each page are listed in Table 1:

1. Comics (created by us)
2. Amazon (www.amazon.com)
3. Goodreads (www.goodreads.com)

Table 1 shows the main attributes of the above mentioned three ontologies. In the following subsections we will talk about each ontology and its design separately.

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors*).

Table 1: Information extracted from websites

Website	Extracted Information
Goodreads	Book, Vendor, Author, Genres, Edition, Review, Story , Character, Number of reviews, Title, Rating, Awards, Language, ISBN, Introduction, Description.
Amazon	Title, Price, Weight of shipment, Hardcover pages, Author, Publisher, Synopsis, Language, Genre , Rating
Comics	Illustrator, Author, Letterer, Inker, Genre, Universe, Team, Comic Age, Panel, Story, Character, Plot, Series, Issue

A. *Comics Ontology*

Concept: The Comics ontology contains concepts and interrelations about the visual narrative of a book. A comic approaches a book concept from a visual perspective. Currently there is no ontology which represents this aspect of a book and in this project we have aimed to create a unique ontology.

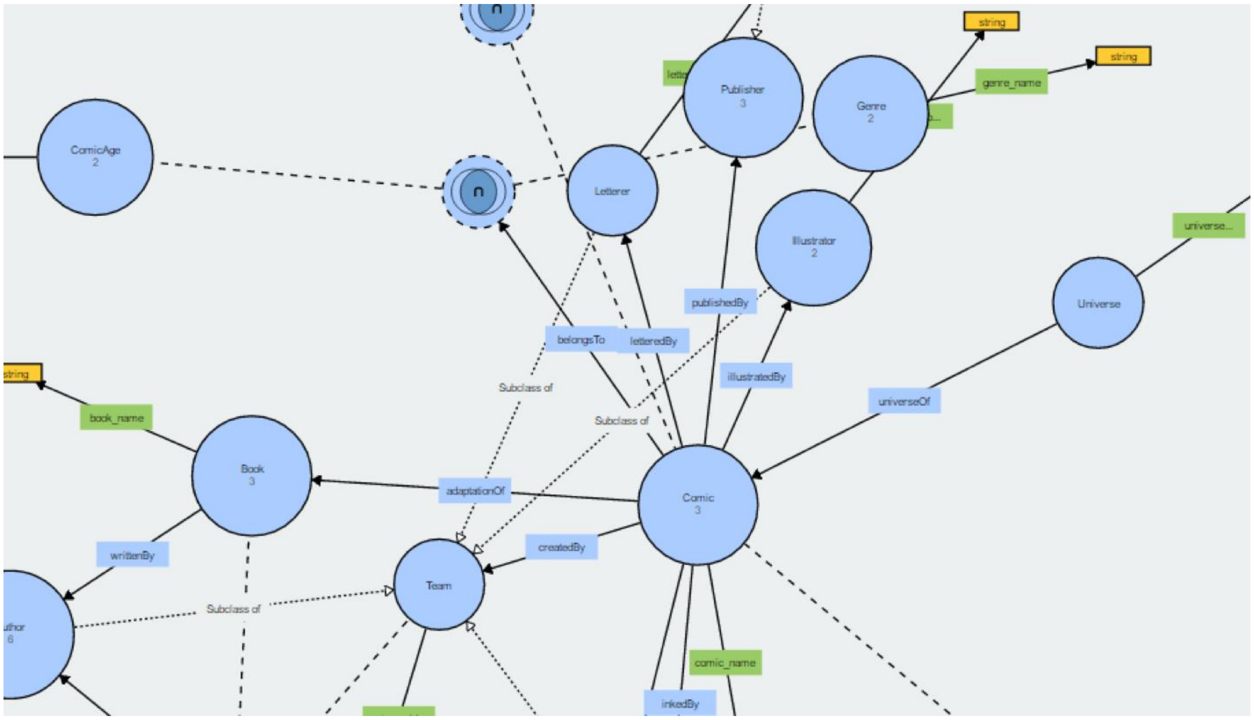
Technical Definition: **Comics** is a medium used to express ideas by images, often combined with text or other visual information. Comics frequently takes the form of juxtaposed sequences of panels of images. Often textual devices such as speech balloons, captions, and onomatopoeia indicate dialogue, narration, sound effects, or other information. Size and arrangement of panels contribute to narrative pacing. Cartooning and similar forms of illustration are the most common image-making means in comics; *fumetti* is a form which uses photographic images. Common forms of comics include comic strips, editorial and gag cartoons, and comic books. Since the late 20th century, bound volumes such as graphic novels, comic albums, and *tankōbon* have become increasingly common, and online webcomics have proliferated in the 21st century.

Semantic Concepts: The Comics ontology has the following semantic concepts:

1. Comic - A visual representation of a book consisting of panels of visually drawn data where each panel consists of some caption of words. A graphic novel is also considered a comic.

2. Universe - A comic might sometimes contain characters who belong to a universe. This universe might be shared among multiple comics in the interest of building a storyline. For example: the characters created by DC comics belong to the DC universe and characters created by Marvel comics belong to Marvel universe.

