UNIVERSITÀ DEGLI STUDI DELL'AQUILA

Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica

CORSO DI LAUREA MAGISTRALE IN INFORMATICA (ASE)

Insegnamento Model Driven Engineering

NAME AND SURNAME	**STUDENT NUMBER**
Luca Francesco Macera	302123
Calogero Carlino	302154

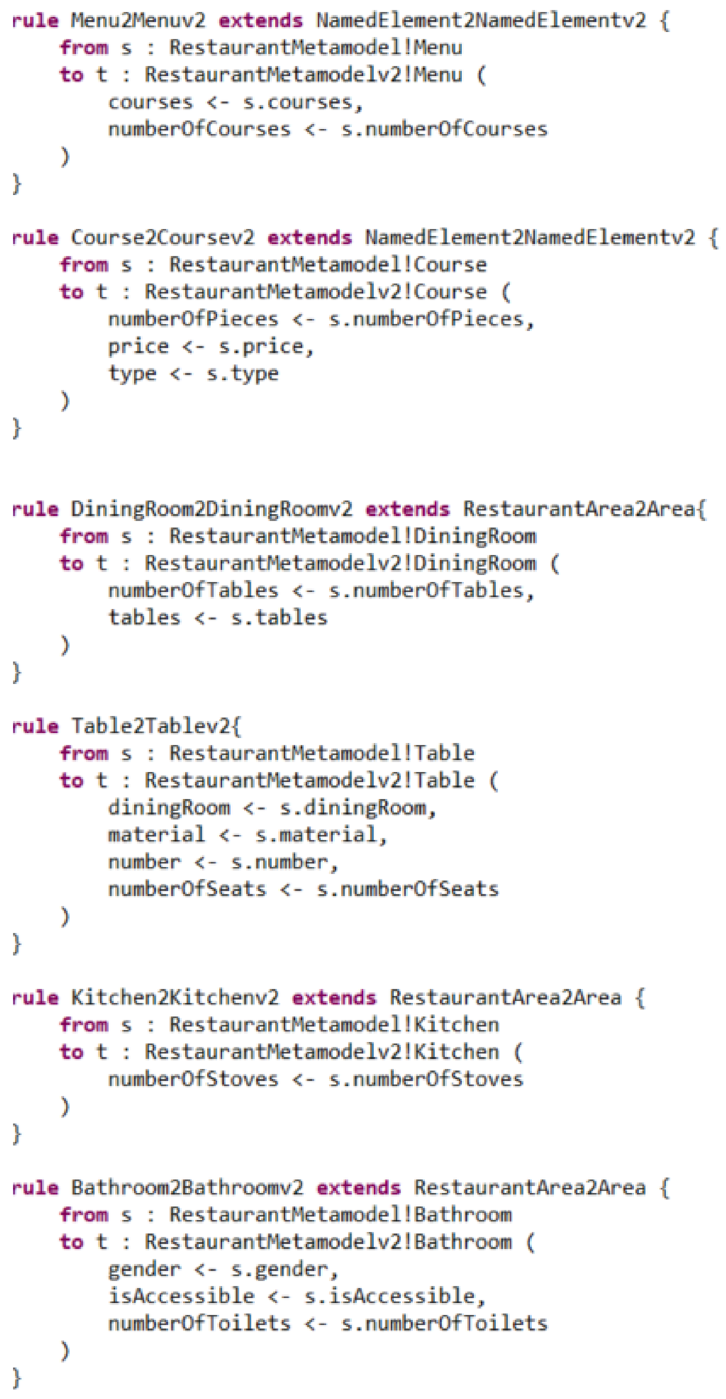
RestaurantMetamodel refactoring

Refactoring operations were performed on the `RestaurantMetamodel` in order to create the `RestaurantMetamodel2`. The changes include:

