First Pratical Work

Jose Vale

a40238@alunos.ipb.pt

Bruno Oliveira

a39230@alunos.ipb.pt

March 20, 2023

Abstract

The objective of this project consists in using the Domain Specific Language ONTODL to specify an ontology describing our knowledge about a domain of our preference. we described the knowledge acquired form the study of this subject. this report contains practicals cases of implementations, concepts, attributes,relations, restrictions and instances about the chosen domain.

1 Introduction

The ontology of a restaurant will be presented, showing a set of concepts within

a domain and the relationships between them and their respective composition.

2 Contextualization

The chosen antology was from an restaurant:

- Concepts:

  - restaurant; -menuitem; -ingredient; -order; -costumer; -employee.

- Attributes:

  -name; -address; -phone; -website; -descripion; -price; -category; -type; -origin; -hour;-date; -price total; -status; -email; -position; -salary for hour;

- Relations:

  -has; -contains; -includes; -is_placed_by; -is_prepared_by; -works_at.

- Individual:
  -restaurant1; restaurant2; restaurante3; MenuItem1; MenuItem2; MenuItem3; Ingredient1; Ingredient2; Ingredient3; Order1; Customer1; Customer2; Customer3; Employee1.

3 Ontology of Restaurant

A brief summary of the graph, we can see that everything starts with the restaurant having the following attributes, such as the name of the restaurant, the contact, the location and also a website where it demonstrates the service provided in it.Restaurant[name:string, address:string, phone:string, website:string].

And like every restaurant, this one has a menu containing items (dishes) with the following attributes (name, type, price and category)

MenuItem[name:string, description:string, price:int, category:string].

The relationship between the restaurant and the menu item is of the “has” type as we can see in the following example.

Restaurant1 = has => MenuItem1;

Each menu item contains ingredients characterized by name, type, origin, and price.
Ingredient[name:string, tpe:string, origin:string, price:int].

With this, we can establish a connection between the menuitem and the ingredient, this relationship will be of the “contains” type, that is, the menu items contain ingredients in their composition.

MenuItem1 = contains => Ingredient1;

In these restaurants there are employees responsible for preparing the menu items (employee), this employee has a name, a position, the hourly wage, and the starting day.

Employee[name:string, position:string, hourly_wage: int, start_date:string]

That is, there is a direct connection between the restaurant and the employee, he works in the restaurant.

Employee1 = works_at => Restaurant1;

When an order is placed, the date and time of the order, the total price, and the status of the order will be recorded. Order[date:string, time:string, total_price:int, status:string];

This request can only be made by the customer, hence the customer's relationship with the request, the same will be made by the customer.

Order1 = is_placed_by => Customer1;

And during the order, the customer is responsible for selecting the ingredients he wants.
Ingredient1 = is_prepared_by => Customer1;

And as the order is placed in the restaurant, there is an important relationship with the customer.

Ex: Customer1 = works_at => Restaurant1;