Figure 1: Comics ontology structure



3. The Justice League comic is one which combines characters from both these universes.

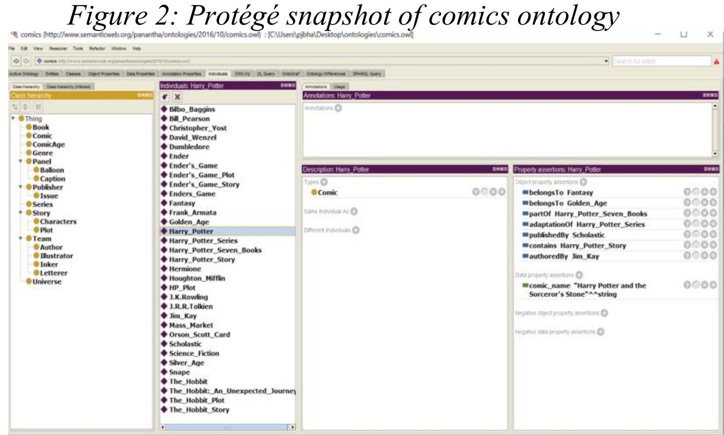
4. Comic Age - A comic typically is classified in terms of age. For example, Comics prevalent during the 70's were called Silver Age Comics and before that it was the Golden Age comics.

5. Publisher - A comic is closely tied to a publisher.

6. Team - A comic is created by a team of people consisting of authors, letterers, inkers and illustrators

   A) Author - A person who writes the main story of a comic in a typical storyboarding textual format.
   B) A Letterer - A person who decides the calligraphy and fonts of a comic text. A comic text also has to be a visually appealing and it is the letterer who decides the format.
   C) Inker - A person who decides the color coding and format of a comic book.
   D) Illustrator - A person who illustrates the art form with actual cartoon like faces

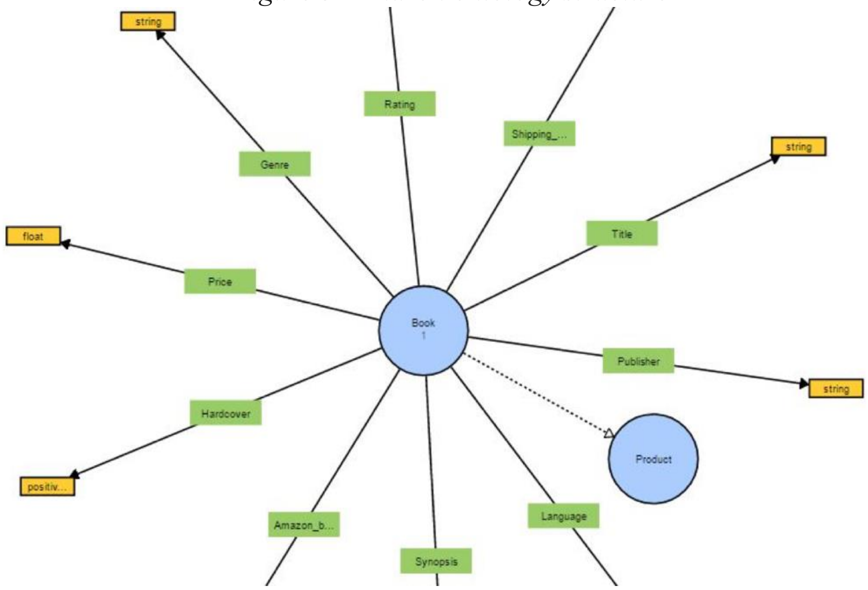
Figure 2: Protégé snapshot of comics ontology



Notable Semantic Relations: The Comics ontology also has the following notable semantic relationships defined (among others):

1. adaptationOf - Comic is an **adaptationOf** a Book. Sometimes a comic can be a standalone story but mostly it is a visual adaptation of a Book. Through this relationship a comic ontology can be related to a Book ontology.
2. universeOf - Universe is **universeOf** a Comic.
3. createdBy - Comic is **createdBy** a Team.
4. consistsOf - Team **consistsOf** Author, Illustrator, Letterer and Inker.

Figure 3: Amazon ontology structure



B. Amazon Ontology

Concept: The Amazon ontology contains concepts and interrelations about the book as a product. Amazon approaches a book concept from a buyer/seller perspective. This ontology has been inspired from amazon.com.

Technical Definition: **Amazon Books** is a chain of retail bookstores owned by online retailer Amazon.com.

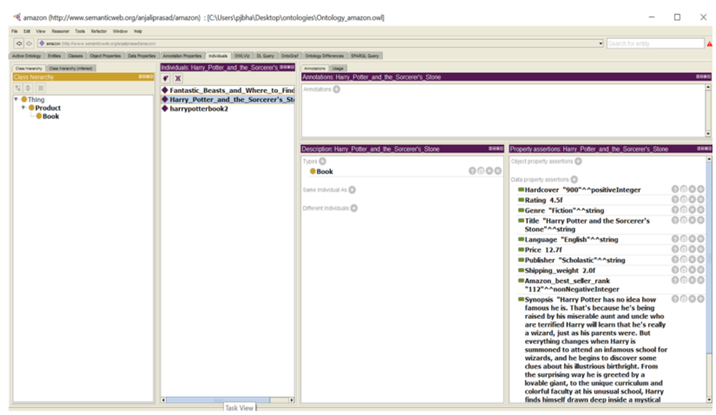
Semantic Concepts: The amazon ontology has the following semantic concepts:

- Product - This can be anything sold on amazon like book, clothes, hair product, etc.
- Book – This is the concept we are interested in. It has attributes like price of the book, weight of the shipment, number of hardcover pages, the author of the book, etc.