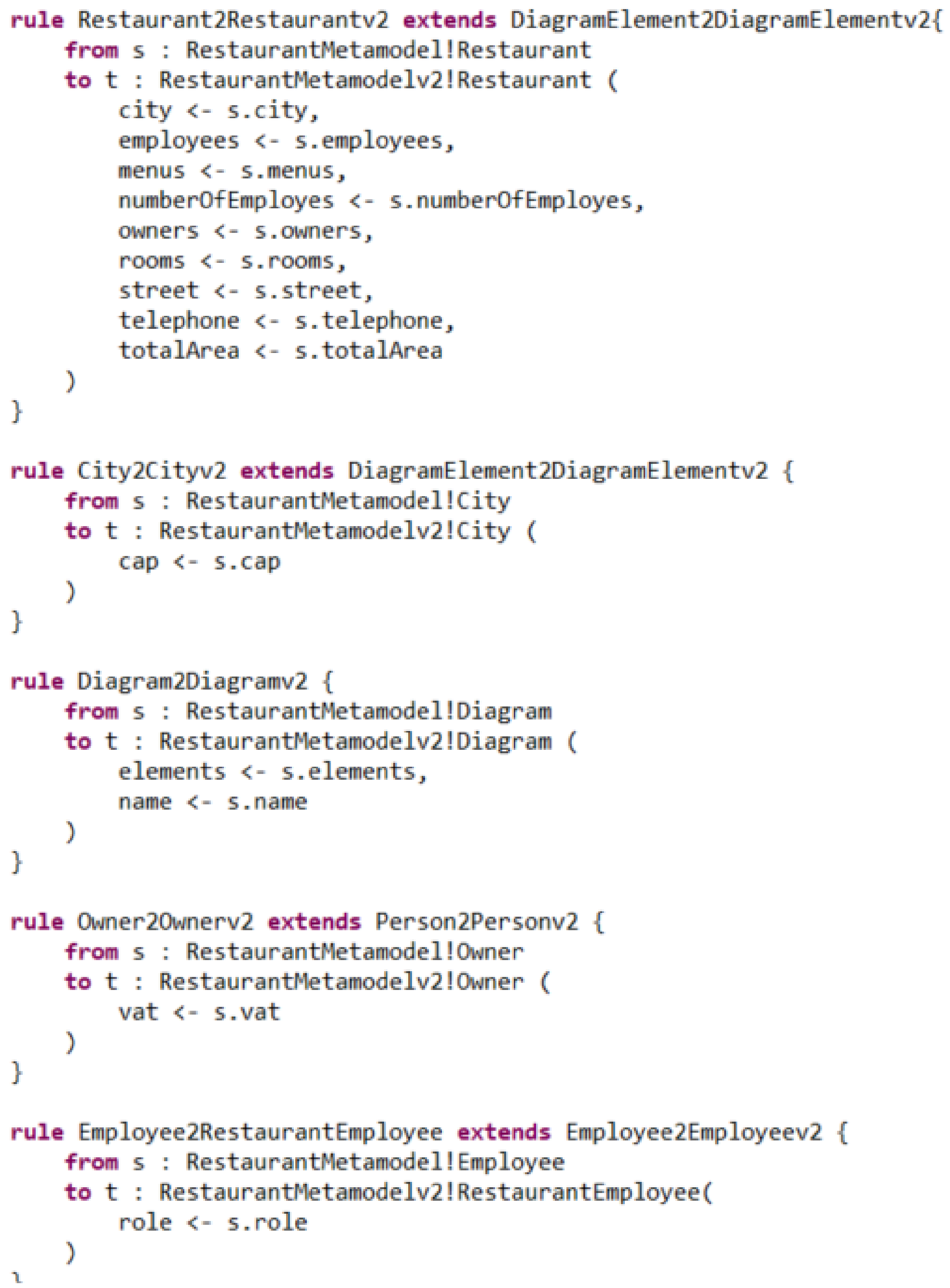
- added `suppliers` attribute to the `Restaurant` metaclass
- removed `region` attribute from the `City` metaclass
- removed the `Region` metaclass
- added `email` attribute to the `Person` metaclass
- added `telephoneNumber` attribute to the `Person` metaclass
- removed `role` attribute from the `Employee` metaclass
- renamed `RestaurantArea` metaclass to `Area`
- created the `Supplier` metaclass, a subclass of `Person`, representing a restaurant supplier belonging to an `industry` (which is a reference)
- created the `Industry` metaclass, which represents an industry consisting of employees working as restaurant suppliers. It includes references to `City`, `Owner`, `Employee` as attributes. The relationships imply that an industry can have multiple owners and employees.
- created the `RestaurantEmployee` metaclass, a subclass of `Employee` representing a restaurant employee with a specific role (e.g., chef, waiter, ecc…). The only new attribute besides those inherited from the `Employee` metaclass is `role`.

