1. Introduction :

Le présent chapitre présente en détail, le processus suivi pour développer notre ontologie baptisée «**OntoADAPT**». En premier lieu, nous décrirons le processus que nous avons suivi pour la construction de l'ontologie. Ce processus est composé de cinq phases, à savoir : spécification des besoins, conceptualisation, formalisation, implémentation et évaluation.
En effet, ces différentes phases sont inspirées à partir de certaines méthodologies étudiées dans le chapitre 1. En deuxième lieu, nous montrerons comment ce processus a été utilisé pour la conception de l'**ontoADAPT**. 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 *SHOIN* qui possède une sémantique claire et procure des services d'inférences et des mécanismes de raisonnement puissants. Par la suite, nous décrirons comment nous avons utilisé l'outil PROTÉGÉ pour le codage d'**OntoADAPT**.
Enfin, nous illustrerons la phase de vérification.

2. Le processus suivi pour la construction de l'OntoADAPT :

Vu l'absence de méthodes standards, voire méthodologie pour le développement d'ontologies, l'emprunt des travaux déjà effectués dans ce sens est inévitable. Le choix de chaque étape dans la démarche s'effectue selon le besoin approprié dans le processus de construction. Les grandes étapes de processus sont inspirées de la méthodologie de construction d'ontologies «**METHONTOLOGY**».

Dans un premier lieu, il sera appliqué tout au long du processus adopté pour le développement de cette ontologie, les conseils de départ suivants :

- Il n'y a pas qu'une seule façon correcte pour modéliser un domaine, il y a d'autres alternatives à adopter en cas de besoin ;
- 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.

2.1 Etape 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. Dans ce présent document sera décrit :

♦ Le domaine de connaissance qui sera représenté par l'ontologie ;

♦ Identifier les objectifs de l'ontologie ;

♦ Déterminer Les futurs utilisateurs de l'ontologie ;

♦ Déterminer les sources d'informations desquelles les connaissances seront obtenues. Ils sont de différentes natures, 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. ;

♦ déterminer La portée de l'ontologie : déterminer la liste des termes candidats du domaine à travers l'analyse des sources d'informations relatif au domaine.

2.2 Etape 2 : La 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 (Fernández-López, *et al.*, 1997) 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 formelles, 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 :

- **Tâche 1 :** construire le glossaire des termes en identifiant tous les termes inclus dans l'ontologie, leurs définitions en langage naturel, leurs synonymes, et leurs acronymes s'ils existent.
- **Tâche 2 :** construire la taxonomie de concepts pour les classifier. Elle est réalisée à travers un choix d'une stratégie d'identification de concepts. Trois stratégies d'identification des concepts doivent être proposées, et qui sont [19] :

- ❖ 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.

- **Tâche 3 :** construire le diagramme de relations binaires où les relations entre les concepts de l'ontologie sont identifiés.
- **Tâche 4 :** construire le dictionnaire de concepts qui inclut principalement les instances de chaque concept, leurs attributs et leurs relations.
- **Tâche 5 :** construire la table de relations binaires en décrivant les relations apparues dans le diagramme de relations binaires.
- **Tâche 6 :** description des attributs apparus dans le dictionnaire de concepts, en spécifiant leur contraintes dans une table d'attributs.
- **Tâche 7 :** description des instances des concepts dans une table d'instances.

La description de chaque tâche est définie au fur et à mesure de la construction de l'ontologie ***OntoADAPT.***

2.3. Formalisation :

Le processus de construction d'ontologie est suivi par une étape de formalisation. Cette étape 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*, qui présente une logique de description très expressive et offre un certain nombre de constructeurs permettant de représenter des concepts (aussi appelés *classes*) d'un domaine d'application et les propriétés (aussi appelées *rôles*), qui représentent des relations binaires pouvant être établies entre ces classes.

Nom du constructeur	Syntaxe	Sémantique
Top (Thing)	T	$T^I = \Delta^I$
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	$\{o_1, o_2, ..., o_N\}$	$\{o_1, o_2, ..., o_N\}^I = \{o_1^I, o_2^I, ..., o_N^I\}$
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	$\forall R_1.C_1$	$(\forall R_1.C_1)^I = \{x \mid \forall x.(x,y) \in R_1^I \text{ then } y \in C_1^I\}$
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	$(\geq n\ R_1)$ $(\leq n\ R_1)$ $(= n\ R_1)$	$(\geq n R_1)^I = \{x \mid \text{card}\{y \mid (x,y) \in R^I{}_1\} \geq n\}$ $(\leq n R_1)^I = \{x \mid \text{card}\{y \mid (x,y) \in R^I{}_1\} \leq n\}$ $(= n R_1)^I = \{x \mid \text{card}\{y \mid (x,y) \in R^I{}_1\} = n\}$

Tableau 4.1 : syntaxe et sémantique de SHOIN.