Notable Semantic Relations: The amazon ontology also has the following notable semantic relationships defined (among others):

- isPartOf - Book is an isPartOf Product. Product has many children. Book has Product as its parent.

Figure 4: *Protégé snapshot of amazon ontology*

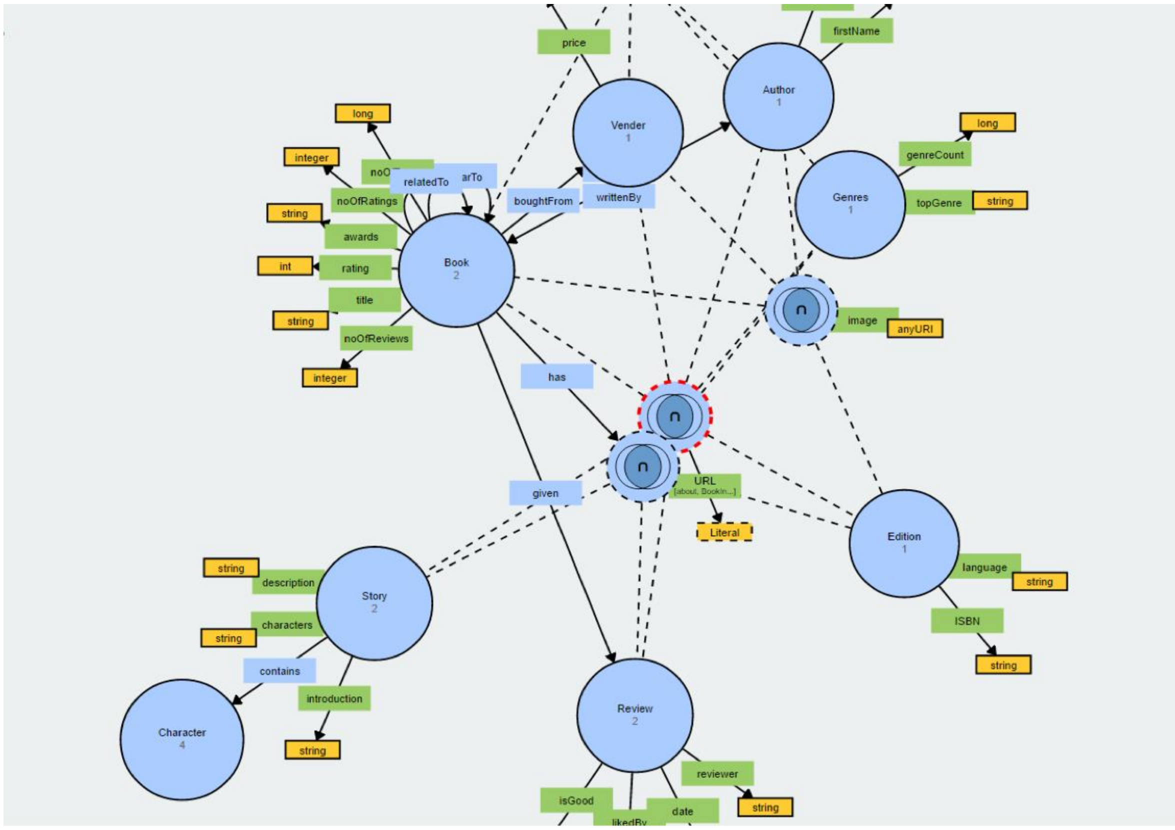


C. Goodreads Ontology

Concept: The goodreads ontology contains concepts and interrelations of the books. Goodreads approaches a book concept from a social networking perspective. The design of this ontology has been inspired from the website goodreads.com.

Technical Definition: Goodreads website allows individuals to freely search Goodreads' extensive user-populated database of books, annotations, and reviews. Users can sign up and register books to generate library catalogs and reading lists. They can also create their own groups of book suggestions, surveys/polls, blogs, and

Figure 6: Goodreads ontology structure



discussions. It has a list of genres with the number of votes given by different users. For example: Harry potter and the sorcerer's stone will have a list of genres and the votes for each genre as shown in the figure 5. Users can have a look at reviews by other users. It also gives a synopsis of the story line of the book. Goodreads also gives description of the characters of the book, the author, edition and the publisher.

On the Goodreads website, users can add books to their personal bookshelves, rate and review books, see what their friends and authors are reading, participate in discussion boards and groups on a variety of topics, and get suggestions for future reading choices based on their reviews of previously read books. Once users have added friends to their profile, they will see their friends' shelves and reviews and can comment on friends' pages. Goodreads features a rating system of one to five stars, with the option of accompanying the rating with a written review. The site provides default bookshelves—read, currently-reading, to-read—and the opportunity to create customized shelves to categorize a user's books.

Figure 5: Genre list of Goodreads

GENRES

Fantasy	42,687 users
Young Adult	13,836 users
Fiction	11,935 users
Fantasy > Magic	3,924 users
Childrens	3,433 users
Adventure	2,269 users
Classics	1,615 users
Science Fiction Fantasy	1,311 users
Childrens > Middle Grade	1,204 users
Novels	976 users

The website facilitates reader interactions with authors through the interviews, giveaways, authors' blogs, and profile information. There is also a special section for authors with suggestions for promoting their works on goodreads.com, aimed at helping them reach their target audience.

