

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire  
Abdelhafid Boussouf Mila

Institut des Sciences et Technologie

Département de Mathématiques et Informatique

## Mémoire préparé en vue de l'obtention du diplôme de Master

En : Informatique

Spécialité: Sciences et Technologies de l'Information et de la  
Communication (STIC)

**Création d'un site web pour un restaurant de futur avec des  
recommandations alimentaires.**

**Préparé par : Louafi Djad**

Mouaissi Mohammed

**Devant le jury**

Dr.Aouag Mouna.

MCB

C.U.Abd Elhafid Boussouf

Président

Dr.Bouzahzah Mounira .

MCB

C.U.Abd Elhafid Boussouf

Rapporteur

Mme.Abderazak Samira.

MAB

C.U.Abd Elhafid Boussouf

Examineur

**Année Universitaire : 2016/2017**

# **REMERCIEMENTS**

*En tout premier lieu, nous remercions DIEU, le Tout Puissant, qui nous a éclairé le bon chemin et nous a aidé à réaliser ce modeste travail dans les meilleures conditions.*

*Nous tenons à remercier notre encadreur, Mme M. Bouzehzeh, pour nous avoir donné l'opportunité de travailler sur ce projet, pour son grand soutien scientifique et moral, pour les conseils, les suggestions et les encouragements qu'elle nous a apportés tout au long de la réalisation de ce mémoire.*

*Nous adressons également nos remerciements à tous nos enseignants, qui nous ont donné les bases de la science.*

*Enfin nous remercions toutes personnes qui ont contribué de près ou de loin à l'achèvement de ce travail.*

***Djad  
Mohamed***

## DEDICACES

*A mes très chers parents qui m'ont beaucoup*

*Aidé, Soutenu, et encouragé ;*

*A mes très chères sœurs*

*A ma petite belle-sœur Fatima*

*A mon frère Dia ;*

*A tous mes amis ;*

*A toute la famille Louafi ;*

*A mon binôme et ami Mohamed avec qui j'ai Partagé ce  
modeste travail ;*

*A tous ceux que j'aime tant et que je n'ai Pas cités.*

*Je dédie ce mémoire ...*

*Djad*

## *Dédicaces*

*Mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, Mon dieu que je te donne tout le louange et que je te demande de me rapprochez chez vous toujours.*

*A mes parents et ma grands mère AZOUZOU : Grâce à leurs tendres encouragements et leurs grands sacrifices, ils ont pu créer le climat affectueux et propice à la poursuite de mes études. Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers eux. Je prie le bon Dieu de les bénir, de veiller sur eux, en espérant qu'ils seront toujours fiers de moi.*

*Mes frères Amar Isslam Ishak et ma chère sœur, ma tante Saïda et Hayat.*

*A mon collègue d'étude Louafi Djad qui je remercie beaucoup de m'encourager et de m'aider de réalisé ce mémoire. Et mon collègue Fedali Imad.*

*A mes amies Reda et Seïf Kssira que mon dieu de les gardent. Et à toute ma famille et à tous ceux que j'aime.*

Mohammed

# Sommaire

---

Liste des figures.....	i
Liste des tableaux.....	iv

## **Introduction générale**

1. Motivations.....	1
2. Objectif du travail et problématique .....	2
3. Organisation de mémoire.....	2

## **Chapitre I : Les Sites Web**

I .1. Introduction .....	5
I .2. Le world wide Web .....	6
I .3. Définition d'un site Web .....	6
I .4. Application sur les sites web .....	6
I .4.1 Sites web personnels .....	6
I .4.2 Sites web d'entreprise .....	7
I .4.3 Sites web de gouvernementaux .....	7
I .5. Les Sites Web statiques.....	8
I .6. Les Sites Web dynamiques.....	9
I .7. Différence entre un site web statique et un site web dynamique.....	10
a) Site statique.....	10
b) Site dynamique .....	11
I .8. Le Web Sémantique.....	11
II .8.1 Outils sémantiques.....	12
I .9. Conclusion .....	14

## **Chapitre II : Les ontologies**

II .1. Introduction .....	16
II .2. Définition.....	17
II .3. Les composants d'une ontologie.....	17
II .3.1. Les classes/ les concepts.....	17
II.3.2. Les relations et les fonctions.....	18
II .3.3. Les axiomes.....	18
II .3.4. Les instances.....	19
II .4. Le cycle de vie des ontologies.....	19

## Sommaire

---

II .4.1. Besoins et évaluation.....	19
II .4.2. Conception et évolution.....	20
II .4.3. Diffusion.....	20
II .4.4. Utilisation.....	20
II .4.5. Gestion.....	21
II .5. Classification des ontologies.....	21
II .5.1 Typologie selon la richesse de la structure interne des ontologies.....	21
a) Vocabulaire contrôlé.....	22
b) Glossaire .....	22
c) Thésaurus.....	22
d) Hiérarchie informelle (is-a).....	22
e) Hiérarchie formelle (is-a).....	22
f) Hiérarchie formelle (is-a) avec instances du domaine.....	22
g) Frame.....	22
h) Ontologies avec restrictions de valeur.....	23
i) Ontologies avec contraintes logiques.....	23
II .5. 2 Typologie selon l'objet de conceptualisation.....	23
a) Ontologies de représentation de connaissances .....	24
b) Ontologies générale ou commune ou générique.....	24
c) Ontologies de niveau supérieur ou de haut niveau.....	24
d) Ontologies du domaine.....	24
e) Ontologies de tâches.....	24
f) Ontologies d'application.....	24
II .5. 3 Typologie selon le niveau de granularité.....	25
a) Granularité fine .....	25
b) Granularité large.....	25
II .5. 4 Typologie selon le niveau de formalisation utilisé.....	25
a) Hautement informelle.....	25
b) Semi-informelle.....	25
c) Semi-formelle.....	25
d) Rigoureusement formelle.....	25

# Sommaire

---

II.6 Conclusion.....	26
<b>Chapitre III : La conception du site web é « Valencia Restaurant »</b>	
III .1. Introduction .....	28
<b>Partie 01 : L’outil informatique</b> .....	29
III .2. Qu’est-ce que UML.....	29
III .2.1 UML le langage de modélisation Unifié .....	29
a) UML est un langage graphique .....	29
b) UML est un support de communication.....	29
c) UML n’est pas une méthode .....	29
d) UML est basé sur un méta-modèle .....	29
e) UML est une norme .....	29
III .2.2 UML est une Unification des méthodes... ..	29
III .3. Diagramme d’uml .....	30
III .3.1 Diagramme de cas d’utilisation .....	31
III .3.1.1. Elément de base de cas d’utilisation .....	31
a) Un acteur .....	31
b) Cas d’utilisation .....	31
c) L’interaction .....	31
III .3.2 Diagramme de séquence.....	31
III .3.1.1 Définitions messages .....	31
a) Message Synchrone.....	32
b) Message asynchrone .....	32
III .3.3 Diagramme d’activité .....	32
III .3.3.1 Eléments de base du diagramme d’activité .....	33
III.4. Le processus de développement .....	34
III.4.1. Processus Unifié .....	34
III.4.2. Processus 2TUP .....	34
III.4.2.1. La branche gauche (fonctionnelle).....	35
III.4.2.2. La branche droite (architecture technique) .....	36

# Sommaire

---

III.4.2.3. La branche du milieu .....	36
III.4.3 Conclusion.....	37
<b>Partie 02 : L'étude de cas.....</b>	<b>37</b>
III.5 L'étude préliminaire .....	37
III .5.1 Le cahier de charge .....	37
a) Présentation du projet.....	37
b) Les grands choix techniques.....	37
c) Recueil des besoins fonctionnels.....	38
III .5.2 Description du contexte .....	38
a) Identification des acteurs.....	38
b) Identification des messages.....	39
III .5.3 Diagramme de contexte dynamique .....	40
III.6 L'étude détaillée .....	41
III.6.1 Diagramme cas d'utilisation .....	41
III.6.2 Diagramme d'activité .....	44
III.6.2 Diagramme de séquence .....	51
III.6 Conclusion.....	59
 <b>Chapitre IV : Développement d'ontologie « Onto-Resto »</b>	
IV .1. Introduction .....	61
IV .2.Le processus suivi pour la construction de l'Onto-Resto.....	62
II .2.1. Description des principes d'ingénierie ontologique et leur application.....	62
IV .3. Les phases du processus de construction de l'ontologie.....	63
IV .3.1 Spécification.....	64
IV .3.2. Conceptualisation.....	64
IV .3.3. Formalisation.....	65
a) Les constructeurs de la logique SHOIN(D) .....	66
b) La partie terminologique (T-BOX) .....	67
c) La partie Assertionnelle (A-BOX) .....	68



# Sommaire

---

IV .3.4. Opérationnalisation.....	68
IV .3.5. Évaluation.....	69
IV .3.5.1 Erreurs d'inconsistance .....	69
a) Erreurs de circularité.....	69
b) Erreurs sémantiques.....	70
c) Erreurs de partition.....	70
IV .3.5.2 Erreurs d'incomplétude.....	70
a) Classification de concepts incomplète .....	70
b) Erreurs de partition.....	71
c) Erreurs de redondance.....	71
IV.4 Construction d'une ontologie pour un restaurant de futur .....	72
IV.4.1 Spécification.....	72
IV.4.2 Conceptualisation.....	72
IV.4.3 Formalisation.....	80
IV.4.1 Construction de T-Box.....	80
IV.4.2 Construction de A-Box.....	81
IV.4.4 Opérationnalisation.....	81
IV.4.4 Evaluation.....	82
IV.5 Conclusion.....	83
<b>Chapitre V : Réalisation</b>	
V .1 Introduction .....	85
V.2 Les outils et les technologies utilisés.....	85
V.2.1 L'éditeur PROTEGE.....	85
V.2.2 Net-Beans IDE.....	85
V.2.3 La technologie JSP.....	85
V.3 Implémentation de l'ontologie.....	86
V.3.1 L'édition de l'ontologie.....	86
V.3.2 L'ajout et suppression d'un class.....	87
V.3.3 L'ajout d'un attribut.....	87
V.3.4 La vérification de l'ontologie.....	88

## Sommaire

---

V.3.5 La génération du code OWL.....	89
V.4 Intégration de l'ontologie.....	89
V.4.1 Scénarios d'utilisation de l'environnement.....	89
V.5 Conclusion.....	103
<b>Conclusion générale.....</b>	<b>104</b>
<b>Bibliographie.....</b>	<b>105</b>

## Listes des figures

<i>Figure I. 1 : Site Web personnel.....</i>	<i>7</i>
<i>Figure I. 2 : Site Web d'entreprise.....</i>	<i>7</i>
<i>Figure I. 3 : Site Web de gouvernement.....</i>	<i>8</i>
<i>Figure I. 4 : Exemple d'un site web statique.....</i>	<i>9</i>
<i>Figure I. 5 : Exemple d'un site Web dynamique .....</i>	<i>10</i>
<i>Figure I. 6 : Site Web statique .....</i>	<i>10</i>
<i>Figure I. 7 : Site Web dynamique .....</i>	<i>11</i>
<i>Figure II. 1 : Le triangle sémantique .....</i>	<i>18</i>
<i>Figure II. 2 : Cycle de vie d'une ontologie .....</i>	<i>19</i>
<i>Figure II. 3 : Classification de Lassila et MCGuinness.....</i>	<i>23</i>
<i>Figure III. 1 : Diagramme de cas d'utilisation .....</i>	<i>31</i>
<i>Figure III. 2 : Représentation d'un message synchrone .....</i>	<i>32</i>
<i>Figure III. 3 : Représentation d'un message asynchrone.....</i>	<i>32</i>
<i>Figure III. 4 : Exemple d'un diagramme d'activité.....</i>	<i>33</i>
<i>Figure III. 5 : Barre de synchronisation.....</i>	<i>34</i>
<i>Figure III. 6 : Le système d'information soumis a deux types de contraintes .....</i>	<i>35</i>
<i>Figure III. 7 : Le processus de développement en Y .....</i>	<i>35</i>
<i>Figure III. 8 : Diagramme de contexte dynamique.....</i>	<i>40</i>
<i>Figure III. 9 : Diagramme de cas d'utilisation.....</i>	<i>41</i>
<i>Figure III. 10 : Diagramme d'activité inscription.....</i>	<i>45</i>
<i>Figure III. 11 : Diagramme d'activité consulter menu.....</i>	<i>46</i>
<i>Figure III. 12 : Diagramme d'activité contacter service restaurant .....</i>	<i>46</i>
<i>Figure III. 13 : Diagramme d'activité authentifier.....</i>	<i>47</i>
<i>Figure III. 14 : Diagramme d'activité créer commande .....</i>	<i>48</i>
<i>Figure III. 15 : Diagramme d'activité consulter facture .....</i>	<i>49</i>
<i>Figure III. 16 : Diagramme d'activité mis à jour menu .....</i>	<i>49</i>
<i>Figure III. 17 : Diagramme d'activité consulter message .....</i>	<i>50</i>
<i>Figure III. 18 : Diagramme de séquence inscription.....</i>	<i>51</i>
<i>Figure III. 19 : Diagramme de séquence consulter menu.....</i>	<i>52</i>
<i>Figure III. 20 : Diagramme de séquence contacter service restaurant .....</i>	<i>53</i>

<i>Figure III. 21 : Diagramme de séquence créer commande</i> .....	54
<i>Figure III. 22 : Diagramme de séquence authentifier</i> .....	55
<i>Figure III. 23 : Diagramme de séquence consulter facture</i> .....	56
<i>Figure III. 24 : Diagramme de séquence mis à jour menu</i> .....	57
<i>Figure III. 25 : Diagramme de séquence consulter message</i> .....	58
<i>Figure IV.1 : Exemple d'erreurs circulaires dans la taxonomie</i> .....	69
<i>Figure IV.2 Exemple d'erreurs de partition dans l'ontologie</i> .....	70
<i>Figure IV.3 Hiérarchies de concepts</i> .....	75
<i>Figure IV.4 Diagramme de relations binaires</i> .....	76
<i>Figure V.1 : Interface de Protégé 3.4.8</i> .....	86
<i>Figure V.2 : Ajouter ou supprimer un class dans Protégé</i> .....	87
<i>Figure V.3 : Ajouter un attribut sous Protégé</i> .....	87
<i>Figure V.4 : Définir type de l'attribut</i> .....	88
<i>Figure V.5 : Exécution des requêtes SPARQL sous Protégé</i> .....	89
<i>Figure V.6. : Page Accueil</i> .....	90
<i>Figure V.7 : Page Inscription</i> .....	91
<i>Figure V.8 : Page authentification</i> .....	92
<i>Figure V.9 : Page client</i> .....	92
<i>Figure V.10 : Page Consulter Menus</i> .....	93
<i>Figure V.11: Page consulter boissons</i> .....	93
<i>Figure V.12 :Page contacter nous</i> .....	94
<i>Figure V.13 : Page Mon Profile</i> .....	95
<i>Figure V.14 : Page Type repas</i> .....	95
<i>Figure V.15 : Page sélectionner Plats</i> .....	96
<i>Figure V.16 : Page quantité Plats</i> .....	96
<i>Figure V.17 : Recommandation d'aliment</i> .....	97
<i>Figure V.18 : Aliment conseillé</i> .....	97
<i>Figure V.19 : Aliment déconseillé</i> .....	97
<i>Figure V.20 : Page sélectionner Boisson</i> .....	98
<i>Figure V.21 : Page sélectionner dessert</i> .....	98
<i>Figure V.22 : Page Evaluation commande</i> .....	99

<i>Figure V.23 : Conseil sur totales protéines.....</i>	<i>99</i>
<i>Figure V.24 : Page Facture Client.....</i>	<i>100</i>
<i>Figure V.25 : Page Paiement.....</i>	<i>101</i>
<i>Figure V.26 : Page Admin.....</i>	<i>102</i>
<i>Figure V.27 : Page Messagerie.....</i>	<i>102</i>

# Liste des tableaux

<b>Chapitre</b>	<b>Tableau</b>	<b>Page</b>
<b>Chapitre I</b>	Tableau 01 : identification des acteurs	38
<b>Chapitre IV</b>	Tableau 2 : Application des principes d'IO	63
	Tableau 3: Syntaxe et sémantique de SHOIN(D)	67
	Tableau 4: Syntaxe des axiomes avec leurs sémantiques.	68
	Tableau 5: Syntaxe des assertions avec leurs sémantiques.	68
	Tableau 6 : Glossaire de termes.	74
	Tableau 7 : Dictionnaire de concepts.	77
	Tableau 8 : Table des relations binaires.	78
	Tableau 9: Table des attributs.	79
	Tableau 10 : Table des axiomes.	79
	Tableau 11: Table des instances.	80
	Tableau 12 : Assertions d'individus.	81

## RÉSUMÉ :

Le Restaurant intelligent est venu comme une solution quand les restaurants classiques deviennent fastidieux, et fournissent des résultats et un rendu moins exact. Le restaurant intelligent permet d'interpréter le contenu des repas choisi par les clients grâce à une ontologie de domaine.

L'objectif de ce mémoire consiste à la construction d'un site web du Restaurant intelligent (Restaurant Valencia) qui donne des recommandations alimentaires en se basant sur les informations disponibles au niveau de l'ontologie Onto-Resto, qui sert à répondre à certains besoins des clients.

**Mots clés :** Site Web, ontologie, recommandations alimentaires.

## Abstract

The Smart Restaurant came as a solution when classic restaurants become tedious, and provide results and less accurate rendering. The intelligent restaurant allows interpreting the content of the meals chosen by the customers thanks to domain ontology.

The objective of this thesis is to build a web site for Valencia Restaurant, which provides food recommendations based on the information available in the Onto-Resto ontology.

**Keywords:** Website, ontology, food recommendations.

## ملخص

جاء المطعم الذكي كحل عندما أصبحت المطاعم التقليدية مملة، وتقدم نتائج أقل دقة. المطعم الذكي قادر على تفسير محتوى وجبات الطعام التي يختارها العملاء من خلال انطولوجيا المجال.

الهدف من هذا العمل هو موقع الكتروني لمطعم ذكي ( مطعم فالنسيا ) الذي يقدم التوصيات الغذائية لزيائنه على أساس المعلومات المتاحة في الانطولوجيا.

**كلمات البحث:** موقع، الانطولوجيا، التوصيات الغذائية.

### 1. Motivations

L'utilisation des Applications web par les entreprises est devenue très adoptée de nos jours dans les différents domaines (banque commerce, économie...), que ce soit pour une publicité ou bien l'offre de services, ou encore le vent et l'achat d'articles. La mise en place d'un site web est motivée par plusieurs raisons nous citons :

- ❖ **La collecte de données :** pour les entreprises, internet représente une formidable opportunité de recueil des données sur leurs clients.
- ❖ **La mise en place d'un supporte aux utilisateurs :** de plus en plus des sociétés utilisent internet comme support privilégié pour le service avant-vente ou après-vente. En effet, avec un site web il est possible de mettre à la disposition des internautes un maximum d'information commerciale ou technique avec moindre coût.

La construction d'un site web demande un system informatique adapté organisé et évolué pour une meilleure conception de site.

Actuellement, il est nécessaire de décrire la sémantique des objets et de l'organisation du domaine concerné, ceci est inscrit dans la structure même du modèle : cette modélisation est appelée « ontologie ». L'un des principaux objectifs de la recherche dans le domaine des ontologies est que le développement de ces ressources ontologiques garantit des outils puissants, fiables, et maintenables pour le codage. La problématique à laquelle nous nous attaquons est donc la création d'une ontologie qui doit être utilisée par un site web.

En réalité, les ontologies sont construites en étroite collaboration avec les utilisateurs, avec lesquels sont mis en place des procédures d'évaluation des ressources dans leur contexte d'usage.

La recherche sur les ontologies qui a débuté vers les années 90 du siècle dernier dans les domaines de l'intelligence artificielle, de l'ingénierie des connaissances, et de la représentation des connaissances, a vu son champ d'application s'élargir de façon considérable, et fait partie des axes de la recherche actuelle, en particulier dans le domaine de la modélisation des systèmes d'information où les recherches utilisant les ontologies sont de plus en plus nombreuses. Les ontologies sont des systèmes formels dont l'objectif est de représenter les connaissances d'un domaine au moyen de concepts, définis et structurés les uns par rapport aux autres. La représentation ontologique des connaissances assure le



maintien de la cohérence des axiomes et de l'intégrité du système, ainsi que l'évolution de la représentation sans changer la structure.

### **1. Objectif du travail et problématique :**

L'objectif de notre travail est la réalisation d'un site web dynamique pour un restaurant de future ou bien un restaurant intelligent, ce terme né aujourd'hui avec la progression de l'informatique et la programmation web, loin des restaurants classiques qui limite l'utilisateur sur des pages statiques publicitaires.

Dans le cadre de ce projet de fin d'études, nous essayerons de créer un site web d'un restaurant (Restaurant Valencia) qui donne des recommandations alimentaires en se basant sur les informations disponibles au niveau d'une ontologie.

En réalité, notre site est basé essentiellement sur la création et l'utilisation d'une ontologie qui essaye de répondre à la majorité des besoins de l'utilisateur et offre plusieurs possibilités intelligentes en calculant le nombre de calories d'un repas, un repas se compose des plats (Algériennes, Italienne, Français, Syrienne), des desserts ou des boissons. Le système restaurant contribue de façon décisive à garantir aux clients la protection sanitaire et la qualité de leur production alimentaire et à mettre en œuvre des stratégies axées sur le risque par une recommandation alimentaire.

Notre application prend en considération deux types de maladie seulement (diabète et l'hypertension), car elles sont connues d'une façon publique et elles représentent une source de souffrance de la majorité des personnes. Mais ce fait ne représente pas un handicap du système puisque on peut ajouter autre maladie comme (cholestérol, obésité, nausée...).

### **2. Organisation de mémoire**

Nous commençons ce travail par une introduction générale qui explique la motivation de ce projet et présente de façon globale les exigences fonctionnelles attendues de ce site.

Afin de bien présenter ce thème, nous organisons notre mémoire en cinq chapitres :

- Le premier chapitre, détermine quelques concepts sur le web, son utilisation et son intérêt.
- Le deuxième chapitre, permet une description détaillée des deux domaines : ontologie et web sémantique.
- Le troisième chapitre est consacré à la conception de notre site. Ce chapitre est divisé en deux parties : la première partie permet un vue générale concernant le langage de

modélisation UML, alors que la deuxième permet la description détaillée des fonctionnalités de notre site.

- Le quatrième chapitre permet la conception de l'ontologie.
- Le cinquième chapitre représente les outils de développements que nous avons utilisés et également quelque interface de l'application.

Enfin, une conclusion générale résume notre travail avec des perspectives qui décrivent les améliorations qui peuvent être apportées à notre application.

# **Chapitre I : Les sites web**

### I.1. Introduction

Le web est le service de consultation de documents sur Internet, le plus connu, le plus récent et aujourd'hui le plus utilisé dans la vie quotidienne, internet est devenu aussi indispensable que la télévision, le téléphone, le réfrigérateur, etc. Lorsque vous arrivez sur un site web, c'est dans un but précis : trouver les réponses à vos questions, partagé, communiqué, s'amuser. L'ère numérique implique l'accès à l'information en direct, répondant au besoin du tout et tout de suite. Une page web contient des informations, généralement pour informer ou faire connaître.

Dans le cadre d'une utilisation privée, une page web permet par exemple de communiquer et de partager des ressources telles que des photos, des vidéos, des messages, etc. Pour accéder à ces ressources, il suffit d'être connecté sur internet, n'importe où dans le monde avec un délai maximal plus court.

Dans ce chapitre nous allons faire un survol du domaine des sites web et leurs utilisations menée par des notions sur les sites web statiques, dynamiques et les web services, puis on termine par une conclusion.

### I.2. Le World Wide Web

Le web, est un moteur principal de l'internet, C'est le service qui a contribué le plus à sa popularité. Le web a été inventé plusieurs années après Internet, mais c'est lui qui a contribué à l'explosion de l'utilisation d'internet par le grand public, grâce à sa facilité d'emploi. Depuis, le Web est fréquemment confondu avec Internet alors qu'il n'est en réalité qu'un de ses services. Le principe de web repose sur l'utilisation d'hyperliens pour naviguer entre des documents appelés « pages web » grâce à un logiciel appelé navigateur. Une page web est ainsi un simple fichier texte écrit dans un langage de description appelé HTML (*Hypertext Markup Language*), permettant de décrire la mise en page du document et d'inclure des éléments graphiques ou bien des liens vers d'autres documents à l'aide des balises. Au-delà des liens reliant des documents formatés, le web prend tout son sens avec le protocole HTTP (*Hypertext Transfer Protocol*), permettant de lier des documents hébergés par des ordinateurs distants appelés serveurs web, par opposition au client que représente le navigateur. Sur internet les documents sont ainsi repérés par une adresse unique, appelée URL (*Uniform Resource Locator*), permettant de localiser une ressource sur n'importe quel serveur du réseau internet.

### I. 3. Définition d'un site web

Un site web est composé d'un ensemble de documents structurés, nommés *pages web*, stockés (hébergés) sur un ordinateur (serveur) connecté au réseau mondial (internet).

Une page web contient essentiellement du texte, et souvent enrichie d'images, de sons, de vidéos et de liens vers d'autres pages web. [1]

### I. 4. Application sur les sites web

Naviguer sur le web, c'est comme se promener dans une jungle infinie. N'importe qui peut publier de l'information sur le web.

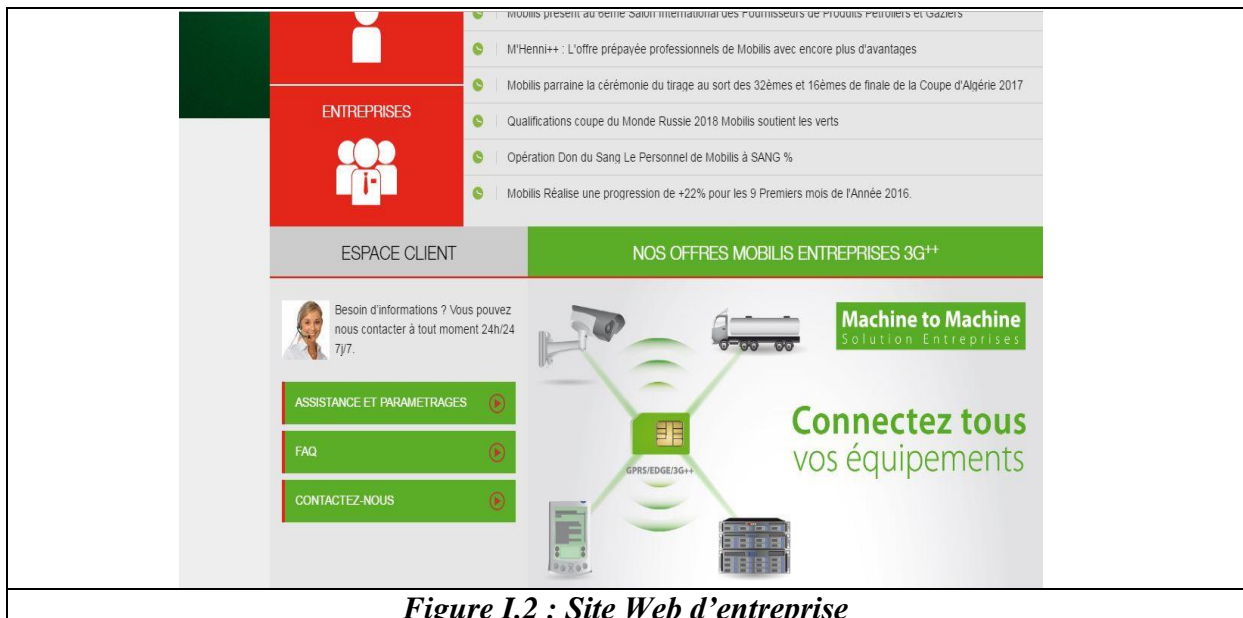
On peut définir plusieurs applications sur le web parmi ceux-ci :

**I.4.1 Des sites web personnels :** Pour que communiquer, publier et mettre des informations sur son actualité, il est indiqué dans la figure suivante (Figure I.1).



**Figure I.1 : Site Web Personnel**

**I.4.2 Des sites web d'entreprise :**Créer un site web d'entreprise est vital, aussi bien pour les jeunes entreprises, PME et auto-entrepreneurs qui souhaitent développer leur activité que pour les sociétés déjà bien installées qui entendent s'attaquer à de nouveaux marchés et augmenter le chiffre d'affaire (la figure I.2).



**Figure I.2 : Site Web d'entreprise**

**I.4.3 Des sites web gouvernementaux :** la figure I.3 montre le site officiel de ministre de l'intérieure d'Algérie.



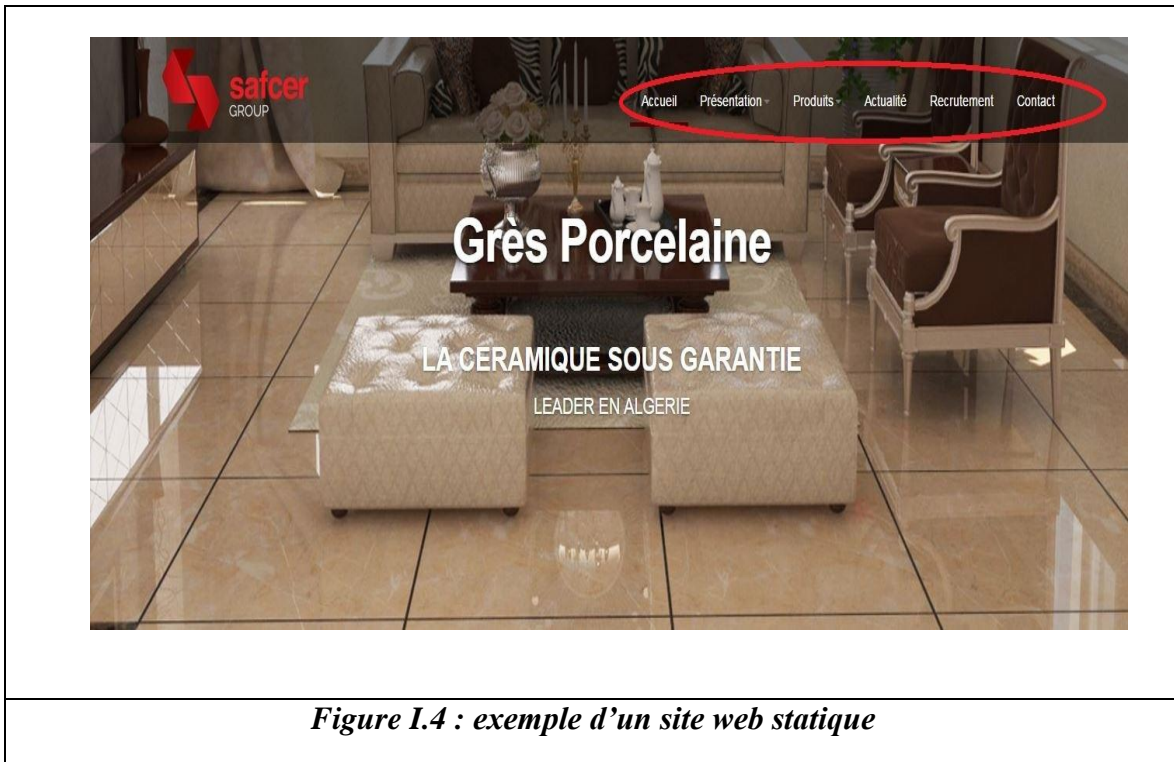
Figure I.3 : Site Web de gouvernement

## I.5. Les sites Web statiques

On entend par page statique, non pas une page sans mouvements ou sans animations, mais une page visible telle qu'elle a été conçue.