SHOIN permet de définir les connaissances selon deux niveaux de description :

le niveau terminologique (TBox) : décrit les concepts et les rôles entre les concepts d'un domaine ;

Le niveau assertionnel (ABox) : décrivant les individus.

2.3.1La partie terminologique (T-BOX)

La T-BOX, ou base terminologique, est un ensemble fini d'axiomes d'inclusions et d'équivalences de classes ainsi que d'axiomes d'équivalence et d'inclusions de rôles. Ils sont utIlisés pour définir les classes et les rôles relatifs à un domaine spécifique. D'où les concepts atomiques et rôles atomiques constituent les entités élémentaires d'une TBOX, les noms débutant par une lettre majuscule désignent les concepts alors que ceux débutant par une lettre minuscule dénomment les rôles. (Tableau 4.2) illustre les axiomes d'inclusions et d'équivalence des classes et des rôles.

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 4.2 : syntaxe des axiomes avec leurs sémantiques.

2.3.2 La partie assertionelle (ABOX) :

Une ABOX contient un ensemble d'assertions sur les individus. Chaque ABOX doit être associé à une TBOX, car les assertions s'expriment en termes de concepts et de rôles de la TBOX.

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 4.3: syntaxe des assertions avec leurs sémantiques

2.4 Implémentation :

L'ontologie obtenue dans l'étape précédente est une ontologie 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.

2.5 Évaluation :

C'est une étape qui peut être effectué 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'ontologiste à construire et à évaluer l'ontologie ;
- *Évaluation de l'ontologie formelle et opérationnelle* : elle est réalisée automatiquement au moyen de l'utilisation d'un raisonneur capable de calculer les inférences dans le but d'améliorer la qualité de l'ontologie.

3. développement d'une ontologie ontoADAPT :

Dans cette section, ***l'Ontoadapt*** sera construite. Selon les critères de classification précédemment évoqués et étudiés, ***Ontoadapt*** est une ontologie qui permet la découverte et la sélection des dispositifs, des media et des services d'adaptations.

Ainsi, il faut noter qu'il n'est pas possible de savoir dès le début, que les termes collectés sont suffisants pour répondre à l'objectif pour lequel l'ontologie a été construite, de nouveaux termes sont ajoutés lorsqu'il est nécessaire, de même d'autres sont retirés lorsqu'ils sont jugés inutiles jusqu'à l'obtention des termes pertinents du domaine.

3.1 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 :**

L'ontologie construite s'inscrit dans le cadre de la découverte et la sélection des dispositifs des media et des services d'adaptations.

- **Objectif :**

Facilite la découverte des propriétés des dispositifs et les services.

- **Utilisateurs :**

Les services intelligents :upnp. L'interface web .

- **Sources d'informations :**

Documents techniques sur la maison intelligente et les services intelligents possibles entre les dispositifs.

- **Portée de l'ontologie :**

DLNA,DMS, DMP, contexte, media …….

3.2 Conceptualisation :

Dans cette étape on distingue les principales tâches suivantes :

3.2.1 Construction du glossaire de termes :

Construire un glossaire de termes est la première tâche à effectuer dans l'étape de Conceptualisation. Il recueille et décrit tous les termes (concepts, instances, attributs, relations entre les concepts, etc.) qui sont utiles et potentiellement utilisables dans l'ontologie. Le tableau 4.4 illustre le glossaire de termes d'OntoADAPT :

NOM	Description
Dispositifs	C'est l'ensemble des dispositifs dans une maison intelligente.
Services	C'est l'ensemble des services d'adaptation.
Technique d'Adaptation	C'est l'ensemble des techniques utilisées pour adapter le media.
DMP	Digital Media Player, Ces équipements lisent des contenus sur les DMS et les affichent. Dans mon exemple, le téléviseur, le lecteur Blue-Ray connecté, l'ampli audio-video, le cadre photo, labox et les lecteurs Windows Media player ou XBMC sont des DMP.
DMS	Digital Media server Ces équipements fournissent des contenus (vidéos, musiques, photos) qui seront lus par des DMPs et affichés par DMRs. Par exemple les serveurs de fichiers, les PC, le cadre photo et la box sont des DMS**.**
DMR	Digital Media rendrer Ces équipements jouent le contenu fourni par un DMC à partir d'un DMS. Les téléviseurs, les appareils audio-vidéo fonctionnent comme des DMR. A noter que le même équipement, par exemple un

