UNIVERSITÉ CÔTE D'AZUR

MSc. Data Science & Artificial Intelligence

Dr. Molka Dhouib

Recipe Management Ontology Analysis Report

Authors:

MD Imtiaz Habib – Hadi Kanaan – Eli Oun

Due: January 22, 2025

Table of Contents

Introduction

This report provides a detailed analysis of two Turtle (TTL) files: Our study examines rbox_ontology.ttl and rbox2.ttl. Our ontology file sets up a recipe management system and shows recipes with their details plus customer information through ingredient references. These files show how to use SPARQL queries to find important insights in the data.

The report is organized into three main sections:

- **Ontology Analysis**: We deeply analyze the elements and connections described in the ontology.
- **Instance Analysis**: An analysis of all information found in the rbox2.ttl file emphasizes recipes, customer details, and ingredient information.
- **SPARQL Queries**: A detailed exploration of the SPARQL queries that find useful patterns in RDF data.

1. Ontology Analysis (`rbox_ontology.ttl`)

The rbox_ontology.ttl file defines the foundational structure of the recipe management system. It introduces classes, properties, and their relationships, which are essential for organizing and querying recipe-related data.

1.1 Classes

The ontology defines three main classes:

- **:Recipe** – Represents a recipe, subclass of schema:Recipe, describing dishes, ingredients, cooking methods, and nutritional information.
- **:Ingredient** – Represents an ingredient used in recipes.
- **:Customer** – Represents customers who order recipes.

These classes form the backbone of the ontology, categorizing recipes, ingredients, and customers.

1.2 Properties

The ontology defines several properties to describe relationships between classes and their attributes. Properties are categorized as follows:

Custom Properties:

- :hasIngredient – Links a recipe to its ingredients.

- :suitableForBudget – Indicates recipe suitability for a specific budget.
- Nutritional content: :proteinContent, :energyContent, :fatContent, :carbohydrateContent.
- :heatLevel – Describes spiciness level.

Schema.org Properties:

- :cookTime – Subproperty of schema:cookTime, specifies cooking duration.
- :cookingMethod, :recipeInstructions, :recipeYield – Provide details about the recipe preparation.

2. Instance Analysis (`rbox2.ttl`)

The rbox2.ttl file provides instances of recipes, customers, and ingredients based on the ontology defined in rbox_ontology.ttl.

2.1 Recipe Instances

Defined recipe instances:

1. **:SpaghettiBolognese** – Spaghetti, tomato sauce, and ground beef; cook time 1 hour; high budget; 50g protein, 800 kcal.
2. **:BritishBeefWellington** – Beef fillet, mushrooms, and pastry; cook time 90 mins; low budget; 64g protein, 947 kcal.
3. **:AuthenticChickenKatsuCurry** – Chicken breast, curry powder, and breadcrumbs; cook time 45 mins; high budget; 65g protein, 1360 kcal.

Each recipe links to ingredients via :hasIngredient and customers via :orderedBy.

2.2 Customer Instances

Defined customer instances:

- :CustomerJohnDoe – Ordered Spaghetti Bolognese.
- :CustomerYvette – Ordered British Beef Wellington.
- :CustomerOscar – Ordered Authentic Chicken Katsu Curry.

2.3 Review Instances

Reviews provided by customers:

- **:Review001** – John Doe: "Amazing Spaghetti Bolognese."
- **:Review002** – Yvette: "Quality ingredients provided for a superb eating experience!"
- **:Review003** – Oscar: "Excellent quality and service consistently."

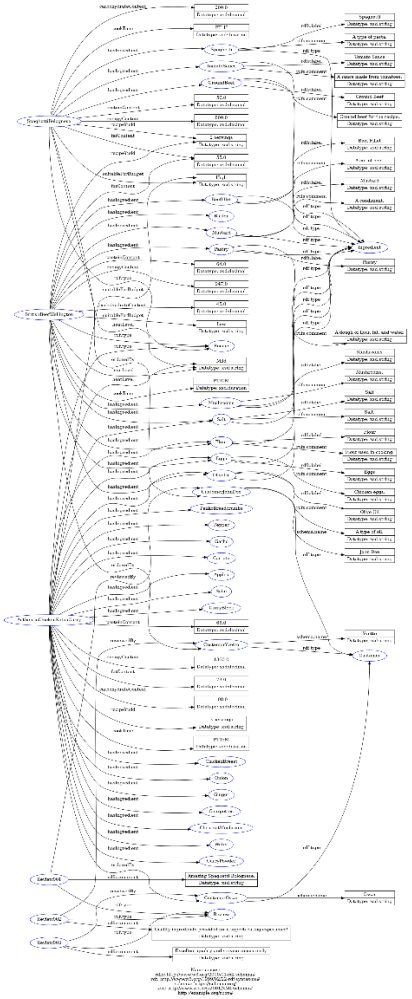
Reviews are linked to customers using :reviewedBy property.