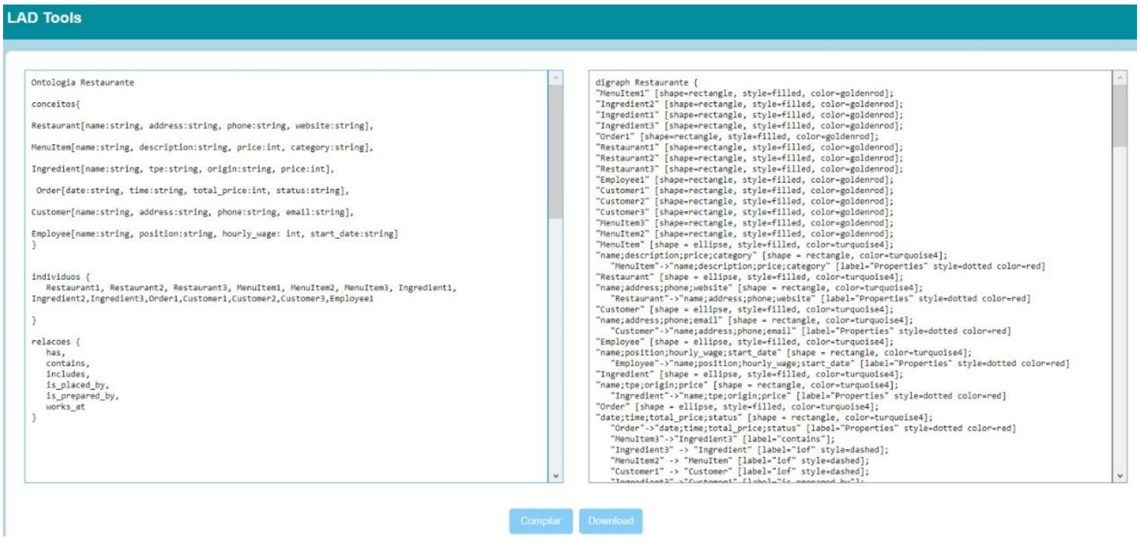
Ontologia Restaurante

```
conceitos{

Restaurant[name:string, address:string, phone:string, website:string],

MenuItem[name:string, description:string, price:int, category:string],

Ingredient[name:string, tpe:string, origin:string, price:int],

 Order[date:string, time:string, total_price:int, status:string],

Customer[name:string, address:string, phone:string, email:string],

Employee[name:string, position:string, hourly_wage: int, start_date:string]

}

 individuos {

         Restaurant1, Restaurant2, Restaurant3, MenuItem1, MenuItem2, MenuItem3,
 Ingredient1, Ingredient2,Ingredient3,Order1,Customer1,Customer2,Customer3,Employee1

 }


 relacoes {

         has,

         contains,

         includes,

         is_placed_by,

         is_prepared_by,

         works_at

 }
```

```
triplos{


Restaurant1 = iof => Restaurant[name='Bistro 123', address='123 Main St',
phone='555-1234', website='bistro123.com'];

Restaurant2 = iof => Restaurant[name='Italian', address= '456 Elm St', phone='555-5678',
website='italiankitchen.com'];

Restaurant3 = iof => Restaurant[name='Sushi House', address='789 Oak St',
phone='555-9012', website='sushihouse.com'];

MenuItem1 = iof => MenuItem[name='Margherita',description='A classic pizza with tomato
sauce and mozzarella cheese',price=' 10.99', category='Pizza'];

MenuItem2= iof => MenuItem[name='Spaghetti carbonara', description='Spaghetti with
bacon',price='12.99', category= 'Pasta'];

MenuItem3 = iof => MenuItem[name='California roll',description='A sushi roll with crab,
avocado, and cucumber',price='8.99', category='Sushi'];

Ingredient1 = iof => Ingredient[name='Tomato', tpe='Fruit', origin='Italy', price='0.50'];

Ingredient2 = iof => Ingredient[name='Mozzarella cheese',tpe='Dairy',origin='Italy',
price='1.00'];

Ingredient3 = iof => Ingredient[name='Bacon', tpe='Meat',origin='USA', price='2.50'];


Order1= iof => Order[date='2022-03-16', time='18:30:00', total_price='45.00',
status='completed'];

Customer1 = iof => Customer[name='John Smith',address='123 Main St', phone='555-1234',
email='john.smith@email.com'];

Customer2 = iof => Customer[name='Estela', address='Rua da Casa', phone='9123412',
email='estela@email.com'];

Customer3 = iof => Customer[name='Lima', address='Main Rua', phone='9123123123',
email='lima@email.com'];

Employee1 = iof => Employee[name='Mary Johnson', position='Chef', hourly_wage='25.00',
start_date='2020-01-01'];
Restaurant1 = has => MenuItem1;

Restaurant2 = has => MenuItem2;

Restaurant3 = has => MenuItem3;

MenuItem1 = contains => Ingredient1;

MenuItem2 = contains => Ingredient2;

MenuItem3 = contains => Ingredient3;

Ingredient1 = is_prepared_by => Customer1;

Ingredient2 = is_prepared_by => Customer1;

Ingredient3 = is_prepared_by => Customer1;

Order1 = is_placed_by => Customer1;

Customer1 = works_at => Restaurant1;

Customer2 = works_at => Restaurant2;

Customer3 = works_at => Restaurant3;

Employee1 = works_at => Restaurant1;

}.
```