	téléviseur, peut fonctionner comme un lecteur (DMP) qui va chercher des contenus sur un serveur DMS ou comme un DMR qui reçoit du contenu d'un DMC.
DMC	Digital Media Controller Ces équipements recherchent des contenus sur des DMS et les jouent sur des DMR. C'est le cas des tablettes ou des mobiles comme dans l'exemple que nous verrons plus loin.
DMP	Digital Media printer Comme le nom l'indique, ces équipements fournissent des services d'imprimantes aux DMP et DMC.
Contexte	l'information élémentaire recherchée : Cette information élémentaire peut avoir des différentes formes : énergie, mémoire ….ect.
Audio	Contenu de media.
Vidéo	Contenu de media.
Texte	Contenu de media.
Service d'adaptation transmodage	permet de changer le type de média. Par exemple : JPEG vers PNG.
Service d'adaptation	Cette classe consiste à transformer par exemple la qualité d'un objet multimédia en une autre qualité.
(…..)	**(…..)**

Tableau 4.4 : glossaire de terme.

3.2.2 Classification des concepts en hiérarchies de concepts :

Cette tâche consiste à définir les relations taxonomiques entre les concepts définis dans leGlossaire de termes.

En souvenir de l'existence de trois stratégies d'identification de concepts, nous avons réellement combiné les trois stratégies pour aboutir aux hiérarchies de la figure 4.1.

En outre, METHONTOLOLY propose d'utiliser quatre relations taxonomiques qui sont sous-classe de qui s'exprime comme suit : Un concept C1 est sous classe d'un autre concept C2, si et seulement, si toute instance deC1 est aussi une instance de C2.

La figure 4.1 illustre les hiérarchies de concepts.

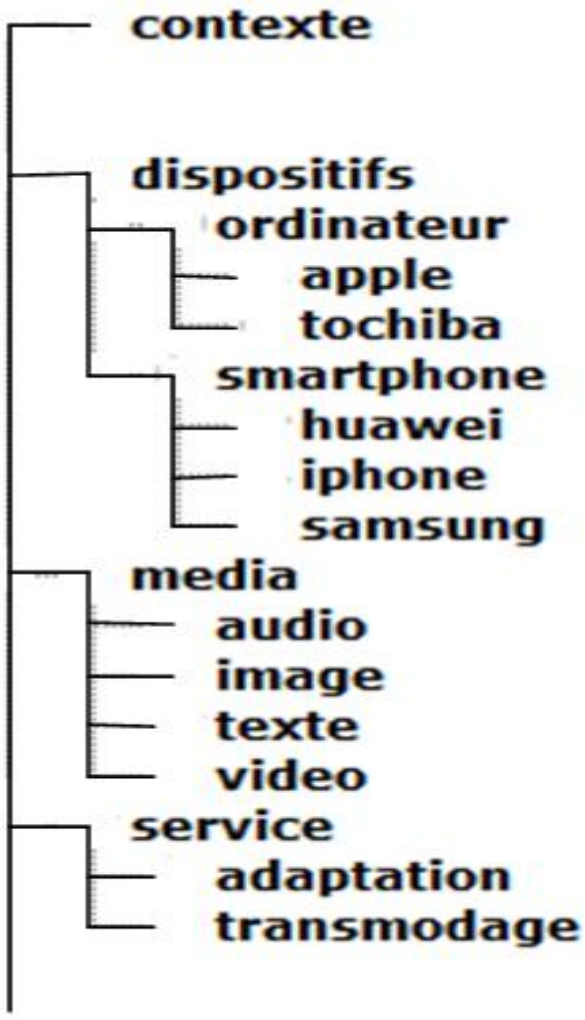

Figure4.1 : classification de concepts.

3.2.3 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 desconcepts 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 (voir figure4.2).