Semantic Concepts: The goodreads ontology has the following semantic concepts:

1. Author – This is the author of the book. It has attributes like name, url, about and image.

2. Edition – This is the edition of the book. It has attributes like ISBN (International Standard Book Number) and language.

3. Story – Story has attributes like description and introduction. It basically gives a brief overview of the story line

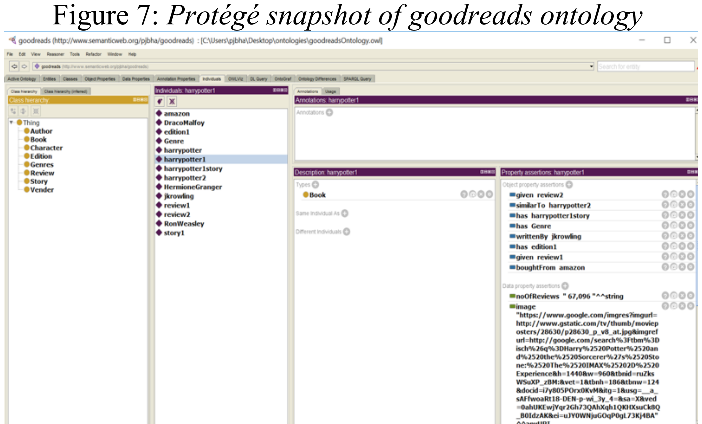
4. Genre – A list of genres with the number of votes given by different users.

5. Reviews – It has reviews concept which basically contains reviews about the book from various users.

6. Character – It is a concept which describes the characters involved in the book.

7. Vendor – Vendor is the online store a user can buy this book from.

Figure 7: *Protégé snapshot of goodreads ontology*



The Goodreads ontology also has the following notable semantic relationships defined (among others):

1. writtenBy - Book is written by an **author**.
2. boughtFrom- Book is brought from a vendor.
3. has – Book has Editions

IV. Problems with Querying Multiple Ontologies

- **Understand each ontology structure**: When we have multiple individual ontologies, we should know the structure of all the ontologies while querying them. In case of querying federated ontology, we need to know the structure of only the merged meta ontology.

- **Conflict in selection of literals:** For example, there is an attribute named title in one ontology. This title is called book name is the other ontology. So there is a conflict between these two attributes.

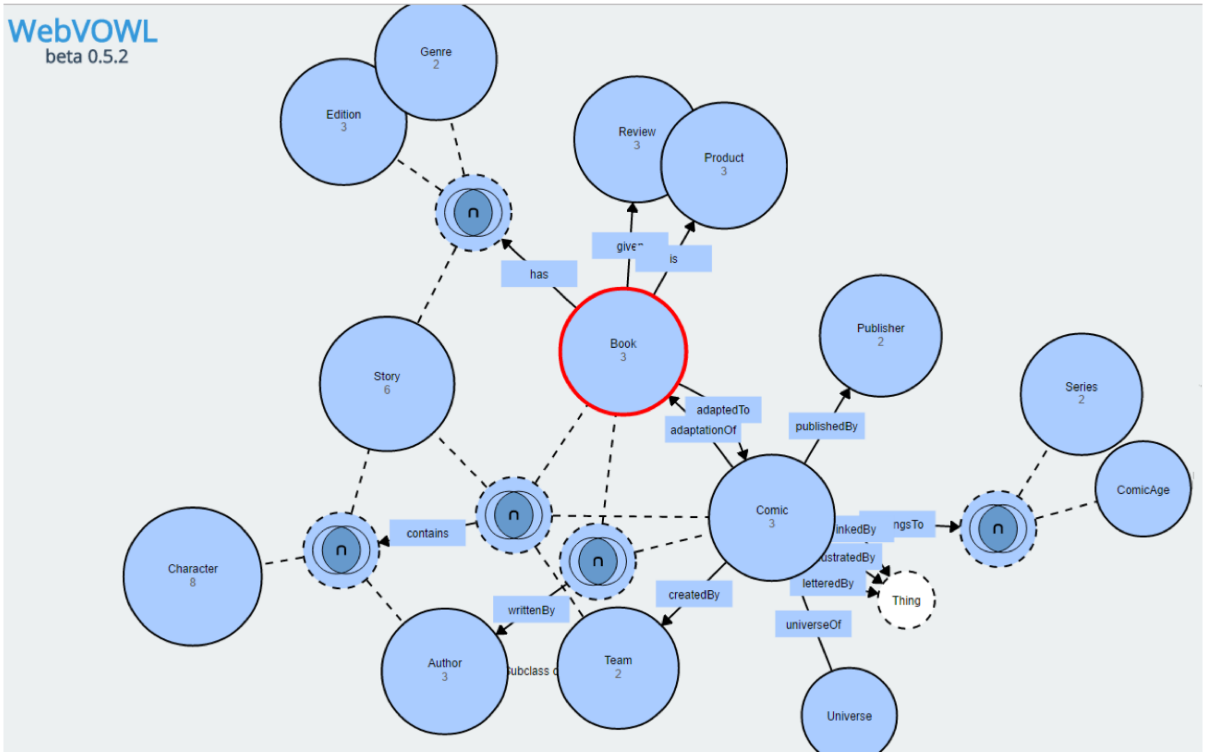
- **Difficult to match literals of different ontologies:** Now let's say we have two ontologies. One has title name "Harry Potter and the soccer's stone" and the other ontology has an attribute named book name. This have value "Harry Potter and the soccer's stone". Now when we try to query these two ontologies, we have to do literal matching which is not an efficient way of comparison and matching.

