

Chapter 1: Domain and Scope of Application

1.1 Introduction.

In this chapter, the domain and application scope of the StoryScape project, a system based on the use of an ontology for the management and exploration of science fiction books, will be presented. The rationale behind the choice of creating an ontology specifically for this domain and the importance of importing other online vocabularies, such as "dbo:book" and "dbo:person," will be explained.

1.2 Choice of Domain

In the context of the StoryScape project, it was chosen to create an ontology based on science fiction books. This choice stems from the distinctiveness and interest of the Science Fiction genre in the world of literature and fiction in general.

In the field of Fiction, there has never been too much emphasis on the distinction based on the length of a piece of writing, regardless of whether it is prose or poetry, novel or biography. However, in the case of Science Fiction (or Sci-Fi, as it was originally known when it was decided to give it a specific connotation as a narrative genre), the length of works has acquired a special significance.

In fact, most Science Fiction writers have produced, in the course of their literary activity, a considerable amount of short or very short stories, some as little as one or two pages long. This characteristic has been made possible precisely by the nature of the science fiction genre. While it would be impossible for other narrative genres (such as sentimental novels, adventure travels, detective investigations, or historical accounts) to tell a coherent story in only a few paragraphs, Science Fiction, dominated by fantasy and with a plausible scientific element, makes it possible to create sensible stories even in very few pages.

The huge number of "very short works" in the science fiction genre has necessitated a more precise cataloging of novels, even considering their length. Such classification allows works to be distinguished on the basis of their narrative impact and size, recognizing the value and surprise that short or very short stories can offer readers.

1.3 Importing Online Vocabularies.

In the process of developing the StoryScape ontology, the choice was made to import other online vocabularies, such as "dbo:book" and "dbo:person." This choice was made to take advantage of the knowledge already available in such vocabularies, allowing us to enrich and expand the information in our ontology.

The "dbo:book" vocabulary provides a basic representation for book-related concepts such as title, author, publication, and literary genre. Using this vocabulary as a reference, additional properties specific to science fiction books can be defined in the StoryScape ontology, such as the scientific field covered, the time period in which the story is set, and the main events.

Similarly, importing the "dbo:person" vocabulary allows enrichment of author information in the StoryScape ontology. In addition to basic information such as the author's name, importing this vocabulary can allow the inclusion of additional details, such as date of birth, most famous works, and other relevant biographical information.

Chapter 2: Ontology and Inference

In the second chapter of the StoryScape project, we focus on ontology and inference, two fundamental aspects of creating a structured and intelligent information system. These elements make it possible to define a consistent ontology model to represent the domain of science fiction books and to use inference to infer new information based on the relationships and properties defined in the ontology.

2.1 Objectives.

The main objectives of StoryScape are as follows:

2.1.1 Define a consistent ontology model.

The first goal of StoryScape is to define a consistent ontology model that represents the domain of science fiction books comprehensively and accurately. The ontology must capture the fundamental concepts, relationships, and properties relevant to describing science fiction books, enabling a structured and unified semantic representation of information.

Although the project is not fully complete, it has been designed with its scalability and extensibility in mind. This means that the ontology and information system can be easily expanded in the future to include additional concepts, relationships, and properties, allowing for greater coverage of the science fiction book domain and management of a larger set of information. This approach provides the flexibility to adapt to the future needs of users and applications using StoryScape.

2.1.2 Support research and exploration of science fiction books.

Another key objective is to provide tools and functionality to support search and exploration of science fiction books within the ontology. This includes the ability to perform complex queries to obtain specific information about books, such as authors, titles, genres, characters, plots, and other relevant attributes.

2.1.3 Fostering interoperability and data integration.

StoryScape aims to promote interoperability and data integration in the context of science fiction books. The ontology is designed to be compatible with other standardized ontologies and vocabularies in the literary and fiction domain. This allows science fiction book information to be linked with other semantic resources available online, facilitating data sharing and enrichment.

2.2 Tools Employed

Several tools for ontology management and inference were employed in the implementation of StoryScape. The main tools used are presented below:

2.2.1 Apache Jena

Apache Jena is a Java framework for creating and managing semantic ontologies. It provides a suite of libraries and tools for loading, processing, storing and querying ontology data. Within StoryScape, Apache Jena has been used for the creation and management of the science fiction book ontology, enabling the manipulation and querying of ontological data in an efficient and scalable manner.