4 Results

The dot program:



```
Ontologia Restaurante

conceitos{

Restaurant[name:string, address:string, phone:string, website:string],

MenuItem[name:string, description:string, price:int, category:string],

Ingredient[name:string, tpe:string, origin:string, price:int],

 Order[date:string, time:string, total_price:int, status:string],

Customer[name:string, address:string, phone:string, email:string],

Employee[name:string, position:string, hourly_wage: int, start_date:string]
}


individuos {
   Restaurant1, Restaurant2, Restaurant3, MenuItem1, MenuItem2, MenuItem3, Ingredient1,
Ingredient2,Ingredient3,Order1,Customer1,Customer2,Customer3,Employee1

}

relacoes {
   has,
   contains,
   includes,
   is_placed_by,
   is_prepared_by,
   works_at
}
```

```
digraph Restaurante {
"MenuItem1" [shape=rectangle, style=filled, color=goldenrod];
"Ingredient2" [shape=rectangle, style=filled, color=goldenrod];
"Ingredient1" [shape=rectangle, style=filled, color=goldenrod];
"Ingredient3" [shape=rectangle, style=filled, color=goldenrod];
"Order1" [shape=rectangle, style=filled, color=goldenrod];
"Restaurant1" [shape=rectangle, style=filled, color=goldenrod];
"Restaurant2" [shape=rectangle, style=filled, color=goldenrod];
"Restaurant3" [shape=rectangle, style=filled, color=goldenrod];
"Employee1" [shape=rectangle, style=filled, color=goldenrod];
"Customer1" [shape=rectangle, style=filled, color=goldenrod];
"Customer2" [shape=rectangle, style=filled, color=goldenrod];
"Customer3" [shape=rectangle, style=filled, color=goldenrod];
"MenuItem3" [shape=rectangle, style=filled, color=goldenrod];
"MenuItem2" [shape=rectangle, style=filled, color=goldenrod];
"MenuItem" [shape = ellipse, style=filled, color=turquoise4];
"name;description;price;category" [shape = rectangle, color=turquoise4];
   "MenuItem"->"name;description;price;category" [label="Properties" style=dotted color=red]
"Restaurant" [shape = ellipse, style=filled, color=turquoise4];
"name;address;phone;website" [shape = rectangle, color=turquoise4];
   "Restaurant"->"name;address;phone;website" [label="Properties" style=dotted color=red]
"Customer" [shape = ellipse, style=filled, color=turquoise4];
"name;address;phone;email" [shape = rectangle, color=turquoise4];
   "Customer"->"name;address;phone;email" [label="Properties" style=dotted color=red]
"Employee" [shape = ellipse, style=filled, color=turquoise4];
"name;position;hourly_wage;start_date" [shape = rectangle, color=turquoise4];
   "Employee"->"name;position;hourly_wage;start_date" [label="Properties" style=dotted color=red]
"Ingredient" [shape = ellipse, style=filled, color=turquoise4];
"name;tpe;origin;price" [shape = rectangle, color=turquoise4];
   "Ingredient"->"name;tpe;origin;price" [label="Properties" style=dotted color=red]
"Order" [shape = ellipse, style=filled, color=turquoise4];
"date;time;total_price;status" [shape = rectangle, color=turquoise4];
   "Order"->"date;time;total_price;status" [label="Properties" style=dotted color=red]
   "MenuItem3"->"Ingredient3" [label="contains"];
   "Ingredient3" -> "Ingredient" [label="iof" style=dashed];
   "MenuItem2" -> "MenuItem" [label="iof" style=dashed];
   "Customer1" -> "Customer" [label="iof" style=dashed];
   "Ingredient3"->"Customer1" [label="is_prepared_by"];
```