M2M Transformation

A Model-to-Model transformation file was created using ATL to migrate models conforming to the `RestaurantMetamodel` metamodel to models conforming to the `RestaurantMetamodel2` metamodel.



```
rule Menu2Menuv2 extends NamedElement2NamedElementv2 {
    from s : RestaurantMetamodel!Menu
    to t : RestaurantMetamodelv2!Menu (
        courses <- s.courses,
        numberOfCourses <- s.numberOfCourses
    )
}

rule Course2Coursev2 extends NamedElement2NamedElementv2 {
    from s : RestaurantMetamodel!Course
    to t : RestaurantMetamodelv2!Course (
        numberOfPieces <- s.numberOfPieces,
        price <- s.price,
        type <- s.type
    )
}


rule DiningRoom2DiningRoomv2 extends RestaurantArea2Area{
    from s : RestaurantMetamodel!DiningRoom
    to t : RestaurantMetamodelv2!DiningRoom (
        numberOfTables <- s.numberOfTables,
        tables <- s.tables
    )
}

rule Table2Tablev2{
    from s : RestaurantMetamodel!Table
    to t : RestaurantMetamodelv2!Table (
        diningRoom <- s.diningRoom,
        material <- s.material,
        number <- s.number,
        numberOfSeats <- s.numberOfSeats
    )
}

rule Kitchen2Kitchenv2 extends RestaurantArea2Area {
    from s : RestaurantMetamodel!Kitchen
    to t : RestaurantMetamodelv2!Kitchen (
        numberOfStoves <- s.numberOfStoves
    )
}

rule Bathroom2Bathroomv2 extends RestaurantArea2Area {
    from s : RestaurantMetamodel!Bathroom
    to t : RestaurantMetamodelv2!Bathroom (
        gender <- s.gender,
        isAccessible <- s.isAccessible,
        numberOfToilets <- s.numberOfToilets
    )
}
```



```
rule Restaurant2Restaurantv2 extends DiagramElement2DiagramElementv2{
    from s : RestaurantMetamodel!Restaurant
    to t : RestaurantMetamodelv2!Restaurant (
        city <- s.city,
        employees <- s.employees,
        menus <- s.menus,
        numberOfEmployes <- s.numberOfEmployes,
        owners <- s.owners,
        rooms <- s.rooms,
        street <- s.street,
        telephone <- s.telephone,
        totalArea <- s.totalArea
    )
}

rule City2Cityv2 extends DiagramElement2DiagramElementv2 {
    from s : RestaurantMetamodel!City
    to t : RestaurantMetamodelv2!City (
        cap <- s.cap
    )
}

rule Diagram2Diagramv2 {
    from s : RestaurantMetamodel!Diagram
    to t : RestaurantMetamodelv2!Diagram (
        elements <- s.elements,
        name <- s.name
    )
}

rule Owner2Ownerv2 extends Person2Personv2 {
    from s : RestaurantMetamodel!Owner
    to t : RestaurantMetamodelv2!Owner (
        vat <- s.vat
    )
}

rule Employee2RestaurantEmployee extends Employee2Employeev2 {
    from s : RestaurantMetamodel!Employee
    to t : RestaurantMetamodelv2!RestaurantEmployee(
        role <- s.role
    )
}
```

