University of Camerino

School of Sciences and Technologies

Master degree in Computer Science (Class LM-18)

Personalized Menu

KEBI - Project

Group members

Samuele Pirani - 131034
Matteo Machella - 131725

Supervisors

Prof. Dr. Knut Hinklemann
Prof. Dr. Emanuele Laurenzi

A.A. 2024/2025

Contents

List of Figures

1. Introduction

The purpose of this document is to describe the entire development process of the project of the course Knowledge Engineering and Business Intelligence. We will discuss about all the tasks to be done, the methodology and technologies used, and a brief project goal description.

In Section 1.1 a brief description of the system to be developed will be reported, while in Section 1.2 all the tasks to be completed for the finalization of the project will be illustrated.

1.1 Project Description

Many restaurants have their menus digitized. Guests can scan a QR code and have the menu presented on their smartphones. A disadvantage is that the screen is very small and it is difficult to get an overview, in particular if the menu is large. However, some guests cannot or do not want every meal, e.g. vegetarians or guests with an allergy. Instead of showing all the meals offered, it would be preferable to show only those meals the guest prefers.

The objective of the project is to represent the knowledge about meals and guest preferences and to create a system that allows for the selection of meals that match the guest preferences.

The knowledge base shall contain information about typical meals of an Italian restaurant, e.g. pizza, pasta, and main dishes. The meals consist of ingredients. There are different types of ingredients, such as meat, vegetables, fruits, or dairy. For each ingredient, there is information on calories.

Guests can be carnivores, vegetarians, calorie conscious, or suffer from allergies, e.g. lactose or gluten intolerance.

1.2 Task List

The following list describes all the tasks to tackle in order to complete the final project:

1. Create different knowledge-based solutions for recommending food depending on the profile of a guest (carnivores, vegetarians, calorie-conscious, suffering from allergies, etc.) using the following representation languages:

- Decision tables (including DRD with sub-decisions and corresponding decision tables);
- Prolog (including facts and rules);
- Knowledge graph/Ontology (including rules in SWRL, queries in SPARQL and SHACL shapes);

2. Agile and ontology-based meta-modeling: adapt BPMN 2.0 to suggest the meals for a given customer. For this, you can re-use or extend the knowledge graph/ontology created in the previous task. One option that you have is to specify the class BPMN Task with a new class and add additional properties, similar to what we have done in class with the Business Process as a Service case. Think of a new graphical notation for the new modeling element, which could be easy to understand for the restaurant manager. Use the triple store interface (Jena Fuseki) to fire the query result.

2. Knowledge Based Solution

In this chapter, the knowledge-based solution realized for the personalized-menu is presented in detail. As described in Section 1.1, the initial interpretation relied on Decision Tables (see Section 2.1) to filter the restaurant's dishes based on guests' preferences.

2.1 Decision Tables

The first phase of the solution is built upon a Knowledge Source, referred to as the *Restaurant Source*. This source feeds the initial decision, called *Menu*, within which the complete dish list of the restaurant is defined. Figure 2.1


```
 { "name": "Tomato Bruschetta",
  "ingredients": [
    { "name": "Bread", "type": "Cereal","calories": 250 },
    { "name": "Tomato", "type": "Vegetable", "calories": 18 },
    { "name": "Garlic", "type": "Vegetable", "calories": 149 },
    { "name": "Basil", "type": "Vegetable", "calories": 22 }
  ],
  "course": "Appetizer",
  "allergens": ["gluten"]
},
{ "name": "Seafood Risotto",
  "ingredients": [
    { "name": "Rice", "type": "Cereal","calories": 130 },
    { "name": "Shrimp", "type": "Fish", "calories": 85 },
    { "name": "Squid", "type": "Fish", "calories": 92 },
    { "name": "Mussels", "type": "Fish", "calories": 86 },
    { "name": "Garlic", "type": "Vegetable", "calories": 149 },
    { "name": "Parlsey", "type": "Vegetable", "calories": 36 }
  ],
  "course": "First",
  "allergens": ["shellfish", "mollusks"]
},
  { "name": "Saltimbocca alla Romana",
  "ingredients": [
    { "name": "Veal slice", "type": "Meat","calories": 110 },
    { "name": "Prosciutto crudo", "type": "Meat", "calories": 150},
    { "name": "Sage", "type": "Vegetable", "calories": 50 },
    { "name": "Butter", "type": "Dairy", "calories": 325 }
  ],
  "course": "Main",
  "allergens": ["lactose"]
},
 { "name": "Tiramisu",
  "ingredients": [
    { "name": "Lady fingers", "type": "Cereal","calories": 200 },
    { "name": "Eggs", "type": "Dairy", "calories": 143},
    { "name": "Mascarpone cheese", "type": "Dairy", "calories": 210 },
    { "name": "Coffee", "type": "Vegetable", "calories": 2 },
    { "name": "Cocoa powder", "type": "Vegetable", "calories": 138 }
  ],
```


Figure 2.1: Dish List defined inside the Menu decision

illustrates this dish list. Once this central decision is established, the system is designed to further refine the menu through specialized decision modules that focus on three main aspects: filtering dishes based on allergenic content, tailoring the selection to match guest dietary preferences, and aligning dishes with caloric requirements.