Ces pages peuvent présenter toute forme de contenu, animations flash, images, musique, vidéo etc... mais elles sont toujours présentées de la même façon. Elles ne changent pas et c'est en ce sens qu'elles sont statiques. [2]

Ce sont des sites réalisés uniquement à l'aide des langages HTML et CSS (*Cascading Style Sheets*). Ils fonctionnent très bien mais leur contenu ne peut pas être mis à jour automatiquement : il faut que le propriétaire du site (le webmaster) modifie le code source pour y ajouter des nouveautés. Les sites statiques sont adaptés pour réaliser des sites « vitrine », pour présenter son entreprise par exemple (la figure I.4) représente les produits de dalle de sol existants dans le groupe safer. [3]



*Figure I.4 : exemple d'un site web statique*

### **I.6. Les Sites Web dynamiques**

En opposition aux pages statiques, les pages dynamiques permettent de présenter les informations de différentes manières selon l'interaction avec le visiteur. Les pages sont alors construites "à la volée" grâce à une programmation conçue par le webmaster.

Le contenu est issu d'une base de données en fonction de critères établis par l'internaute puis mis en page en temps réel. C'est le cas, par exemple, d'un site e-commerce : présentation des articles par thèmes, couleurs, prix etc...

C'est également le cas des blogs et des forums vus sur la figure I.5 (forum de Djelfa) où les visiteurs peuvent participer au contenu du site. C'est aussi le cas d'un système de mises à jour [2].



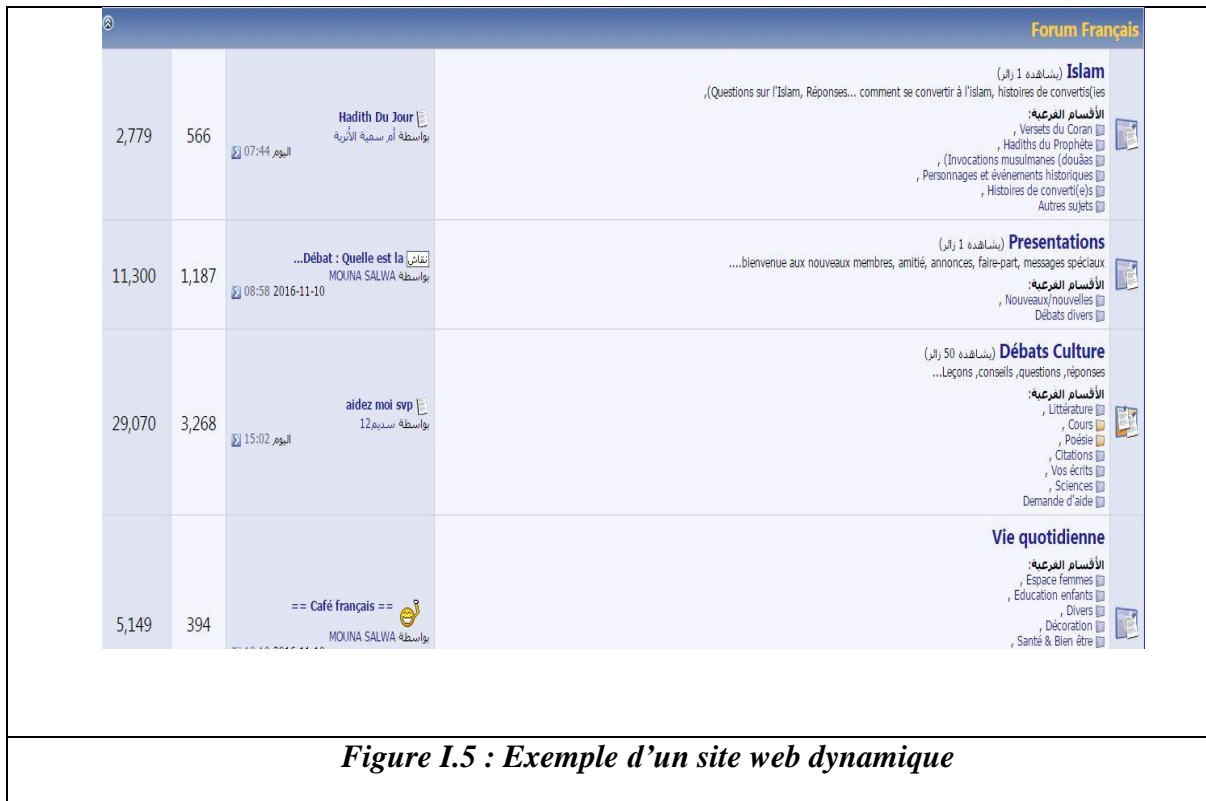


Figure I.5 : Exemple d'un site web dynamique

## I.7 Différence entre un site web statique et un site web dynamique :

- a) **Site statique** : les pages du site ne sont pas modifiables par des utilisateurs. Le site est donc rempli et mis à jour par l'administrateur qui le fait depuis son poste de travail. Une fois le site mis à jour sur l'ordinateur de l'administrateur, celui-ci devra être envoyé sur le site via FTP. Le site est dit statique car les pages HTML qui le compose sont toujours identiques entre deux visites sans mise à jour. Le serveur donc n'a pas besoin de éléments de Scripting. [4]

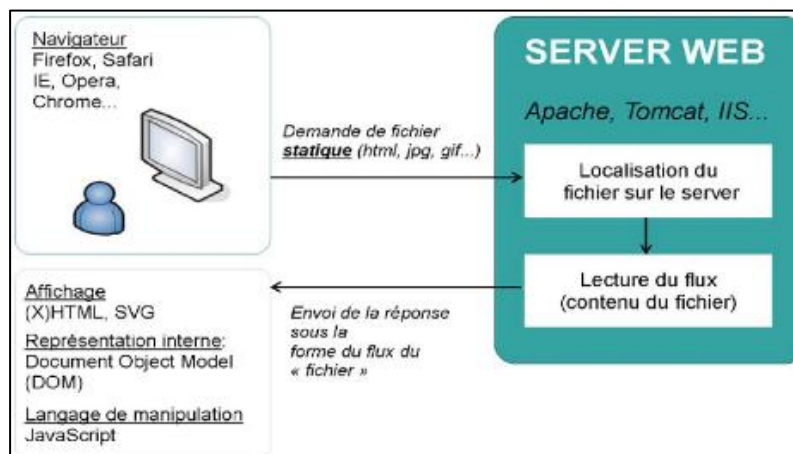
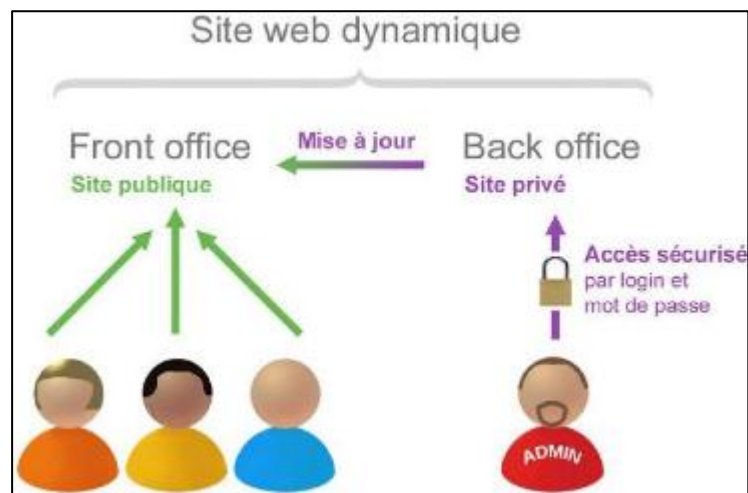


Figure I.6 : site web statique

- b) **Site dynamique** : les pages du site qui le compose peuvent être modifiables par les visiteurs. De plus, entre deux visites sur un même site, le contenu de la page peut être différent sans action de l'administrateur du site Internet. Les grandes applications de ce type de site sont : les forums, les Wiki (Wikipédia étant le plus grand représentant du genre) et tous les sites communautaires (Facebook, Twister, hi5, etc.). Le serveur qui fait fonctionner le site utilise une technologie de Scripting (comme PHP, Ruby, Python ou Perl) ainsi qu'une base de données comme MySQL. [4]



*Figure I.7 : site web dynamique*

### I.8. Le Web Sémantique

A sa création par Tim Berners Lee, au début des années 1990, le web était exclusivement destiné à partager des informations sous forme de pages HTML, affichables par un logiciel « Navigateur web », et généralement destinées à être lues par un utilisateur humain.

Très rapidement, on s'est rendu compte que cette conception du web était bien trop limitée, et ne permettait pas un réel partage du savoir. Les limites du Web actuel sont nombreuses, nous mentionnons :

- ✓ L'hétérogénéité des formats d'information (Word, PDF ...).
- ✓ La nature dynamique et évolutive de l'information.
- ✓ L'absence de sémantique ou structure de l'information elle-même, etc.

Une des solutions existantes correspond à la vision du Tim Berners-Lee qui a défini le Web sémantique comme suit : "Le Web sémantique est une extension du Web actuel (prolongation du Web actuel), dans laquelle l'information reçoit une signification bien définie, améliorant les possibilités de travail collaboratif entre les ordinateurs et les personnes."

Le Web sémantique désigne un ensemble de technologies visant à rendre le contenu des ressources du World Wide Web accessible et utilisable par les programmes et agents logiciels grâce à un système de métadonnées formelles utilisant notamment la famille de langages développés par le W3C.

Il s'agit d'arriver à un Web intelligent, où les informations ne seraient plus stockées mais comprises par les ordinateurs afin d'apporter à l'utilisateur ce qu'il cherche vraiment.

D'après la définition de Tim Berners-Lee, le Web sémantique permettra (contrairement au Web actuel vu comme un Web syntaxique) de rendre le contenu sémantique des ressources Web interprétables, non seulement par l'homme mais aussi par la machine.

Le web sémantique se veut un web de demain dont le contenu peut être appréhendé et exploité par des machines. Résumons les objectifs du web sémantique dans les points suivants :

- ✓ Créer un espace d'échange de ressource entre machines permettant l'exploitation de grand volume d'informations et de services variés.
- ✓ Libérer l'utilisateur d'une bonne partie de son travail de recherche et d'exploitation des résultats.
- ✓ Rendre explicite les relations sémantiques qui existent entre unités d'information.
- ✓ Obtenir des annotations sémantiques plus riches, compatibles avec les standards du Web, tel que le XML, qui fournit un format universel de stockage et d'échange.[5]

### **I.8.1 Outils sémantiques :**

Un outil d'annotation sémantique est un outil logiciel qui permet d'insérer et de gérer des annotations sémantiques liées à au moins une ressource donnée. Dans le cadre du Web Sémantique, les outils d'annotation sémantique utilisent une ontologie, ou tout au moins un modèle formel, qui formalise et structure les annotations produites en fonction des concepts et des contraintes définis dans cette ontologie. Un outil d'annotation sémantique peut aussi être utilisé pour peupler une ontologie, i.e. pour instancier la base de connaissance contenant les instances de l'ontologie de référence. [5]

Nous citons ci-dessous quelques exemples d'outils :

- ❖ Annotea
- ❖ MnM
- ❖ KIM
- ❖ S-CREAM

### **I.9 Conclusion**

Nous avons abordé plusieurs notions sur les web en passant par les différents concepts de site web statique, dynamique, web sémantique, pour pouvoir comprendre tout ce qu'il concerne le web et son environnement.

Dans le prochain chapitre en va parlera des ontologies.

## **Chapitre II : Les ontologies**

### II.1. Introduction

Une ontologie est une représentation partagée et consensuelle entre les collaborateurs qui a pour but de se mettre d'accord sur un sujet particulier avec un objectif commun. Le but est de définir un ensemble de connaissances dans un domaine donné. Exploitée notamment sur le terrain du développement logiciel, elle explicite un vocabulaire en définissant les termes nécessaires pour partager la connaissance liée à ce domaine.

Les ontologies sont employées généralement dans l'intelligence artificielle. Nous allons utiliser cette notion pour faciliter le raisonnement et pour répondre aux problématiques de notre projet. Autrement dit, pour avoir la possibilité de fournir des conseils alimentaires pertinents et personnalisés à un consommateur ou un client.

Ce chapitre représente un tour exhaustif qui concerne les principaux axes sur lesquels s'articule le domaine des ontologies.

Premièrement on commence par un survol historique qui décrit les origines des ontologies. Puis, nous allons citer la définition et les composants de l'ontologie ainsi son cycle de vie. Enfin, nous terminera ce chapitre par une conclusion.

### II .2. Définition

Dans le cadre de l'intelligence artificielle NEECHES et ses collègues furent les premiers à proposer une définition à savoir « une ontologie définit les termes et les relations de base de vocabulaire d'un domaine, ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire ». [6]

En 1993, GRUBER propose la définition suivante : « une ontologie est une spécification explicite d'une conceptualisation ». [7]

En 1995, GUARINO a modifié légèrement la définition de GRUBER, et la définit par : « une ontologie est une spécification partielle et formelle d'une conceptualisation ». [8]

En 1997, ces deux dernières définitions sont regroupées dans celle de BORST comme « une ontologie est définie comme étant une spécification formelle d'une conceptualisation partagée ». [9]

- ✓ **Conceptualisation** : le modèle abstrait d'un phénomène du monde réel par identification des concepts clefs de ce phénomène.
- ✓ **Explicite** : le type de concepts utilisés et les contraintes liés à leur usage sont définis explicitement.
- ✓ **Formelle** : une ontologie doit être traduite en langage interprétable par une machine.
- ✓ **Partagée** : capture la connaissance consensuelle, c'est-à-dire non réservée à quelques individus, mais partagée par un groupe ou une communauté.

### II .3. Les composants d'une ontologie

Les ontologies définissent, le sens des termes et les relations entre eux, elles fournissent un vocabulaire commun. Bien que les ontologies puissent être très différents surtout au niveau du traitement de leurs composants de base telles que les choses, les relations..., elles caractérisent un même univers.

#### II .3.1. Les classes/ les concepts

Les classes ou concepts, représentent des types d'objets. Elles, sont habituellement organisées en taxinomies soumis aux mécanismes d'héritage. Elles peuvent être structurées en une large taxinomie, avec un grand nombre de hiérarchisations ou sans taxinomie.

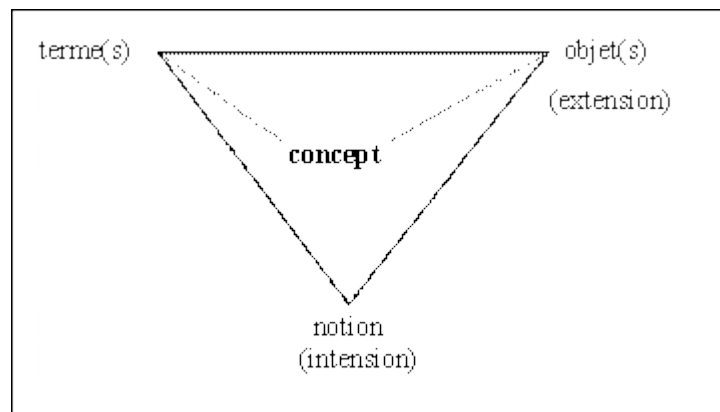


Les concepts sont utilisés dans leur sens large. Ils peuvent être de natures très différentes abstraits ou concrets, élémentaires (électrons) ou composés (atome), réels ou fictifs.

Il arrive que les définitions des ontologies aient été diluées, en ce sens que les taxinomies sont considérées comme des ontologies complètes. [10]

Un concept peut donc décrire une tâche, une fonction, une action, une stratégie, un processus de raisonnement, etc.

Selon PINTO, Il est souvent fait référence aux concepts en tant qu'union de classes et d'instances, alors que chacun des constituants de l'ontologie est considéré comme un fragment de connaissance. [11]



*Figure II.1 : Le triangle sémantique.*

### II .3.2. Les relations et les fonctions

L'interaction entre les notions d'un domaine définit les relations, On trouve des relations binaires, tertiaires

Certaines relations binaires entre des objets sont considérées comme des rôles comme le définit BORGIDA. [12]

Les fonctions sont des cas particuliers de relations dans lesquelles un élément de la relation est défini à partir des autres éléments.

### II .3.3. Les axiomes

Les axiomes désignent des vérités indémontrables qui doivent être admises. Ce sont des affirmations considérées comme évidentes sans preuve. Ils permettent de contraindre les valeurs de classes ou d'instances.

### II .3.4. Les instances

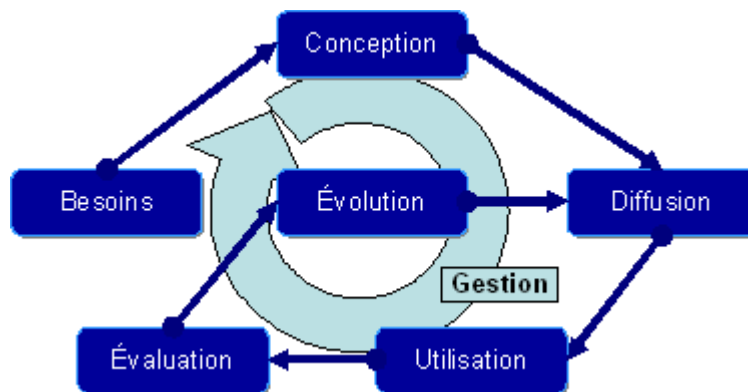
Les instances représentent les éléments des concepts et des relations dans un domaine donné. [13]

### II .4. Le cycle de vie des ontologies

Les ontologies utilisées comme des composants logiciels dans des systèmes répondant à des objectifs opérationnels différents, leur développement doit s'appuyer sur les mêmes principes que ceux appliqués en génie logiciel [14].

Les activités liées aux ontologies sont d'une part des activités de gestion de projet (planification, contrôle, assurance qualité), et d'autre part des activités de développement (spécification, conceptualisation, formalisation). S'y ajoutent des activités transversales de support telles que l'évaluation, la documentation, la gestion de la configuration [15].

Un cycle de vie inspiré du génie logiciel est présenté dans. Chaque étape de ce cycle de vie pose des problèmes de recherche. [14]



*Figure II.2 : Cycle de vie d'une ontologie*

#### II .4.1. Besoins et évaluation

Dans l'utilisation d'une ontologie, des problèmes méthodologiques de recueil d'information, d'analyse et d'identification peuvent surgir. Afin d'éviter ces problèmes dans l'utilisation d'une ontologie, un état des lieux initial approfondi doit être élaboré dans l'étape de détection des besoins, car cette étape ne peut reposer sur des études précédentes ou des retours d'utilisation comme dans le cas de l'activité d'évaluation.

### II .4.2. Conception et évolution

Les phases de conception initiale et d'évolution ont en commun un certain nombre de points :

- ✓ Spécification des solutions ;
- ✓ Conceptualisation et modélisation ;
- ✓ Formalisation (logiques de description, graphes conceptuels, formalismes du web sémantique RDF, RDF(S) et OWL) ;
- ✓ Intégration de ressources existantes ;
- ✓ Implantation (graphes conceptuels, logiques de description).

Le choix de représentation et de conceptualisation faits dans l'ontologie représente un problème de conception et d'évolution. Notons aussi que l'évolution pose le problème de la maintenance de ce qui repose déjà sur l'ontologie. L'ontologie est à la fois un ensemble évolutif et un ensemble de primitives pour décrire des faits et des algorithmes sur ces faits. Ses changements donc, ont un impact direct sur tout ce qui a été construit sur la base de cette ontologie. Le maintien de la cohérence dans une ontologie est un des points clés dans son utilisation. La maintenance de l'ontologie soulève donc des problèmes d'intégration technique et des problèmes d'intégration aux usages. Les domaines de la maintenance comme l'historique et la gestion des versions, la propagation des changements après modification, sont des questions importantes de la recherche actuelle.

### II .4.3. Diffusion

Le déploiement et à la mise en place de l'ontologie interviennent dans la phase de diffusion, où les problèmes sont fortement contraints par l'architecture des solutions. Pour l'application Web on utilisera des technologies adéquates pour le partage de fichiers, nous pouvons utiliser des architectures distribuées ou l'architectures Peer to Peer. Dans toutes les architectures dans les serveurs web, services web, Peer to Peer, etc. la distribution des ressources et leur hétérogénéité du point de vue syntaxiques, sémantiques, protocolaires, contextuelles, ou autres posent des problèmes de recherche sur l'interopérabilité et le passage à l'échelle (larges bases, optimisation d'inférences, propagation de requêtes, etc.).

### II .4.4. Utilisation

Les activités reposant sur la disponibilité de l'ontologie, comme l'annotation des ressources, la résolution de requête, la déduction de connaissances, etc. constituent la phase d'utilisation.

Les problèmes de la conception des interactions avec l'utilisateur et de leur ergonomie concernant les interfaces dynamiques, les profils et contextes d'utilisation, sont posés dans toutes ces activités. Sur ce point, l'ontologie apporte à la fois de nouvelles solutions et de nouveaux problèmes.

### II .4.5. Gestion

L'existence d'une activité permanente de gestion et planification est importante pour assurer une pérennité dans le travail de suivi et la politique globale pour la détection, la préparation l'évaluation des itérations d'un cycle et s'assurer que l'ensemble de ces causes améliorent les systèmes d'information.

### II .5. Classification des ontologies

Les ontologies peuvent être classifiées en fonction de plusieurs dimensions. Parmi celle-ci, nous examinons quatre, à savoir :

- ✓ La richesse de la structure interne des ontologies ;
- ✓ L'objet de conceptualisation ;
- ✓ Le niveau de granularité ;
- ✓ Le niveau de formalisation de la représentation des connaissances.

#### II .5.1 Typologie selon la richesse de la structure interne des ontologies

LASSILA et MCGUINNESS proposent une classification des ontologies selon la richesse de leur structure interne. Telle qu'il lustrée à la figure II.3, cette classification consiste en un continuum allant du vocabulaire contrôlé à l'ontologie avec des contraintes logiques. Lorsque nous nous déplaçons tout le long du continuum, le sens des termes devient plus précis et le degré de formalisme augmente ce qui réduit l'ambiguïté. Ce critère est connu aussi dans d'autres travaux « le degré d'engagement sémantique » qui correspond au niveau des spécifications formelle permettant de restreindre l'interprétation de chaque concept et ainsi d'en donner la sémantique. Autrement dit, le degré d'engagement sémantique fait en particulier référence au niveau sémantique des connaissances que l'ontologie représente.

Les principales catégories de cette classification sont :

- ✓ Vocabulaire contrôlé ;
- ✓ Glossaire ;
- ✓ Thésaurus ;
- ✓ Hiérarchie informelle (is-a) ;

- ✓ Hiérarchie formelle (is-a) ;
- ✓ Hiérarchie formelle (is-a) avec instances du domaine ;
- ✓ Frame ;
- ✓ Ontologies avec restrictions de valeur ;
- ✓ Ontologies avec contraintes logiques.

a) **Vocabulaire contrôlé** : est une liste finie de termes définis par un groupe de personnes où une communauté. La signification des termes n'est pas forcément définie et il n'y a pas d'organisation logique entre les termes. Les catalogues sont des exemples de cette catégorie.

b) **Glossaire** : est une liste de termes avec leurs significations spécifiées en langage naturel. Cette représentation apporte plus d'informations car une personne peut lire la définition, cependant elle n'est pas interprétable par l'ordinateur.

c) **Thésaurus** : il fournit une sémantique supplémentaire entre les termes tels que la relation de synonymie. Mais, il n'offre pas la structure hiérarchique explicite.

d) **Hiérarchie informelle (is-a)** : est une hiérarchie où la notion vague de généralisation et de spécialisation est fournie bien que ce ne soit pas une hiérarchie qui correspond à la stricte notion de subsomption. Ce qui revient à dire que la structure de ce type d'hiérarchie est basé sur la proximité des concepts.

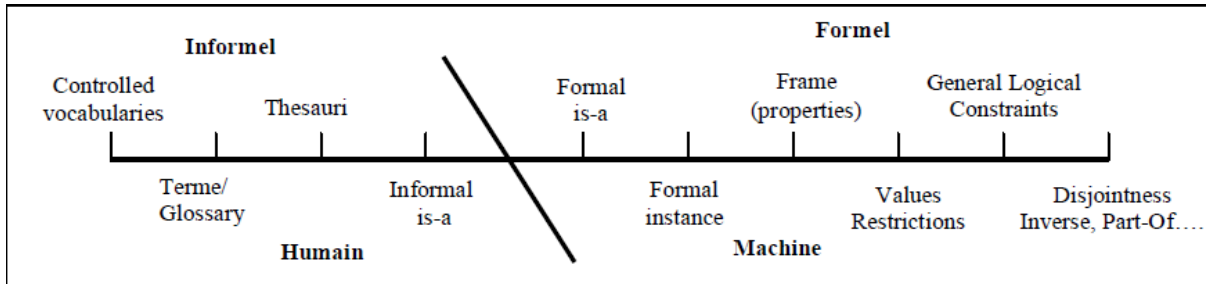
e) **Hiérarchie formelle (is-a)** : est une hiérarchie dont la structure est déterminée par des relations de subsomption. Dans cette catégorie, si A est une super -classe de B, alors si un objet est une sous-classe de B, il est nécessairement le cas qu'il est ainsi une sous -classe de A. De même, si A est une super-classe de B, alors si un objet est une instance de B, alors il est nécessairement le cas qu'il est une instance de A. Cette ontologie ne peut inclure que des noms des classes.

f) **Hiérarchie formelle (is-a) avec instances du domaine** : similaire à la catégorie précédente mais incluant des instances.

g) **Frame** : est une ontologie incluant des classes avec leurs propriétés. Le fait d'inclure des propriétés dans la description d'une classe devient intéressante dans la mesure où ces Propriétés pouvant être héritées par les sous classes de la taxonomie formelle « is a ».

**h) Ontologies avec restrictions de valeur** : sont des ontologies pouvant contenir des restrictions sur les valeurs des propriétés.

**i) Ontologies avec contraintes logiques** : sont les ontologies les plus expressives. Ces ontologies pouvant contenir des contraintes entre constituants (exemple relations) définies dans un langage logique. L'otologiste peut exprimer les contraintes dans un langage d'ontologie expressif tel qu'OWL.



*Figure II.3 Classification de LASSILA et MCGUINNESS.*

Peu de gens considèrent les quatre premières catégories (le vocabulaire contrôlé, le glossaire, le thésaurus et la hiérarchie informelle is-a) comme des ontologies et leur justification est qu'ils fournissent une conceptualisation consensuelle d'un domaine. À ce titre, la notion d'ontologie est diluée.

En revanche, Les autres qui sont les plus nombreux, les considèrent comme des ressources terminologiques. Notamment, ils préfèrent avoir une hiérarchie formelle incluse avant de considérer quelque chose comme une ontologie. Dans ce même ordre d'idée, Gomez et ses collègues ont distingué les ontologies légères (*light weight ontologies*) et les ontologies lourdes (*heavy weight ontologies*). Les ontologies légères incluent des concepts comprenant des propriétés et qui sont organisés en taxonomies avec des relations conceptuelles. Tandis que les ontologies lourdes ajoutent aux ontologies légères des axiomes et des restrictions clarifiant le sens. Les ontologies lourdes modélisent un domaine de façon plus profonde avec plus de restrictions sur la sémantique du domaine. [16]

### II .5. 2 Typologie selon l'objet de conceptualisation

GOMEZ et ses collègues proposent une classification selon le sujet de conceptualisation des ontologies. Cette classification est une extension des travaux de, ceux de VAN, et ceux GUARINO :

- ✓ Ontologies de représentation de connaissances ;
- ✓ Ontologies générique / générale / commune ;
- ✓ Ontologies de haut niveau / de niveau supérieur ;
- ✓ Ontologies du domaine ;
- ✓ Ontologies de tâche ;
- ✓ Ontologies d'application.

**a) Ontologies de représentation de connaissances:** elles regroupent les primitives utilisées pour formaliser les connaissances sous un paradigme de représentation de connaissances.

L'exemple le plus expressif est l'ontologie de frame qui intègre les primitives de représentation des langages à base de *frames* : classes, instances, facettes, propriétés/slots, relations, etc.

**b) Ontologies générale ou commune ou générique:** les connaissances modélisées dans ce type d'ontologie doivent être générales pour être réutilisées dans différents domaines.

Elle Comprend le vocabulaire relatif au temps, espace, unités, etc.

**c) Ontologies de niveau supérieur ou de haut niveau :** ce type d'ontologies modélise des concepts de haut niveau auxquels ces derniers doivent être reliés au sommet des ontologies de plus bas niveaux. Cependant, il existe plusieurs ontologies de haut niveau qui se différencient par le critère utilisé pour classer les concepts généraux de la taxonomie.

**d) Ontologies du domaine:** elles sont réutilisables au sein d'un domaine donné, mais pas d'un domaine à un autre. Les connaissances représentées dans ce type d'ontologies sont spécifiques à un domaine particulier. Elle fournit un vocabulaire d'un domaine spécifique au travers de concepts et de relations qui modélisent les principales activités, les théories du domaine en question. Les concepts et les relations des ontologies de domaine sont souvent des spécialisations de concepts et des relations définis dans des ontologies de haut niveau.

**e) Ontologies de tâches:** ce type d'ontologie est utilisé pour décrire un vocabulaire relatif à une tâche ou une activité générique (faire un diagnostic, planifier une activité . . .) en spécialisant certains termes des ontologies de haut niveau. Ces ontologies fournissent un ensemble de termes au moyen desquels on peut décrire, au niveau générique, comment résoudre un type de problème.

**f) Ontologies d'application :** Ce sont les ontologies les plus spécifiques. Contrairement à l'ontologie de domaine, l'ontologie d'une application donnée ne peut pas être réutilisée pour d'autres applications. Elle contient les connaissances requises pour une application

particulière. Ce type d'ontologie décrit des concepts qui dépendent à la fois d'un domaine particulier et d'une tâche particulière. Par conséquent, elle spécialise souvent des ontologies de domaine et des ontologies de tâches pour une application donnée. [16]

### II .5. 3 Typologie selon le niveau de granularité

On peut distinguer les ontologies selon le niveau de détail utilisé lors de la conceptualisation de l'ontologie en fonction de l'objectif opérationnel envisagé pour l'ontologie, deux catégories peuvent être identifiées : granularité fine et granularité large :

**a) Granularité fine** : elle correspond à une ontologie très détaillée, possédant ainsi un vocabulaire plus riche capable d'assurer une description détaillée des concepts pertinents d'un domaine ou d'une tâche ;

**b) Granularité large** : elle correspond à un vocabulaire moins détaillé. Les ontologies génériques possèdent une granularité large, compte-tenu du fait que les notions sur lesquelles elles portent peuvent être raffinées par des notions plus spécifiques. [16]

### II .5. 4 Typologie selon le niveau de formalisation utilisé

Les ontologies peuvent être distinguées en fonction du degré de formalisme utilisé pour les exprimer. USCHOLD et GRUNINGER proposent une classification contenant les quatre catégories : Hautement informelles ; Semi-informelles ; Semi-formelles et rigoureusement formelles.