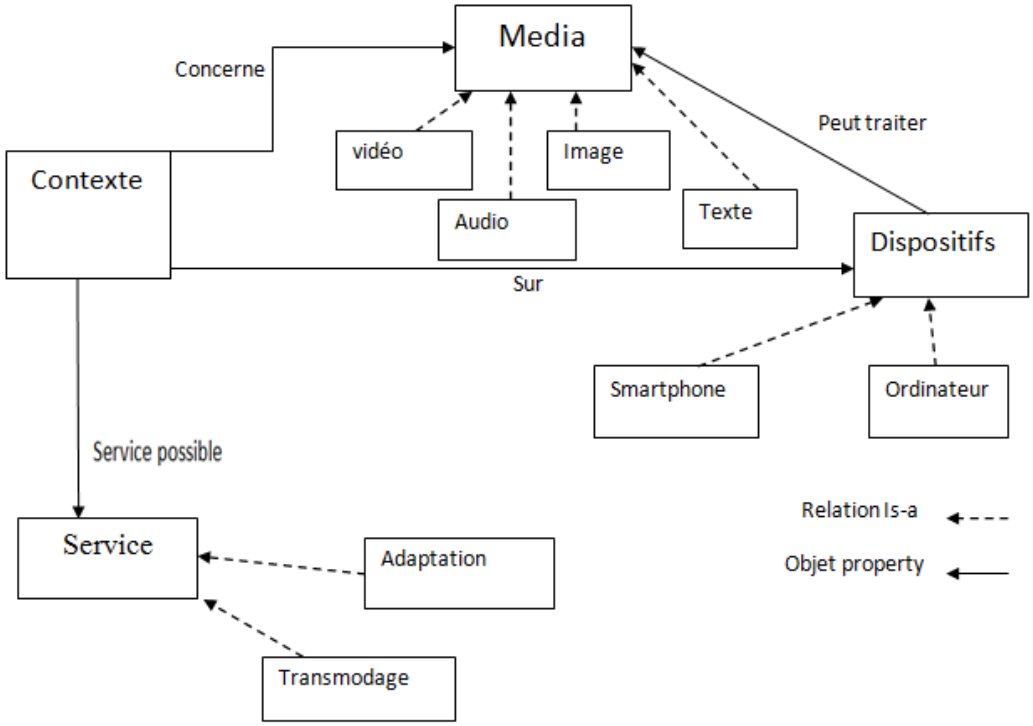

Figure 4.2 : le diagramme de relations binaires

3.2.4 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.

Nom de concept	attributs	instances	relations
Media	Format taille		
Contexte	Energie de batterie c Taille de mémoire c		Concerne sur
Audio	Format taille	Aiff M4p mmf	
Vidéo	Format taille	Mp4 M4v SVI WMV	
Image	Format taille	Bmp Jpg png	
Texte			
Dispositifs			Peut traiter
Ordinateur	Energie de batterie o Cpu o		
Smartphone	Energie de batterie s Cpu s		
Apple		MacBook 1.1 MacBook 3.1 MacBook 5.2	
Toshiba			
Huawei		Huawei y5	
i phone		i phone 4s i phone 5s	
Samsung		Samsung i9082 Samsung s5360 Samsung s6810	

Tableau 4.5 : dictionnaire de concepts.

3.2.5Construction 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 utilisé 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.

Nom de la relation	Concept source	Concept cible	Card source	Card cible
concerne	contexte	media	(1.1)	(0.n)
Peut traiter	dispositifs	media	(1.1)	(0.n)
Sur	contexte	dispositifs	(1.1)	(1.1)
Service possible	contexte	service	(1.1)	(0.1)
(…)	(…)	(…)	(…)	(…)

Tableau 4.6 : table des relations binaires.

3.2.6Construction de la table d'attributs :

La table des attributs comporte une description détaillée des attributs inclusdans le dictionnaire de concepts, et l'ensemble de contraintes et de restrictions sur ces valeurs.

Nom de l'attribut	Nom du concept	Type de valeur	Cardinalité
Cpu o	ordinateur	Name	(1.1)
Energie de batterie s	Smartphone	float	(1.n)
Taille de mémoire c	contexte	Name	(1.1)
Energie de batterie c	contexte	float	(1.n)
Format	media	Name	(1.1)
Taille	media	float	(1.1)
Energie de batterie	ordinateur	float	(1.n)
Taille mémoire s	Smartphone	Name	(1.1)
Taille mémoire o	ordinateur	Name	(1.1)
Energie de batterie s	Smartphone	float	(1.n)
Cpu s	Smartphone	Name	(1.1)
(…)	(…)	(…)	(…)

Tableau 4.7 : table d'attributs.

3.2.7Construction 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.

Nom de l'instance	Nom de concept	attributs	valeurs
Aiff	audio		
Huawei y5	huawei		
I phone 4s	i phone	Cpu s Energie de batterie s Taille de mémoire s	
I phone 5s	i phone	Cpu s Energie de batterie s Taille de mémoire s	
M4p	audio		
M4v	vidéo		
MacBook 1.1	Apple	Cpu o Energie de batterie	2.0 GHZ

		Taille de mémoire o	
MacBook 5.2	Apple	Cpu o Energie de batterie Taille de mémoire o	2.13GHZ
MacBook 3.1	Apple	Cpu o Energie de batterie Taille de mémoire o	2.2 GHZ
Mmf	audio	Taille format	
Mp4	vidéo	Taille format	
SVI	vidéo	Taille format	
Wma	audio	Taille format	
Wmv	vidéo		
Samsung i9082	Samsung	Cpu s	1.2 GHZ
Samsung s5360	Samsung	Cpu s	Qualcomm MSM7227 Snapdragon - 800 MHz
Samsung s6810	Samsung	Cpu s	Cortex A9 - 1 GHz
(.....)	(……)	(……)	(……)

Tableau 4.8 : table des instances

3.3 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* composé de deux parties T-BOX et A-BOX.