We have also made abstract rules to assist in defining subclasses of metaclasses, such as `Person` and `RestaurantArea`

```
abstract rule NamedElement2NamedElementv2 {
    from s : RestaurantMetamodel!NamedElement
    to t : RestaurantMetamodelv2!NamedElement(
        name <- s.name
    )
}

abstract rule DiagramElement2DiagramElementv2 extends NamedElement2NamedElementv2 {
    from s : RestaurantMetamodel!DiagramElement
    to t : RestaurantMetamodelv2!DiagramElement
}


abstract rule Person2Personv2 extends DiagramElement2DiagramElementv2 {
    from s : RestaurantMetamodel!Person
    to t : RestaurantMetamodelv2!Person(
        birthDate <- s.birthDate,
        birthPlace <- s.birthPlace,
        fiscalCode <- s.fiscalCode,
        gender <- s.gender,
        surname <- s.surname,
        email <- s.createEmail(),
        telephoneNumber <- OclUndefined
    )
}

abstract rule Employee2Employeev2 extends Person2Personv2 {
    from s : RestaurantMetamodel!Employee
    to t : RestaurantMetamodelv2!Employee(
        contractExpirationDate <- s.contractExpirationDate,
        contractSignDate <- s.contractSignDate,
        salary <- s.salary
    )
}

abstract rule RestaurantArea2Area extends NamedElement2NamedElementv2 {
  from s : RestaurantMetamodel!RestaurantArea
  to t : RestaurantMetamodelv2!Area (
    area <- s.area,
    perimeter <- s.perimeter
  )
}
```

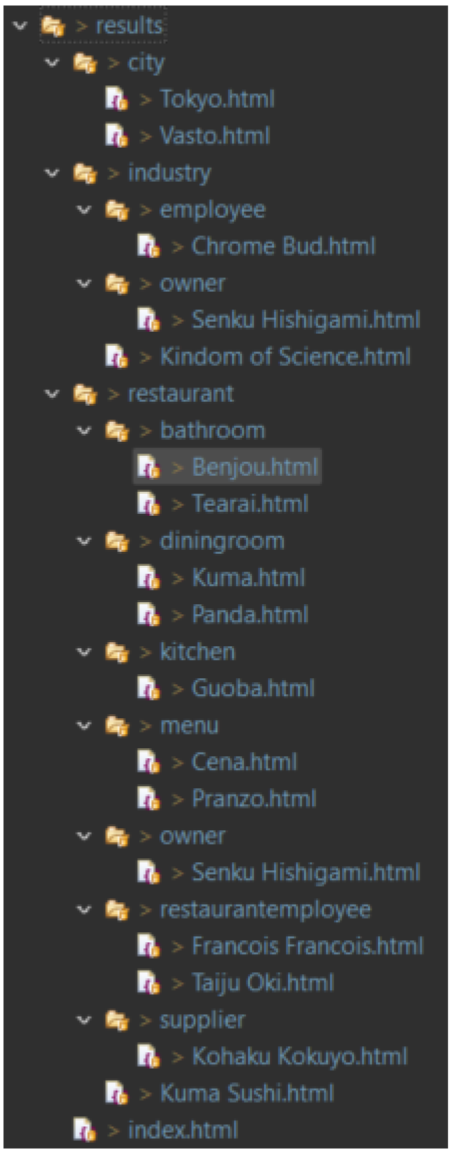
and a helper to compute a standard email for the `Person` metaclass.

```
helper context RestaurantMetamodel!Person def : createEmail() : String =
    (self.name+'.'+self.surname+'@gmail.com').toLower();
```

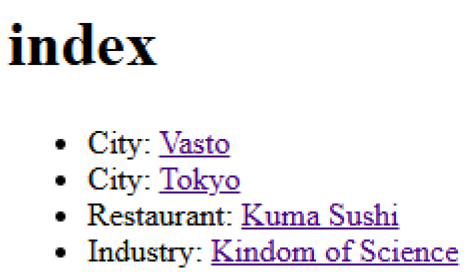
M2T Transformation

The Model-to-Text transformation was created using Acceleo to generate informative HTML pages for the Restaurant domain. All the HTML pages take their name from the respective model element and are organized with respect to the structure of the model.