- a) *Hautement informelle* : elle est exprimée en langue naturelle (sémantique ouverte).
- b) *Semi-informelle* : elle est exprimée dans une forme restreinte et structurée de langage naturel.
- c) *Semi-formelle* : elle est exprimée dans un langage artificiel défini formellement.
- d) *Rigoureusement formelle* : l'ontologie est exprimée dans un langage contenant une sémantique formelle, des théorèmes, et des preuves pour vérifier les propriétés telles que la validité et la complétude.

Il est évident qu'il est difficile de faire la différence, pour une ontologie considérée, entre ces quatre différentes classifications et de choisir celle qui correspond. [16]



### II.6 Conclusion

Nous avons vu dans ce chapitre un rapide aperçu sur les notions d'ontologies et le Web Sémantique. Notre intérêt est de définir les ontologies en présentant leurs composants, utilisations, typologies, ainsi que leurs constructions et les outils nécessaires pour leur édition et l'inférence.

## **Chapitre III : La conception du site web**

### III.1 Introduction

Après avoir décrit les deux domaines, concernés par notre travail, nous avons fixé l'objectif de notre application et déterminée les choix techniques et les outils nécessaires pour sa réalisation.

L'approche utilisée dans la conception de notre travail est l'approche objet. Elle consiste à obtenir un modèle informatique d'une collection d'éléments d'une partie du monde réel en un ensemble d'entités appelées objets. Cette approche présente plusieurs avantages dont les principaux sont : la modularité, l'extensibilité et la réutilisation.

Pour décrire les fonctionnalités de notre site web, nous avons choisie l'utilisation du langage de modélisation UML et plus précisément nous avons fixé l'utilisation des trois diagrammes : diagramme de cas d'utilisation, diagramme d'activité et diagramme de séquence.

A travers ce chapitre nous allons présenter le langage de modélisation UML et déterminer ses différents diagrammes, ensuite, nous allons passer à la description des différentes fonctionnalités de notre application en utilisant les trois diagrammes cités dans les paragraphes précédentes.

Nous allons terminer ce chapitre par une conclusion qui permet le passage vers le chapitre suivant.

### *Partie 01 : L'outil informatique*

#### **III.2 Qu'est-ce que UML ?**

##### **III.2.1. UML le langage de modélisation unifié**

UML est la forme contractée de Unified Modeling Language qui peut se traduire en français par le langage unifié pour la modélisation. Il fournit les fondements pour spécifier, construire, visualiser et décrire les éléments d'un modèle. Il se base sur une sémantique précise et sur une notation graphique expressive. C'est un moyen d'exprimer des modèles et faisant abstraction de leur implémentation, c'est à dire le modèle fourni par UML est valable pour n'importe quel langage de programmation.

##### **a) UML est un langage graphique**

C'est un ensemble de graphisme très complet. Sa notation graphique permet d'exprimer visuellement une solution du problème. L'aspect formel de sa notation limite les ambiguïtés et les incompréhensions. Son aspect visuel facilite la comparaison et l'évaluation des solutions.

##### **b) UML est un support de communication**

Il facilite l'expression et la communication de modèles en fournissant un ensemble de symboles (la notation) et de règles qui régissent l'assemblage de ces symboles (la syntaxe et la sémantique).

##### **c) UML n'est pas une méthode**

UML n'est pas une méthode ou un processus : UML est développé pour permettre la modélisation des systèmes d'une manière standard et pas pour être une méthode de conception ou d'analyse, pour ça il lui manque la démarche.

##### **d) UML est basé sur un méta-modèle**

Un modèle de plus haut niveau qui définit les éléments d'UML (les concepts utilisables) et leur mode d'utilisation (leur signification et leur mode d'utilisation).

##### **e) UML est une norme**

UML est devenu une référence en matière de langages de description des systèmes et d'aide à la conception.

##### **III.2.2. UML est une unification des méthodes**

UML est né de la fusion des trois méthodes ont le plus influencé la modélisation objet au milieu des années 90 :

- OMT de James Rumbaugh (General Electric) fournit une représentation graphique des aspects statique, dynamique et fonctionnel d'un système
- OOD de Grady Booch, définie pour le Département de Défense, introduit le concept de paquetage (package)
- OOSE d'Ivar Jacobson (Ericsson) fonde l'analyse sur la description des besoins des utilisateurs (cas d'utilisation, ou use cases).

### III.3 Diagrammes d'UML

UML 2.0 comporte treize types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information. Ils se répartissent en deux grands groupes :

- **Diagrammes structurels ou diagrammes statiques (UML Structure)**
  1. Diagramme de classes (Class diagram)
  2. Diagramme d'objets (Object diagram)
  3. Diagramme de composants (Component diagram)
  4. Diagramme de déploiement (Deployment diagram)
  5. Diagramme de paquetages (Package diagram)
  6. Diagramme de structures composites (Composite structure diagram)
- **Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior)**
  1. Diagramme de cas d'utilisation (Use case diagram)
  2. Diagramme d'activités (Activity diagram)
  3. Diagramme d'états-transitions (State machine diagram)
  4. Diagrammes d'interaction (Interaction diagram)
  5. Diagramme de séquence (Sequence diagram)
  6. Diagramme de communication (Communication diagram)
  7. Diagramme global d'interaction (Interaction overview diagram)
  8. Diagramme de temps (Timing diagram)

Ces diagrammes, d'une utilité variable selon les cas, ne sont pas nécessairement tous produits à l'occasion d'une modélisation. Les plus utiles pour la maîtrise d'ouvrage sont les diagrammes d'activités, de cas d'utilisation, de classes, de séquence

Les diagrammes de composants, de déploiement et de communication sont surtout utiles pour la maîtrise d'œuvre à qui ils permettent de formaliser les contraintes de la réalisation et la solution technique.

### III .3.1. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il est utilisé dans les deux étapes de capture des besoins fonctionnels et techniques.

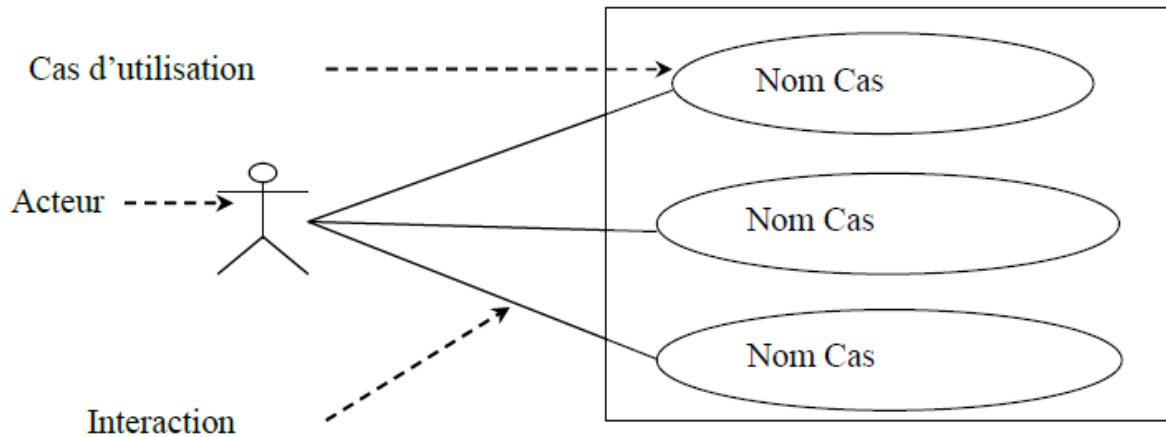


Figure III.1 : Diagramme de cas d'utilisation

#### III .3.1.1. Éléments de base du diagramme de cas d'utilisation

**a. Un acteur :** représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages éventuellement porteurs de données.

**b. Cas d'utilisation :** ensemble d'actions réalisées par le système, en réponse à une action d'un acteur (voir figure III.1).

**c. L'interaction :** l'interaction entre l'acteur et le cas d'utilisation est représentée comme une association pouvant porter des multiplicités (voir figure III.1).

#### III .3.2. Diagramme de séquence

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur. IL indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre.

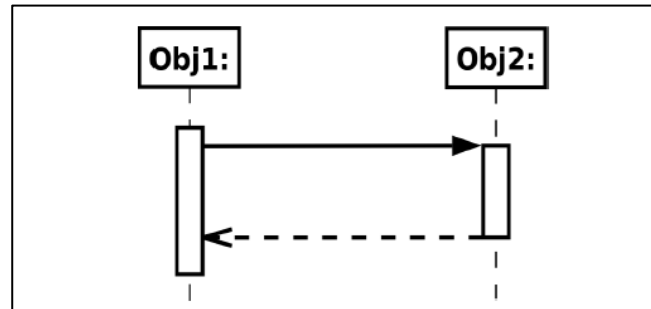
##### III .3.2.1. Définition d'un message

Un message définit une communication particulière entre des lignes de vie. Plusieurs types de messages existent, les plus communs sont :

- ✓ L'envoi d'un signal ;
- ✓ L'invocation d'une opération ;

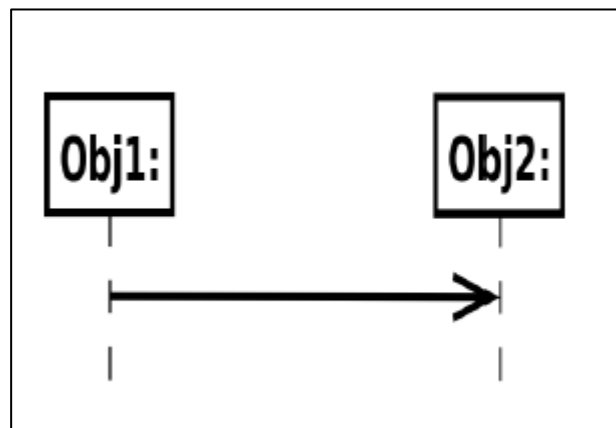
✓ La création ou la destruction d'une instance.

**a. Message synchrone :** bloque l'expéditeur jusqu'à prise en compte du message. Le flot de contrôle passe de l'émetteur au récepteur. Graphiquement, un message synchrone se représente par une flèche en traits pleins et à l'extrémité pleine partant de la ligne de vie d'un objet expéditeur et allant vers celle de l'objet cible.



*Figure III.2 : Représentation d'un message synchrone*

**b. Message asynchrone :** n'interrompt pas l'exécution de l'émetteur. Le message peut être pris en compte par le récepteur à tout moment ou ignoré. Graphiquement, un message asynchrone se représente par une flèche en traits pleins et à l'extrémité ouverte partant de la ligne de vie d'un objet expéditeur et allant vers celle de l'objet cible.



*Figure III.3 : Représentation d'un message asynchrone*

### III .3.3. Diagramme d'activités

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une

méthode ou le déroulement d'un cas d'utilisation. On peut attacher un diagramme d'activités à n'importe quel élément de modélisation afin de visualiser, spécifier, construire ou documenter le comportement de cet élément.

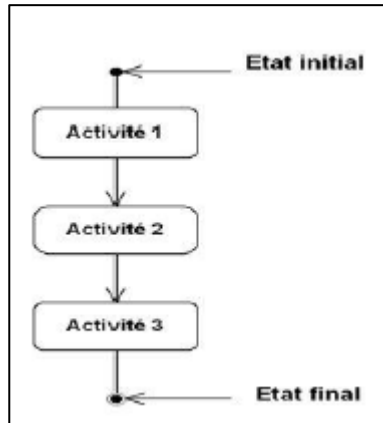


Figure III.4 : Exemple diagramme d'activité.

### III .3.3.1. Eléments de base du diagramme d'activités

- **Activité** : représente une exécution d'un mécanisme, un déroulement d'étapes séquentielles.
- **Transition** :

La transition automatique : déclenchée par la fin d'une activité et provoquent le début immédiat d'une autre.

La transition gardée : le passage à l'activité suivante n'est possible que si la condition de la transition est vérifiée.

Synchronisation : il est possible de synchroniser les transitions à l'aide des "barres de synchronisation"

Barre de synchronisation : Une barre de synchronisation permet d'ouvrir et fermer des branches parallèles au sein d'un flot d'exécution : les transitions qui partent d'une barre de synchronisation ont lieu en même temps.

On ne franchit une barre de synchronisation qu'après réalisation de toutes les transitions qui s'y rattachent. [17]



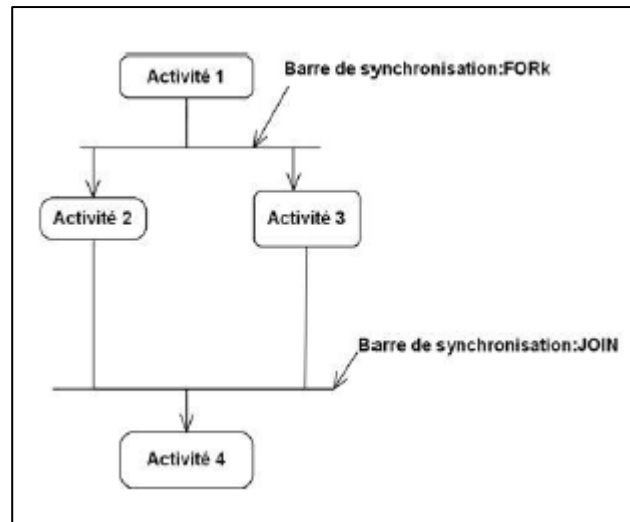


Figure III.5 : Barres de synchronisation.

### III.4. Le processus de développement

Il existe deux processus de développements un processus de développement bien défini : le processus en Y (processus 2TUP), qui va de la détermination des besoins fonctionnels attendus du système jusqu'à la conception et le codage final, il définit une séquence d'étapes, en partie ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant. et aussi le processus unifié.

#### III.4.1. Processus Unifié

Le Processus Unifié (PU) est un processus de développement O.O de logiciel construite sur UML ; elle est itérative et incrémentale, centrée sur l'architecture, conduite par les cas d'utilisation et pilotée par les risques.

Un tel processus se déroule selon les 4 phases suivantes :

- a. **La pré-étude** : c'est ici qu'on évalue la valeur ajoutée du développement et la capacité technique à le réaliser (étude de faisabilité).
- b. **L'élaboration** : sert à confirmer l'adéquation du système aux besoins des utilisateurs et à livrer l'architecture de base.
- c. **La construction** : sert à livrer progressivement toutes les fonctions du système.
- d. **La transition** : déployer le système sur des sites opérationnels. [18]

#### III.4.2. Processus 2TUP

On dit du processus **UP** qu'il est générique c.à.d. qu'il définit un certain nombre de critères de développement, que chaque société peut par la suite personnaliser afin de créer son propre

processus plus adapté à ses besoins. C'est dans ce cadre que l'entreprise *Valtech* a créé la méthode **2TUP** « 2 Track Unified Process ».

Le processus **2TUP** est un processus qui répond aux caractéristiques du Processus Unifié. Il apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes.

« 2 Track » signifie littéralement que le processus suit deux chemins. Il s'agit des « chemins fonctionnels » et « d'architecture technique », qui correspondent aux deux axes de changement imposés au système d'information.



Figure III.6 : Le système d'information soumis à deux types de contraintes.

A l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des deux branches. Cette fusion conduit à l'obtention d'un processus de développement en forme de Y, comme illustré par la Figure III.7.

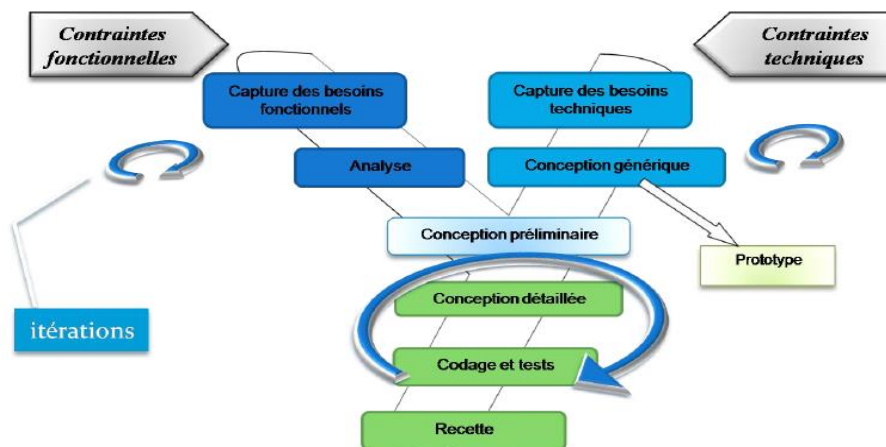


Figure III.7 Le processus de développement en Y.

### III.4.2.1. La branche gauche (fonctionnelle)

Capitalise la connaissance du métier de l'entreprise. Elle constitue généralement un investissement pour le moyen et le long terme.

Les fonctions du système d'information sont en effet indépendantes des technologies utilisées.

Cette branche comporte les étapes suivantes :

- La capture des besoins fonctionnels, qui produit un modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie, au plus tôt le risque de produire un système inadapté aux utilisateurs.
- L'analyse, qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en termes de métier.

### III.4.2.2. La branche droite (architecture technique)

Capitalise un savoir-faire technique. Elle constitue un investissement pour le court et moyen terme. Les techniques développées pour le système peuvent l'être en effet indépendamment des fonctions à réaliser. Cette branche comporte les étapes suivantes :

- La capture des besoins techniques. Les outils et le matériel sélectionné ainsi que la prise en compte des contraintes d'intégration avec l'existant.
- La conception générique, qui définit ensuite les composants nécessaires à la construction de l'architecture technique.

### III.4.2.3. La branche du milieu

Cette branche comporte les étapes suivantes :

- La conception préliminaire, elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer.
- La conception détaillée, qui étudie ensuite comment réaliser chaque composant.
- Le codage, qui produit ses composants et teste au fur et à mesure les unités de code réalisées.
- L'intégration, consiste à valider les fonctionnalités du système développé.

Le processus 2TUP préconise un cycle de vie en Y qui dissocie et parallélise la résolution des problèmes fonctionnelles et techniques. Le cycle de vie de 2TUP s'apparente à un cycle en cascade mais introduit une forme itérative à certaines tâches. Il n'est pas certain que ce cycle en cascade s'apparente réellement à une approche Agile. Par contre, 2TUP préconise des formes de recherche de qualité et de performance intéressantes telles que les services réutilisables et la conception générique (Framework et Design pattern). [18]

### III.4.3 Conclusion

Le langage de modélisation UML a passé son utilité concernant la modélisation des applications de gestion ou les sites web, donc il représente un moyen indispensable pour décrire les fonctionnalités du site web de notre restaurant de futur.

### *Partie 02 : L'étude de cas << Site web de restaurant >>*

#### III.5. L'étude préliminaire :

Etude préliminaire ou (pré-étude) est la première étape de notre processus de développement. Elle survient à la suite d'une décision de démarrage de projet, consiste à effectuer un premier repérage des besoins fonctionnelle et non fonctionnelle, en considérant le système comme une boîte noire, afin d'étudier sa place dans le système métier plus global de l'entreprise. Après avoir identifié les acteurs qui interagiront avec le system, il sera développé un premier modèle UML de niveau contexte, pour pouvoir établir précisément les frontières fonctionnelles du system.

##### III.5.1 Le cahier de charge :

###### **a. Présentation du projet**

Le système à développer est un site web dynamique en interaction avec une ontologie pour faire des recommandations alimentaires tel que le calcul des calories pour chaque repas commandé par le client.

Les objectifs principaux de notre system sont :

- ✓ Donner a l'internaute la possibilité d'accéder au site et consulter les actualités de restaurant.
- ✓ Permettre de consulter les menus.
- ✓ Permettre d'inscrire en ligne.
- ✓ Permettre de faire des recommandations alimentaires aux clients.
- ✓ Permettre de calculer le nombre de calories de repas.

###### **b. Les grands choix techniques**

Outil de modélisation : UML (langage de modélisation unifié).

L'architecture de l'application : client/Ontologie.

La plateforme disponible :PC+Windows+ environnement de travail HTML, CSS+JSP+ environnement de travail pour l'ontologie (protéger).

### C. Recueil des besoins fonctionnels

Ce sont les actions que doit effectuer le système en réponse à une demande, d'après nos études et recherches, nous a conduits aux besoins fonctionnels suivant :

- ❖ L'internaute peut s'inscrire en ligne comme un membre
- ❖ L'internaute peut consulter le menu
- ❖ L'internaute peut contacter le service de restaurant pour plus d'informations
- ❖ Le client peut effectuer une commande
- ❖ Le client peut consulter la facture et faire payer en ligne.

### d. Recueil des besoins opérationnels

Définissez les exigences opérationnelles comme un ensemble d'exigences fonctionnelles axées sur des objectifs précis. Généralement, les spécifications opérationnelles s'appliquent aux éléments suivants :

- ✓ Fonctionnalité pour l'utilisateur final.
- ✓ Réduction des temps de réponse.
- ✓ temps de disponibilité.
- ✓ Réduction des taux d'erreurs.
- ✓ Archivage et conservation des informations.

### III.5.2 Description du contexte :

#### a. Identification des acteurs

Les acteurs	Description	Les actions
L'internaute	personne qui visite le site pour voir les actualités de restaurant	-S'inscrire dans le site -Consulter le menu -Contacter le service de restaurant
Client	un membre dans le site	-Créer commande -Consulter les conseils Alimentaires
Admin	Administrateur du site web	-Lire et répondre aux messages reçus -Mis à jour Menu

Tableau 01 : identification des acteurs

### **b. Identification des messages**

Le system reçoit les messages suivant :

#### **Internaute**

- ✓ Accéder au site
- ✓ Consulter le Menu
- ✓ S'inscrire dans le site
- ✓ Contacter le service restaurant

#### **Client**

- ✓ Créer commande
- ✓ Voir conseils alimentaires
- ✓ Consulter facture
- ✓ Faire le paiement
- ✓ Valider commande
- ✓ Annuler commande

#### **Admin**

- ✓ Lire et répondre aux messages reçus
- ✓ Mis à jour Menu

Le system émet les messages suivants :

- Notification des inscriptions.
- Liste des menus.
- Les actualités.
- Information sur le restaurant.
- Recommandations alimentaires
- Totale de calories pour chaque repas
- Information pour contacter l'administrateur

III.5.3 Diagramme de contexte dynamique :

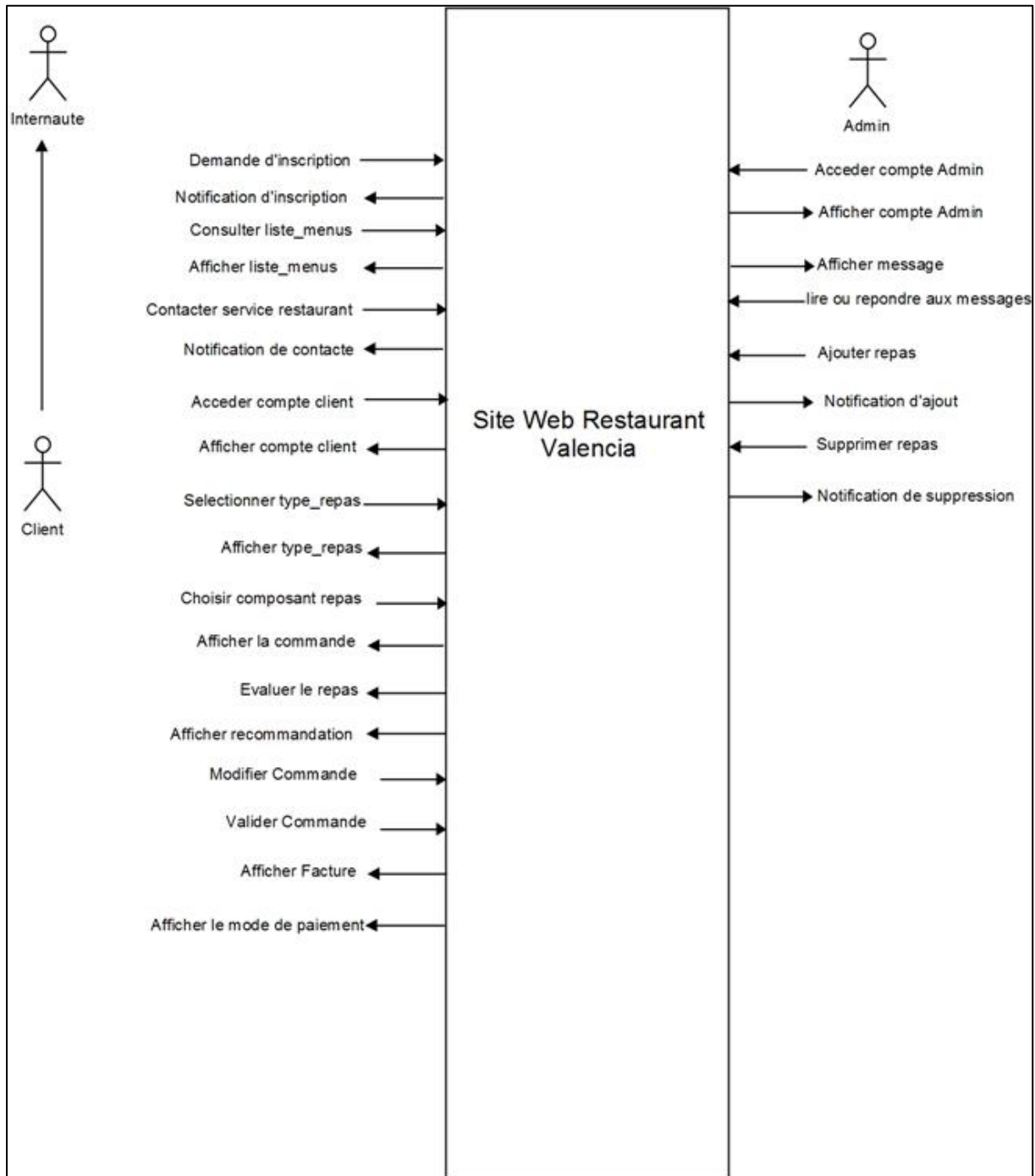
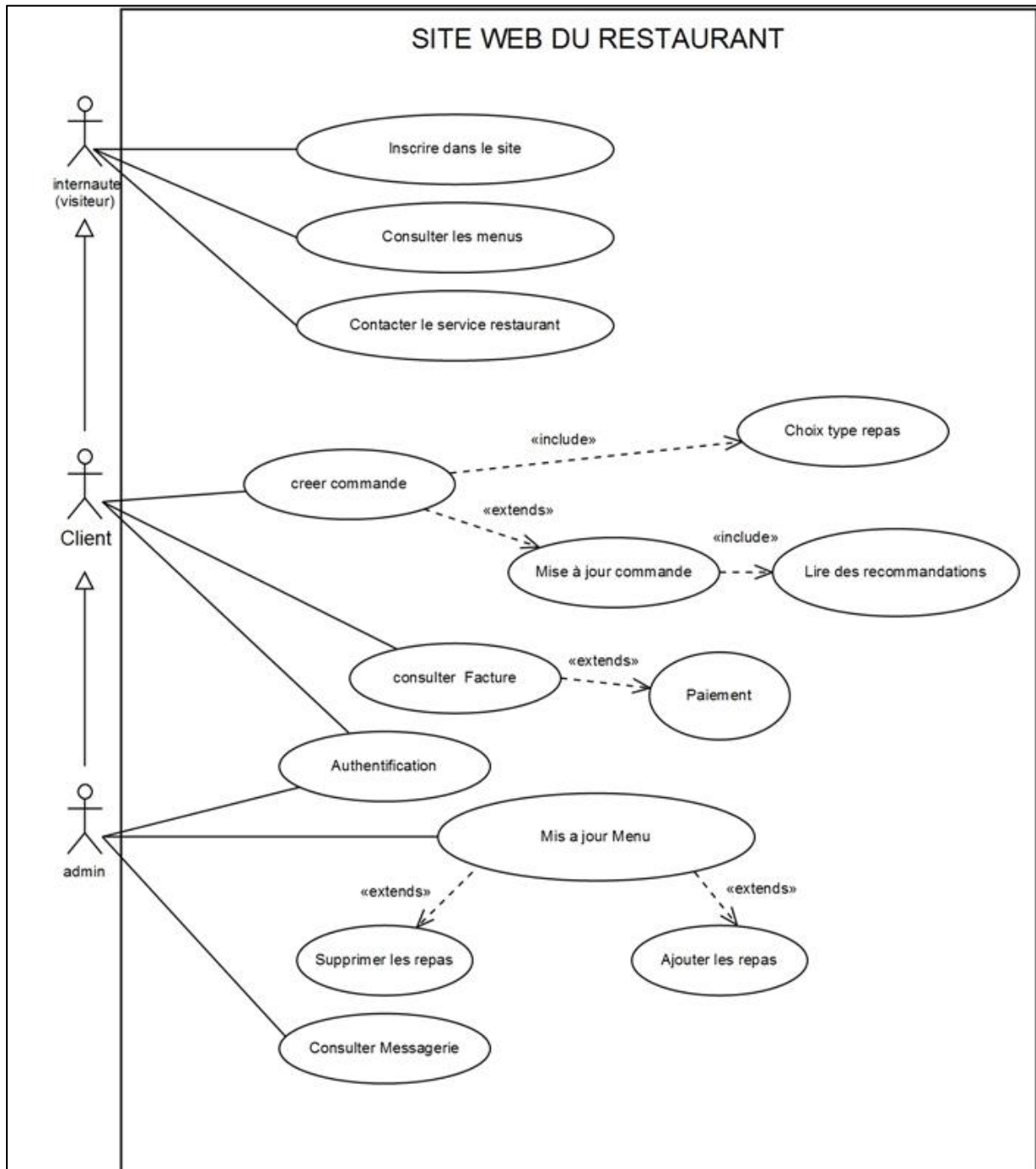


Figure III.8 : Diagramme de contexte dynamique

### III.6. L'étude détaillée :

#### III.6.1 Diagramme cas d'utilisation :

Maintenant que nous avons identifié les cas d'utilisation et leurs acteurs, nous allons les représenter graphiquement sur un diagramme de cas d'utilisation qui représente le fonctionnement du système vis-à-vis de l'utilisateur :



*Figure III.9 : Diagramme cas d'utilisation*



### ❖ *Cas d'utilisation « Inscription »*

Objectif : la possibilité d'effectuer des inscriptions en ligne.