The module that handles allergenic content takes into account common allergens such as lactose, eggs, peanuts, tree nuts, fish, shellfish, mollusks, soy, and gluten. These allergens have been identified as the most significant due to their frequent occurrence and potential to affect a considerable portion of the clientele. In parallel, the solution considers the guest's dietary profile, taking into account preferences that range from carnivorous diets to vegetarian, vegan, and fish-based diets, thereby capturing a broad spectrum of culinary tastes. In addition, the system evaluates the caloric aspect by distinguishing between various nutritional levels. It categorizes dishes in relation to their energy content by defining ranges such as Light or Diet Friendly (200 to 400 kcal), Moderate or Balanced (200 to 700 kcal), Hearty or Energy Rich (200 to 1000 kcal), and High Calorie (above 1000 kcal). This structured approach ensures that a wide range of input data is available for the decision-making process, which ultimately enhances the precision of the personalized recommendations.

The implementation of these decision processes is illustrated by several figures. Figure 2.2

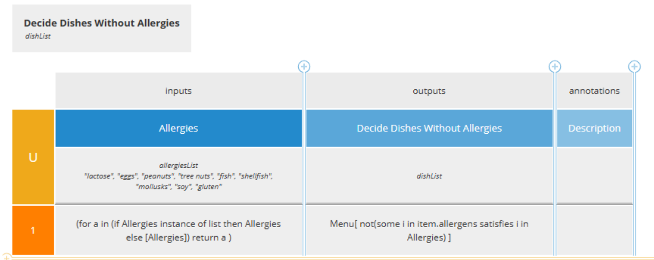

Figure 2.2: Filter Allergies Decision Table

presents the decision table used to filter out dishes containing allergens, ensuring that no item contradicts the specific health or dietary constraints of the guest. Similarly, Figure 2.3

shows the decision table that matches dishes to the guest's declared profile, and Figure 2.4

details the decision table for aligning the dishes with the guest's caloric profile. These sub-decisions operate sequentially and their outcomes are subsequently merged by a main decision module known as *Filter Remaining Menu*. The purpose of this module is to reconcile the individual outputs of the allergenic, dietary profile, and caloric filters so that a coherent set of options is forwarded to the final decision table, named *Suggest Personalized Menu*. The final decision table is responsible for displaying the tailored menu to the restaurant customer. This overall approach is encapsulated in the Decision Model Diagram presented in Figure 2.5.

Decide Dishes Match Guest Profile
dishList

U	inputs	outputs	annotations
	Guest Profile	Decide Dishes Match Guest Profile	Description
	dietProfileType "carnivor", "vegetarian", "vegan", "fish-based"	*dishList* *nullMenu*	
1	"carnivor"	Menu[some i in item.ingredients satisfies i.type = "Meat"]	
2	"vegetarian"	Menu[every i in item.ingredients satisfies not(i.type = "Meat" or i.type = "Fish")]	
3	"vegan"	Menu[every i in item.ingredients satisfies not(i.type = "Meat" or i.type = "Fish" or i.type = "Dairy")]	
4	"fish-based"	Menu[some i in item.ingredients satisfies i.type = "Fish"]	

Figure 2.3: Filter Guest Profile Decision Table

Decide Dishes Match Calories Profile
dishList

U	inputs	outputs	annotations
	Calories Profile	Decide Dishes Match Calories Profile	Description
	caloriesProfileType "Light/Diet Friedly [200 - 400 kcal]", "Moderate/Balanced [200 - 700 kcal]", "Hearty/Energy rich [200 - 1000 kcal]", "High - Calorie [1000+ kcal]"	*dishList* *nullMenu*	
1	"Light/Diet Friedly [200 - 400 kcal]"	Menu[sum(item.ingredients.calories) in [200..400]]	
2	"Moderate/Balanced [200 - 700 kcal]"	Menu[sum(item.ingredients.calories) in [200..700]]	
3	"Hearty/Energy rich [200 - 1000 kcal]"	Menu[sum(item.ingredients.calories) in [200..1000]]	
4	"High - Calorie [1000+ kcal]"	Menu[(sum(item.ingredients.calories) in [200..1000]) or (sum(item.ingredients.calories) > 1000)]	

Figure 2.4: Filter Calories Profile Decision Table

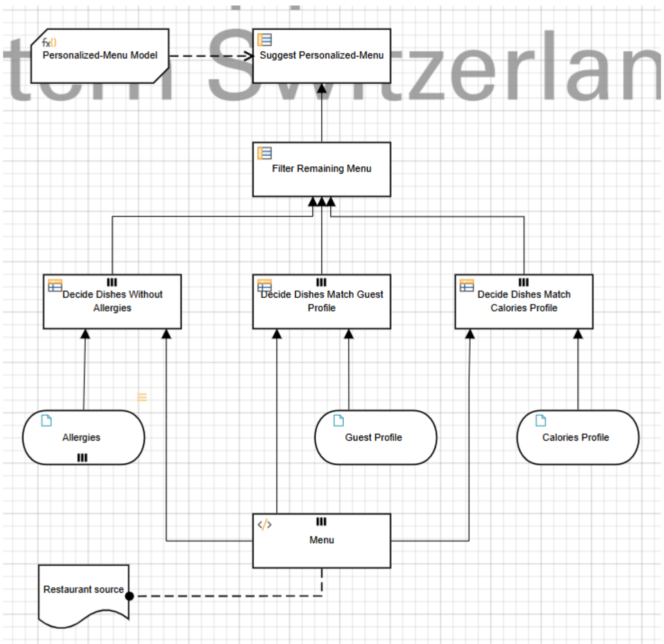

Figure 2.5: Decision Model Diagram

3. Agile and ontology-based meta-modeling

4. Conclusions