The transformation produces the necessary HTML pages containing all the key information about the Model with cross links between pages and a quick way to navigate them via breadcrumbs.

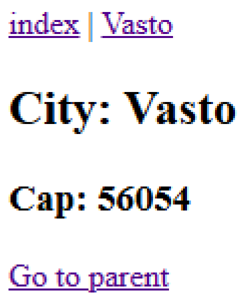


```
> results
  > city
    > Tokyo.html
    > Vasto.html
  > industry
    > employee
      > Chrome Bud.html
    > owner
      > Senku Hishigami.html
    > Kindom of Science.html
  > restaurant
    > bathroom
      > Benjou.html
      > Tearai.html
    > diningroom
      > Kuma.html
      > Panda.html
    > kitchen
      > Guoba.html
    > menu
      > Cena.html
      > Pranzo.html
    > owner
      > Senku Hishigami.html
    > restaurantemployee
      > Francois Francois.html
      > Taiju Oki.html
    > supplier
      > Kohaku Kokuyo.html
    > Kuma Sushi.html
  > index.html
```



index

- City: Vasto
- City: Tokyo
- Restaurant: Kuma Sushi
- Industry: Kindom of Science

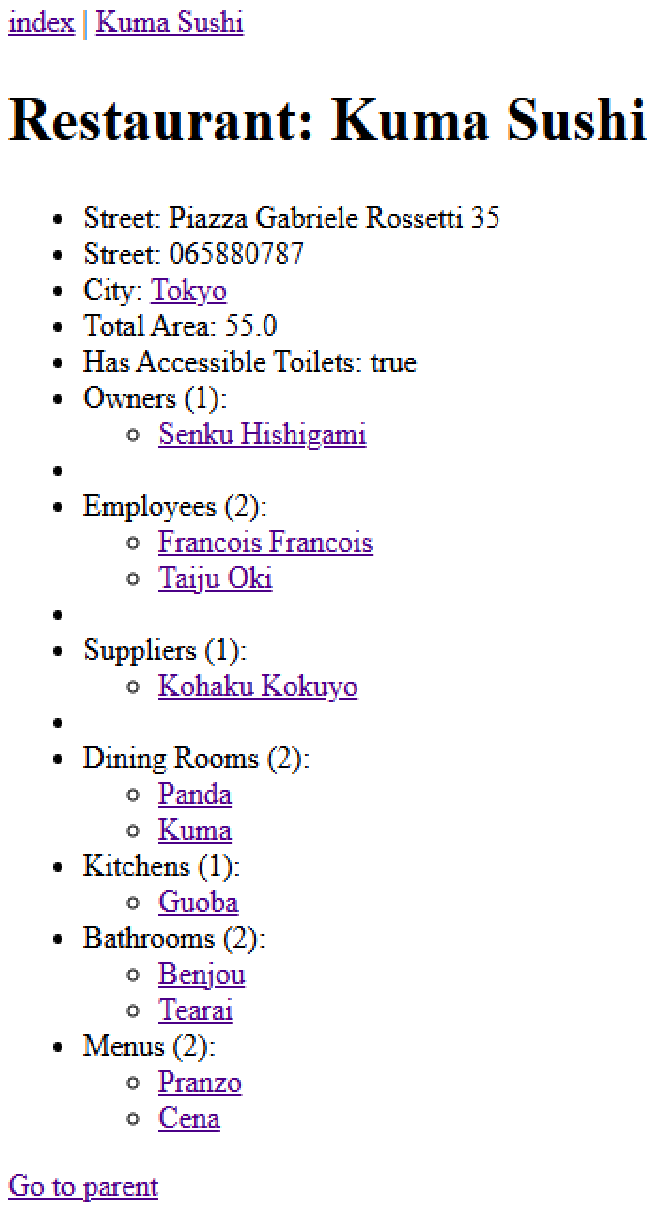


index | Vasto

City: Vasto

Cap: 56054

Go to parent

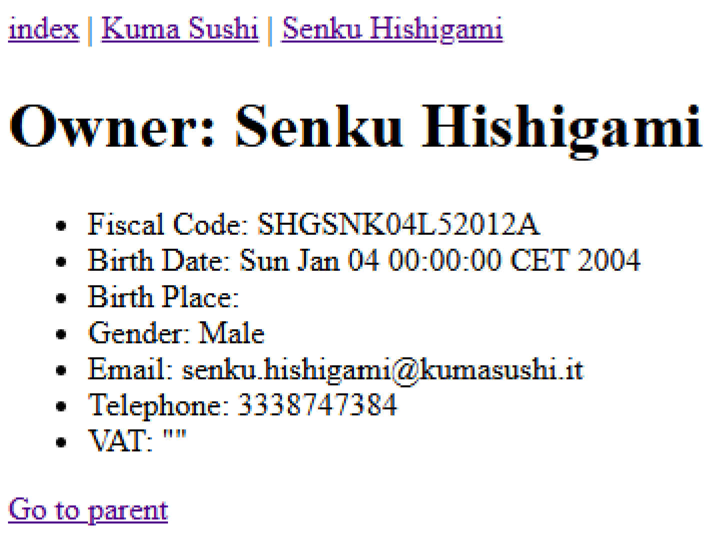


index | Kuma Sushi

Restaurant: Kuma Sushi

- Street: Piazza Gabriele Rossetti 35
- Street: 065880787
- City: Tokyo
- Total Area: 55.0
- Has Accessible Toilets: true
- Owners (1):
  - Senku Hishigami
-
- Employees (2):
  - Francois Francois
  - Taiju Oki
-
- Suppliers (1):
  - Kohaku Kokuyo
-
- Dining Rooms (2):
  - Panda
  - Kuma
- Kitchens (1):
  - Guoba
- Bathrooms (2):
  - Benjou
  - Tearai
- Menus (2):
  - Pranzo
  - Cena

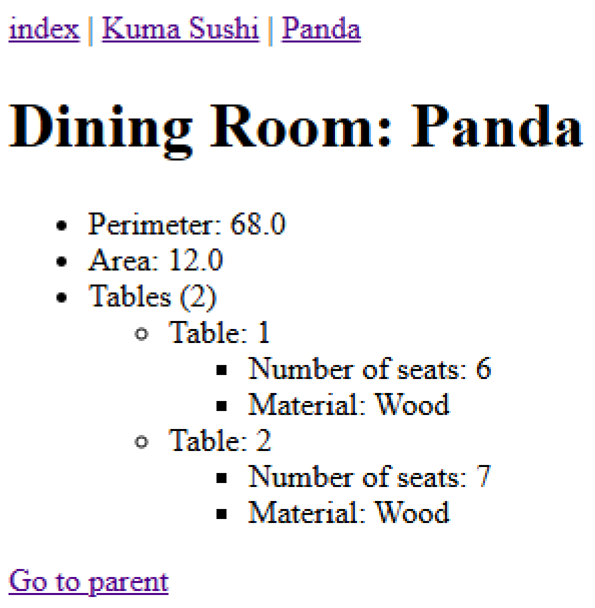
Go to parent


index | Kuma Sushi | Senku Hishigami

Owner: Senku Hishigami

- Fiscal Code: SHGSNK04L52012A
- Birth Date: Sun Jan 04 00:00:00 CET 2004
- Birth Place:
- Gender: Male
- Email: senku.hishigami@kumasushi.it
- Telephone: 3338747384
- VAT: ""