#### **Scénario :**

- 1) Le system affiche le formulaire d'inscription.
- 2) L'internaute saisie les informations nécessaires nom, prénom etc...
- 3) L'internaute confirme son formulaire.
- 4) Le system affiche la réussite d'inscription.

#### **Scénario alternatif :**

L'internaute n'a pas rempli certains champs obligatoires. Le system les détecte et lui propose de les remplir à nouveau.

#### **Exception :**

L'internaute décide d'annuler l'inscription, il abandonne.

### ❖ *Cas d'utilisation« consulter les menus»*

Objectif : donner la possibilité de consulter différent repas qui existe dans le site.

#### **Scénario :**

- 1) Le system affiche la liste des repas.
- 2) Le client consulte les menus.

#### **Exception :**

Si le client n'est pas intéressé par le menu. Il quitte la page

### ❖ *Cas d'utilisation « Contacter le service du restaurant »*

Objectif : permet aux utilisateurs de contacter facilement le service de restaurant pour informer sur la qualité de ses offres où renseigner sur un besoin.

#### **Scénario :**

- 1) Le system affiche le bouton qui permet le contact « CONTACT ».
- 2) Le client appuyé sur le bouton « CONTACT ».
- 3) Le client écrire le message et l'envoie au service restaurant.

#### **Scénario alternatif :**

Le client oublier de saisie un champ. Le service propose de remplir des informations obligatoires (mail, mot de passe si les informations correctes alors reprendre le scénario aux point 1.

**Exception** : si l'information n'est pas remplie alors, l'opération échouer.

### ❖ *Cas d'utilisation « authentification »*

Objectif : permet d'identifier le membre du site pour pouvoir accéder à des services privilégiés réservés uniquement aux internautes préalablement inscrits comme membre.

#### **Scénario :**

- 1) Le client saisit son nom d'utilisateur et mot de passe.
- 2) Le client valide la saisie.
- 3) Le system confirme l'identifiant.
- 4) Le system ouvre la session client.

#### **Scénario alternatif :**

S'il n'est pas un membre le system lui demande de s'inscrire.

Le membre saisit un nom d'utilisateur ou un mot de passe incorrect le system affiche un message d'erreur et revient à l'étape 1.

#### **Exceptions :**

Si le membre se trompe alors le system refuse sa demande

### ❖ *Cas d'utilisation « créer commande »*

Objectif : permet aux clients de faire le choix des repas (boisson, plat, dessert).

#### **Scénario :**

- 1) Le membre choisit les composants de son repas (boisson, plats, dessert).
- 2) Le membre saisit la quantité pour chaque composant.
- 3) Le client valide son choix.

#### **Scénario alternatif :**

S'il n'est pas un membre le system lui demande de s'inscrire dans le site, ou authentifié s'il est un membre.

### ❖ *Cas d'utilisation « consulter facture »*

Objectif : permet aux clients de consulter la facture.

#### **Scénario :**

- 1) Le membre crée une commande
- 2) Le membre valide la commande et consulte les recommandations faites par le système.
- 3) Le système affiche la facture et le mode de paiement.

### Scénario alternatif :

S'il n'est pas un membre le system lui demande de s'inscrire ou authentifié s'il est un membre.

Le membre créer une commande et le system affiche la facture de client.

#### ❖ *Cas d'utilisation* « Mis à jour Menu »

Objectif : permet à l'administrateur de faire une mise à jour de repas par un ajout ou suppression des repas.

### Scénario :

- 1) Admin choisi la mise à jour des repas.
- 2) Admin remplir les informations de repas.
- 3) Le system vérifie les informations et enregistrer les modifications.

#### ❖ *Cas d'utilisation* « consulter messagerie »

Objectif : permet à l'administrateur de site de consulter les différents messages envoyés par les internautes ou bien par les clients, et supprimer les messages indésirables.

### Scénario :

- 1) Admin accéder au courrier.
- 2) Admin lire et répondre aux messages.

### III.6.2 Diagramme d'activité

Le diagramme d'activité représente les règles d'enchaînement des actions et décisions au sein d'une activité. Il peut également être utilisé comme alternative au diagramme d'états pour décrire la navigation dans un site Web

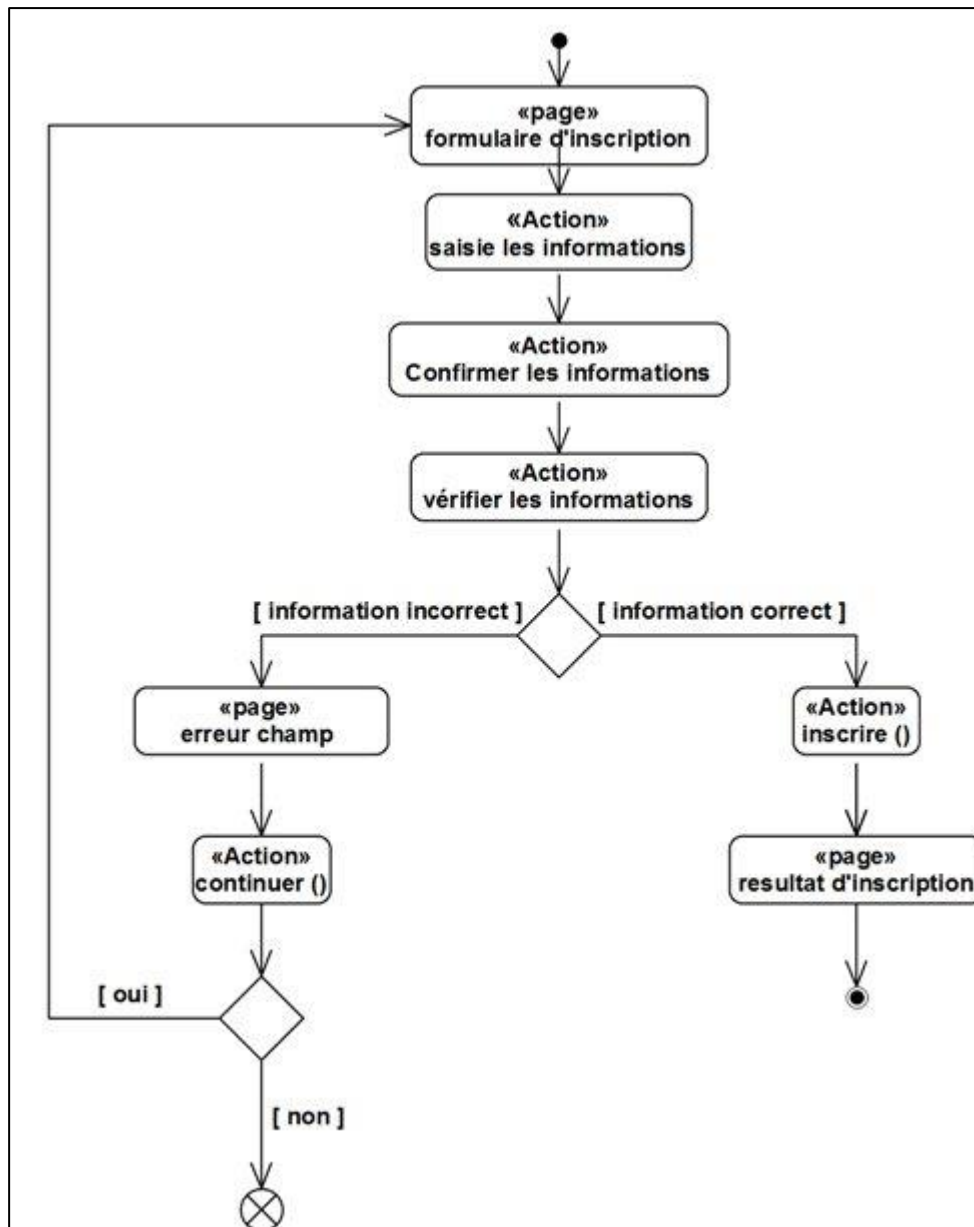


Figure III.10 : Diagramme d'activité cas d'utilisation inscription

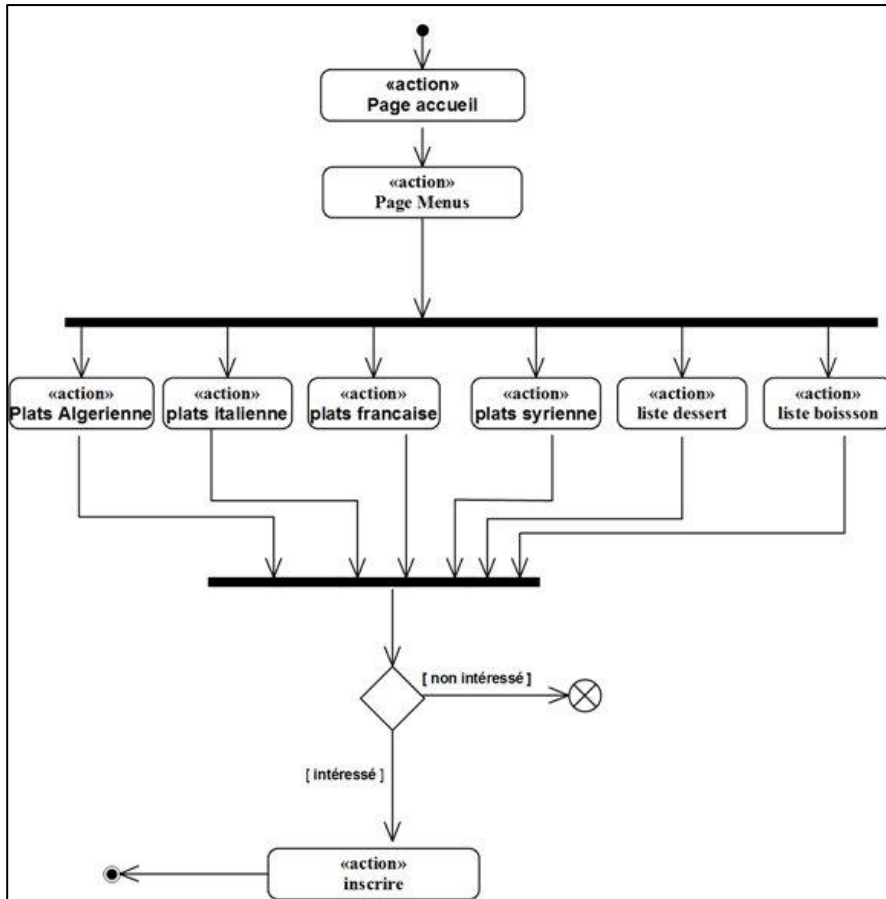


Figure III.11 : Diagramme d'activité cas d'utilisation Consulter le menu

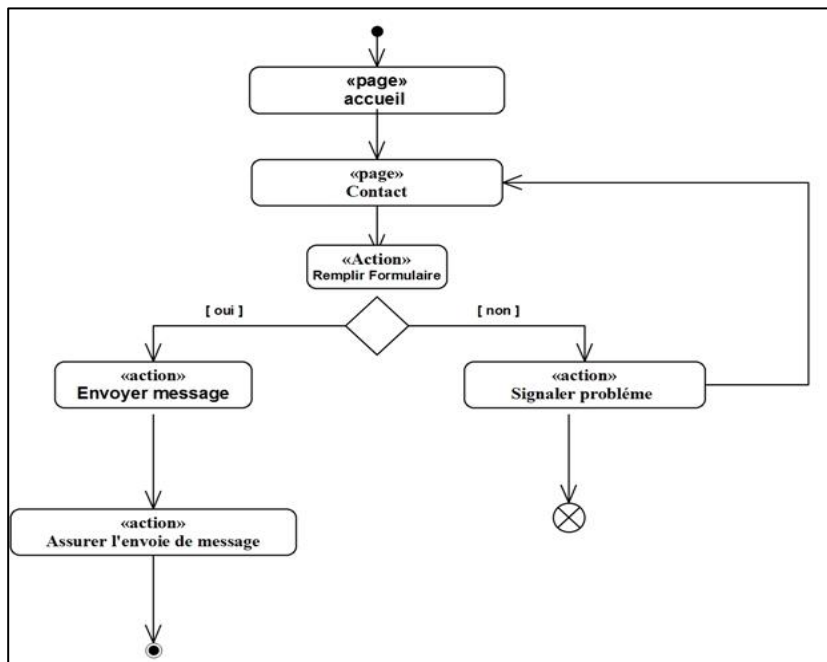


Figure III.12 : Diagramme d'activité cas d'utilisation contacter le service du restaurant

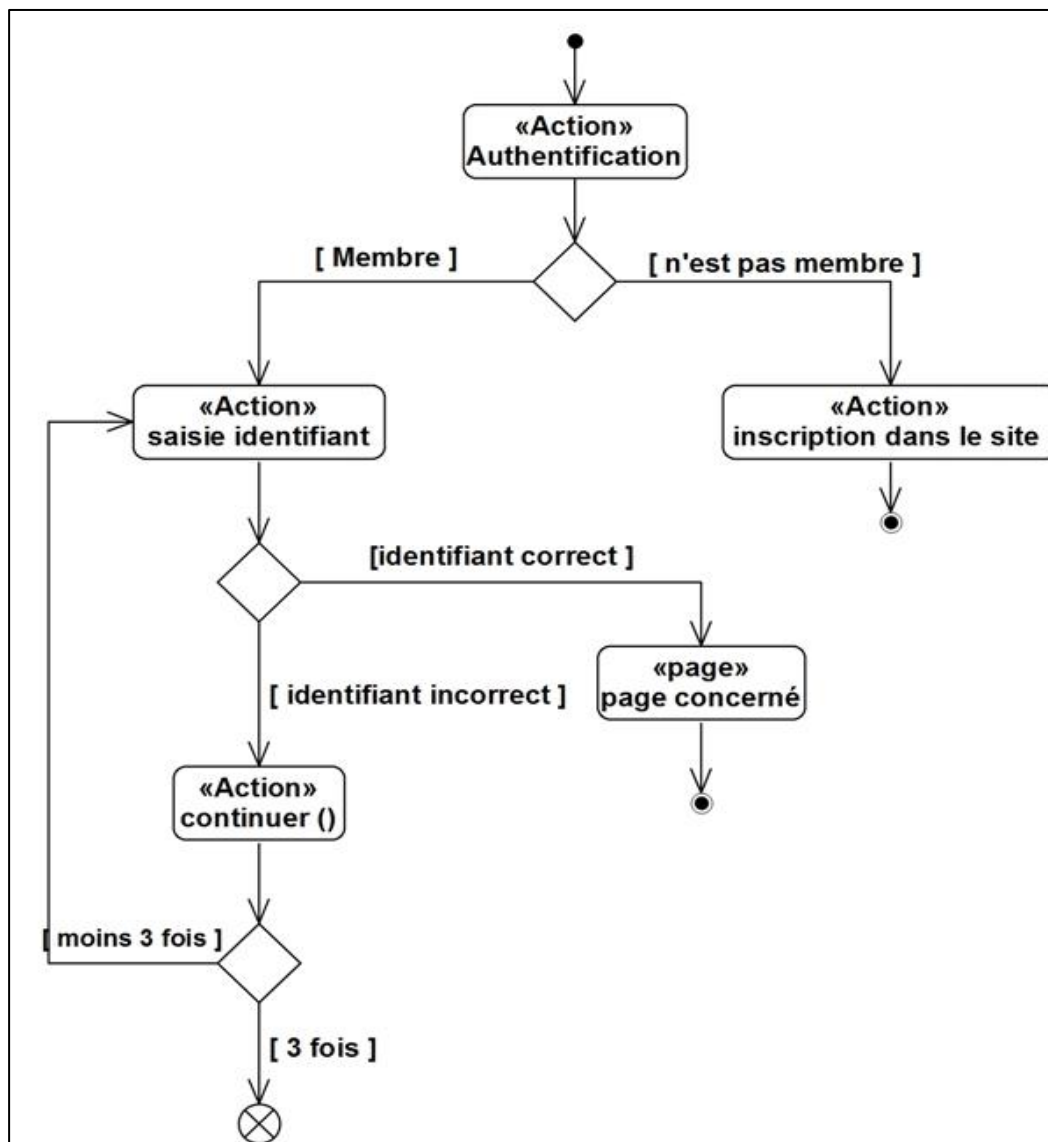


Figure III.13 : Diagramme d'activité cas d'utilisation authentifier

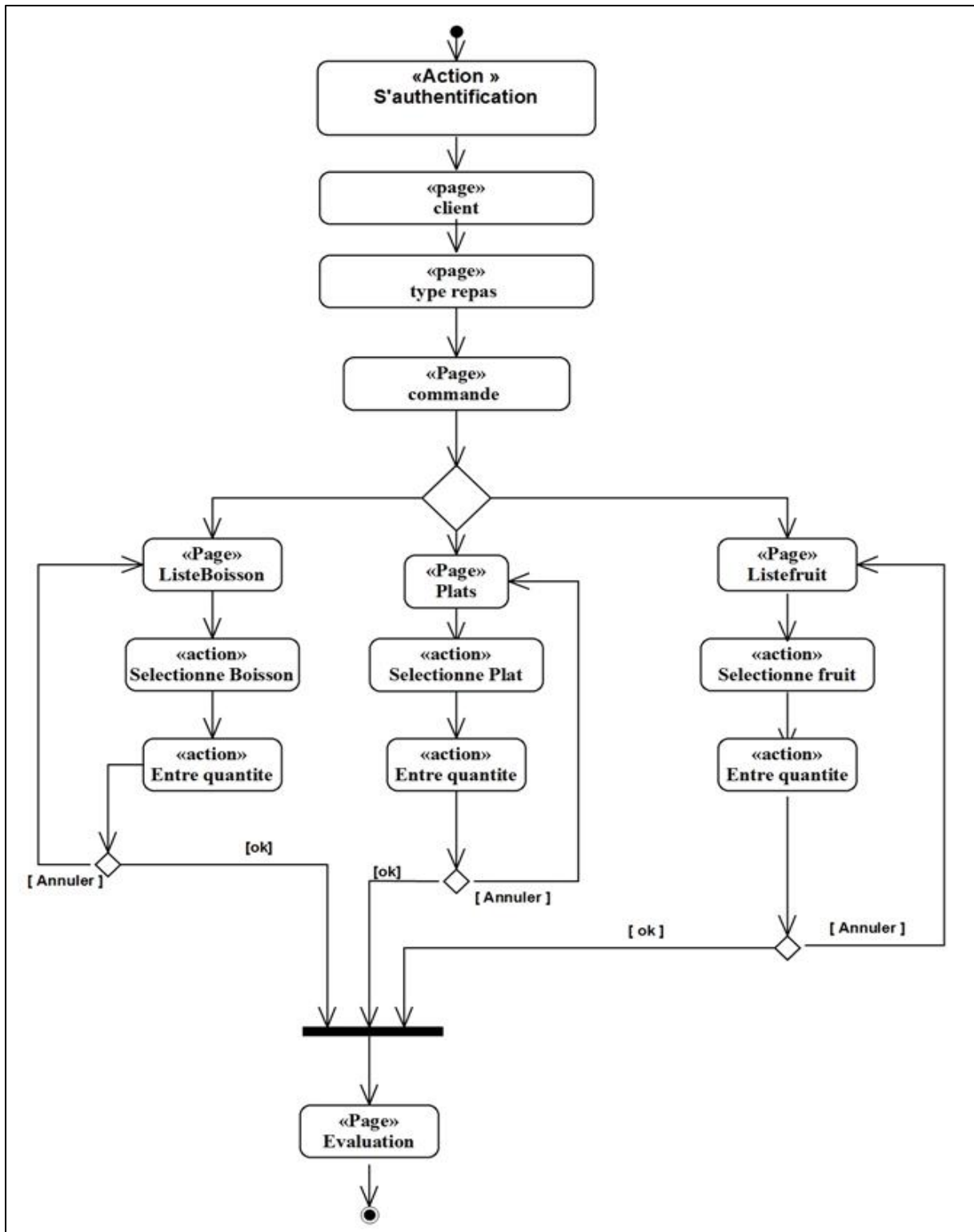


Figure III.14 : Diagramme d'activité cas d'utilisation créer une commande

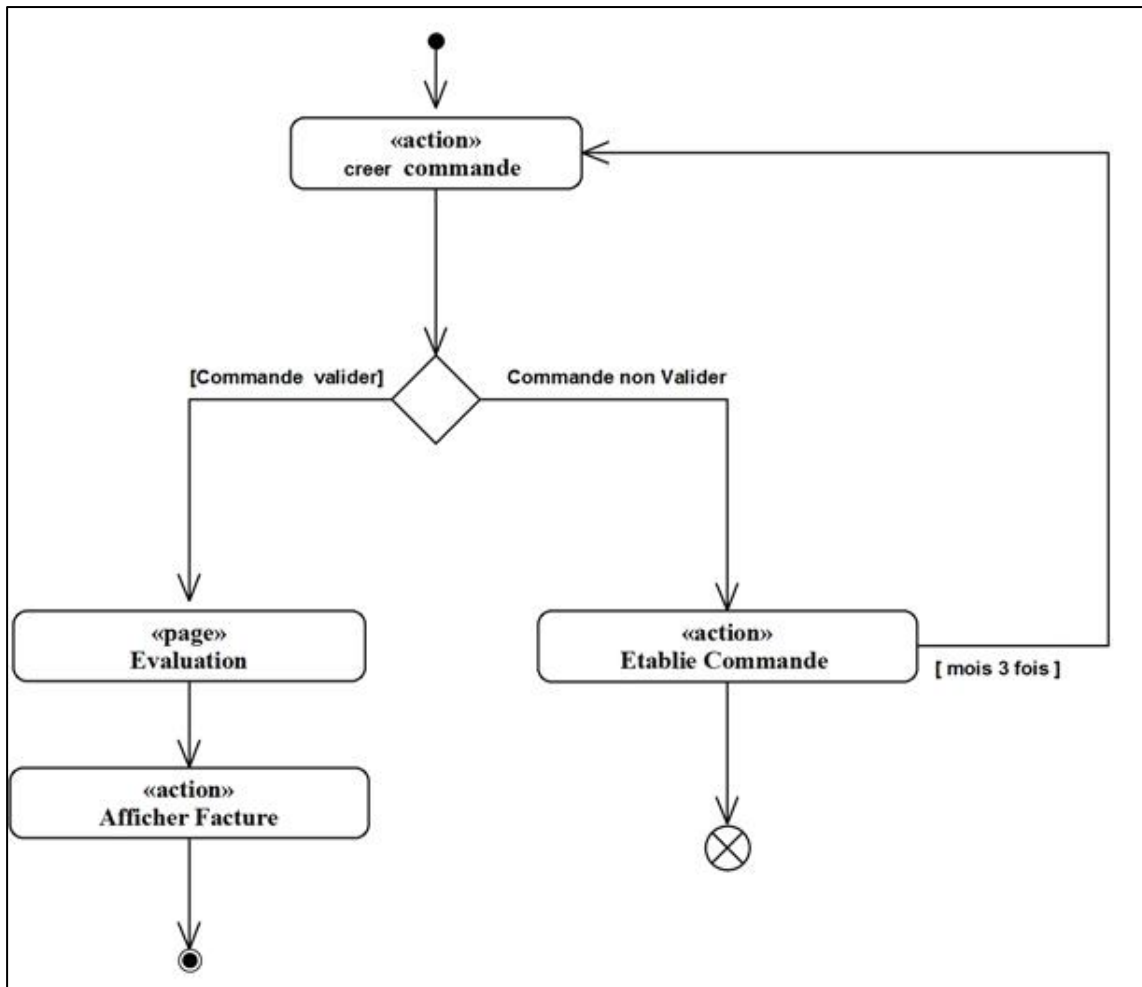


Figure III.15 : Diagramme d'activité cas d'utilisation consulter facture

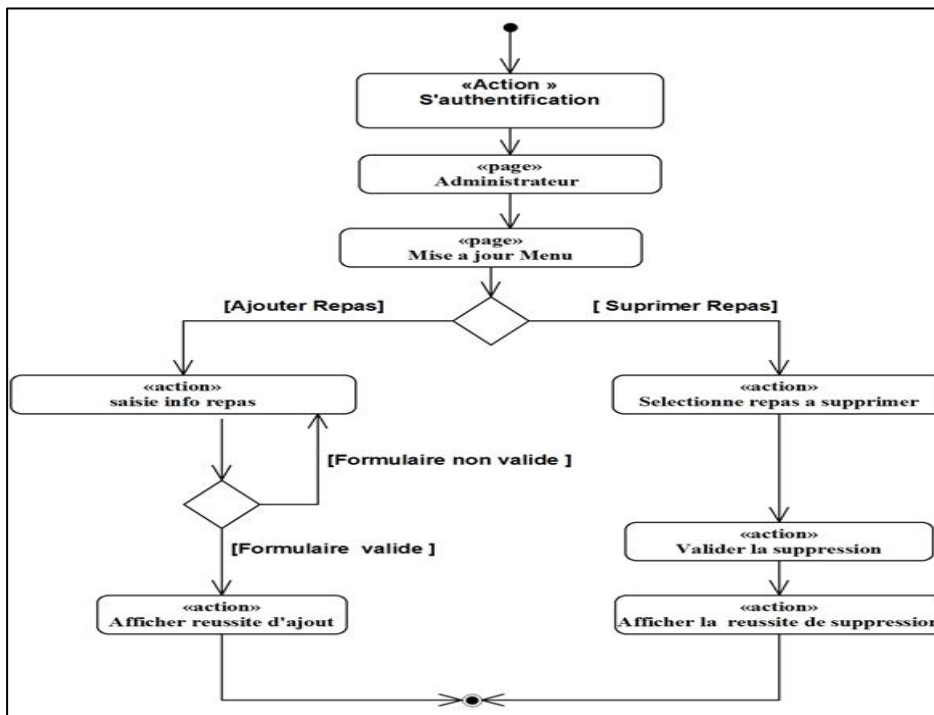


Figure III.16 : Diagramme d'activité cas d'utilisation mis à jour menu



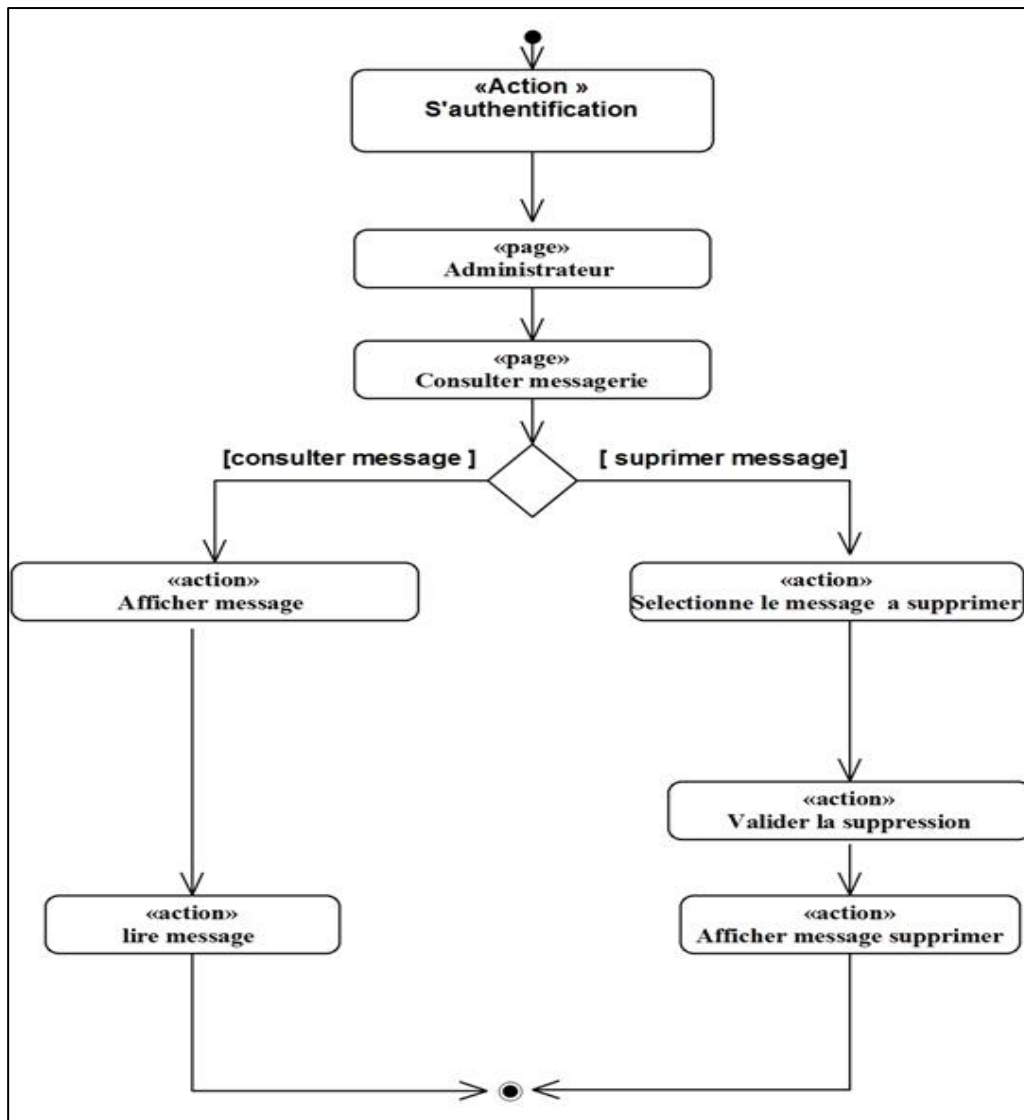


Figure III.17 Diagramme d'activité cas d'utilisation consulter message

III.6.3 Diagramme de séquence :

Dans cette figure on a illustre les diagrammes de séquences des cas d'utilisation suivant :