2.4 Ingredient Instances

Defined ingredient instances include:

- :Spaghetti, :TomatoSauce, :GroundBeef, :OliveOil, :BeefFillet, :Mushrooms, :Mustard, :Salt, :Eggs, :Flour, :Pastry.

Each ingredient is described using rdfs:label and rdfs:comment.



3. SPARQL Queries Explanation

SPARQL (SPARQL Protocol and RDF Query Language) is used to extract specific information from RDF datasets using pattern matching.

Query 1: Retrieve All Recipes with Their Ingredients and Suitable Budget

```
SELECT ?recipe ?ingredient ?budget
WHERE {
   ?recipe a :Recipe ;
         :hasIngredient ?ingredient ;
         :suitableForBudget ?budget .
}
ORDER BY ?recipe
```

Purpose: Generates a list of recipes, ingredients, and budget suitability.

Query 2: Find Recipes with Protein Content Greater Than 50 Grams

```
SELECT ?recipe ?protein
WHERE {
   ?recipe a :Recipe ;
         :proteinContent ?protein .
   FILTER (?protein > 50.0)
}
ORDER BY DESC(?protein)
```

Purpose: Identifies high-protein recipes.

Query 3: Count Ingredients Used in Each Recipe

```
SELECT ?recipe (COUNT(?ingredient) AS ?numIngredients)
WHERE {
   ?recipe a :Recipe ;
         :hasIngredient ?ingredient .
}
GROUP BY ?recipe
ORDER BY DESC(?numIngredients)
```

Purpose: Determines ingredient count per recipe.

Query 4: Find Recipes Reviewed by Customers and Their Comments

```
SELECT ?customer ?recipe ?comment
WHERE {
   ?review a :Review ;
         :reviewedBy ?customer ;
         rdfs:comment ?comment .
   ?recipe :orderedBy ?customer .
}
ORDER BY ?customer
```

Purpose: Extracts reviews linked to recipes and customers.

Query 5: Calculate Total Energy Content for Recipes Under a Specific Budget

```
SELECT ?budget (SUM(?energy) AS ?totalEnergy)
WHERE {
   ?recipe a :Recipe ;
         :suitableForBudget ?budget ;
         :energyContent ?energy .
   FILTER (?budget = "Low"^^xsd:string)
}
GROUP BY ?budget
```

Purpose: Calculates total energy for low-budget recipes.

4. Discussion and Conclusion

4.1 Strengths

- Modularity: Organizing the ontology followed established standards to keep its three component types - classes, properties, and individuals - distinct.
- Reusability: Our ontology functions better with existing semantic web standards by using schema.org properties.
- Detailed Descriptions: The ontology supplies complete data about recipes such as their nutrition facts, preparation steps, and listing of all ingredients.

4.2 SPARQL Query Strengths

- Flexibility: Through these queries we see how SPARQL supplies users the ability to easily extract targeted data from their RDF repositories.
- Aggregation and Filtering: Powerful data analysis results from using both COUNT and SUM functions together with the FILTER statement.
- Sorting and Grouping: The ORDER BY and GROUP BY rules help you sort and group information so you can easily understand the results.

4.3 Limitations

- Needs advanced relationships (e.g., dietary restrictions).
- Limited customer data.

Table of Participation

Participant	Role	Contribution
Hadi	Ontology Design	Authored the ontology, defining the classes, properties, and relationships.
Hadi	Instance Backbone	Created the backbone of instances for recipes, customers, and ingredients.
Imtiaz	Instance Analysis	Analyzed the provided instances, focusing on their structure and content.
Imtiaz	Data Gathering	Gathered data to populate the ontology and instances.
Eli	SPARQL Queries	Designed SPARQL queries.
Eli	Report Writing	Prepare the report.