Go to parent

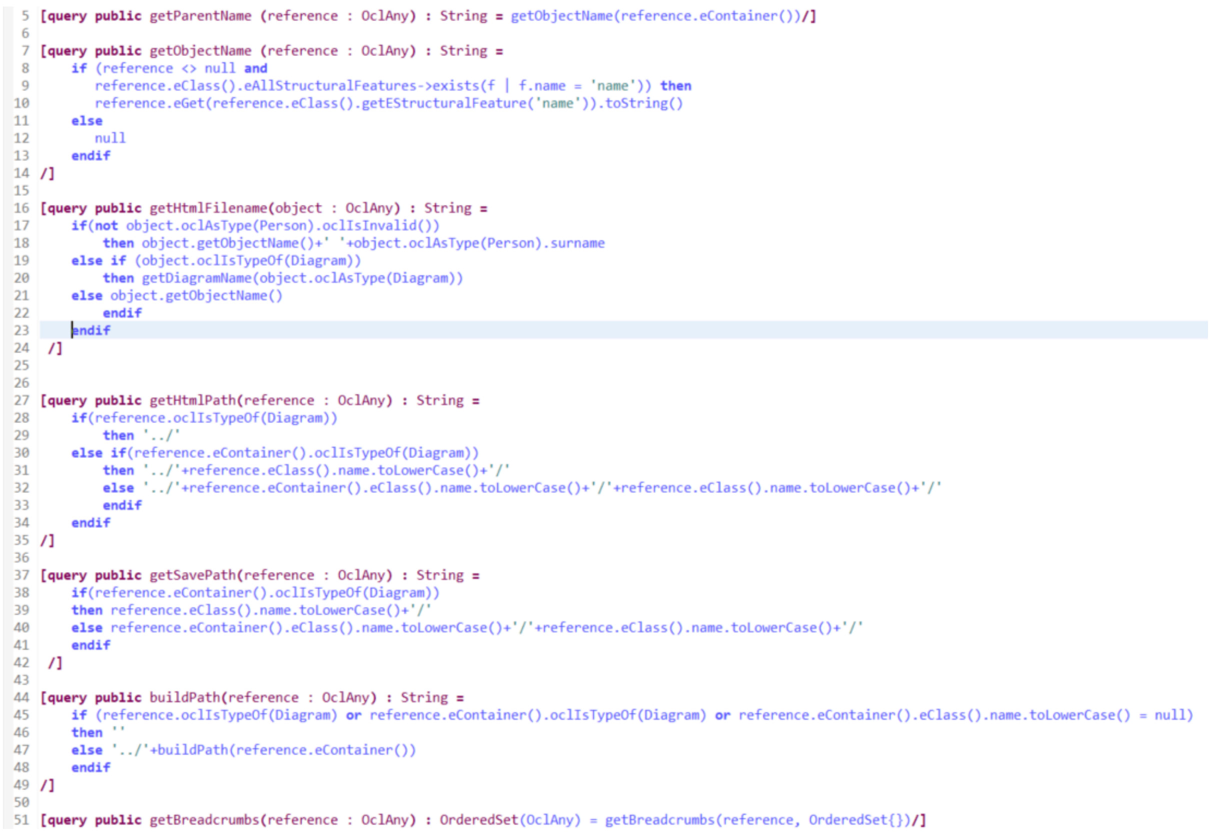

index | Kuma Sushi | Panda

Dining Room: Panda

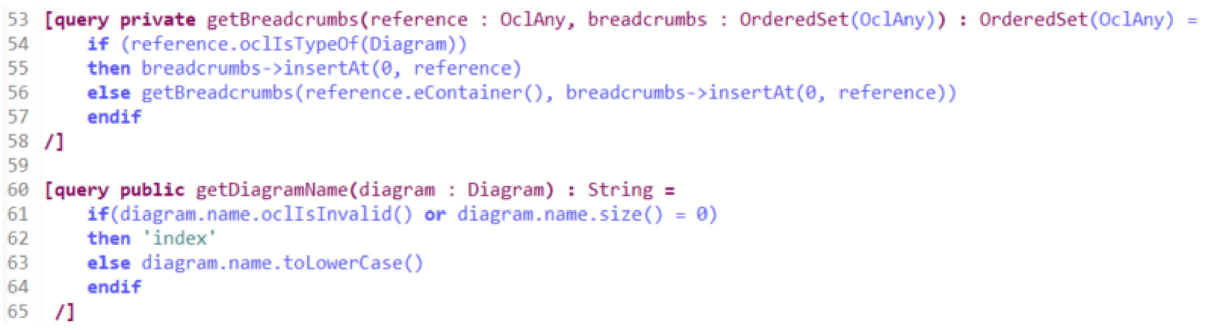
- Perimeter: 68.0
- Area: 12.0
- Tables (2)
  - Table: 1
    - Number of seats: 6
    - Material: Wood
  - Table: 2
    - Number of seats: 7
    - Material: Wood

Go to parent

The logic behind the creation of these pages uses some Acceleo queries as shown in the images below.



```
[query public getParentName (reference : OclAny) : String = getObjectName(reference.eContainer())/]

[query public getObjectName (reference : OclAny) : String =
    if (reference <> null and
       reference.eClass().eAllStructuralFeatures->exists(f | f.name = 'name')) then
       reference.eGet(reference.eClass().getEStructuralFeature('name')).toString()
    else
       null
    endif
/]

[query public getHtmlFilename(object : OclAny) : String =
    if(not object.oclAsType(Person).oclIsInvalid())
        then object.getObjectName()+' '+object.oclAsType(Person).surname
    else if (object.oclIsTypeOf(Diagram))
        then getDiagramName(object.oclAsType(Diagram))
    else object.getObjectName()
        endif
    endif
 /]


[query public getHtmlPath(reference : OclAny) : String =
    if(reference.oclIsTypeOf(Diagram))
        then '../'
    else if(reference.eContainer().oclIsTypeOf(Diagram))