Figure 1: Convertion Lad tools

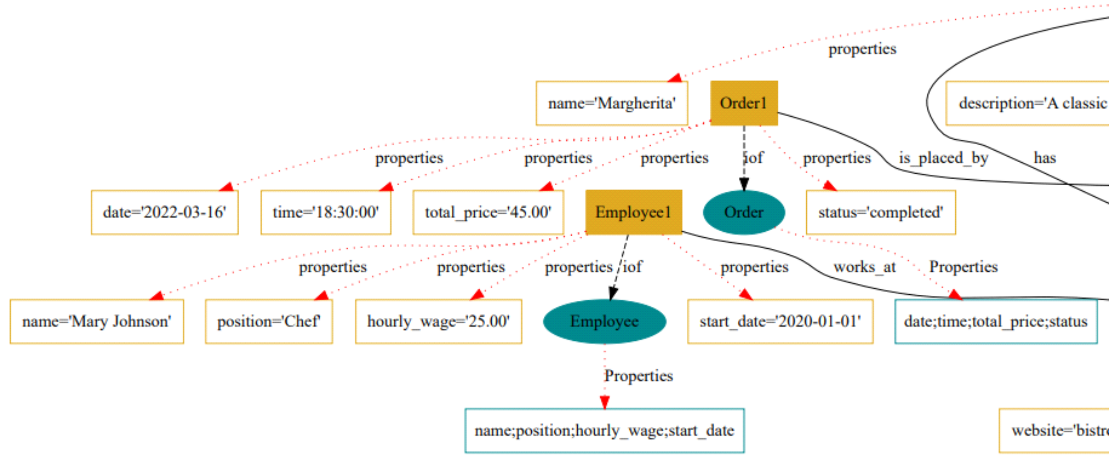


Figure 2:Employee and order

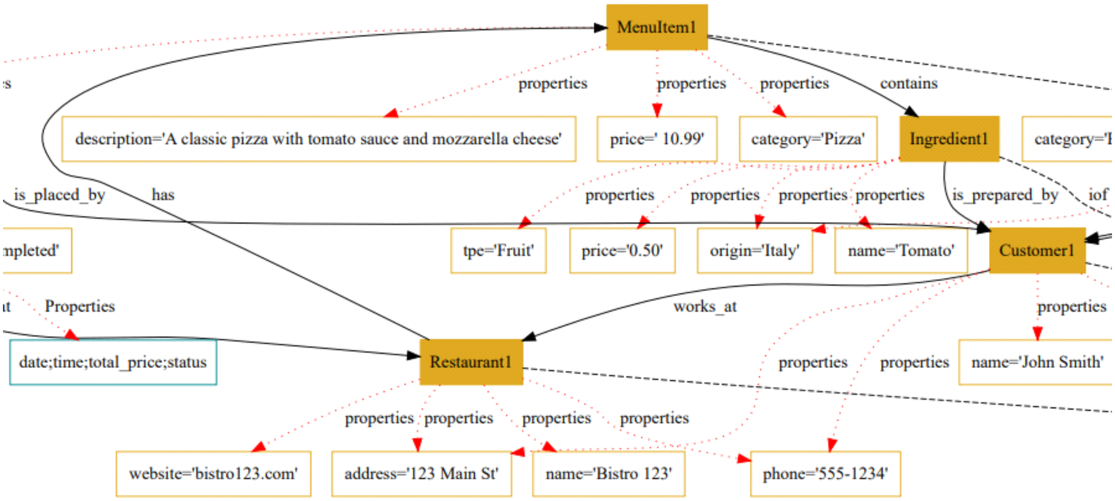


Figure 3 : MenuItem, Ingredient,Restaurant and Customer

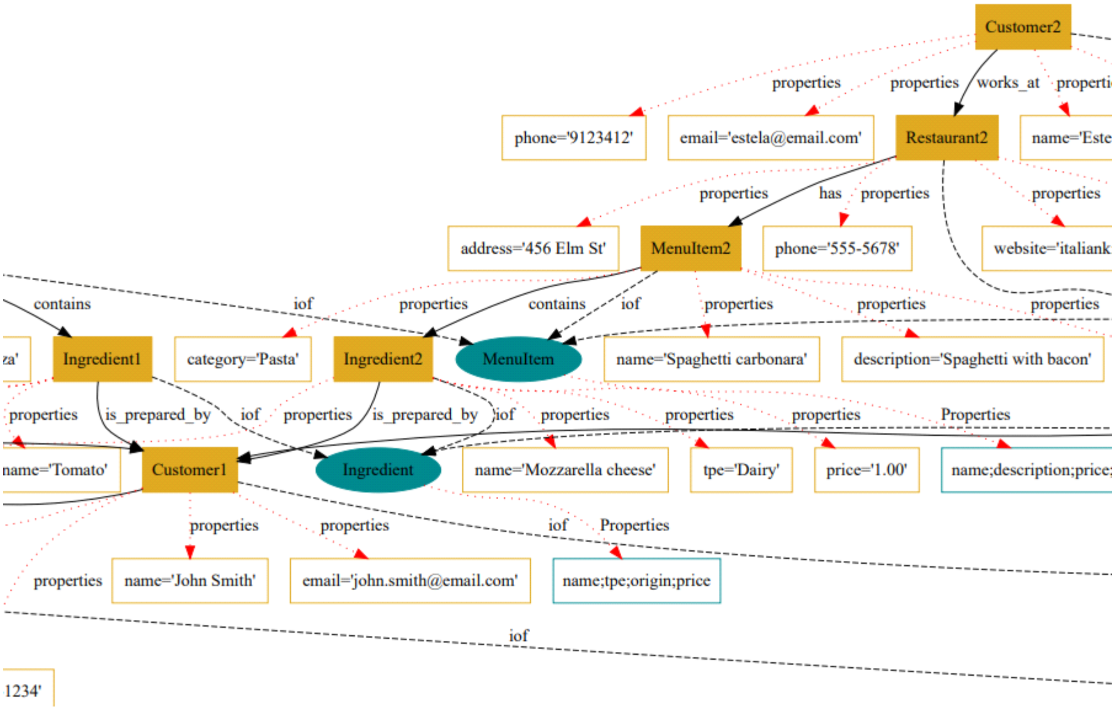


Figure 4 : MenuItem, Ingredient and Restaurant

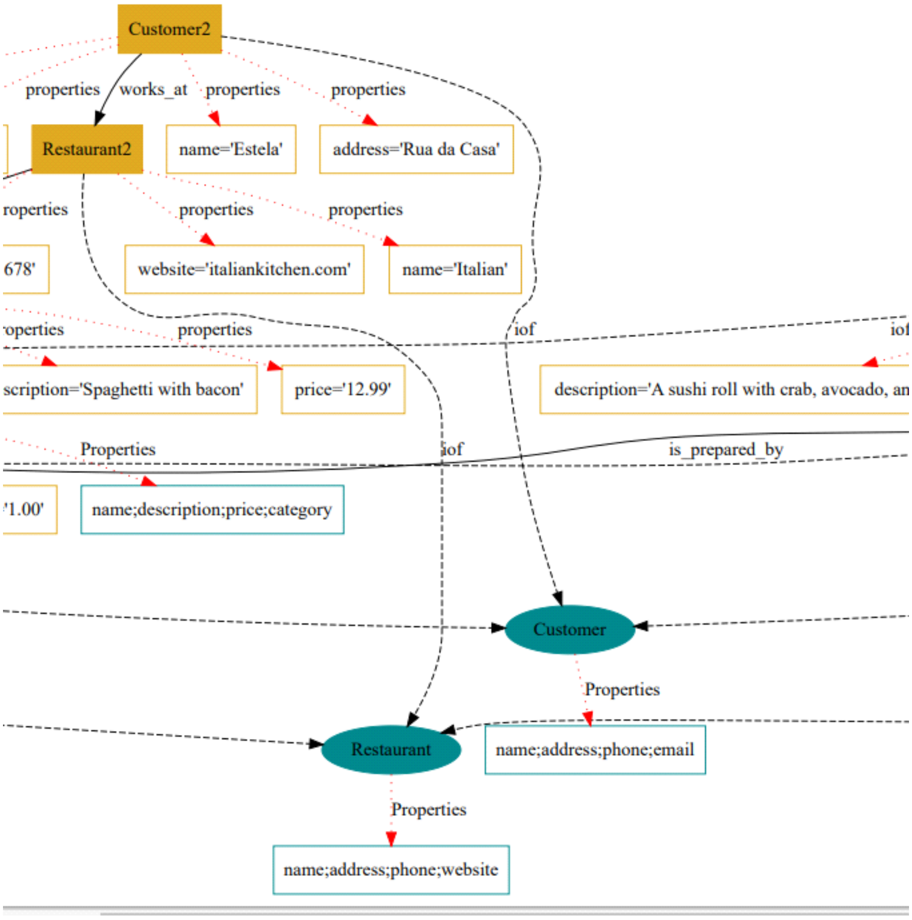