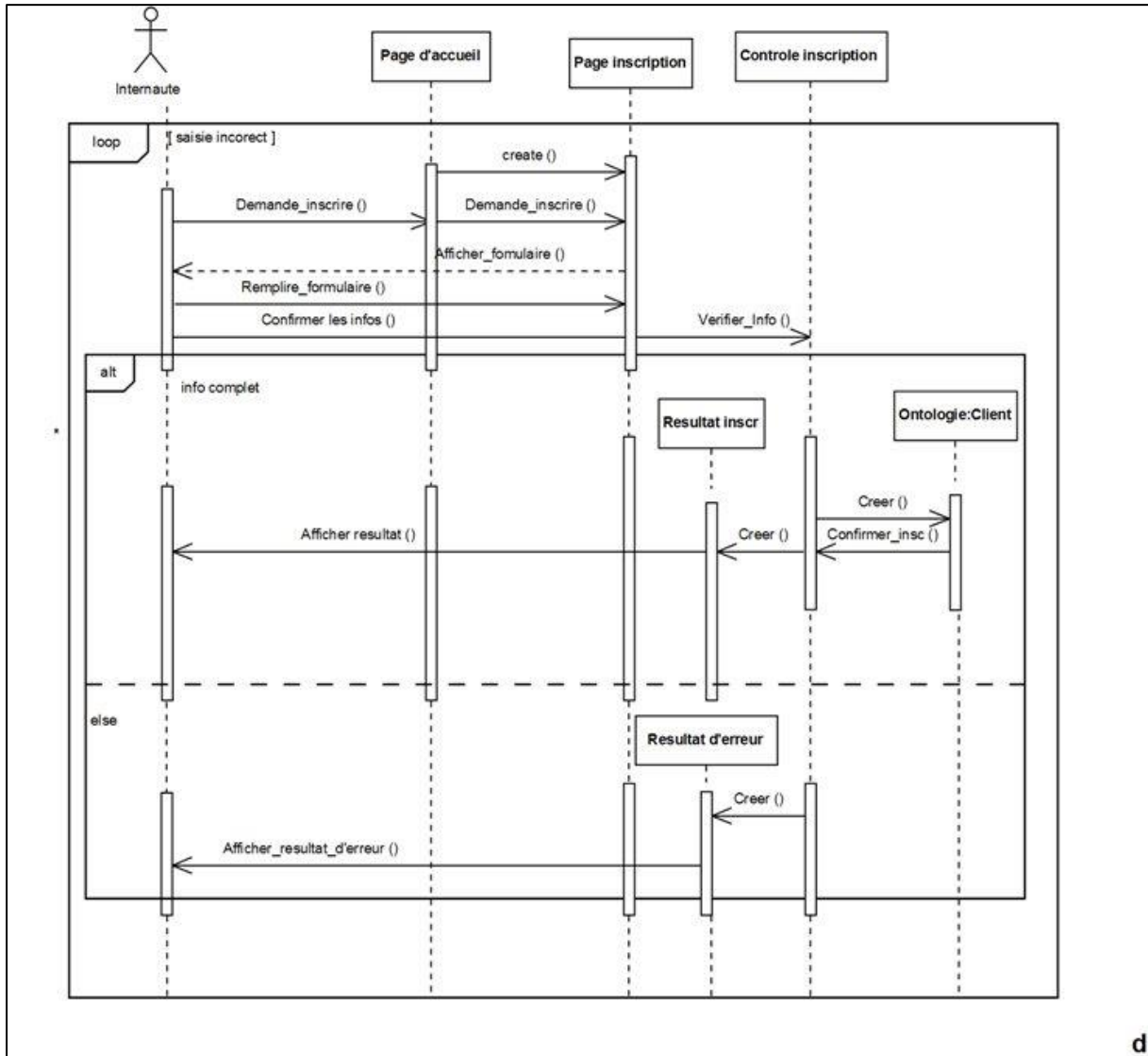
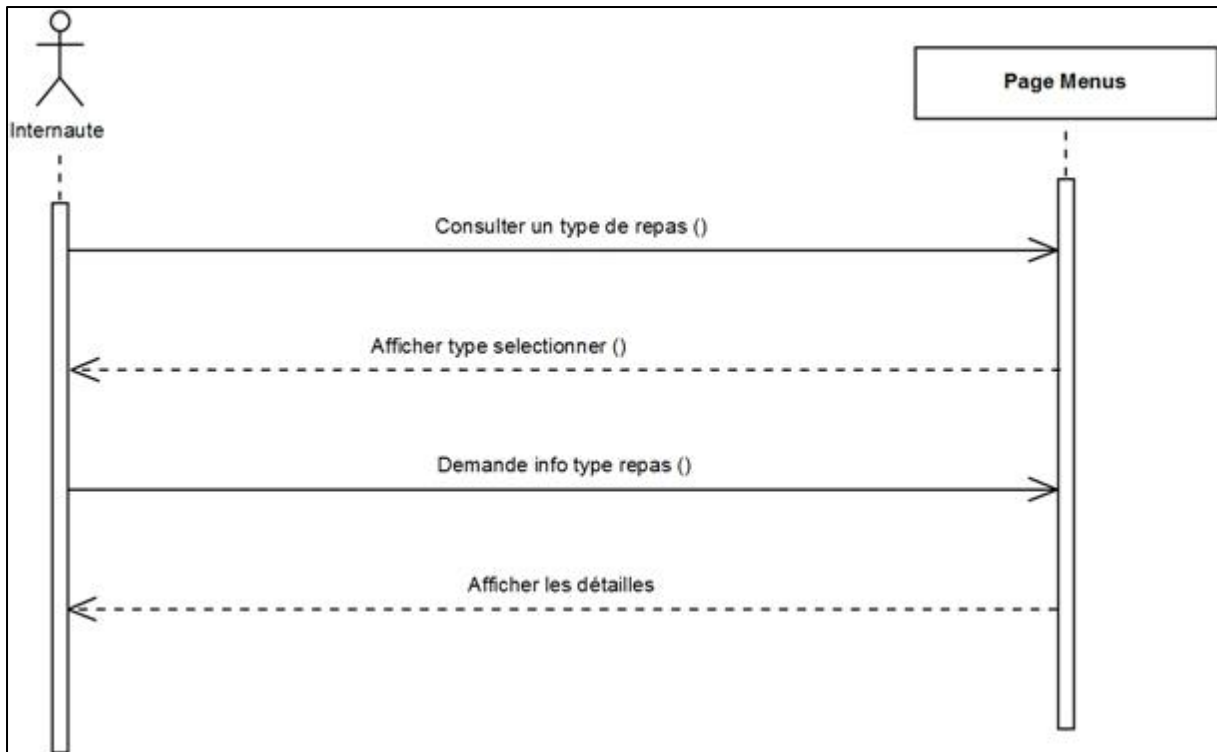


Figure III.18 Diagramme de séquence du cas d'utilisation inscription.



**Figure III.19** Diagramme de séquence du cas d'utilisation « consulter le menu ».

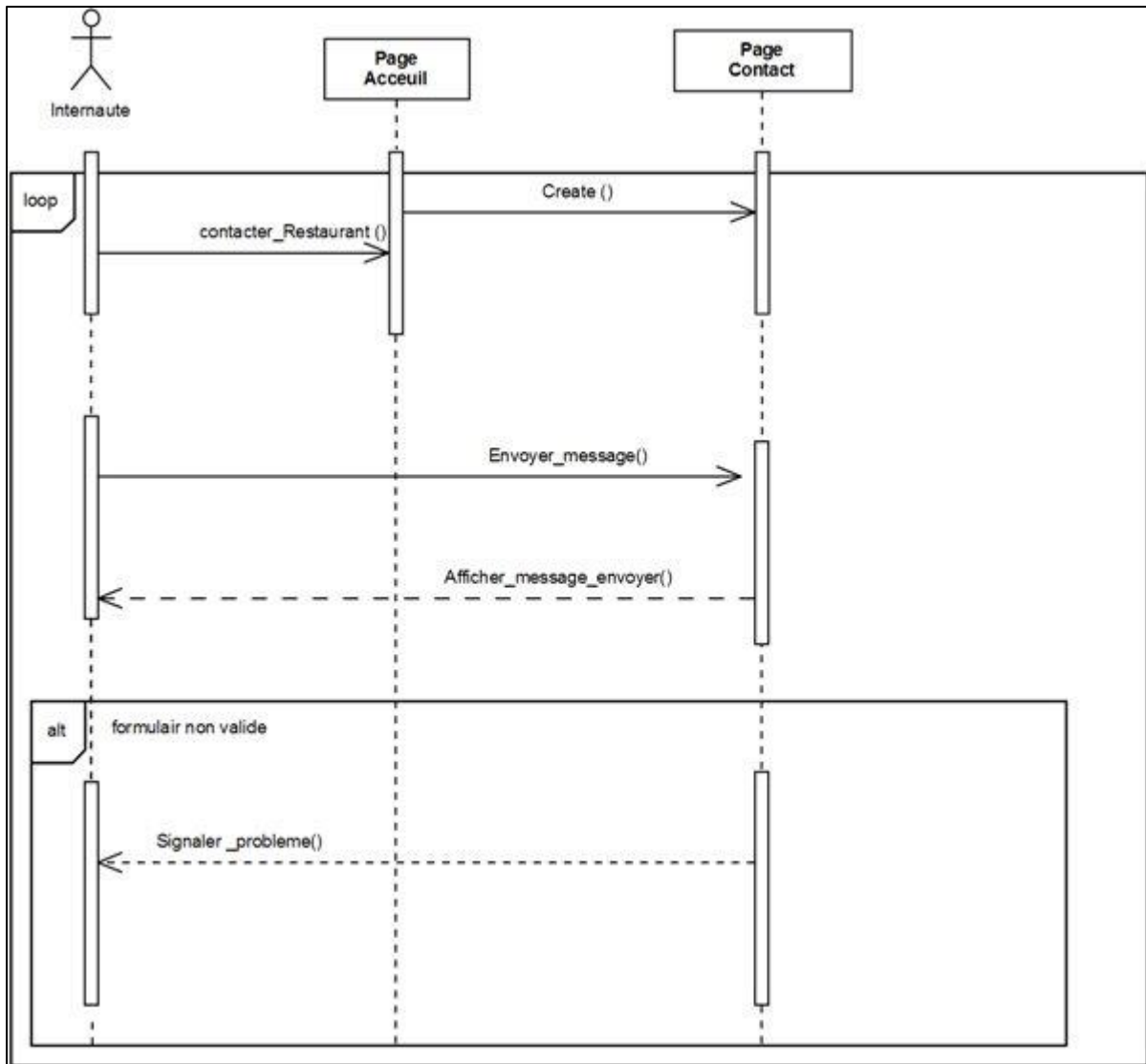


Figure III.20 Diagramme de séquence du cas d'utilisation contacter le service du restaurant.

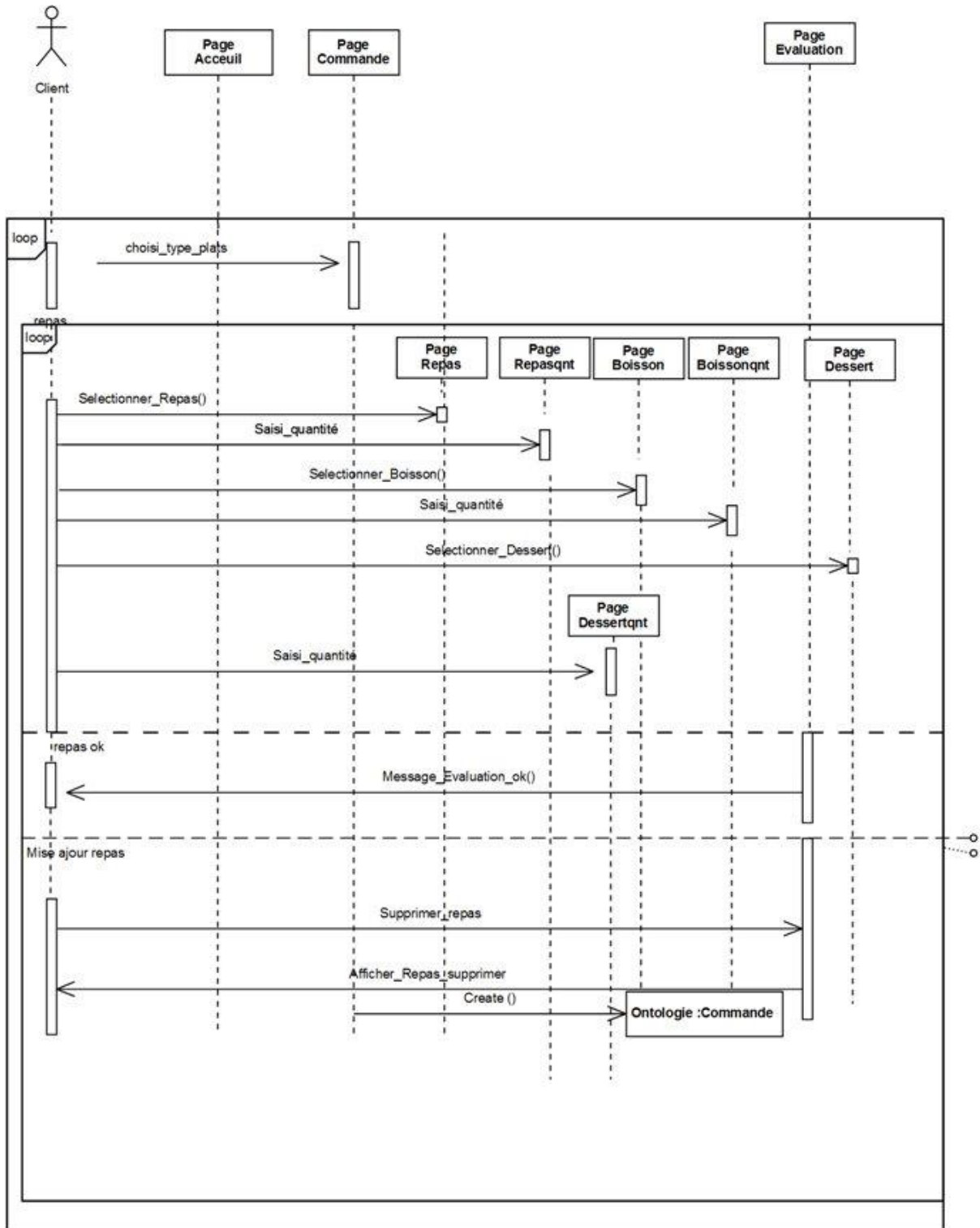


Figure III.21 Diagramme de séquence du cas d'utilisation « créer une commande ».

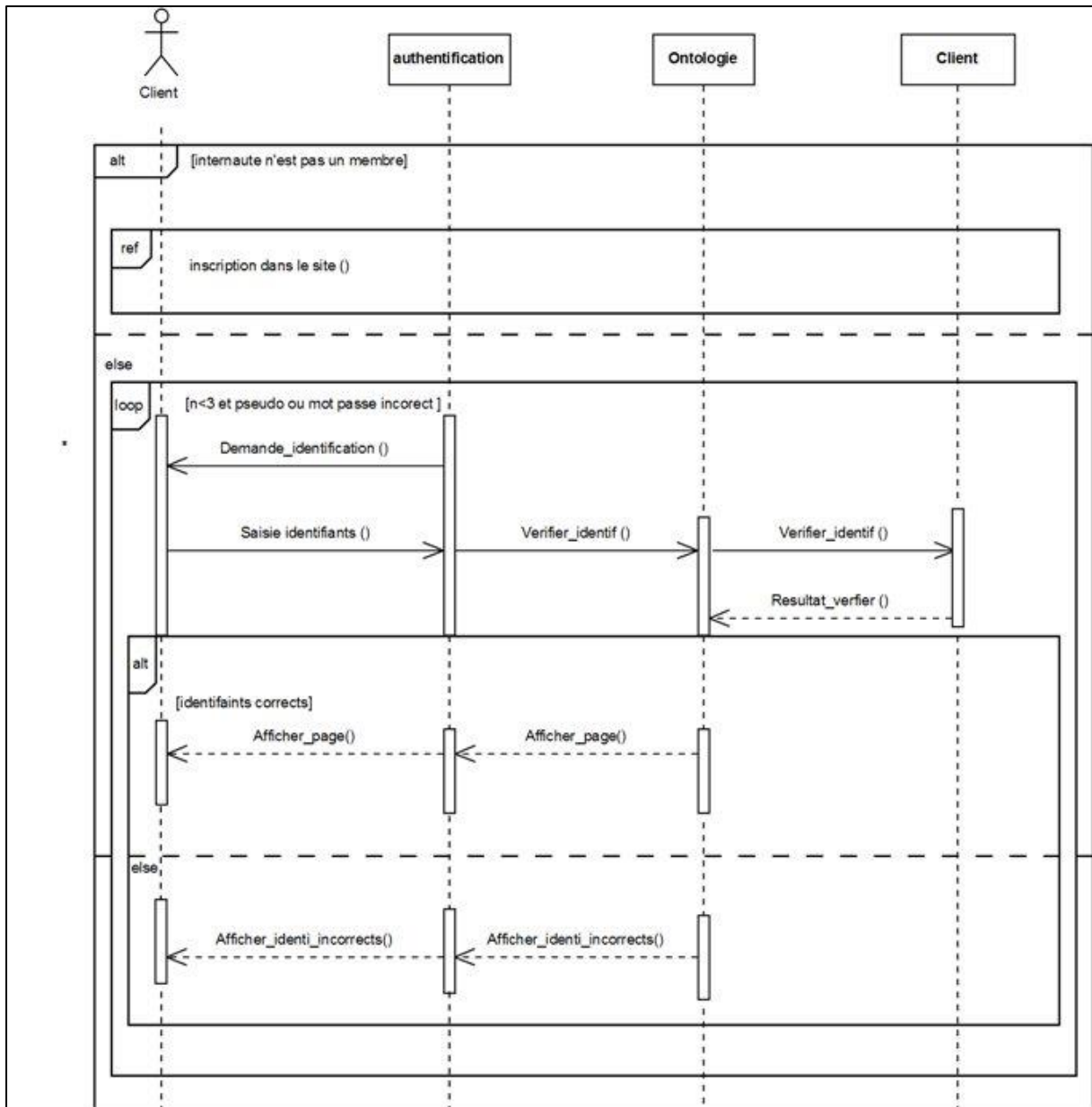


Figure III.22 Diagramme de séquence du cas d'utilisation authentifier.

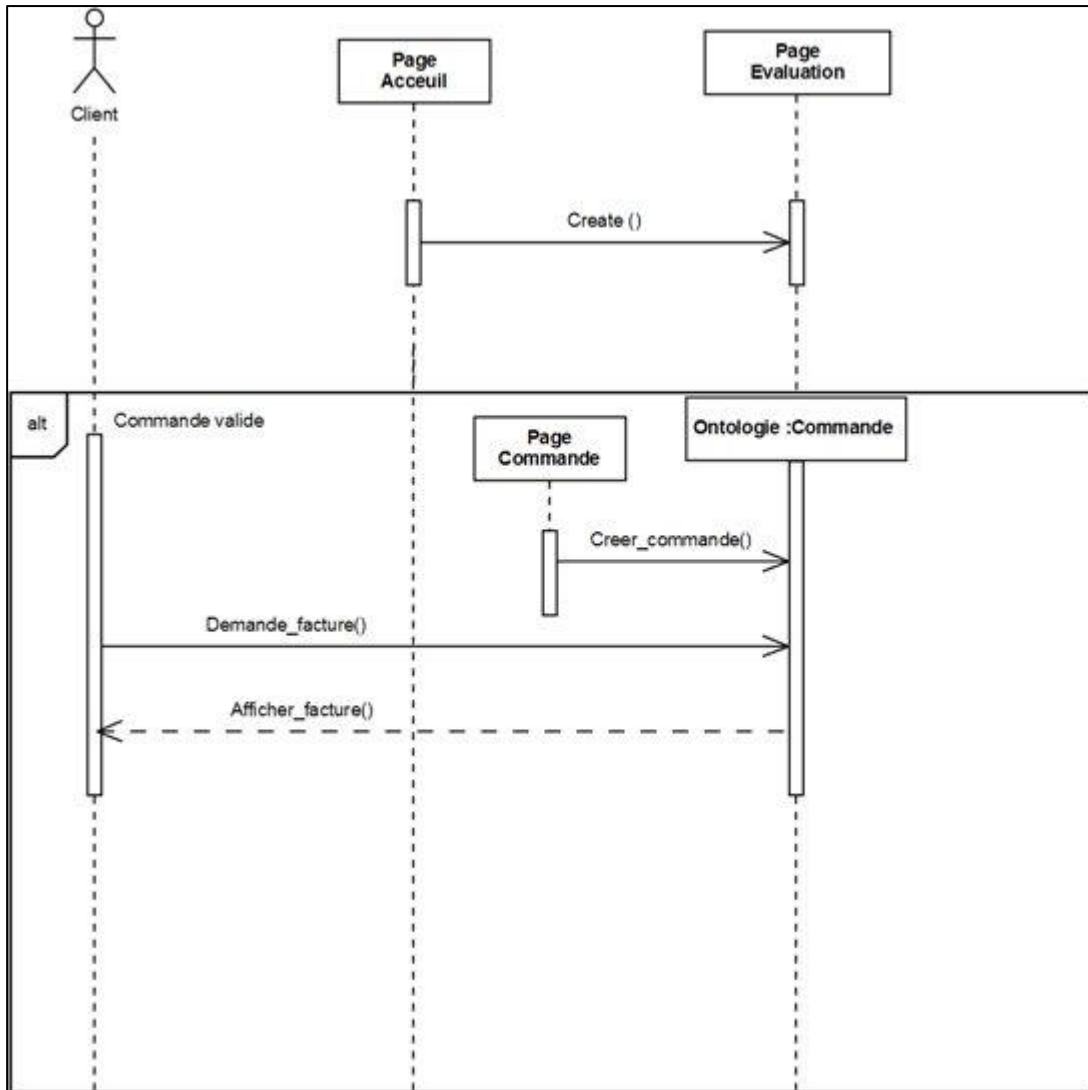


Figure III.23 Diagramme de séquence du cas d'utilisation consulté facture.

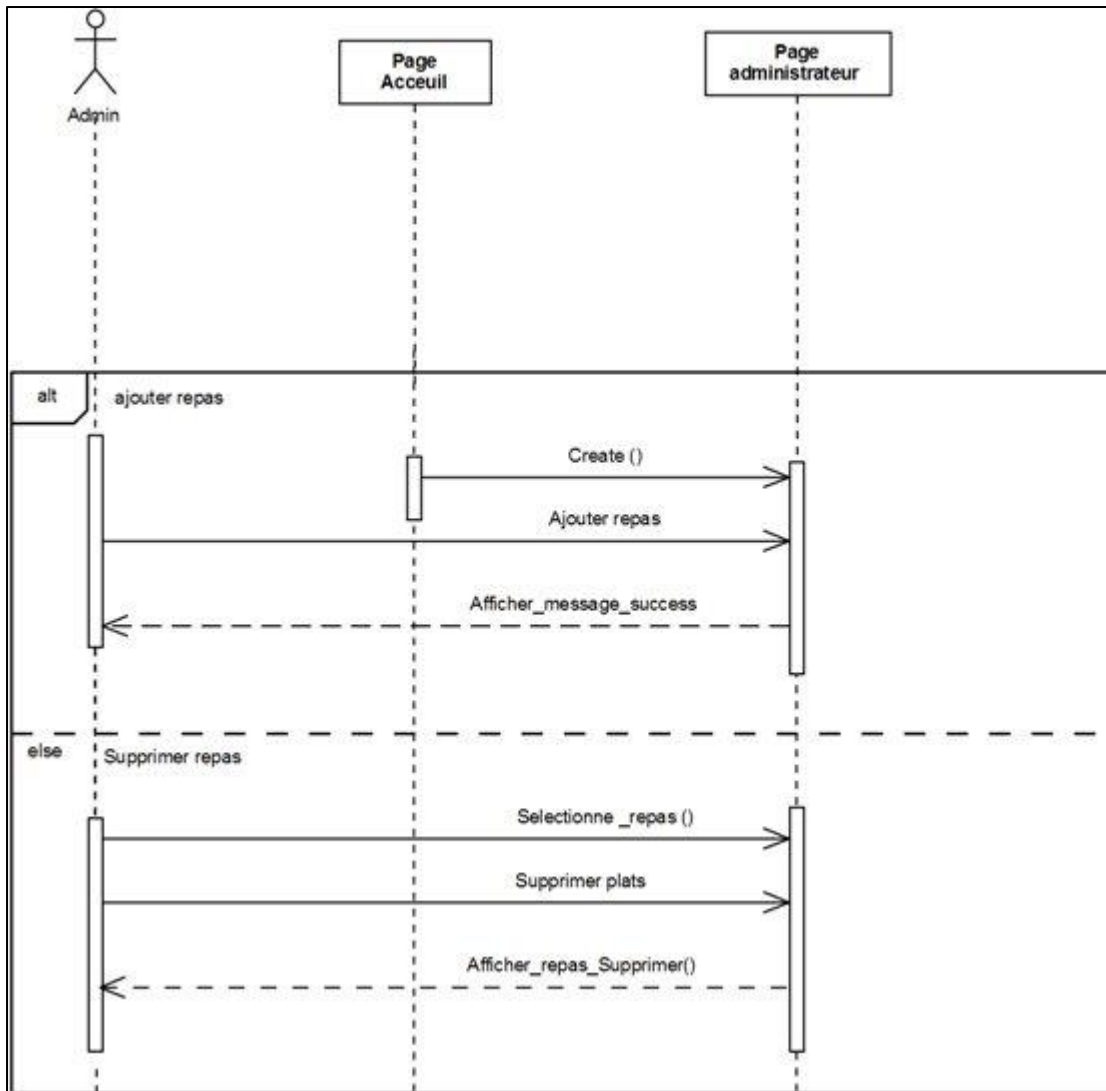
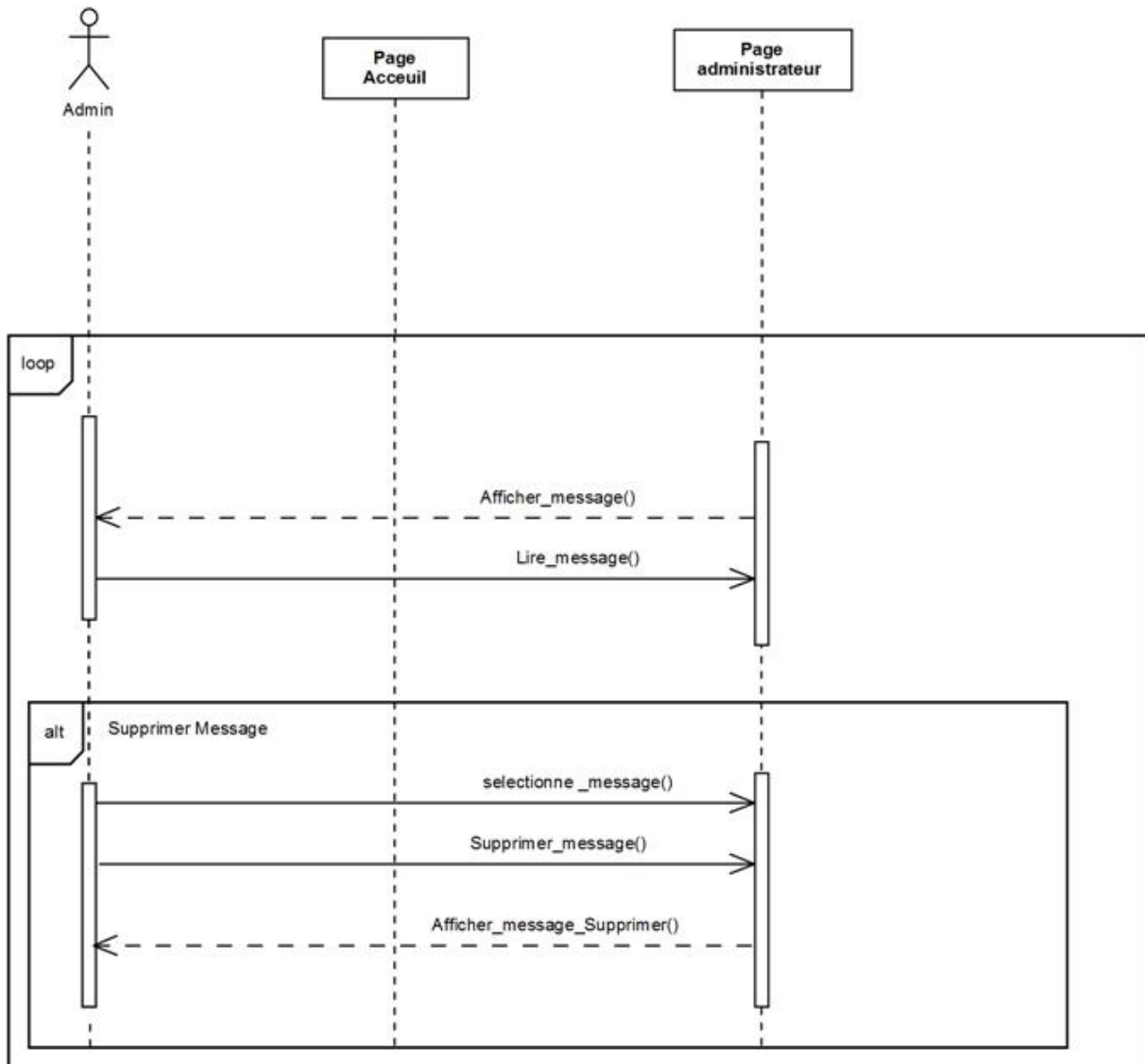


Figure III.24 Diagramme de séquence du cas d'utilisation mis à jour menu.





**Figure III.25** Diagramme de séquence du cas d'utilisation consulter message.

Donc, les différents diagrammes décrits dans ce chapitre servent à la description des fonctions de notre site

### III.7 Conclusion

Au cours de cette analyse nous avons donné une vue générale sur l'UML. Nous avons montré le principe de fonctionnement et les relations qui existent entre les différents acteurs et leurs interactions avec le système. Nous avons aussi expliqué les diagrammes de cas d'utilisation, d'activité, de séquence, qui est nécessaires pour notre projet

Nous avons limités notre étude à ces trois diagrammes car les données de notre système vont être décrits et exploités en utilisant une ontologie de domaine.

## **Chapitre IV : Développement d'ontologie**

### IV.1 Introduction

Le présent chapitre présente la partie qui concerne la construction d'une ontologie pour notre restaurant de futur, nommé « Onto-Resto ».

Dans la 1<sup>ère</sup> partie, nous décrirons le processus que nous avons suivi pour la construction de l'ontologie. Ce processus est composé de cinq- phases : spécification des besoins, conceptualisation, formalisation, opérationnalisation et évaluation. En effet, ces différentes phases sont inspirées à partir de certaines méthodologies utilisées par plusieurs travaux qui existent déjà.

Dans la deuxième partie de ce chapitre nous montrerons comment ce processus a été utilisé pour la conception de l'Onto-Resto. Nous verrons alors, comment l'ontologie a été conceptualisée, en utilisant un ensemble de représentations intermédiaires, semi-formelles de METHONTOLOGY. Le résultat de cette phase est une ontologie conceptuelle. Pour la formaliser, nous avons fait recours à la logique de description qui possède une sémantique claire et procure des services d'inférences et des mécanismes de raisonnement puissants.

### IV.2 Le processus suivi pour la construction de l'Onto-Resto

Bien qu'il n'existe pas une méthode ou méthodologie standard pour construire une ontologie, et que chaque équipe utilise la méthode qu'elle propose, nous avons utilisé un processus basé sur certains travaux existants.

Tout d'abord, Nous appliquons tout au long du processus que nous avons adopté pour le développement de notre ontologie les conseils suivants [19] :

- ❖ Il n'y a pas qu'une seule façon correcte pour modéliser un domaine, il y a toujours des alternatives viables.
- ❖ Le développement d'une ontologie est nécessairement un processus itératif.
- ❖ Les concepts dans une ontologie doivent être très proches des objets (physiques ou logiques) et des relations dans le domaine d'intérêt. Fort probablement, ils sont des noms (objets) et des verbes (relations) dans les phases qui décrivent le domaine.