3.3.1Construction de TBOX :

concept	définition	Relation de subsomption
dispositifs	(∃acpu. cpu)∩(∃ a batterie. énergie de batterie)∩(∃ a mémoire. Taille de mémoire) ∩(≥1 peut traiter. media)	Dispositifs ⊆ Thing
media	(∃ a format. format) ∩ (∃ a taille. Taille)	media ⊆ Thing
contexte	(∃ a taille. Taille de mémoire) ∩(≥1 concerne. media) ∩ (=1 sur. dispositifs) (≥1 service possible. service)	contexte ⊆ Thing
ordinateur	dispositifs ∩ (∃acpu o. cpu o)∩(∃ a batterie o. énergie de batterie)∩(∃ a mémoire. Taille de mémoire) ∩(≥1 peut traiter. media)	service ⊆ Thing
(….)	(…..)	(….)

Tableau 4.9 : définition de concepts(TBOX).

En ce qui concerne les rôles, seront définis en donnant les couples des concepts sources et cibles de chacune, c'est-à-dire le couple (domaine, Co-domaine) .

Rôles	**couple (domaine, Co-domaine)**
Concerne	(contexte, média)
Peut traiter	(dispositifs, média)
Service possible	(contexte, service)
Sur	(contexte, dispositifs)

Tableau 4.10 : définition des rôles (TBOX)

3.3.2Construction d'ABOX :

Les instances seront décries en utilisant le langage assertionnel de la manière suivante :

A (C) : pour spécifier qu'A est une instance de la classe C.

Audio	Aiff(audio)
Apple	MacBook 1.1(apple)
Image	Bmp (image)
Samsung	Samsung i9082(samsung)
Audio	M4p(audio)
Apple	MacBook 3.1(apple)
Audio	Mmf(audio)
Image	Jpg(image)

Iphone	i phone 5s(iphone)
Huawei	Huawei y5(Huawei)
Video	M4v(video)
Image	Jpg(image)
(…..)	(…….)

Tableau 4.11 Description assertionnelle des individus(ABOX)

3.4 Implémentation :

Après avoir conçu l'ontologie, elle doit être opérationnelle pour qu'elle soit concrètement manipulable dans un système informatique. A cette fin, l'ontologie doit être spécifiée dans un langage de représentation de connaissance doté de capacités d'inférences. Pour ce faire, le choix s'est porté sur **OWL** qui représente un langage de codification utilisé pour implémenter les ontologies, mais aussi pour toutes les fonctionnalités sémantiques qu'il permet et qui sont plus riches que celles des langages **RDFS & DAML+OIL.**

3.4.1 Présentation de l'éditeur PROTEGE :

PROTEGE est une interface modulaire développée par le « Stanford Médical Informatics » au sein de l'université de Stanford, permettant l'édition, la visualisation, le contrôle, l'extraction à partir de sources textuelles, et la fusion semi-automatique d'ontologie. Le modèle de représentation de connaissances de PROTEGE est issu du paradigme de frames et contient des classes (pour modéliserles concepts), des slots (pour modéliser les propriétés des concepts), et des facettes (pour définir les valeurs des propriétés et des contraintes sur ces valeurs), ainsi que des instances des classes. L'architecture de PROTEGE se compose de deux parties : « le model » et « la vue ». Le model est le mécanisme de représentation interne pour les ontologies et les composants, tandis que la vue fournit une interface utilisateur pour l'affichage et la manipulation du model sous-jacent.

L'interface très complète ainsi que l'architecture logicielle ouverte et extensible permettant l'insertion de plug-ins, ont grandement participé au succès de PROTEGE. Ces plug-ins pouvant apporter de nouvelles fonctionnalités entre autre « **OWLVIZ** » qui permet de gérer

des représentations sous forme graphique, « **PROMPT** » qui est dédié la gestion des plusieurs ontologies, en particulier à leur fusion [14].