Figure 5 : Customer and Restaurant

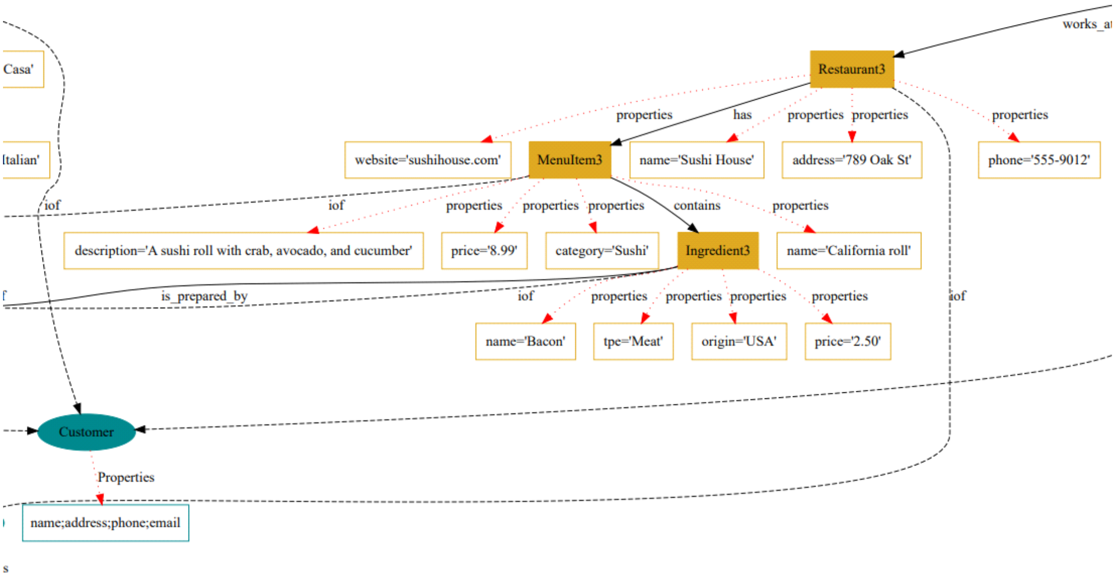


Figure 6: Customer

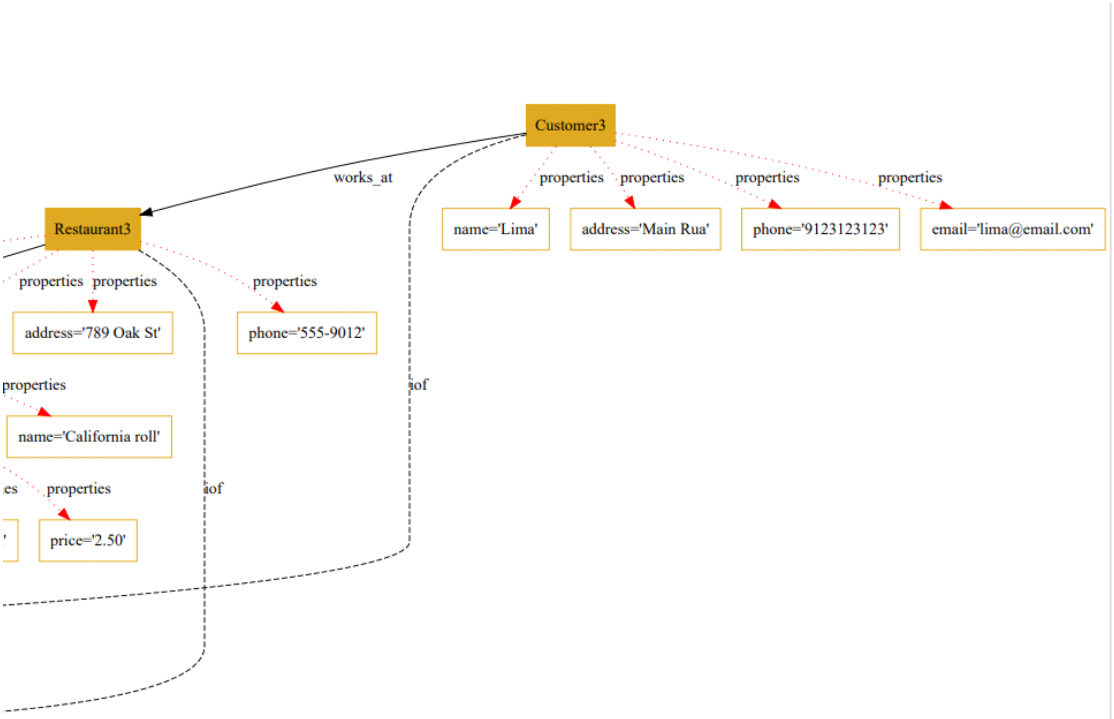


Figure 7 : Restaurant and Customer

5 Conclusion

With this ontology created, it was concluded that ONTODL could express complex concepts and relationships in a precise and structured way, and could easily understand and modify ontologies as the understanding of a domain evolves. Overall, ONTODL is an important tool for ontology development and knowledge management, allowing organizations and individuals to better capture and share knowledge in their fields.

References

[1] https://websites.fraunhofer.de/CTS2-LE/index.php/OntoDL_Tutorial:_Defining_a_Terminology