#### IV.2.1 Description des principes d'ingénierie ontologique et leur application

Nous appliquons les principes de construction des ontologies, permettant la construction et l'évaluation d'ontologie, dans la réalisation des différentes phases du processus de construction d'ontologies.

- ❖ **Clarté/Objectivité:** L'ontologie doit fournir la signification des termes définis en fournissant des définitions <objectives> ainsi qu'une documentation en langage naturel.
- ❖ **Complétude/ Perfection:** Une définition complète (exprimée par des conditions nécessaires et suffisantes) est préférable à une définition partielle (définie seulement par des conditions nécessaires).
- ❖ **Cohérence:** Une ontologie cohérente doit permettre des inférences conformes à ces définitions. Les axiomes doivent être consistants.
- ❖ **Extensibilité monotone maximale:** De nouveaux termes généraux et spécialisés devraient être inclus dans l'ontologie d'une façon qui n'exige pas la révision des définitions existantes.
- ❖ **Biais de codage minimal:** l'ontologie doit être conceptualisée indépendamment de tout langage d'implémentation. Il faut minimiser les choix de représentations faits pour permettre le partage des connaissances, contenues dans l'ontologie, entre différentes applications utilisant des langages de représentation différents.

- ❖ **Engagements ontologiques minimaux:** l'ontologie doit faire aussi peu d'hypothèses (réclamations) que possible sur le monde qu'elle modélise. Elle doit contenir un vocabulaire partagé mais ne doit pas être une base de connaissances comportant des connaissances supplémentaires sur le monde à modéliser. Autrement dit, elle doit choisir de représenter un point de vue en particulier sur le domaine qui l'intéresse. [20]
- ❖ **Distinction ontologique:** les classes dans une ontologie devraient être disjointes. Le critère utilisé pour isoler le noyau de propriétés considérées comme invariables pour une instance d'une classe est appelé le critère d'Identité. Ce critère permet de spécialiser les concepts tant qu'il est possible. [21]
- ❖ **Modularité:** il faut minimiser les couplages entre les modules.
- ❖ **Distance sémantique minimale :** Il faut minimiser la distance entre les concepts enfants de même parents. Les concepts similaires sont groupés et représentés comme des sous-concepts d'un concept, et devraient être définis en utilisant les mêmes primitives, considérant que les concepts qui sont moins similaires sont représentés plus loin dans la hiérarchie.
- ❖ **Normaliser les noms :** Il faut normaliser les termes chaque fois que c'est possible.

Selon notre vue, ces principes sont appliqués de la façon suivante (tableau 2):

Spécification	Conceptualisation	Formalisation
-Engagements ontologiques minimaux	-Clarté/Objectivité -Complétude/ Perfection -Cohérence -Biais de codage minimal -Distinction ontologique -Modularité minimale -Distance sémantique minimale -Normaliser les noms	-Cohérence -Extensibilité monotone maximale

**Tableau 2 : Application des principes d'IO.**

### IV.3 Les phases du processus de construction de l'ontologie

les phases qui permettent la construction d'une ontologie opérationnelle partant des connaissances brutes d'un domaine sont :

- Spécification
- Conceptualisation
- Formalisation

- Opérationnalisation
- Évaluation.

### IV.3.1. Spécification

Pour commencer le développement de l'ontologie, une première importante étape doit être effectuée. Elle consiste à établir un document informel de spécification de besoins écrit dans un langage naturel. Ce document décrit :

- Le domaine de connaissance qui sera représenté par l'ontologie ;
- L'objectif de l'ontologie à créer pour le domaine considéré ;
- Les utilisateurs futurs de l'ontologie ;
- Les sources d'informations desquelles les connaissances seront obtenues. Ils sont de nature différentes et variées, par exemple : les interviews avec les experts du domaine, les documents techniques (publications scientifiques, livres), les observations, les ontologies existantes qui peuvent être réutilisés... etc. ;
- La portée de l'ontologie : déterminer la liste des termes candidat du domaine à travers l'analyse des sources d'informations relatif au domaine.

### IV.3.2. Conceptualisation

Sans doute, l'étape de conceptualisation est la plus importante dans le processus de développement de l'ontologie. Elle mérite une attention particulière car elle détermine le reste de la construction de l'ontologie.

Cette phase est inspirée de la méthodologie METHONTOLOGY [22] parce qu'elle est spécifiée de manière très détaillée. Elle consiste à organiser et à structurer, à partir des sources d'informations, les connaissances du domaine en utilisant un ensemble de représentations intermédiaires, semis formels, sous forme de tableaux et graphes, indépendamment du mécanisme de formalisation utilisé pour représenter l'ontologie. A la fin nous obtenons une ontologie conceptuelle (model conceptuel).

Les principales tâches suivantes sont réalisées :

- ❖ **Tâche1** : construire le glossaire des termes en identifiant tous les termes inclus dans l'ontologie, leurs définitions en langage naturel, leurs synonymes et leurs acronymes ;
- ❖ **Tâche2** : construire la taxonomie de concepts. Elle est réalisée à travers un choix d'une stratégie d'identification de concepts. Trois stratégies d'identification des concepts ont été proposées par [23] :

- **Approche ascendante (bottom-up strategy)** : les concepts les plus spécifiques sont identifiés, par la suite, ils sont généralisés en concepts plus abstraits.
- **Approche descendante (top-down strategy)**: les concepts les plus abstraits sont identifiés, par la suite, ils sont spécialisés en plus spécifiques.
- **Approche centrifuge (middle-out strategy)**: les concepts les plus importants sont identifiés (centraux), par la suite, ils sont généralisés et spécialisés comme il est nécessaire.

En fait, aucune de ces trois stratégies n'est fondamentalement meilleure que les autres. L'approche à adopter dépend fortement du point de vue personnel du concepteur d'ontologie sur le domaine.

- ❖ **Tâche3** : construire le diagramme de relations binaires où les relations entre les concepts de l'ontologie sont identifiées.
- ❖ **Tâche4** : construire le dictionnaire de concepts qui inclut principalement les instances de chaque concept, leurs attributs et leurs relations.
- ❖ **Tâche5** : décrire en détail dans une table de relations binaires les relations apparues dans le diagramme de relation binaires.
- ❖ **Tâche6** : décrire les attributs apparus dans le dictionnaire de concepts en spécifiant leurs contraintes dans une table d'attributs.
- ❖ **Tâche7** : spécifier les axiomes sur les concepts dans une table d'axiomes logiques.
- ❖ **Tâche8** : décrire les instances des concepts dans une table d'instances.

### IV.3.3. Formalisation

Le processus de construction d'ontologie se poursuit par une étape de formalisation. Elle consiste à formaliser l'ontologie conceptuelle obtenue à l'étape précédente afin de faciliter sa représentation ultérieure dans un langage formel et opérationnel. Nous utilisons comme formalisme de représentation de connaissance la logique de description en s'appuyant sur la syntaxe de type SHOIN(D). Ce choix est motivé par le fait que SHOIN(D) est une logique de description très expressive, qui offre un certain nombre de constructeurs permettant de représenter des concepts (aussi appelés classes) d'un domaine d'application. Ainsi, elle est la logique sur laquelle se base le langage OWL-DL qui est utilisé par la suite pour la codification de l'ontologie. Le résultat de cette étape est une ontologie formelle.



### a) Les constructeurs de la logique SHOIN(D) :

La logique propose les constructeurs : S désigne la logique ALC additionnée de R+ (ensemble de rôles transitifs), H permet d'exprimer l'inclusion des rôles, O qui permet la description de concepts par énumération d'individus nommés, I permet la description des rôles inverses, N permet d'exprimer des contraintes de cardinalités sur les rôles et une dernière extension, symbolisée par la lettre (D) , ajoute le support de types de données (voir le tableau 3).

$\Delta^I$  : est un ensemble arbitraire non vide d'individus ;

$\Delta_D^I$  : est un second domaine d'interprétation disjoint avec  $\Delta^I$  , représente l'ensemble des valeurs de types de données prédéfinis tels que les entiers, les chaînes de caractères, etc.

$D$ : est un type de données tel que  $D^D \subseteq \Delta_D^I$  ;

$I$  : est une fonction d'interprétation que fait correspondre à un concept  $C$  un sous ensemble

$C^I \subseteq \Delta^I$  , à une relation  $R$  un sous ensemble de  $R^I \subseteq \Delta^I * \Delta^I$  , à un attribut  $U$  un sous ensemble  $U^I$

$\subseteq \Delta^I * \Delta_D^I$  , à une instance  $o$  :  $o^I \in \Delta^I$  et à un littérale  $t$  :  $t^I \in \Delta_D^I$  .

Nom du constructeur	Syntaxe	Sémantique
Top (Thing)	$\top$	$\top^I = \Delta^I$
Buttom (Nothing)	$\perp$	$\perp^I = \Phi$
Le concept atomique (primitif)	$A$	$A^I \subseteq \Delta^I$
Le nom de classe	$C_1$	$C_1^I \subseteq \Delta^I$
L'intersection (conjonction)	$C_1 \cap C_2$	$(C_1 \cap C_2)^I = C_1^I \cap C_2^I$
L'union (disjonction)	$C_1 \cup C_2$	$(C_1 \cup C_2)^I = C_1^I \cup C_2^I$
La négation	$\neg C_1$	$(\neg C_1)^I = \Delta^I \setminus C_1^I$
Énumération	$\{o_1, o_2, \dots, o_N\}$	$\{o_1, o_2, \dots, o_N\}^I = \{o_1^I, o_2^I, \dots, o_N^I\}$
Le quantificateur existentiel	$\exists R_1 . C_1$	$(\exists R_1 . C_1)^I = \{x \mid \exists y . (x,y) \in R_1^I \text{ and } y \in C_1^I\}$
Le quantificateur universel	$\forall R_1 . C_1$	$(\forall R_1 . C_1)^I = \{x \mid \forall y . (x,y) \in R_1^I \text{ then } y \in C_1^I\}$
Restriction à une valeur	$\ni R_1 . o_1$	$(\ni R_1 . o_1)^I = \{x \mid (x, o_1^I) \in R_1^I\}$
Contraintes de cardinalité non quantifié Minimum/Maximum sur une relation	$(\geq n R_1)$ $(\leq n R_1)$ $(= n R_1)$	$(\geq n R_1)^I = \{x \mid \text{card } \{y \mid (x,y) \in R_1^I\} \geq n\}$ $(\leq n R_1)^I = \{x \mid \text{card } \{y \mid (x,y) \in R_1^I\} \leq n\}$ $(= n R_1)^I = \{x \mid \text{card } \{y \mid (x,y) \in R_1^I\} = n\}$
Le quantificateur existentiel	$\exists U_1 . D$	$(\exists U_1 . D)^I = \{x \mid \exists y . (x,y) \in U_1^I \text{ and } y \in D^D\}$
Le quantificateur universel	$\forall U_1 . D$	$(\forall U_1 . D)^I = \{x \mid \forall y . (x,y) \in U_1^I \text{ then } y \in D^D\}$
Contraintes de cardinalité non quantifié Minimum/Maximum sur un attribut	$(\geq n U_1)$ $(\leq n U_1)$ $(= n U_1)$	$(\geq n U_1)^I = \{x \mid \text{card } \{y \mid (x,y) \in U_1^I\} \geq n\}$ $(\leq n U_1)^I = \{x \mid \text{card } \{y \mid (x,y) \in U_1^I\} \leq n\}$ $(= n U_1)^I = \{x \mid \text{card } \{y \mid (x,y) \in U_1^I\} = n\}$
Le rôle inverse	$R_1^-$	$(R_1^-)^I = (R_1^I)^-$

Tableau 3 Syntaxe et sémantique de SHOIN(D)

Les concepts sont soit atomiques soit des expressions s'appuyant sur des constructeurs de classe. SHOIN(D) permet de définir les connaissances suivant deux niveaux de description : le niveau terminologique (T-Box) décrit les concepts et les rôles entre les concepts d'un domaine alors que le niveau assertionnel (A-Box), décrit les assertions sur les individus.

**b) La partie terminologique (T-BOX) :**

La T-BOX, ou base terminologique, est un ensemble fini d'axiomes d'inclusions (équivalences) de classes, d'axiomes d'inclusions (équivalence) de rôles et d'axiomes de transitivité. Ils sont utilisés pour décrire (axiomes d'inclusions) ou définir (axiomes d'équivalences) les classes relatives à notre domaine, en utilisant les constructeurs fournis par les LDs.

Les axiomes de classes consistent en deux classes séparées par le symbole d'inclusion  $\subseteq$  (axiome d'inclusion ou encore GCI : General Concept Inclusion) ou par le symbole

d'équivalence  $\equiv$  (équivalent à  $\subseteq$  dans les deux directions  $\subseteq$  et  $\supseteq$ ) (voir le tableau 4). Même chose pour les axiomes de rôles, ils consistent en deux noms de rôles séparés par le symbole  $\subseteq$  ou  $\equiv$  (voir le tableau 4) sachant que l'apparence de ces deux rôles dans le même axiome réfère soit à deux attributs soit à deux relations. On peut trouver aussi l'axiome de transitivité d'une relation Trans. Les axiomes que nous avons utilisés sont de la forme :

Nom de l'axiome	Syntaxe de l'axiome	Sémantique de l'axiome
inclusion de concepts	$C_1 \subseteq C_2$	$C_1^I \subseteq C_2^I$
équivalence de concepts	$C_1 \equiv C_2$	$C_1^I \equiv C_2^I$
inclusion de relations	$R_1 \subseteq R_2$	$R_1^I \subseteq R_2^I$
équivalence de relations	$R_1 \equiv R_2$	$R_1^I \equiv R_2^I$
transitivité d'une relation	$Trans(R_1)$	$R_1^I = (R_1^I)^+$

*Tableau 4 Syntaxe des axiomes avec leurs sémantiques.*

### c) La partie Assertionnelle (A-BOX)

La A-Box, ou base assertionnel, contient un ensemble d'assertions sur les individus. Ils sont soit des appartenances à une classe soit des valeurs de rôles (relations et attributs) (voir le tableau 5).

Les assertions d'individus qu'on va utiliser sont de la forme :

Syntaxe de l'assertion	Sémantique de l'assertion
$C_1(o_1)$	$o_1^I \in C_1^I$
$R_1(o_1, o_2)$	$(o_1, o_2) \in R_1$
$U_1(o_1, t)$	$(o_1, t) \in U_1$

*Tableau 5 Syntaxe des assertions avec leurs sémantiques.*

### IV.3.4. Opérationnalisation

L'ontologie obtenue dans l'étape précédente est formelle. Le but de cette étape consiste à encoder l'ontologie formelle en un langage d'implémentation d'ontologies pour la rendre opérationnelle. Elle est réalisée en faisant recours à un outil d'édition d'ontologies. Cette étape achève donc le processus de construction de l'ontologie.

### IV.3.5.Évaluation

C'est une étape qui peut être effectuée en parallèle avec les trois étapes précédentes (conceptualisation, formalisation et opérationnalisation). Elle est effectuée au niveau conceptuel qu'au niveau formel et opérationnel :

- Évaluation de l'ontologie conceptuelle : rappelons que les principes d'ingénierie ontologique sont utilisés afin de guider l'otologiste à construire et à évaluer l'ontologie. Par conséquent, les principes respectés dans l'étape de conceptualisation sont eux même utilisés pour l'évaluation de l'ontologie conceptuelle.
- Évaluation de l'ontologie formelle et opérationnelle : elle est réalisée automatiquement au moyen de l'utilisation d'un raisonneur capable de calculer les inférences afin de supporter le processus de construction et d'améliorer la qualité de l'ontologie.

Par ailleurs, il existe une liste de critères de vérification et de validation de l'ontologie, proposé par les auteurs de la méthode METHONTOLOGY.

Les erreurs qui peuvent être détectés à la phase d'évaluation sont classées dans les catégories suivantes [24] :

#### IV.3.5.1. Erreurs d'inconsistance

##### a) Erreurs de circularité :

Elles se produisent quand une classe est définie comme spécialisation ou généralisation d'elle même (voir figure V.1). Selon le nombre de relations impliquées, ces erreurs peuvent être classées comme : erreurs de circularité à distance zéro (une classe avec elle-même), à distance 1 et à distance n.

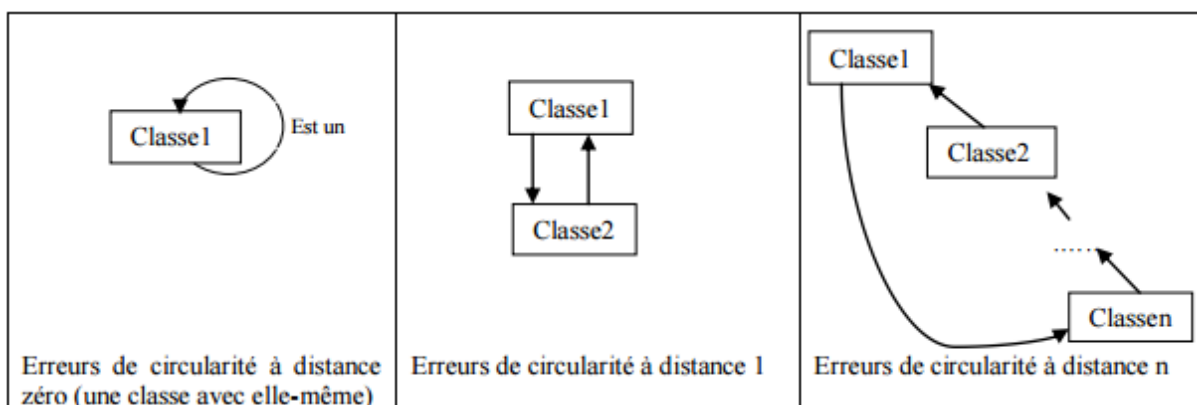


Figure IV.1 : Exemple d'erreurs circulaires dans la taxonomie.

### b) Erreurs sémantiques :

Elles se produisent habituellement parce que l'ingénieur ontologique fait une classification sémantique incorrecte, c.-à-d., il classe un concept comme sous-classe d'une classe d'un concept auquel il n'appartient pas vraiment. Le même problème peut arriver lors de la classification d'instances.

### c) Erreurs de partition :

La classification des concepts peut être définie de façon disjointe (une décomposition disjointe), complète (décomposition exhaustive) ou disjointe et complète (partition). Les erreurs suivantes sont définies (voire figure IV.2) :

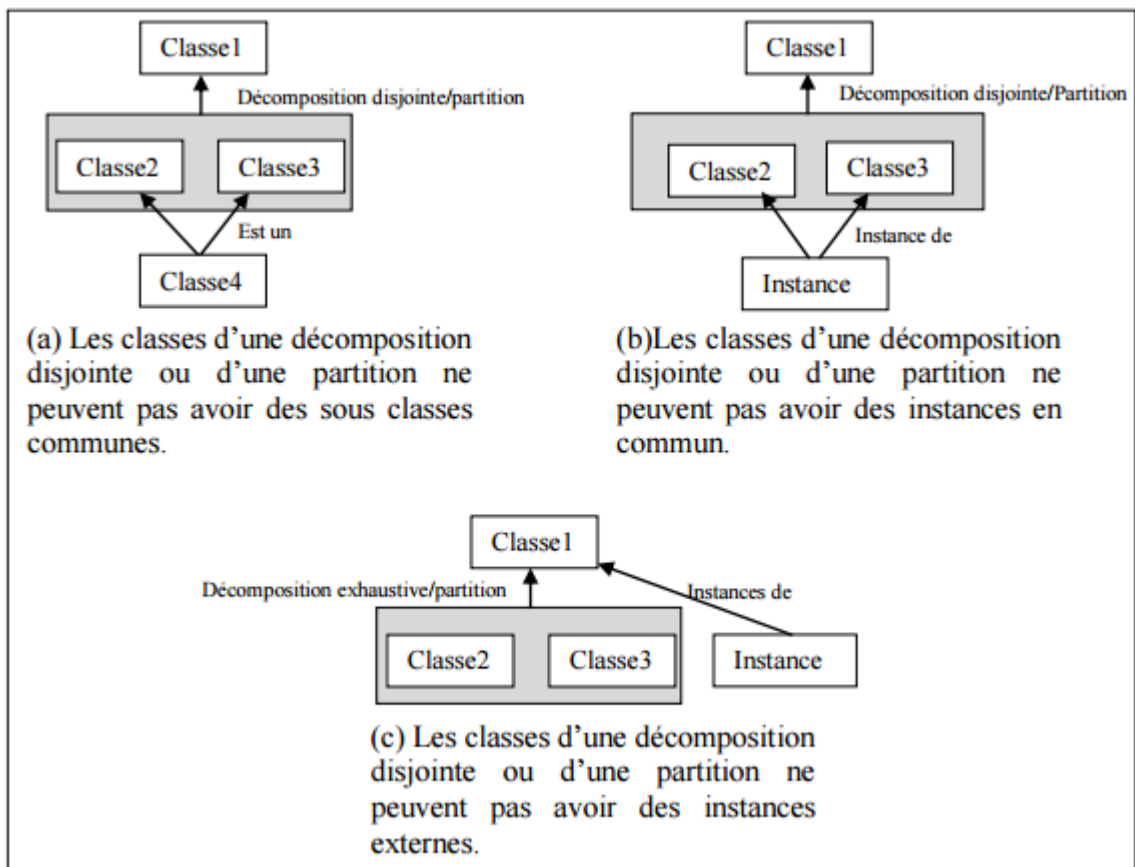


Figure IV.2 Exemple d'erreurs de partition dans l'ontologie.

### IV.3.5.2. Erreurs d'incomplétude :

- a) Classification de concepts incomplète :** généralement, cette erreur se produit chaque fois que les concepts sont classifiés sans qu'ils n'aient tous été pris en compte, c.-à-d. quand des concepts existants dans le domaine ont été négligés.

**b) Erreurs de partition** : Les classifications de concepts peuvent être définies d'une façon disjointe (décomposition disjointe), complète (décomposition complète), ou disjointe et complète (partition). Deux types d'erreurs sont définis : (a) l'ingénieur ontologique définit un ensemble de sous-classe d'une classe donnée, mais omet de modéliser dans la taxinomie que les sous-classes sont disjointes alors qu'elles devraient l'être ; (b) L'ingénieur ontologique définit une partition de classe pour un ensemble de sous-classes qui ne sont pas complètement classées et devraient l'être.

### **c) Erreurs de redondance**

Elles se produisent au niveau de la taxonomie quand on redéfinit des expressions qui ont déjà été explicitement définies ou qui peuvent être déduites à partir d'autres définitions. On distingue les types de redondance suivants :

- ❖ **Redondance de la relation « est un »**
- ❖ **Redondance de la relation « instance de »**
- ❖ **Définitions formelles identique pour certaines classes**
- ❖ **Définitions formelles identique pour certaines instances**

Nous présenterons dans la deuxième partie de ce chapitre comment nous avons appliqué ce processus pour le développement de l'Onto-Resto.

### IV.4. Construction d'une ontologie pour un restaurant de futur

Dans cette partie, nous construisons Onto-Resto. Selon les critères de classification étudiés dans la 1<sup>ère</sup> partie, Onto-Resto est une ontologie de domaine d'un restaurant de futur.

Nous suivons le processus décrit précédemment afin de la construire.

#### IV.4.1. Spécification

Une ontologie ne peut être construite qu'après la phase de spécification. Il s'agit d'établir un document informel de spécification de besoins. Au niveau de ce document, nous décrivons l'ontologie à construire à travers les cinq aspects suivants :

❖ **Domaine de connaissance**

Domaine de restauration.

❖ **Objectif**

L'objectif de l'ontologie du Restaurant du Futur est de modéliser le contexte d'un restaurant : matériel (disposition des buffets, des plats...) et le contenu (le menu, catégories...). Une ontologie des repas est aussi nécessaire pour décrire les repas impliqués dans les menus. Ces modèles sont spécifiques à notre prototype et utilisables pour la restitution du menu à l'utilisateur. Pour cette modélisation, nous nous sommes appuyés sur les données acquises concernant les repas et les menus.

❖ **Utilisateurs**

Client, internaute, administrateur.

❖ **Sources d'informations**

Documents techniques en rapport avec le domaine de restaurant.

❖ **Portée de l'ontologie**

Client, administrateur, repas, Structure de restaurant, Document, etc.

#### IV.4.2. Conceptualisation

Une fois la majorité des connaissances acquise, on doit les organiser et les structurer en utilisant des représentations qui sont faciles à comprendre et indépendantes de tout langage d'implémentation. Cette phase contient plusieurs étapes qui sont :

### ➤ Construction du glossaire des termes

Ce glossaire recueille et décrit tous les termes qui sont utiles dans l'ontologie finale. Le tableau ci-dessous (Tableau 6) fournit une liste détaillée des différents termes utilisés dans l'ontologie :

Nom	Description
<b>Repas</b>	C'est un aliment liquide ou solide.
<b>Menu</b>	La liste des repas disponible.
<b>Commande</b>	Sont les éléments demander pour servir à table.
<b>Boisson</b>	C'est un liquide destiné à la consommation par le client peut être gazeuses ou énergétique ou minérale.
<b>Régime</b>	C'est ensemble de prescriptions concernent les aliments destinés à rétablir la santé
<b>Cuisinier</b>	C'est la personne qui prépare les repas
<b>Dessert</b>	Dernière partie de repas peut être fruit ou noix
<b>Calories</b>	C'est une unité de mesure qui détermine la quantité de chaleur dégagée dans un aliment
<b>Poids</b>	C'est une mesure son unité KG
<b>Taille</b>	C'est une mesure son unit CM
<b>Date et heure</b>	Représente la date et heure de facture
<b>Recommandation</b>	Proposer des plats ou des conseils alimentaires
<b>Facture</b>	C'est un document de comptabilité générale qui prouve une commande
<b>Catégories</b>	Représente les différentes catégories des repas
<b>Type repas</b>	Déjeuner ou dîner
<b>Restaurant algérienne</b>	C'est restaurant qui propose des plates algériennes.
<b>Restaurant française</b>	C'est un restaurant qui propose des plats français
<b>Restaurant Syrienne</b>	C'est un restaurant qui propose des Plats syrienne
<b>Restaurant italiennes</b>	C'est un restaurant qui propose des lats italiennes
<b>Régime diabète</b>	C'est tous les conseils concert les aliments nécessaires pour une personne diabète
<b>Régime hypertension</b>	C'est tous les conseil concert les aliments nécessaires pour un



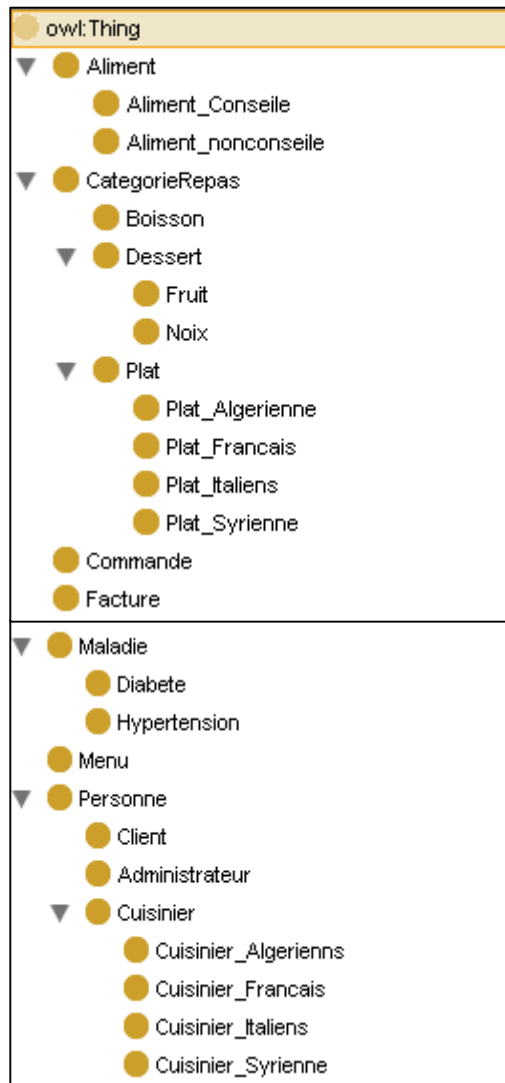
	personne a un hypertension
<b>Client</b>	C'est une personne qui passe des commandes
<b>Plat</b>	Un ensemble des aliments

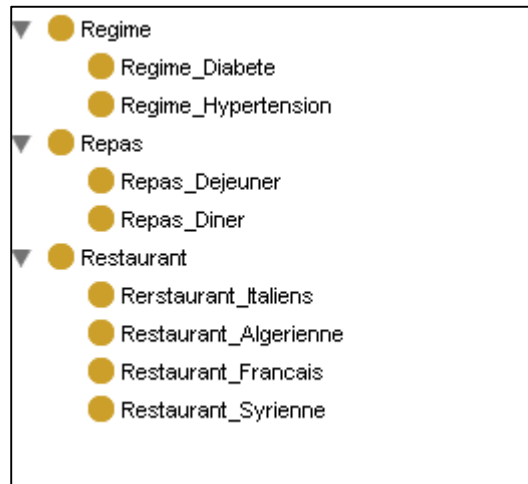
*Tableau 6 Glossaire de termes.*

### ➤ Classification des concepts en hiérarchies de concepts

Cette tâche consiste à définir les relations entre les concepts définis dans le glossaire de termes.

La figure IV.3 illustré les hiérarchies de concepts.