2.2.2 Pellet OpenIlet

Pellet OpenIlet is a Web Ontology Language (OWL) inference engine that implements several reasoning and inference techniques. It has been employed within StoryScape to perform inference on the ontology of science fiction books in order to infer new information or verify the consistency of axioms defined in the ontology. Pellet OpenIlet allows leveraging rule-based reasoning and logical inference to obtain richer and more meaningful results from the ontology.

2.2.3 GSON

GSON is a Java library for serializing and deserializing JSON objects. It has been used within StoryScape for handling data in JSON format. GSON allows Java objects to be converted to JSON strings and vice versa, facilitating interaction with other components of the system that use the JSON format for data exchange.

Here is the description of Chapter 3 with the required sub-chapters:

Chapter 3: Usage Scenarios

In the third chapter of the StoryScape project, we will explore different usage scenarios of the information system based on the science fiction book ontology. These scenarios show how users can interact with the application to perform various activities related to browsing, viewing, and searching for books.

3.1 Full Catalog Consultation.

In the full catalog consultation scenario, users wish to view the entire catalog of science fiction books available in the ontology. This scenario provides a general overview of the books present and allows users to explore the entire domain of science fiction books.

Example query:

This query retrieves all the books in the ontology along with their corresponding titles and authors. The returned results allow users to view a complete list of science fiction books in the catalog.

3.2 Viewing the Details of a Book

In the Viewing Details of a Book scenario, users wish to obtain more detailed information about a specific book. This scenario allows users to explore the details of a book, such as title, author, synopsis, genre, and other relevant information.

Sample query:

This query retrieves the details of a specific book identified by the URI of the book (replace ``<bookURI>`` with the actual URI of the desired book). Results returned include the book title, author, synopsis, and genre, providing users with an in-depth view of information specific to a selected book.

3.3 Searching for Books by Name

In the book search by name scenario, users wish to find a specific book by entering its name or a related keyword. This scenario gives users the ability to search for books by name or a keyword, facilitating the discovery of specific books within the catalog.

Sample query:

This query retrieves books that match a keyword specified in the book title. You can replace "keyword" with the actual keyword for the search. The results returned include books that meet the search criteria, such as the associated title and author.

Chapter 4: Description of Responsibilities.

In the fourth chapter of the StoryScape project, we provide a detailed description of the responsibilities identified during the design phase of the system. Following Object-Oriented Programming methodologies and the Model-View-Controller (MVC) method, we identify and define the key responsibilities of the different components of the system.

4.1 Model Responsibilities.

In the design of StoryScape, the Model represents the central part of the application that manages the science fiction book ontology data. The Model's responsibilities include:

- Load and maintain the science fiction book ontology.
- Manage queries and access to the ontology data.
- Provide methods for adding, modifying, or deleting information in the ontology.
- Manage inference rules to obtain more complete and meaningful results.

4.2 Responsibilities of the View

The View represents the user interface of the StoryScape application, through which users interact with data in the ontology. The View's responsibilities include:

- Displaying the science fiction book ontology in a way that is intuitive and understandable to users.
- Present query results in an appropriate and easily readable format.
- Provide navigation and filtering tools to explore the ontology effectively.
- Allow users to interact with the data, such as adding comments or ratings to books.

4.3 Controller Responsibilities.

The Controller is the connection point between the Model and the View and manages the interactions between them. The Controller's responsibilities include:

- Receiving user input from the View and translating it into actions to be performed on the Model.
- Coordinating query requests from the Model and getting the results to display in the View.
- Handle user events and actions, such as selecting a book or performing a search.
- Maintain the state of the application and ensure consistency between the Model and the View.

Following this division of responsibilities, the design of StoryScape respects the principles of separation of responsibilities and modular organization. This enables better maintenance, extensibility, and scalability of the system, allowing changes to be made or new features to be added more efficiently.

5. Implementation of Responsibilities.

In the fifth chapter of the StoryScape project, we will focus on the implementation of the responsibilities identified in the previous chapter. We will describe the key components of

the system, illustrating how the responsibilities were actually implemented through the use of specific classes, interfaces, and enumerations.