3.4.2 Les étapes de l'édition de L'ontoADAPT sous PROTEGE :

Après avoir installé et lancé PROTEGE sous Windows, une fenêtre principale s'ouvre. Cette dernière fournit un ensemble d'onglets. Parmi les plus importants et qui seront fréquemment utilisés pour l'édition des composants de l'ontologie sont :

- **L'onglet Classes** : affiche l'hiérarchie de classes de l'ontologie. Il permet aux développeurs de créer et d'éditer les classes et affiche le résultat de la classification.
- **L'onglet Objet Properties** : est utilisé pour créer et éditer les relations entre les différentes classes de l'ontologie.
- **L'onglet Data Properties** : permet de créer les attributs de chaque classe.
- **L'onglet Individuals** : est utilisé pour créer et éditer les instances.

La figure suivante (Figure 4.3) présente l'interface principale de PRTEGE (version 4.3) ainsi que ces différents onglets.

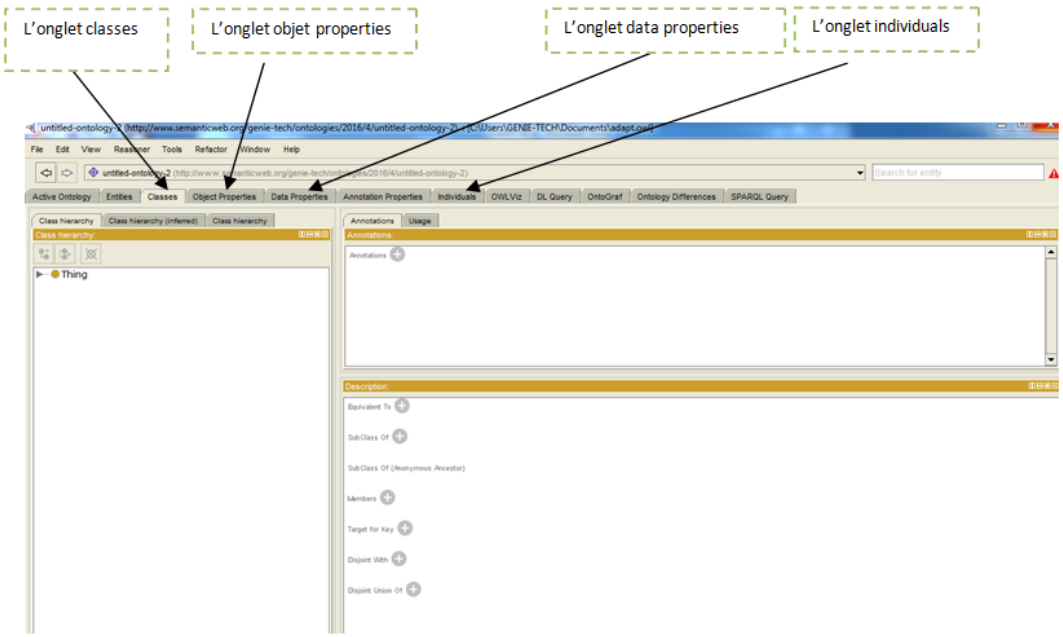

Figure 4.3 : interface de protégé.

3.4.2.1 Création des classes et la hiérarchie des classes :

L'onglet **Classes,** permet d'éditer toutes les classes et les hiérarchies de l'ontologie spécifiées dans l'étape de conceptualisation. Une classe universelle « **owl : thing** » est utilisée comme racine pour cette hiérarchie, et la création des sous-classes se fait par le choix de la classe mère, suivi par un simple clic sur le bouton de création des sous-classes de la classe sélectionnée.

La figure suivante **(figure 4.4),** présente comment créer, supprimer et commenter un concept sous PROTEGE.