- **Limited content exposure:** The individual ontologies cannot give us all the content that we require. For example, comics ontology cannot give reviews of the book as it lacks that information, Amazon ontology cannot give the comics that are adapted from the book, etc. Therefore, there is limited content exposure when it comes to different ontologies.

- **Some queries cannot be answered:** Due to missing pieces of information in the individual ontologies, there can be many cases when the queries doesn't give the desired results and may fail. For example, When we try to find the price of the book from which the comic has been adapted, we try to match the name of the book from both the ontologies (comic ontology and amazon ontology). Now we are trying to do string matching. What if there is a slight difference in the title of the book saved in the two ontologies ex. "Harry Potter and the sorcerer's stone" and "Harry Potter and the sorcerers stone". Then this query will fail to give proper results.

- **More complex queries:** Complex queries on multiple ontologies are very inefficient.

Figure 8: Meta Ontology



V. FEDERATED ONTOLOGY

To overcome the issues with collection of individual ontologies, we will create a meta ontology that integrates all these three ontologies (comics, amazon and goodreads). Following are some of the steps taken to create meta ontology

- **Built common parent classes:** Book becomes the common parent of comics ontology, goodreads ontology and amazon ontology.

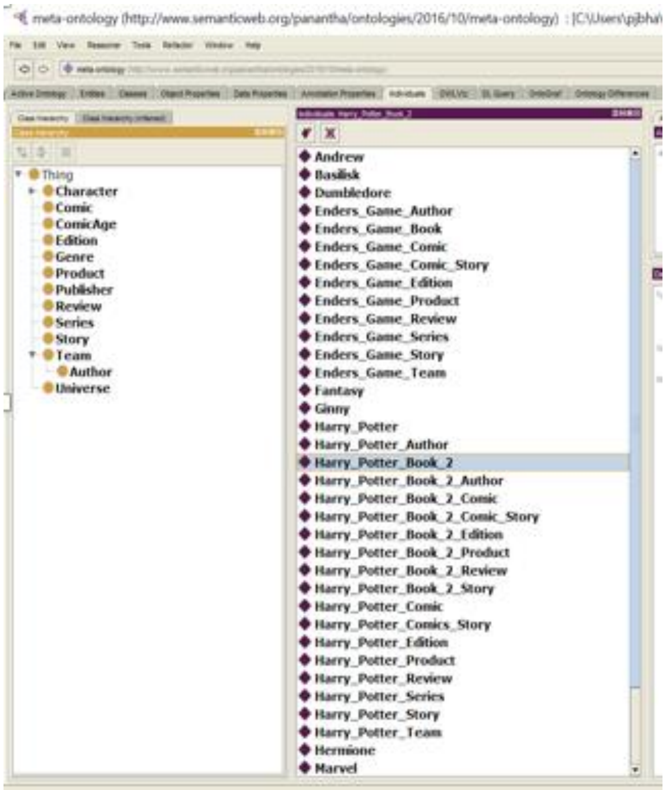
- **Removed redundant classes and unified classes:** Title of the book is present in all the three ontologies. We have unified these three attributes into a single title attribute. We have even unified common classes. For example, there is author class in the goodreads ontology and in comics ontology, but the comic author is not the same as the book author. These two will have separate instances. So we cannot eliminate any one of them. We will make a unified class named author and use separate instances of this class for both the braches of our meta ontology.

- **Maintained distinction without falsely combining classes:** Some classes may look same but they are not supposed to be combined. For example, goodreads have a list of genres for a book. This list may be different from the genre of the comic which are adaptation of this book. We are supposed to retain both these classes and not combine them.

- **Resolution of conflicts and ambiguities:** When we tried to merge different ontologies, we came across some conflicts and ambiguities which required resolution. For example: comics ontology has "writtenBy" relationship between author and the comic and goodreads ontology has "written_by" relationship between author and the book. We made sure that such ambiguities were eliminated.

- **Semantic resolution:** There can be many classes and relationships that are having different name but same meaning. We resolved such ambiguities as well. For example, writtenBy and authoredBy has the same meaning.

Figure 9: Protégé snapshot of goodreads ontology



VI. QUERYING META ONTOLOGY AND COLLECTION OF INDIVIDUAL ONTOLOGIES

In this section we will see how the meta ontology and collection of different ontologies behave when we fire SPARQL queries.

SPARQL is an RDF query language, that is, a semantic query language for databases, able to retrieve and manipulate data stored in Resource Description Framework (RDF) format.

We will use the concept of edit distance to compare the two types of ontologies. In this paper, **edit distance** is interpretted to be a way of quantifying how dissimilar ontologies are to one another by counting the minimum number of operations required to give the result of the similar query. Lesser the number of operations (tripples used) in a query , lesser is the edit distance. Low edit distance means better efficiency that means the ontolology with lower edit distance is better than ontology with higher edit distance when we see from the efficiency perspective.