        then '../'+reference.eClass().name.toLowerCase()+'/'
        else '../'+reference.eContainer().eClass().name.toLowerCase()+'/'+reference.eClass().name.toLowerCase()+'/'
        endif
    endif
/]

[query public getSavePath(reference : OclAny) : String =
    if(reference.eContainer().oclIsTypeOf(Diagram))
    then reference.eClass().name.toLowerCase()+'/'
    else reference.eContainer().eClass().name.toLowerCase()+'/'+reference.eClass().name.toLowerCase()+'/'
    endif
 /]

[query public buildPath(reference : OclAny) : String =
    if (reference.oclIsTypeOf(Diagram) or reference.eContainer().oclIsTypeOf(Diagram) or reference.eContainer().eClass().name.toLowerCase() = null)
    then ''
    else '../'+buildPath(reference.eContainer())
    endif
/]

[query public getBreadcrumbs(reference : OclAny) : OrderedSet(OclAny) = getBreadcrumbs(reference, OrderedSet{})/]
```



```
[query private getBreadcrumbs(reference : OclAny, breadcrumbs : OrderedSet(OclAny)) : OrderedSet(OclAny) =
    if (reference.oclIsTypeOf(Diagram))
    then breadcrumbs->insertAt(0, reference)
    else getBreadcrumbs(reference.eContainer(), breadcrumbs->insertAt(0, reference))
    endif
/]

[query public getDiagramName(diagram : Diagram) : String =
    if(diagram.name.oclIsInvalid() or diagram.name.size() = 0)
    then 'index'
    else diagram.name.toLowerCase()
    endif
 /]
```