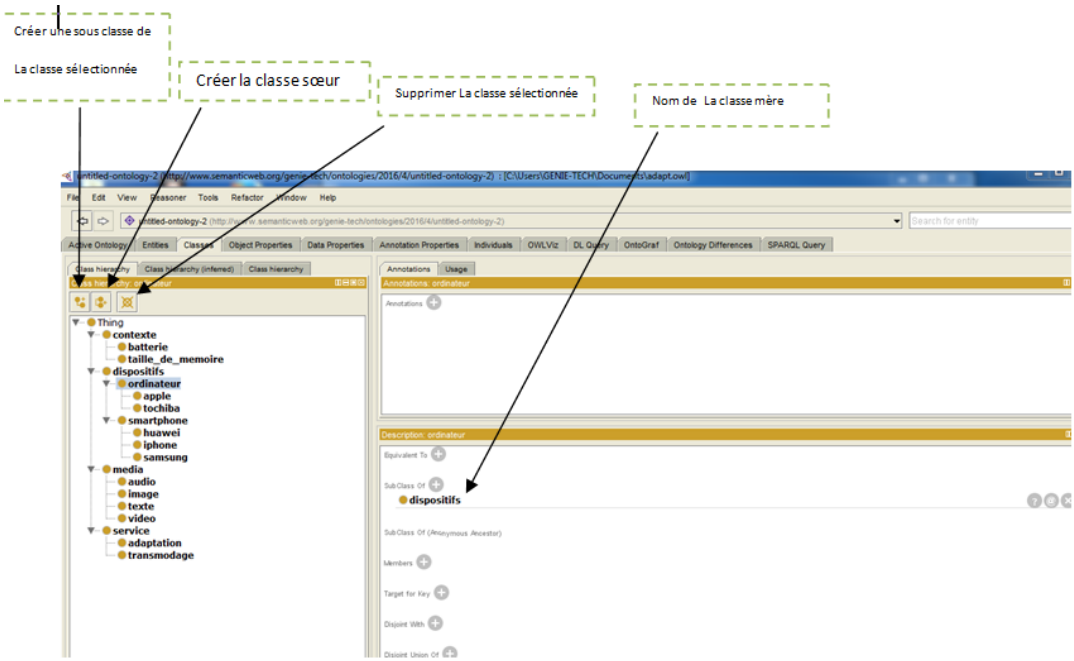

Figure 4.4 : édition de concepts

3.4.2.2 Création des propriétés (ObjectsProperties) :

La création des relations est l'étape qui suit l'étape de construction des classes. Une relation permet de relier les classes « concepts » entre elles. La création d'une relation entre deux classes est possible par une simple clic sur le bouton **Object Properties.** Cetteaction permet de générer une nouvelle fenêtre **(figure 4.5)** où sera spécifie le nom de la relation. Le domaine, le Co-domaine, la relation inverse si elle existe ainsi que les caractéristiques de la relation (functional, symétrique, transitive…).

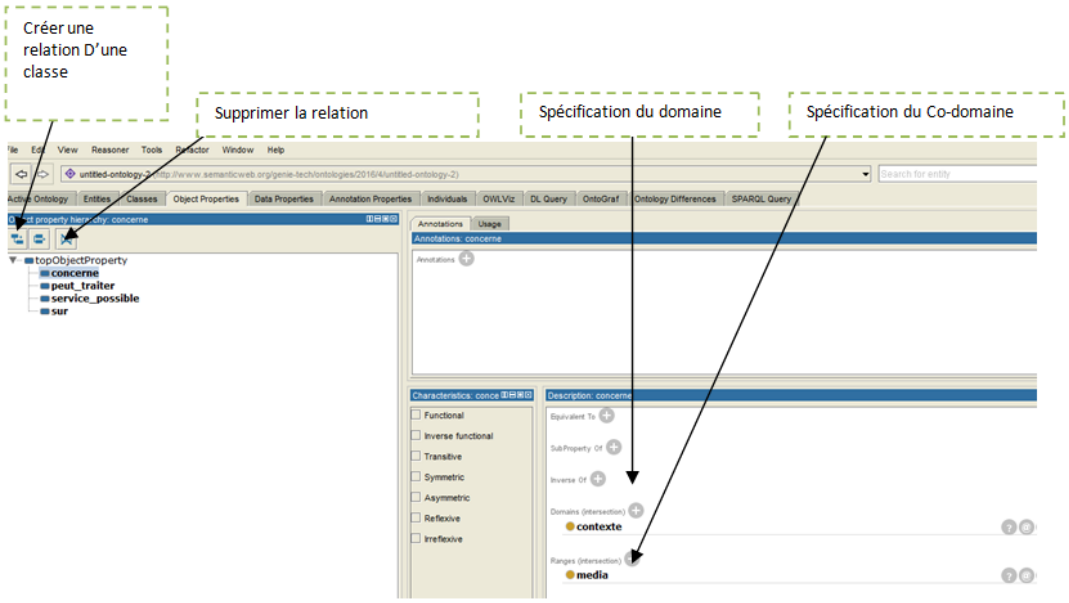

Figure 4.5 : création des propriétés pour une classe.

3.4.2.3 Création des attributs (Data Properties) :

Un attribut permet de relier des individus à des valeurs de données. Une valeur étant un élément d'un des types de données prédéfinis (float, string, Int, etc…).

Pour créer un attribut d'une classe, il faut cliquer sur le bouton de création d'une nouvelle propriété : **Data Properties**. Cette action génère une fenêtre comme montre la figure ci-après **(Figure 4.6)** où le nom de l'attribut, le domaine, le type associé, les valeurs permises s'ils ont connues sont spécifiées.