The following are four queries fired on meta ontology (federated ontology) and collection of individual ontologies (amazon, comic and goodreads).

A. *Query 1: Ratings of books with comics written by Jim Kay*

We want to find the ratings of the books which are adapted as a comic authored by Jim kay. We will fire a simple SPARQL query on federated ontology and we will fire a federated query on the collection of individual ontologies. Figure 10 shows the federated query for collection of different individual ontologies. Figure 11 shows SPARQL query for meta ontology

Figure 10: SPARQL for individual ontologies for query 1

```
PREFIX a: <http://www.semanticweb.org/amazon#>
PREFIX c: <http://www.semanticweb.org/ontologies/2016/10/comics.owl#>
PREFIX g: <http://www.semanticweb.org/goodreads#>
select ?ans where
{
    ?bk g:rating ?ans.
    ?bk g:title ?title.
    ?b c:book_name ?title.
    ?c c:adaptationOf ?b.
    ?c c:authoredBy ?a.
    ?a c:author_name ?str.
    Filter  regex(?str,"Jim Kay")
}
```

Figure 11: SPARQL for meta ontology for query 1

```
PREFIX x: <http://www.semanticweb.org/ontologies/2016/10/meta-ontology#>
SELECT (str(?ans) as ?rating) WHERE
{
    ?b x:rating ?ans.
    ?c x:adaptationOf ?b.
    ?c x:createdBy ?t.
    ?t x:illustrator_name ?i.
    Filter regex(?i,"Jim Kay")
}
```

We know that edit distance is the number of operations. We can see that edit distance of query for individual ontologies is 7 and the edit distance of query for meta ontology is 5. Clearly meta ontology gives an efficient result as compared to collection of individual ontologies.

B. *Query 2: Comics with same genre as that of book*

We want to find the comics with the same genre as that of the book from which it is inspired. We will fire a simple SPARQL query on federated ontology and we will fire a federated query on the collection of individual ontologies. Figure 12 shows the federated SPARQL query for collection of different individual ontologies. Figure 13 shows SPARQL query for meta ontology

We know that edit distance is the number of operations. We can see that edit distance of query for individual ontologies is 8 and the edit distance of query for meta ontology is 5. Clearly meta ontology gives an efficient result as compared to collection of different ontologies.

Figure 12: SPARQL for individual ontologies for query 2

```
PREFIX a: <http://www.semanticweb.org/amazon#>
PREFIX c: <http://www.semanticweb.org/ontologies/2016/10/comics.owl#>
PREFIX g: <http://www.semanticweb.org/goodreads#>

select ?ans where
{
    ?b1 g:title ?n.
    ?c c:adaptationOf ?b.
    ?b c:book_name ?n.
    ?c c:belongsTo ?genre.
    ?genre c:genre_name ?g.
    ?b1 g:has ?g1.
    ?g1 g:topGenre ?g.
    ?c c:comic_name ?ans
}
```

Figure 13: SPARQL for meta ontology for query 2

```
PREFIX x: <http://www.semanticweb.org/panantha/ontologies/2016/10/meta-ontology#>
SELECT ?ans WHERE
{
    ?c x:comic_name ?ans.
    ?c x:hasGenre ?n.
    ?c x:adaptationOf ?b.
    ?b x:has ?g.
    ?g x:topgenre ?n
}
```

It is interesting to note that the genre class is different for the comics and goodreads. We did not merge them to be one. Although we haven't combined them, still the edit distance of the query fired on meta ontology is less than the edit distance of the query fired on the collection of individual ontologies. This clearly shows meta ontology is better as edit distance is less despite preserving distinction.

C. ***Query 3:*** *JK Rowling books available for less than 100$*

We want to find the books written by JK Rowling available for less than 100$. We will fire a simple SPARQL query on federated ontology and we will fire a federated SPARQL query on the collection of different ontologies. Figure 14 shows the federated query for collection of different individual ontologies. Figure 15 shows SPARQL query for meta ontology

The edit distance of both the queries is 7. The price attribute is present only in amazon when we consider the set of three individual ontologies. Although the book can be sold by many vendors. So when we consider collection of individual ontologies, we need to consider many vendor ontologies, which is not feasible as we have to keep in mind the structure of each and every ontology. When such situation arises in meta ontology, we have a single ontological structure which merges all the ontologies we need. So this leads to much better content exposure as meta ontology is going to consider books available by other vendors as well. We can see that the edit distance is same for both but the meta ontology has better content exposure than the collection of individual ontologies.