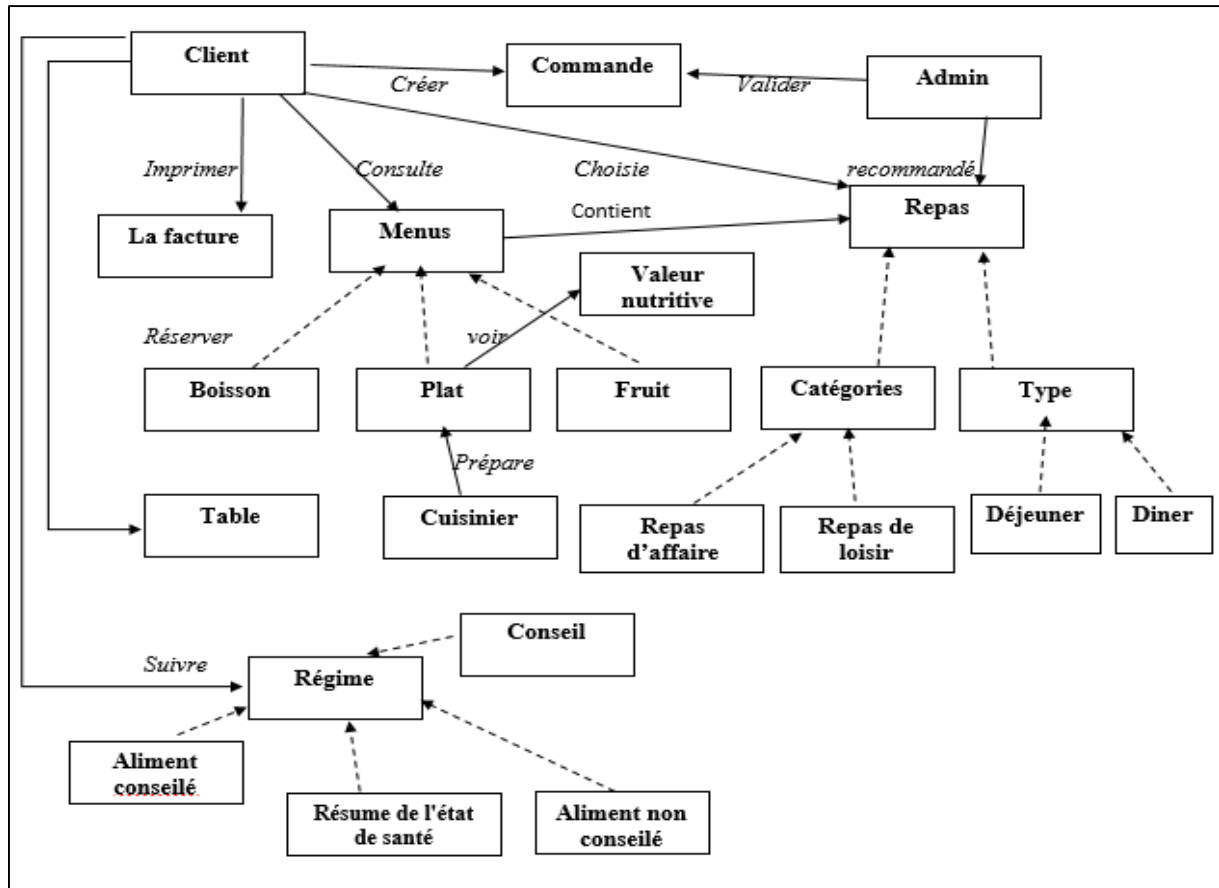




*Figure IV.3 Hiérarchies de concepts.*

### ➤ Construction de diagramme de relations binaires

Une relation binaire permet de relier deux concepts entre eux (un concept source et un concept cible). « Si R est une relation entre deux concepts C1 et C2 alors pour tout couple d'instances des concepts C1 et C2, il existe une relation de type R qui lie deux instances de C1 et C2 ». Cette tâche permet de représenter d'une manière graphique les différentes relations qui existent entre les divers concepts que ce soit de même ou de différentes hiérarchies.



*Figure IV.4 Diagramme de relations binaires.*

### ➤ Construction d'un dictionnaire de concepts

Une fois la taxonomie de concepts et le diagramme de relations binaires sont effectués, il faut donc spécifier les propriétés qui décrivent chaque concept de la hiérarchie dans un dictionnaire de concepts. Un dictionnaire de concepts contient tous les concepts du domaine, leurs synonymes, leurs acronymes, leurs attributs et leurs relations. Les relations spécifiées pour chaque concept sont celle où leur domaine est le concept lui-même (voir Tableau 7).

Nom concept	Attributs	Les relations	Synonymes
-Client	-Nom -prénom -datenaissance -age -email -tel -adresse -taille -poids	-suivre -créer -imprimer -consulter -choisie	
Repas	-n°repas -nombrecalories -valeur nutritive -prix		
Menus	-n°menus	-Contient	
Admin	-id -motpasse	-Valider -recommandé	-Le géré
Régime			
Plats	-n°plat -composantplat		
Boisson	-nomboisson -prixboisson		
Dessert	-nomdessert -prixdessert		
Facture	-n°facture -dateheure	-etablir le	
Catégories repas	-nomcategories		
Type repas	-		
Cuisinier	-idcuisinier	- Prépare	
Commande	-n°cammande -prix totale		
Date et Heure	-date_heure		

**Tableau 7 Dictionnaire de concepts.**

➤ **Construction de la table des relations binaires**

Le but de cette tâche consiste à construire une table de relations binaires décrites en détail. Pour chaque relation utilisée dans le diagramme des relations binaires, nous définissons le nom de la relation, le nom des concepts sources et cibles, le nom de la relation inverse et les cardinalités source et cible (voir Tableau 8).

## Chapitre IV : Développement d'ontologie

Nom de la relation	Concept source	Concept cible	Card source	Card cible	Relation inverse
Créer	Client	Commande	1..1	1.. n	Créer par
Consulter	Client	Menus	1..1	1..n	Consulter par
Suivre	Client	Régime	1..n	1..1	Suivie par
Valider	Admin	Commande	1..1	1..n	Valider par
Recommandé	Admin	Repas	1..1	1..n	Recommandé par
Choisie	Client	Repas	1..1	1..n	Choisie par
Imprimer	Client	Facture	1..n	1..1	Imprimer par

*Tableau 8 Table des relations binaires.*

### ➤ Construction de la table d'attributs

La table des attributs (voir le tableau 9) comporte une description détaillée des attributs inclus dans le dictionnaire de concepts, et l'ensemble de contraintes et de restrictions sur ces valeurs

Nom attribut	Type de valeur	Card (max/min)	Valeur par défaut	Domaine de valeurs
Nom	String	1..1	-	-
Prenom	String	1..n	-	-
Email	String	1..n	-	-
Age	Int	1..1	-	-
Datenaissance	String	1..1	-	-
Adresse	String	1..n	-	-
N°repas	Int	1..1	-	-
Nombrecalories	Int	1..1	-	-
Valeur nutritive	Int	1..n	-	-
idadmin	String	1..1	-	-

Motpasse	String	1..1	-	-
----------	--------	------	---	---

*Tableau 9 Table des attributs.*

➤ **Construction de la table des axiomes :**

Dans cette étape, nous définissons les concepts au moyen d'expressions logiques. Pour chaque axiome, il faut spécifier la description de l'axiome en langage naturel, l'expression logique qui décrit formellement l'axiome en logique du premier ordre, les concepts, les relations et les variables utilisées (voir le tableau 10).

Nom du concept	Description	Expression logique	Concepts	Relation	Variables
Personne	Chaque personne est soit un client, soit un administrateur soit cuisinier.	$\forall x \text{Personne}(X) \Rightarrow \text{Client}(X) \vee \text{admin}(X) \vee \text{Cuisinier}(X)$	Personne, Admin, Client		X
Repas	Chaque repas est soit un plat, soit un boisson, soit un dessert.	$\forall x \text{ Repas } (X) \Rightarrow \text{Plats}(X) \vee \text{Boisson } (X) \vee \text{Dessert } (X)$	Plats, boisson, Dessert	Contient	X
Maladie	Maladie diabète ou Maladie hypertension	$\forall x \text{ Maladie } (X) \Rightarrow \text{diabète}(X) \vee \text{Hypertension}(X)$			X

*Tableau 10 : Table des axiomes.*

➤ **Construction de la table des instances**

La table des instances décrit les instances connues ; qui sont déjà identifiées dans le dictionnaire de concepts. Pour chaque instance, il faut spécifier le nom de l'instance, le nom du concept où elle appartienne, ses attributs et les valeurs qui lui y sont associés. Le tableau 11 ci-après illustre quelques instances créées.

Nom de l'instance	Nom du concept	Attributs	Valeurs
-------------------	----------------	-----------	---------

Masculin	Sexe		
Féminin			
Plats	Repas		
Boisson			
Dessert			
Hypertension	Maladie		
Diabète			

*Tableau 11 Table des instances.*

### IV.4.3. Formalisation

Comme cité auparavant, dans cette étape, nous utilisons le formalisme de la logique de description pour formaliser le modèle conceptuel que nous avons obtenu dans l'étape de conceptualisation. Le résultat est une base de connaissances en logique de description SHOIN(D) composé de deux parties T-BOX (voir le tableau 11) et A-BOX (voir le tableau 12).

#### IV.4.3.1 Construction de T-Box :

//---les axiomes suivants permettent de décrire et définir les classes de l'ontologie  
 $Personne \sqsubseteq T \cap (\exists nom.String) \cap (\exists prenom.String) \cap (\exists dateNaissance.String) \cap (\exists adresse.String) \cap (\exists age.String) \cap (\exists taille.Int) \cap (\exists Poids.Float) \cap (\exists Maladie.String) \cap (=1 Choisie) \sqcap (\geq 0 Suiv) \cap (\geq 0 passe)$

//---les axiomes suivants permettent de définir le domaine, le co-domaine et les caractéristiques des relations---//

$\exists a PourMaladie. T \sqsubseteq Client$

$T \sqsubseteq \exists a PourRegime$

(.....)

//---les axiomes suivants permettent de définir le domaine et le co-domaine d'un exemple d'un attribut qui est nom. La même chose est réalisée sur toutes les attributs de chaque concept---//

$\exists nom. T \sqsubseteq Personne$

$T \sqsubseteq \forall nom.string$

$\exists age. T \sqsubseteq Personne$

$T \subseteq \forall \text{ age.int}$

(.....)

### IV.4.3.2 Construction d'A-Box :

Sexe	Masculin
	Féminin
Repas	Plats
	Boisson
	Dessert
Maladie	Hypertension
	Diabète

*Tableau 12 Assertions d'individus.*

Jusqu'à présent, nous avons attribué un modèle formel à l'ontologie Onto-Resto dans cette section. Dans la section suivante, nous détaillons son implémentation en utilisant les outils qui ont été développés pour cet objectif

### IV.4.4 Opérationnalisation

L'ontologie que nous avons obtenue à ce stade est une ontologie formelle, il nous reste à opérationnaliser cette ontologie pour qu'elle soit concrètement manipulable dans un système informatique. Par conséquent, elle doit être spécifiée dans un langage de représentation de connaissance doté de capacités d'inférences. Pour ce faire, nous utilisons le logiciel PROTÉGÉ OWL 3.4.8.

### IV.4.5 Évaluation de l'ontologie



Il est conçu pour raisonner sur les logiques de descriptions et accepte en entrée un fichier OWL. Une fois l'ontologie chargée, il effectue les inférences sur la T-Box et la A-Box.

### **IV.5 Conclusion**

## **Chapitre IV : Développement d'ontologie**

---

Pour conclure ce chapitre, nous pouvons dire que le processus suivi pour la construction d'Onto-Resto nous a permis de réussir finalement à construire une ontologie de domaine. Cette dernière est prête à une future évaluation au sein de notre site de restaurant de futur.

## **Chapitre V : Réalisation**

### V.1 Introduction

Après avoir présenté notre conception dans les deux chapitres précédents, nous allons présenter, dans ce chapitre, notre travail d'implémentation qui consiste premièrement à éditer notre ontologie sous l'éditeur PROTEGE, puis à l'intégrer dans une application web avec JSP (Java Server Page) utilisant éditeur Net-Beans.

Nous commençons d'abord par la présentation des outils et des technologies utilisés dans le développement de notre environnement. Ensuite, nous présentons une vue globale sur l'édition de ontologie sous l'éditeur PROTEGE. Enfin, nous présentons des scénarios d'utilisation de notre environnement.

### V.2 Les outils et les technologies utilisés

#### V.2.1 L'éditeur PROTEGE

Suite aux explications de différents éditeurs cités dans le chapitre 4 et dans l'objectif de générer automatiquement le code OWL correspondant à notre ontologie, nous avons choisi PROTEGE version 3.4.8 parce que c'est l'éditeur le plus flexible et qui est muni d'une interface souple permettant la mise à jour de l'ontologie.

#### V.2.2 Net-Beans IDE

L'IDE Net-Beans est un outil puissant, gratuit, et multi plateformes. Il permet l'intégration des outils nécessaires au développement d'une application.

Pour notre projet, nous avons utilisé la version 8.1 facilitant le développement des applications web.

#### V.2.3 La technologie JSP

Les JSP sont une des technologies de la plate-forme Java EE les plus puissantes. Elles permettent aux développeurs de créer des vues dynamiques. La technologie JSP se base sur le modèle MVC (Modèle, Vue, Contrôleur) qui permet de séparer la partie de traitement de la partie présentation, où :

- ❖ **Modèle** : représente les données et les traitements à appliquer à ces données, c'est à dire la logique programmatique (le code java).

- ❖ **Vue** : représente des pages JSP exécutées côté serveur permettant d'appeler des parties de code Java via des balises ressemblant aux balises de la présentation HTML.
- ❖ **Contrôleur** : représente des servlets qui sont des objets qui permettent de capter des requêtes réalisées par un client, de personnaliser une réponse par l'appel des traitements effectués par le modèle et d'ordonner le retour à la vue et d'afficher le résultat au client.

### V.3 Implémentation de l'ontologie

Après la conception de notre ontologie du domaine, nous l'avons édité sous PROTEGE pour générer automatiquement le code OWL correspondant.

#### V.3.1 L'édition de l'ontologie

Au démarrage de protégé, la fenêtre suivant va apparaître au premier lieu :

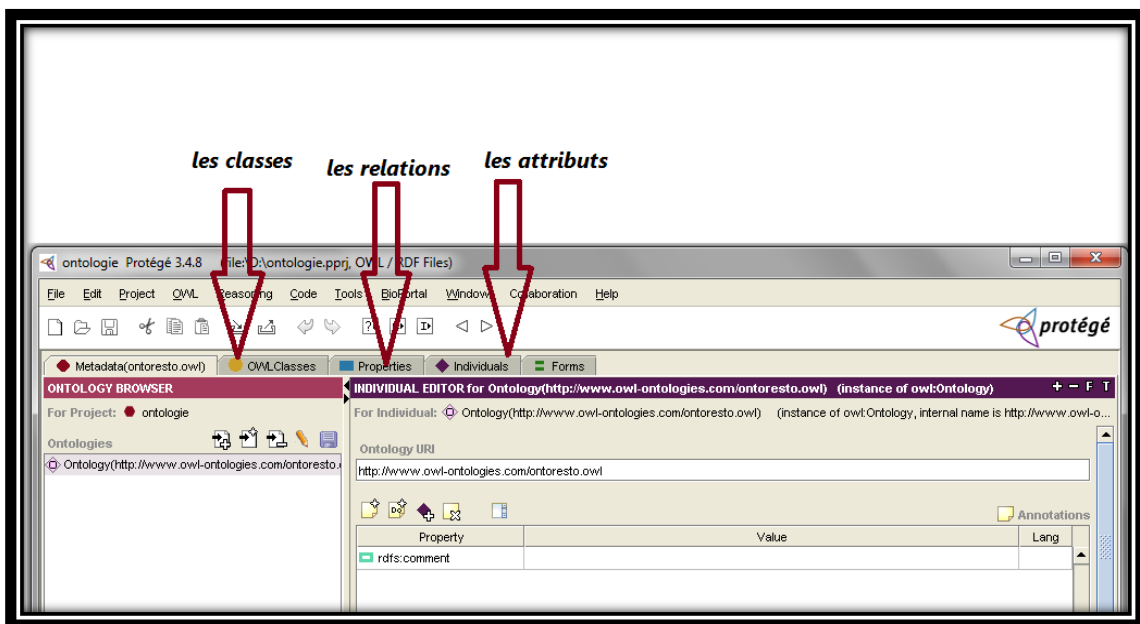


Figure V.1 : Interface de Protégé 3.4.8

### V.3.2 L'ajout et suppression d'un class

L'ajout et suppression d'un class demande d'indiquer son nom et de définir son type d'opération à réaliser création ou suppression. La figure suivante explique le concept sous PROTEGE :

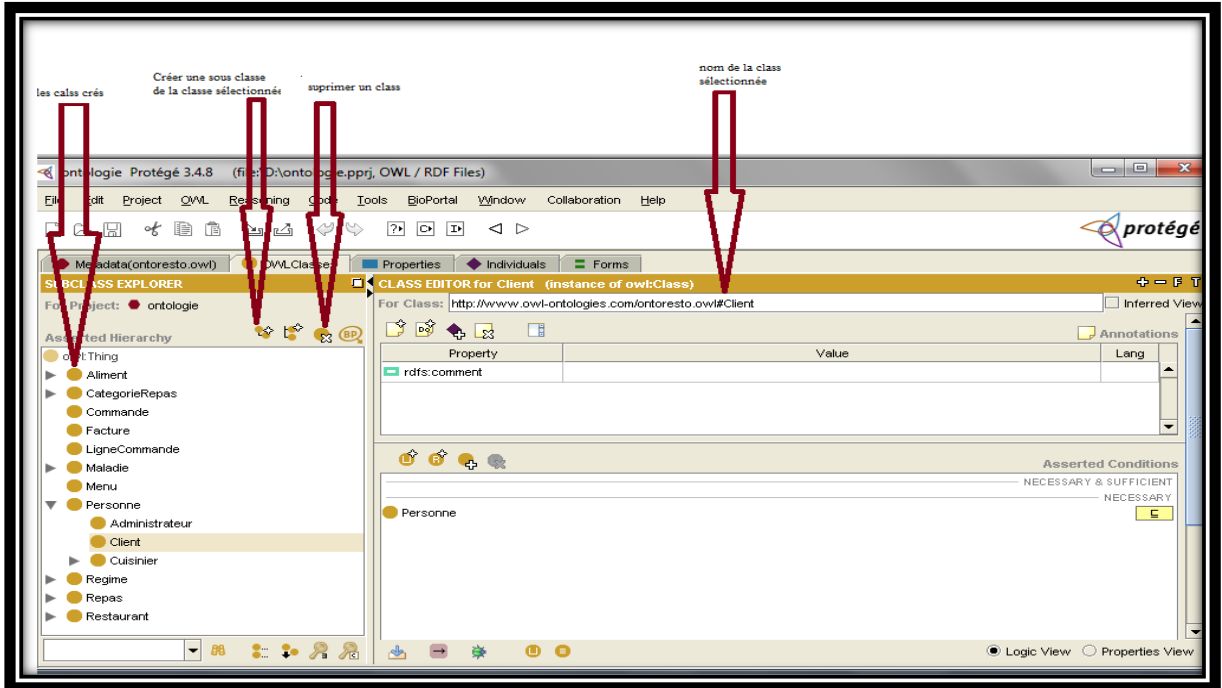


Figure V.2: Ajouter ou supprimer un class dans Protégé

### V.3.3 L'ajout d'un attribut

L'ajout d'un attribut exige de préciser son nom, son type et son concept d'appartenance. Les deux figures suivantes montrent comment ajouter un attribut.

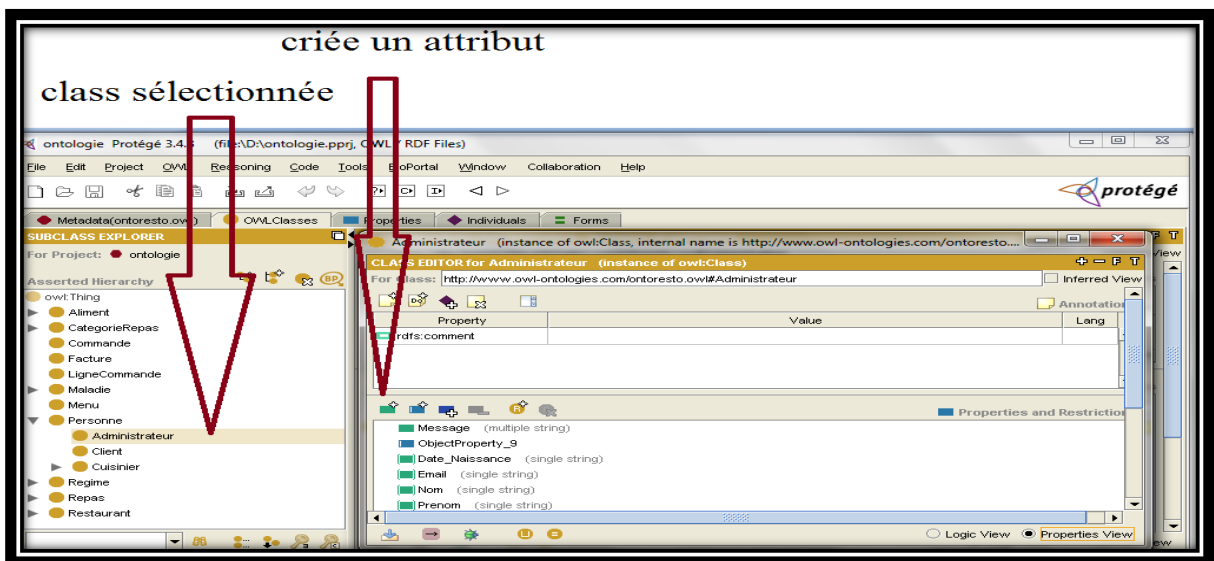
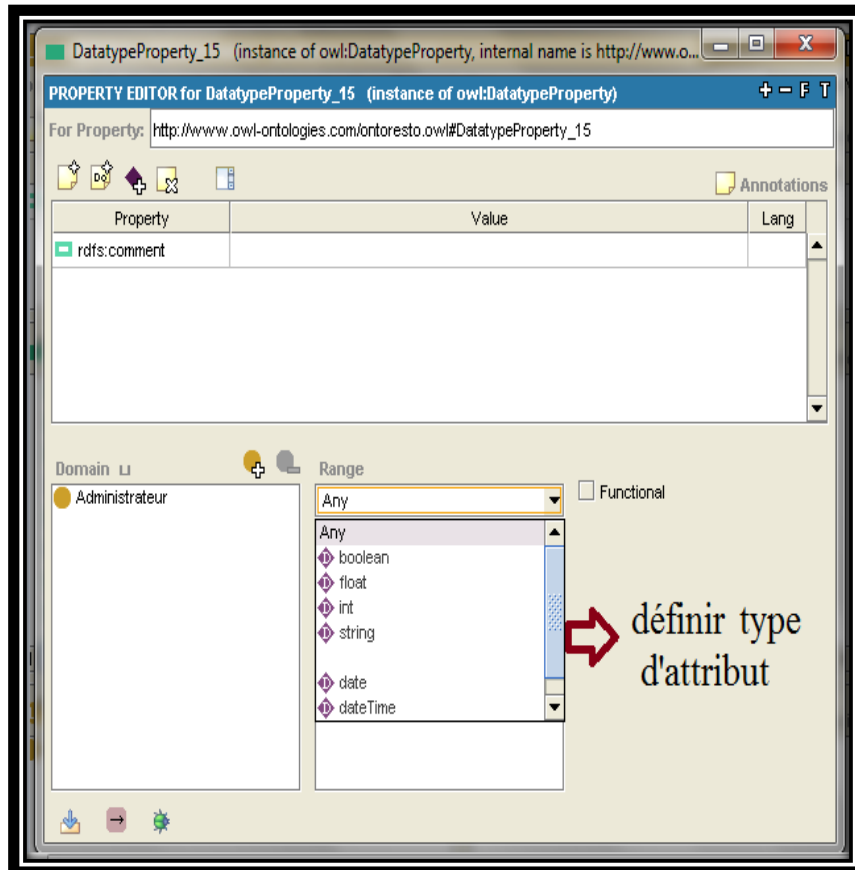


Figure V.3 : ajouter un attribut sous Protégé

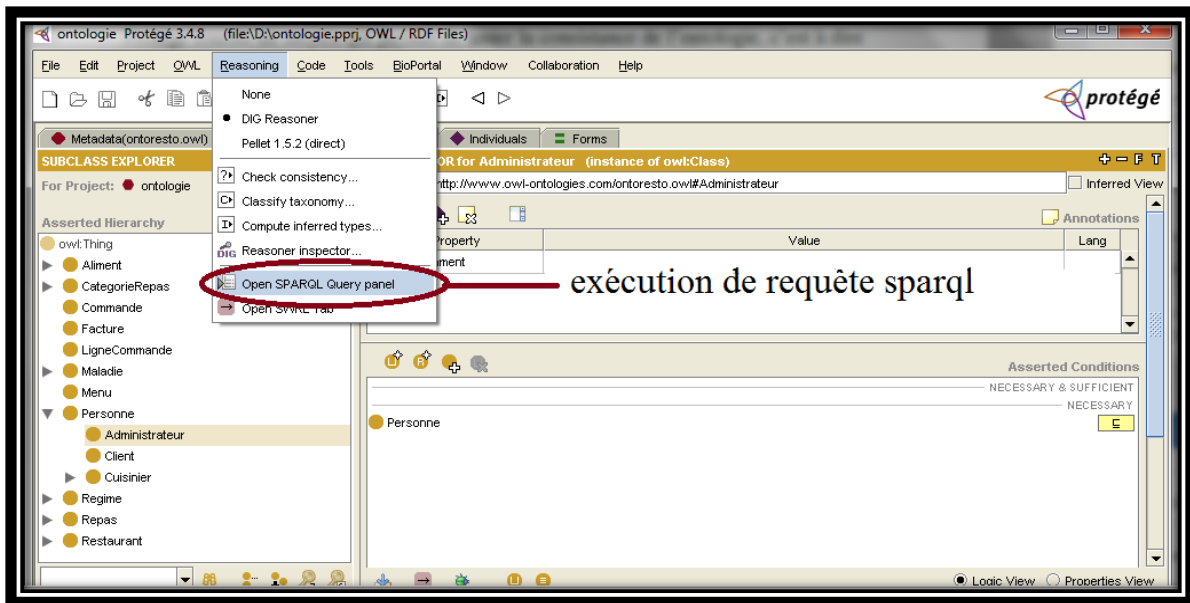
La définition du type de l'attribut sous PROTEGE montré dans la figure suivant (Figure V.4) :



*Figure V.4: Définir type de l'attribut*

### V.3.4 La vérification de l'ontologie

Protégé fournit un moyen qui permet de tester la consistance de l'ontologie, c'est-à-dire grâce à son raisonneur, nous pouvons détecter l'existence des erreurs dans notre ontologie par exécution des requêtes SPARQL:



*Figure V.5 : exécution des requêtes SPARQL sous Protégé*

### V.3.5 La génération du code OWL

Quant l'opération d'ajout des concepts, des relations et des attributs est terminée, nous obtenons un fichier OWL qui sera utilisé par notre application dans la connections de base de connaissance avec le site web de restaurant.

### V.4 Intégration de l'ontologie

Afin d'interroger notre ontologie nous avons développé un environnement de commander un plat à distance qui permet aux utilisateurs de naviguer dans l'ontologie, de trouver des plats, chercher des repas correspondants à la santé de consommateur. D'après l'étude des besoins des utilisateurs (administrateur, client, internaute) présenté dans le chapitre précédant, nous avons développé cet environnement dont il adapte avec chaque utilisateur. Dans ce qui suit nous essayons de faire des scénarios d'utilisation pour chaque utilisateur.

#### V.4.1 Scénarios d'utilisation de l'environnement

Dans cette partie nous présentons les scénarios d'utilisation de l'environnement :

##### ➤ *L'interface pour les internautes :*

Au démarrage de notre environnement, l'internaute peut se connecter ou s'inscrire s'il n'est pas encore inscrit ou contacte le service restaurant. Dans cette application nous avons présenté les figures suivantes:



### ❖ Page d'accueil :

Une page d'accueil s'affiche pour permettre à l'utilisateur d'accéder aux différents services offerts par le site en cliquant sur le bouton correspondant

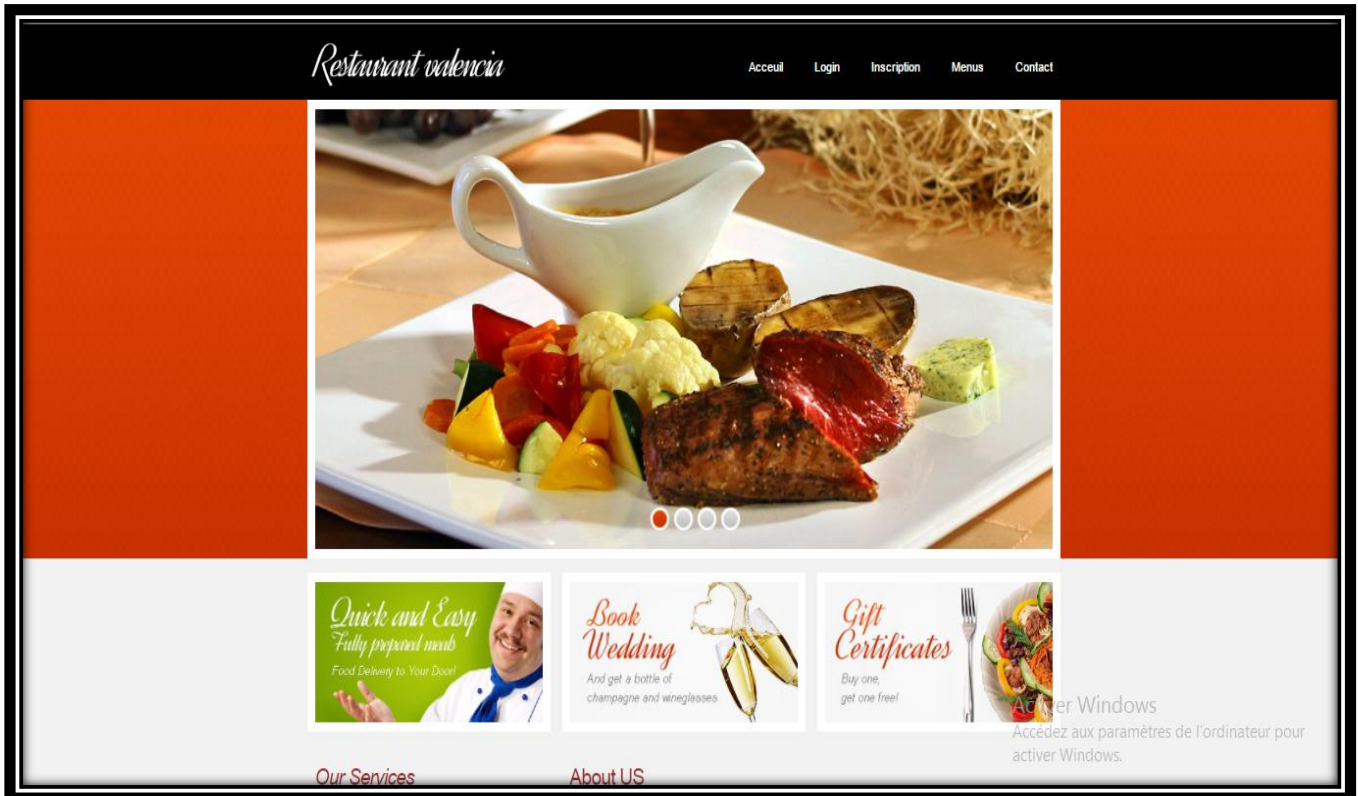


Figure V.6. Page Accueil.

- ❖ **Page d'inscription** : Elle contient un formulaire d'inscription dans lequel n'importe quel visiteur peut s'inscrire pour devenir client afin de faire leur commande en ligne.