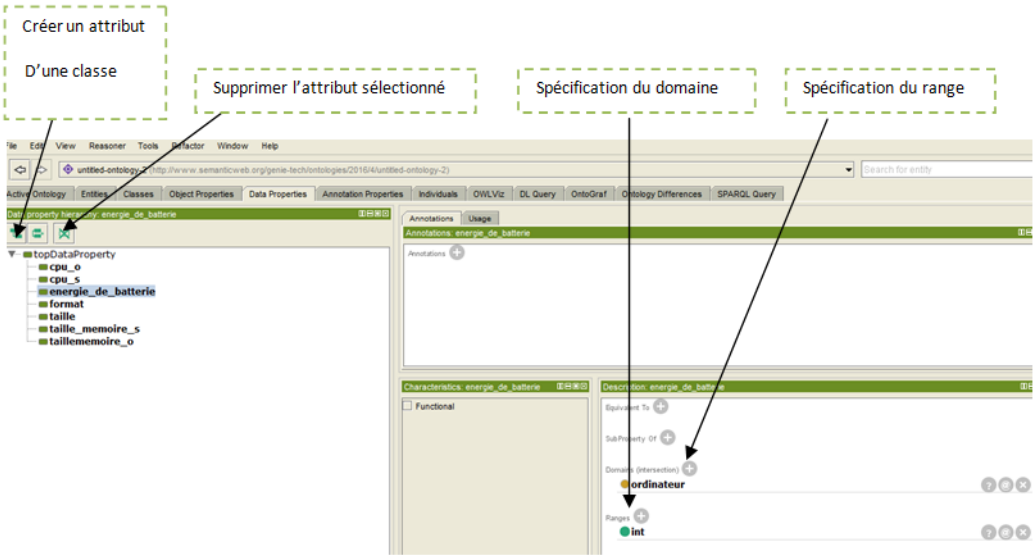

Figure 4.6 création des attributs.

3.4.2.4 Création des individus (Individuals) :

La dernière étape consiste à créer les instances (individus) des classes. Donc, la définition d'une instance individuelle d'une classe exige :

- Choisir une classe ;
- Créer une instance individuelle de cette classe ;
- La renseigner avec les valeurs des attributs.

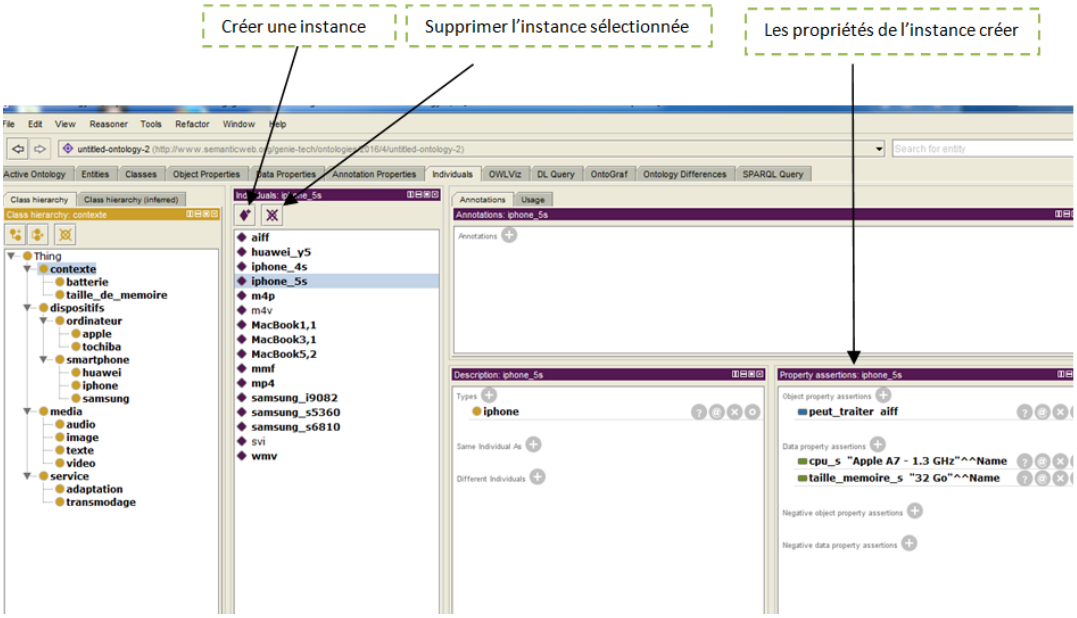

Figure 4.7 création des instances (individus) d'une classe.

Conclusion :

À travers tout ce que nous avons réalisé dans ce chapitre, nous pouvons dire que le processus suivi pour la construction d'OntoADAPT, nous a permis de réussir finalement à construire une ontologie de domaine, semi formelle.