5.1 Construction of the Ontology Model.

In the construction phase of the Ontology Model, several classes and interfaces were implemented to manage the ontology of science fiction books. The main components involved include:

- The `DefaultRDFModelBuilder` class, responsible for building a basic RDF model for the ontology.
- The `InferredRDFModelBuilder` class, which extends `DefaultRDFModelBuilder` and adds support for rule-based inference.
- The `Controller` class, which acts as an intermediary between the Ontology Model and the application view.
- The `RDFModelBuilder` interface, which defines common methods for building the RDF model.

These components work together to build and manage the science fiction book ontology model, providing a consistent and structured representation of the data.

5.2 Maintaining and Executing SPARQL Queries.

To support ontology queries, several classes and interfaces have been implemented for maintaining and executing SPARQL queries. The main components involved include:

- The `QuerySelector` class, responsible for selecting SPARQL queries to execute based on user requests.
- The `OntologySPARQLExec` class, which executes SPARQL queries on the ontology and returns the results.
- The `SelectionQueries` enumeration, which contains a set of predefined queries for querying the ontology.
- The `Query` and `SPARQLExecutor` interfaces, which define methods for creating queries and executing SPARQL queries, respectively.

These components enable complex queries on the science fiction book ontology, obtaining specific information such as authors, titles, genres, characters, plots, and other relevant attributes.

5.3 Parsing of Query Results.

After the queries are executed, the results are processed using specific classes and interfaces for data parsing. The main components involved include:

- The `JSONData` class, which represents the data resulting from queries in JSON format.
- The `JSONParser` class, which parses the JSON data and converts it into a format that can be used by the application.
- The `ParsedData` and `Parser` interfaces, which define methods for representing the processed data and parsing the JSON data, respectively.

These components allow SPARQL query results to be extracted and manipulated, making them accessible for presentation to the user.

5.4 State Management and Graphical Interface Presentation.

To manage the state of the application and present the graphical interface, several classes and interfaces have been implemented. The main components involved include:

- The `AppController` class, which handles user events and actions, coordinating interactions between the Ontology Model and the application view.
- The `BookDetails` and `BookList` classes, which handle the display of a book's details and the list of books, respectively.
- The `Render` interface, which defines methods for presenting data to the user.

These components manage the user interface, display book details, and provide an intuitive navigation experience within the science fiction book ontology.

Chapter 6: Using the application

In the final chapter of this report, we will take a final general overview of the StoryScape application and highlight its strengths and why it should be used. StoryScape was designed with the goal of providing a comprehensive platform for exploring and browsing science fiction books through a well-structured ontology.

One of the main distinguishing features of StoryScape is its ontology of science fiction books, which enables a structured and unified semantic representation of information. Through the use of well-defined concepts, relationships, and properties, users can explore the world of science fiction in an accurate and detailed manner. The ontology not only captures the core concepts of science fiction books, but also allows users to import other online vocabularies such as `dbo:book` and `dbo:person`, further expanding the context and completeness of information.

StoryScape offers a range of features to support search and exploration of science fiction books. Users can perform complex queries to obtain specific information about books,

such as authors, titles, genres, characters, plots, and other relevant attributes. With support for SPARQL queries, the application provides accurate and relevant results, enabling users to obtain a broad overview of the world of science fiction books.

In addition to search and exploration, StoryScape also offers an intuitive and engaging user interface. Users can browse the complete catalog of books, view details of each book, and navigate through related information. The application's well-designed GUI and status management enable smooth navigation and a pleasant user experience.

Another strength of StoryScape is its scalability and extensibility. Although the project is not yet complete, it was developed with a long-term vision, allowing for future expansions and improvements. Through the use of Object Oriented Programming methodologies and the Model View Controller method, the code is well structured and modular, making it easy to add new features and extend the application.

In conclusion, StoryScape represents a powerful tool for exploring and browsing science fiction books. With its well-defined ontology, complex queries, intuitive user interface, and scalable application, users can immerse themselves in the fascinating world of science fiction, discovering new stories, authors, and genres. StoryScape provides an engaging and enriching experience for science fiction literature lovers, allowing them to explore a universe of endless possibilities.

Running the project

```
$ ./gradlew :gui:run
```


Appendix A: Asserted Ontology



Appendix B: Inferred Ontology