Figure 14: SPARQL for individual ontologies for query 3

```
PREFIX a: <http://www.semanticweb.org/amazon#>
PREFIX c: <http://www.semanticweb.org/ontologies/2016/10/comics.owl#>
PREFIX g: <http://www.semanticweb.org/goodreads#>

select (str(?n)as ?books) where
{
    ?b a:Title ?n.
    ?bo g:title ?n.
    ?bo g:writtenBy ?athr.
    ?athr g:LastName ?a.
    Filter regex(?a,"Rowling").
    ?b a:Price ?p.
    Filter(?p<=100)
}
```

Figure 15: SPARQL for meta ontology for query 3

```
PREFIX x: <http://www.semanticweb.org/ontologies/2016/10/meta-ontology#>
SELECT distinct (str(?pr) as ?price)
WHERE
{
    ?b x:title ?ans.
    ?b x:is ?p.
    ?p x:price ?pr.
    Filter (?pr<=20).
    ?b x:writtenBy ?n.
    ?n x:author_name ?str.
    Filter regex(?str,"J.K.Rowling")
}
```

D. ***Query 4:*** *Comics having same characters as that of its book*

We have to find the books and comics inspired by the book having the same characters. We will fire a simple SPARQL query on federated ontology and we will fire a federated query on the collection of different ontologies. Figure 16 shows the federated query for collection of different individual ontologies. Figure 17 shows SPARQL query for meta ontology.

Figure 16: SPARQL for individual ontologies for query 4

```
PREFIX a: <http://www.semanticweb.org/amazon#>
PREFIX c: <http://www.semanticweb.org/ontologies/2016/10/comics.owl#>
PREFIX g: <http://www.semanticweb.org/goodreads#>

select ?ans where{
  ?story2 g:contains ?char.
  ?char g:name ?n.
  ?story1 c:contains ?char1.
  ?char1 c:characters ?n.
  ?b1 g:has ?story2.
  ?c c:contains ?story1.
  ?c c:adaptationOf ?b.
  ?b c:book_name ?bn.
  ?b1 g:title ?bn.
  ?c c:comic_name ?ans
}
```

We can see that edit distance of query for individual ontologies is 10 and the edit distance of query for meta ontology is 5. In meta ontology query we are trying to do instance matching which is a better way of querying than literal matching which we have to do in the case of collection of individual ontologies to get result. In this case also its evident that meta ontology is more efficient and gives better results than collection of individual ontologies.

Figure 17: SPARQL for meta ontology for query 4

```
PREFIX x: <http://www.semanticweb.org/ontologies/2016/10/meta-ontology#>
SELECT distinct (str(?ans) as ?comics)
WHERE
{
    ?c x:comic_name ?ans.
    ?c x:contains ?story1.
    ?story1 x:contains ?char.
    ?b x:has ?story2.
    ?story2 x:contains ?char.
    ?c x:adaptationOf ?b
}
```

VII. RICHNESS IN META ONTOLOGY

- **Meta ontology can answer rich set of queries:** As we have seen in the previous section, meta ontologies give betters results for rich and complex queries. Meta ontology handles complex query in the right way. We do not need to do literal matching which makes sure that we don't miss results just because of slight mismatch of strings.

- **Minimal operations compared to querying multiple ontologies:** We have seen in the previous section that queries fired on meta ontology have lower edit distance as compared to queries fired on collection of individual ontologies. This shows that the number of operation to query meta ontology as compared to collection of individual ontologies is usually low making the meta ontology a better choice.

- **Comparison based on instances rather than literals:** We have seen in the previous section query 3 that meta ontology makes sure that we are comparing the instances rather than literals. This results in better and more accurate results.

VIII. RESULTS

The results for the experiment is mentioned in the table 2.

Table 2: Edit distance of meta ontology and collection of individual ontologies.

	Query 1	Query 2	Query 3	Query 4
Meta Ontology	5	5	7	5
Collection of individual ontologies	7	8	7	10

Brief analysis of the results:

1. **Query 1**: The edit distance of query 1 fired on meta ontology (5) is lower than the edit distance of query 1 fired on collection of individual ontologies (7) because the number of operations required to retrieve data in meta ontology is lesser than its counterpart in this case.

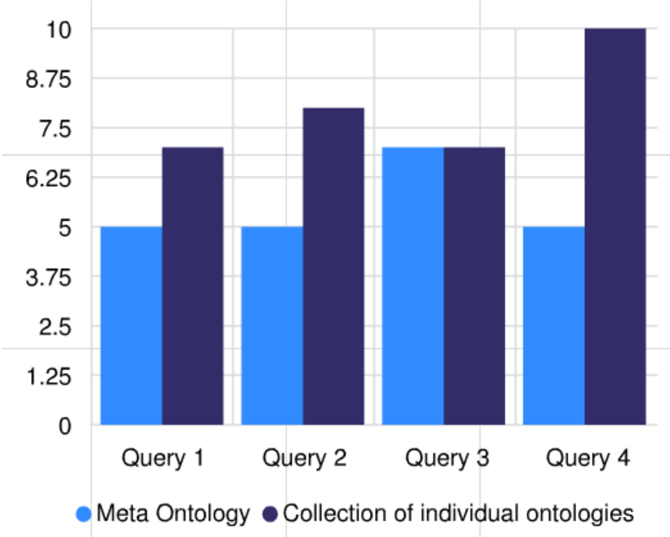
2. **Query 2**: The edit distance of query 2 fired on meta ontology (5) is lower than the edit distance of query 2 fired on collection of individual ontologies (8) because the number of operations required to retrieve data in meta ontology is lesser than its counterpart in this case. The genre class is different for the comics and goodreads. Eventhough we haven't combined the two classes, still the edit distance of the query fired on meta ontology is less than the edit distance of the query fired on the collection of individual ontologies.