*Restaurant valencia* Accueil Login Inscription Menu Contact

### Créer votre Compte

Nom :

Prenom :

Date Naissance : 01 - 01 - 2017

Username

Mot de passe :

Sexe : Masculin

Poids :

Taille :

Maladie : Diabete

Adresse

Taille :

Maladie : Diabete

Adresse

Email :

Telephone +213 :

Copyright © Restaurant valencia All Rights Reserved  
Design by Louafi Djad & Mouaissi Mohammed

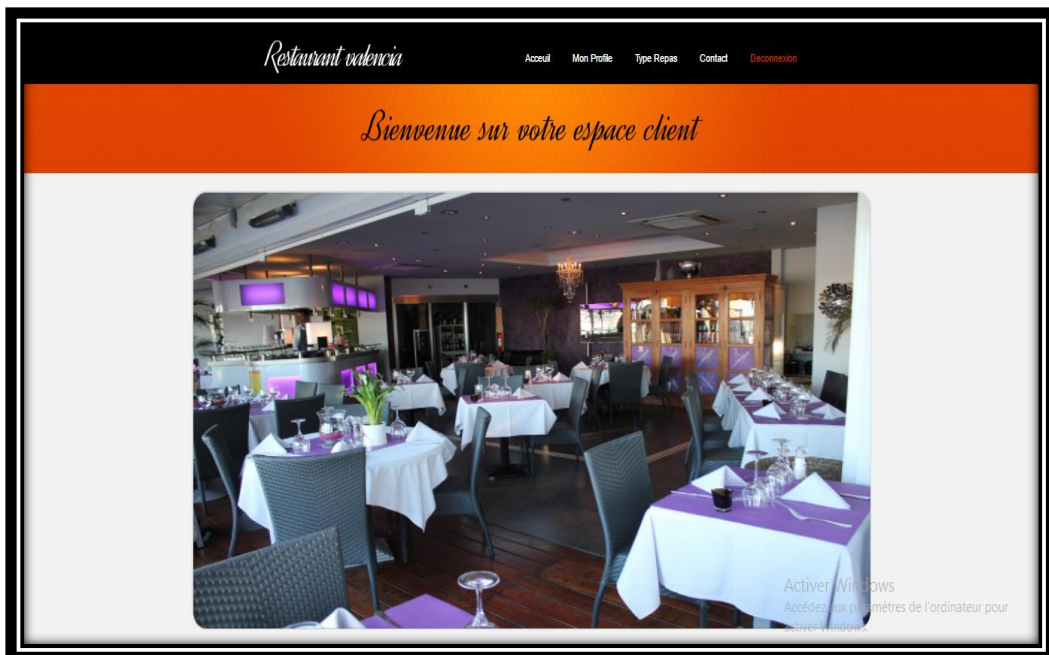
*Figure V.7 : Page Inscription*

- ❖ **Page d'authentification** : l'interface d'authentification permet aux clients d'accéder à son espace par le biais de son login et son mot de passe. Comme montre la figure suivante :



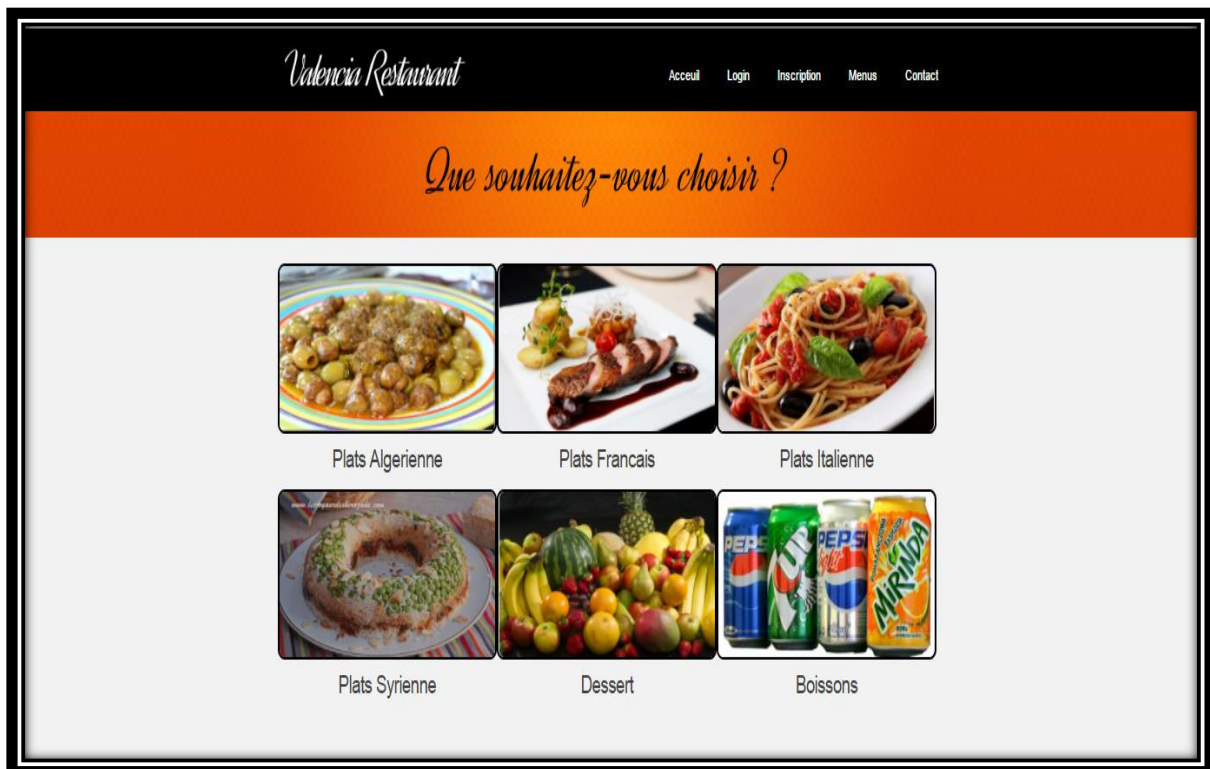
*Figure V.8. Page authentification.*

- ❖ **Page Client** : une ontologie valide est prête d'être chargée par le système pour être interrogée. Une fois le site est actif, l'interface présentée par la figure V.9 est affichée.



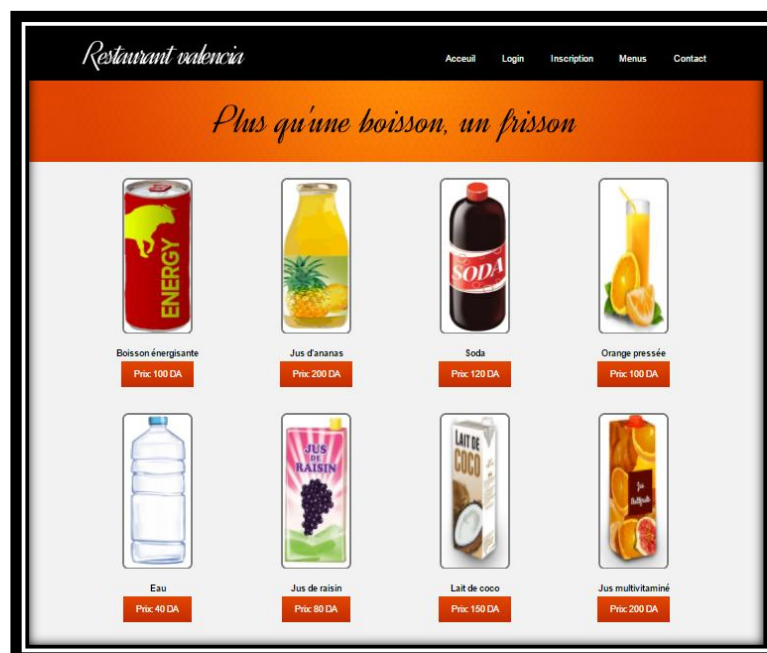
*Figure V.9 : Page Client*

**Page consulter menu :** La page de consulter menu de ce site représente le catalogue des différents types de repas.



*Figure V.10: Consulter Menus*

En cliquant sur n'importe quel choix, Elle représente l'image de chaque aliment et son prix.  
Par exemple boisson :



*Figure V.11 : Page consulter boissons*

- ❖ **Page contact** : permet aux internautes ou clients de contacter administration du site sur n'importe quel problème en met le nom et email et message puis en appuyant sur le bouton Send.



*Figure V.12 : Page contacter nous*

### ➤ L'interface pour les clients :

Dans cette interface on va voir toutes les possibilités associées à nos repas. On peut ajouter ou supprimer un type repas), voir profile, créer commande, voir l'évaluation de la commande, contacter service restaurant, voir les aliments conseillés et les aliments déconseillés pour chaque cas (Diabète, Hypertension, Non maladie).

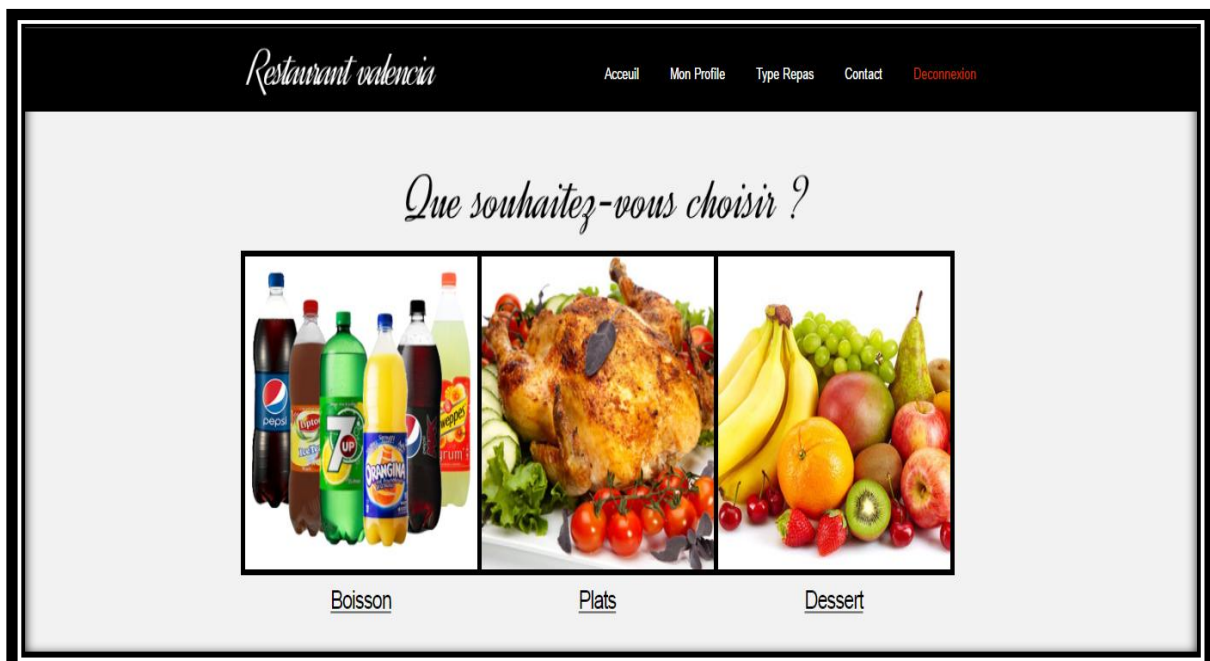
### ❖ *Page Mon Profile :*

Cette page contient tous les informations des clients (nom, prénom, date naissance etc....)



*Figure V.13 page Mon Profile*

### ❖ **Page Type repas :** permet aux clients de choisir le type de repas qu'ils souhaitent sélectionner



*Figure V.14 : Page Type repas*

- ❖ **Page sélectionner repas** : permet aux clients de faire choisie le repas en appuyer sur le bouton ajouter puis en saisie la quantité souhaitée après en clique sur valider

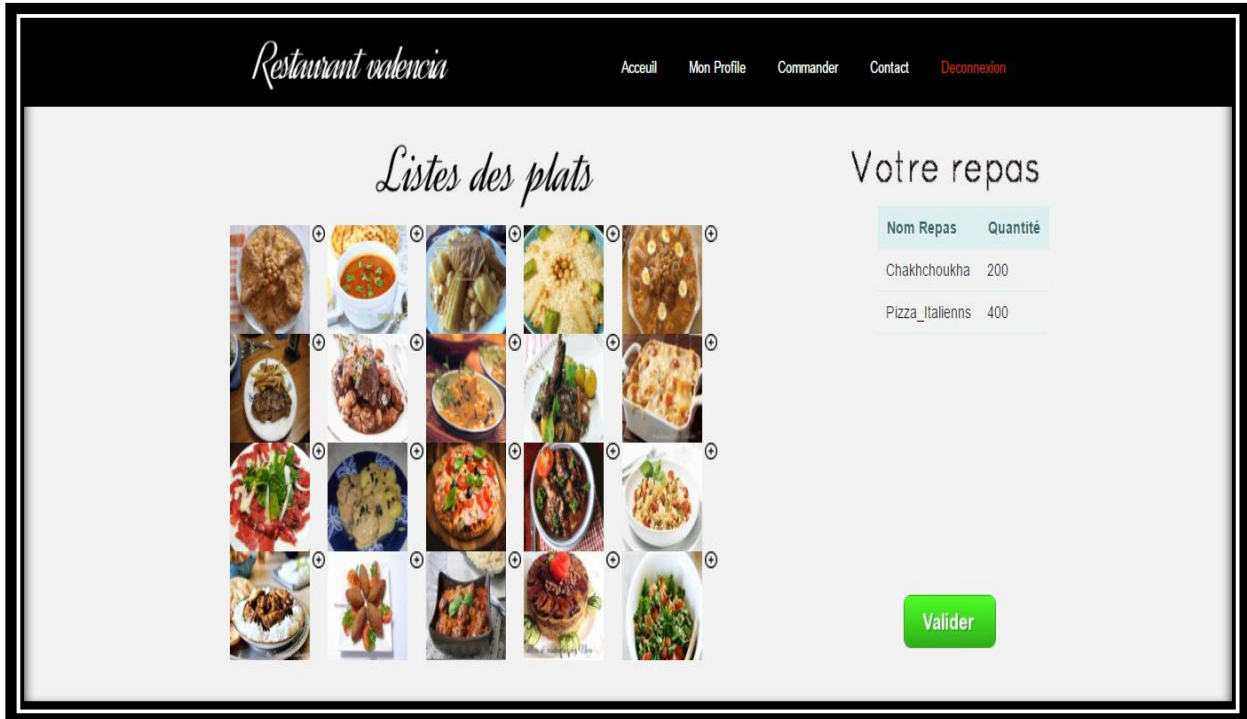


Figure V.15 : Page Sélectionner Plats

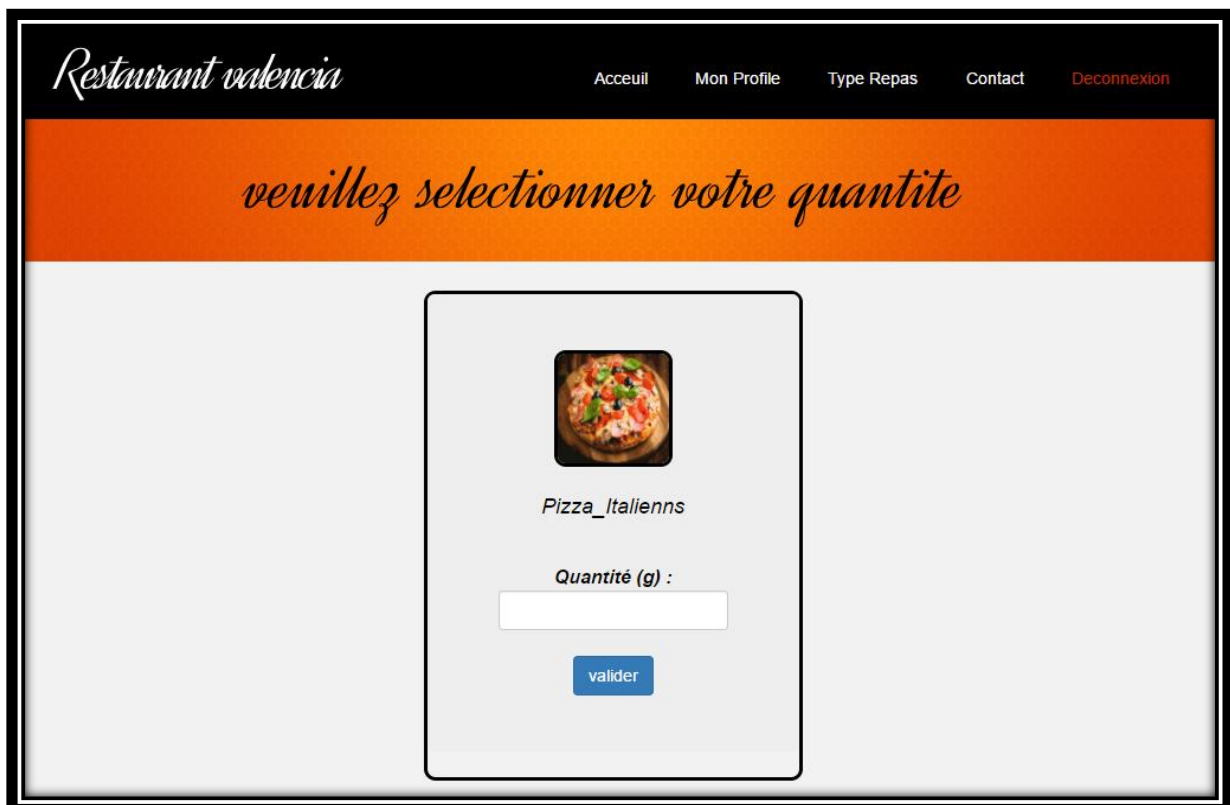
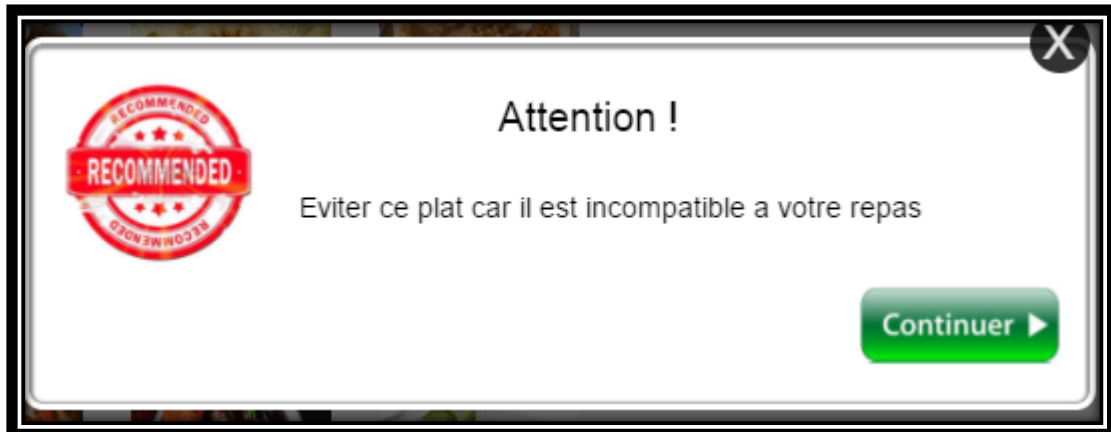
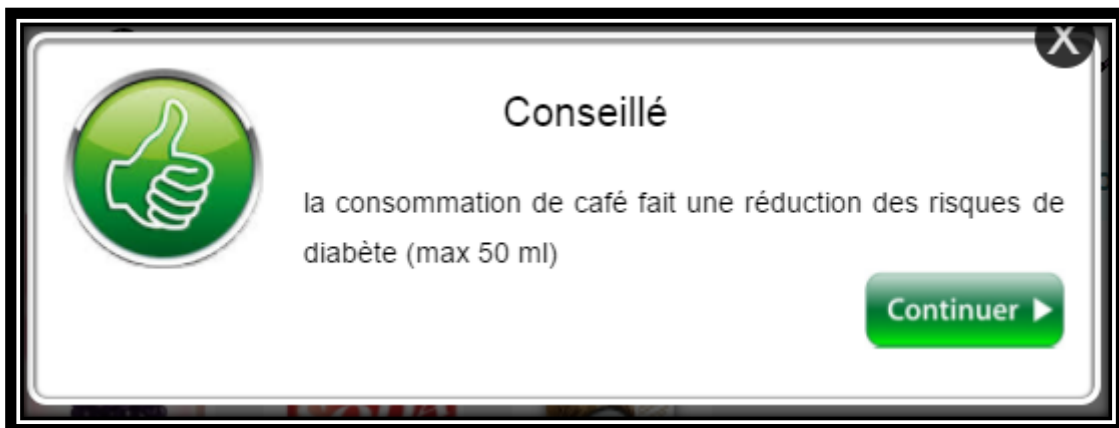


Figure V.16 : Page quantité plat.

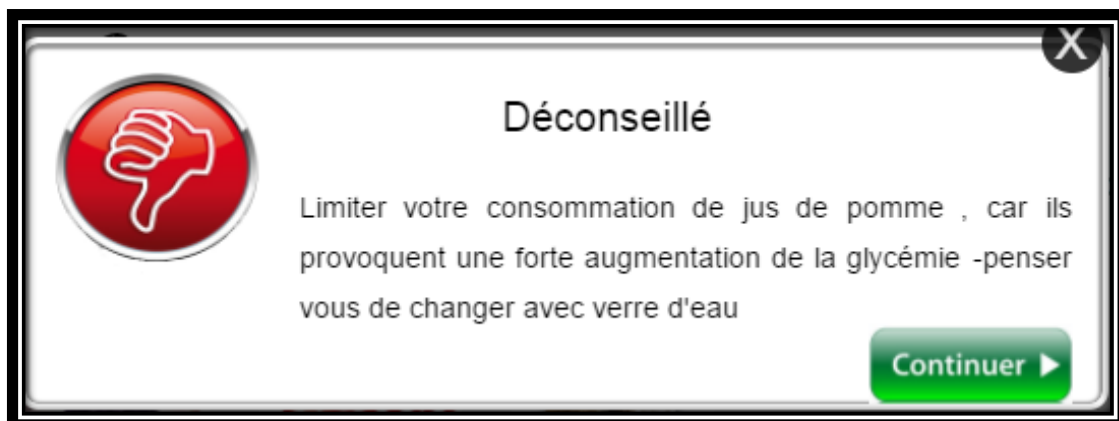
- ❖ **Page recommandation** : Si le type de repas (plats, boisson ou dessert) compatible alors afficher conseil sinon afficher recommandation.



*Figure V.17 Recommandation d'aliment*



*Figure V.18 Aliment Conseillé*



*Figure V.19 Aliment Déconseillé*



- ❖ **Page sélectionner Boisson** : Permet aux clients de faire choisir n'importe quelle boisson en saisie la quantité et en valider



Figure V.20 : Page Sélectionner Boisson

- ❖ **Page Sélectionner Dessert** : permet aux clients de faire choisir le dessert en saisie la quantité et en valider



Figure V.21 Page Sélectionner dessert

- ❖ **Page Evaluation commande** : permet aux clients de voir l'évaluation de repas. Et les conseils associés sur le nombre de calories protéines, Lipide etc....

*Restaurant valencia* Accueil Type Repas Contact Deconnexion Retour

*Bienvenue sur votre espace client*

Composition de votre repas

Energie	1781.8 kcal	✓ 🔍
Protéines	81.8 g	✗ 🔍
Glucides	93.35 g	✓ 🔍
Lipides	113.6 g	✗ 🔍
Sel	1.26 g	✓ 🔍

Votre repas

Nom Repas	Quantité	
Teboula	100	✗
Jus_de_pomme	150	✗
Jus_raisin	200	✗
Le_Steak_frites	500	✗
Pomme	200	✗

Facture

*Figure V.22 : Page Evaluation commande*

- Si en appuyer sur la croix rouge exemple (protéines) pour voir c'est quoi le problème le système affiche la fenêtre suivante :

**Proteines : 81.8 g**

Votre repas est trop riche en protéines, pensez à le compenser avec les autres repas de la journée (qui devront comprendre moins de protéines).

Un apport excessif en protéines peut être nocif pour les reins.

Vous pouvez parvenir à réduire votre consommation en diminuant les portions de viandes, poissons ou œufs, qui sont les principales sources de protéines dans notre alimentation.

*Figure V.23 : conseil sur totale protéines*

- **Page Facture** : permet aux clients de voir leur facture en ligne pour payer ou imprimer.

Nom : louafi  
Prenom : djad  
Date de commande : 12/05/2017

### Facture

Liste des repas	Prix (DA)
Teboula	150.0
Jus_de_pomme	150.0
Jus_raisin	80.0
Le_Steak_frites	250.0
Pomme	500.0

---

Total	1130.0 DA
TVA	200 DA
Total NET	1330.0 DA



*Figure V.24 : Page Facture Client*

- **Page paiement** : permet aux clients de payer les factures par chèque ou carte visa ou Mastercard pour faire la livraison



The image shows a payment page with a green background. At the top, the word "Paiement" is centered. Below it, a white-bordered box contains the "Mode de paiement" section. This section has three radio button options: "VISA" (with the Visa logo), "MasterCard" (with the MasterCard logo), and "Paiement par chèque". The "Paiement par chèque" option is selected. Below the radio buttons are three input fields: "N°de carte", "Code sécurité", and "Nom du porteur". The "Nom du porteur" field contains the text "Nom indiqué sur la carte". At the bottom of the white-bordered box is a dark green button with the word "PAYER" in white capital letters.

*Figure V.25 : Page Paiement*

- **L'interface pour Administrateur :** permet à l'administrateur de consulter les messages reçus ou mettre à jour de repas



Figure V.26 : Page Admin



Figure V.27 : Page Messagerie

### V.5 Conclusion

Dans ce chapitre, nous avons présenté l'implémentation de notre application. Nous avons commencé par la présentation des outils et des technologies utilisés dans ce travail. Ensuite, nous avons présenté une vue globale de l'édition des ontologies sous PROTEGE, nous avons créé la hiérarchie des concepts, ses attributs, les relations entre les concepts, et nous terminons cette phase par la génération du code OWL qui a été intégré dans notre application. Enfin, nous avons présenté notre environnement avec des résultats d'interrogation de notre ontologie.

### Conclusion générale

Le Restaurant intelligent est proposé comme une solution quand les restaurants classiques deviennent fastidieux, et fournissent des résultats et un rendu moins exact. Le Restaurant intelligent permet aux machines d'interpréter le contenu grâce à l'ontologie qui offre une représentation formelle. Et comme l'apprentissage en ligne est un des domaines du Web, il a la même problématique du web et donc, il peut bénéficier de ses améliorations.

Dans l'objectif d'améliorer la qualité de restaurant, nous avons proposé une ontologie qui joue le rôle d'une base de connaissances pour notre environnement dans lequel nous l'avons intégré et interrogé.

Afin de proposer notre solution, nous avons présenté un état de l'art sur les sites web et les ontologies, nous avons montré la relation entre ces deux domaines et comment notre restaurant peut bénéficier des avantages du web et aussi comment les ontologies peuvent aider les clients pour faire des choix en ligne. Nous avons développé un site web qui interroge une ontologie en utilisant l'environnement java Net-Beans et les pages JSP. Concernant notre ontologie nous avons utilisé la méthodologie METHONTOLOGY pour la conception. Ensuite, nous l'avons édité sous l'éditeur PROTEGE. Pour exploiter notre ontologie nous avons choisi le langage d'interrogation des ontologies SPARQL et le Framework Jena.

La mise en œuvre de ce projet permet aux clients de naviguer sur le site d'évaluer leurs plats, et permet au consommateur d'indexer leurs propres nourritures, et de protéger la santé des clients en donnant un ensemble de recommandations alimentaires.

On peut dire que le présent travail peut être limité par le fait d'avoir prendre en considération deux maladies seulement diabète et hypertension, Mais il peut être enrichi par l'ajout d'autres maladies.

[1] : <http://www.dzaircom.com/linternet-et-le-web/>

[2] : <http://www.morphemzero.be/sujets/site-statique-dynamique/>

[3] : <https://openclassrooms.com/courses/concevez-votre-site-web-avec-php-et-mysql/introduction-a-php>

[4] : *Mémoires de fin d'étude Rima et Hanane SAOUCHI Conception et réalisation d'un site web dynamique pour un magazine en ligne Université Mentouri Constantine - Licence en informatique option académique 2011*

[5] : *T. Berners-Lee, J. Hendler et O. Lasilla, The Semantic Web. Scientific American, 284 (2001).*

[6] : *Neeches, Finin T, Fikes R.E, Gruber T.R, Senator T et Swartou W.R. « Enabling technology for knowledge sharing » AI Magazine. Vol.12, no 3, 1993.*

[7] : *Gruber T. (1993). A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2), 199-220.*

[8] : *Guarino N. (1997b). Understanding, building and using ontologies. International J. Human-Computer Studies, 46, 293-310.*

[9] : *Borst W. N. (1997). Construction of Engineering Ontologies. Center for Telematica and Information Technology, University of Twente, Enschede, NL.*

[10] : *Studer et al. 1998, Studer R., Benjamins R. et Fensel D. (1998). Knowledge Engineering : Principles and Methods. Data Knowledge Engineering.*

[11] : *H. Sofia Pinto, J.P. Martins, Reusing Ontologies, aifbhermes.aifb.uni-karlsruhe.de/AAAI2000/ CameraReady/HPinto00.pdf.*

[12] : *BORG96, A. Borgida. "On the relative expressiveness of description logics and predicate logics". Artificial Intelligence, volume 82, number 1-2, pages 353-367, 1996.*

[13] : *Blas et al. 98, BLAZQUEZ M. FERNANDEZ M., GARCIA-PINAR J. M. & GOMEZ-PEREZ A., Building Ontologies at the Knowledge Level using the Ontology Design Environment, in Proceedings of the Banff Workshop on Knowledge Acquisition for Knowledge-based Systems, 1998.*



- [14] : F. Furst, "L'ingénierie ontologique". *Rapport de recherche N°02-07. (2002).*
- [15] : M. Blazquez, M. Fernandez, J. Garcia-Pinar. et A. Gomez-Perez, "Building Ontologies at the Knowledge Level using the Ontology Design Environment", in *Proceedings of the Banff Workshop on Knowledge Acquisition for Knowledge-based Systems, (1998).*
- [16] : Melle Souheila KHALFI. (2009). *CONSTRUCTION D'UNE ONTOLOGIE POUR LA PRISE EN CHARGE DES PATIENTS À DOMICILE. Mémoire de magister, Université Mentouri de Constantine, Faculté des sciences de l'ingénieur, département informatique.*
- [17] : Mémoires de fin d'étude Mehamda Rima et Taloub Amina, *Développement d'un système d'information pour la gestion des ressources humaines la méthode UML, 2012*
- [18] : P. Roques et F. Vallée, *UML en action, 4<sup>ième</sup> édition 2007.*
- [19] : *Ontology Development 101 – A guide to creating your first ontology. KSL Technical Report, Stanford University, Stanford, CA, USA.*
- [20] : *Toward principles for the design of ontologies used for knowledge sharing. In N. Guarino, & R. Poli (Ed.), International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation.*
- [21] : *Stratified Ontologies: the case of physical objects. ECAI96. Workshop on Ontological Engineering , 5-15.*
- [22] : *METH ONTOLOGY: From Ontological Art Towards Ontological Engineering. Spring Symposium on Ontological Engineering of AAAI , 33–40.*
- [23] : *Ontologies: Principles, Methods and Applications. Knowledge Engineering Review , 11 (2), 93–155.*
- [24] : *Ontological Engineering (with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web). Springer.*