3. **Query 3:** The edit distance of both the types of ontologies is same (7) but the meta ontology has better content exposure than the collection of individual ontologies. Meta ontology leads to much better content exposure as it is going to consider books available by other vendors as well. We can see that the edit distance is same for both but the meta ontology has better content exposure than the collection of individual ontologies.

4. **Query 4:** The edit distance of query 4 fired on meta ontology (5) is lower than the edit distance of query 4 fired on collection of individual ontologies (10) because the number of operations required to retrieve data in meta ontology is lesser than its counterpart in this case. In meta ontology query we are trying to match instances which is a better way of querying than literal matching which we have to do in the case of collection of individual ontologies to get result.

We can infer from the results of the queries that the meta ontologies are more stable, gives better results, higher precision and higher recall as compared to the collection of individual ontologies.

Figure 17: Results bar graph



IX. Conclusion

Ontologies are becoming an increasingly popular way of organizing data and are used in multiple domains. We have shown the value of creating federated ontology from various individual ontologies in the domain of books by answering queries which would have required exploring multiple web pages to answer.

Let's also look at the following inferences from our research:

1. **Federation of ontologies is better:** We have proved that the federated ontologies are better than collection of individual ontologies. We have used edit distance metric to compare these two types.

2. **We need a meta-ontology to answer diverse and richer set of queries:** We saw that the collection of individual ontologies fails to give results for rich and diverse queries. Meta ontologies can handle such queries better.

3. **The federated ontology performs better in terms of edit-distance and ambiguity resolution:** We have seen that the federated ontology's edit distance is usually less than the edit distance of the collection of individual ontologies proving that the former is more efficient.

4. **Federation provides maximum content exposure:** We also witnessed that the federated ontology provides better content exposure by combining all the ontologies in one meta ontology.

X. Future Work

Many other books ontologies can be integrated with our federated ontology. We can make a much richer meta ontology by crawling all the top rated books website and integrating with our meta ontology. Querying this ontology will provide excellent results with high recall as well as high precision.

When we increase the number of ontologies to be integrated into a federated ontology, we can no longer manually integrate these ontologies therefore we will require automated tools to for linking.

In future, we will use better ontology integration method by following the flow chart in figure 18.

Figure 18: Integration process

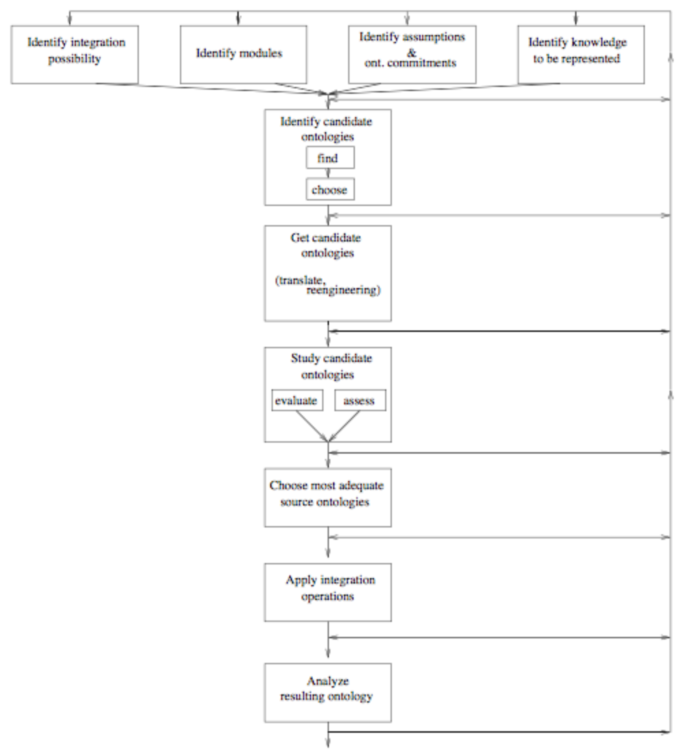


Figure 1: The integration process

We will also make a web page which would show the users the query result of the federated ontology.

XI. References

[1] Liyang Yu. A Developer's Guide to the Semantic Web. Springer, 2011.

[2] Sunitha Ramanujam, Anubha Gupta, Latifur Khan, Steven Seida, and Bhavani M. Thuraisingham. A relational wrapper for RDF reification. In Trust Management III, Third IFIP WG 11.11 International Conference, IFIPTM 2009 , West Lafayette, IN, USA, June 15-19, 2009. Proceedings, pages 196–214, 2009.

[3] http://owl.cs.manchester.ac.uk/publications/talks-and-tutorials/protg-owl-tutorial/

[4] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. Scientific American, 284(5):34–43, May 2001

[5] https://jena.apache.org/tutorials/sparql.htmlY. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[6] https://en.wikipedia.org/wiki/Resource_Description_Framework.

[7] http://www.cambridgesemantics.com/semantic-university/sparql-by-example

[8] Federated Ontology Search for the Medical Domain, Vasco Calais Pedro, Lucian Vlad Lita, Stefan Niculescu , Bharat Rao and Jaime Carbonell

[9] Ontology Merging for Federated Ontologies on the Semantic Web by **Gerd Stumme and Alexander Maedche**

[10] http://www.slideshare.net/GeorgeSam3/federated-ontology-based