

## **Rapport de Travail Pratique**

### **Module : RCR - Représentation des Connaissances et Raisonnement**

2<sup>ème</sup> année Cycle Supérieur (2CS)

Option : Systèmes Intelligents et Données (SID)

Groupe : 2SD1

---

### **Thème:**

Ontologie pour la Gestion des Systèmes de Santé  
Intelligents

---

#### **Réalisé par:**

- ABOUD Ibrahim
- BOUYAKOUB Rayane

#### **Proposé par:**

- Mme. BOUSSAHA Ryma

---

Année universitaire : 2024 - 2025

# Table des matières

<b>Table des matières.....</b>	<b>1</b>
<b>I. Partie Théorique.....</b>	<b>2</b>
1. Ontologie.....	2
1.1. Definition.....	2
1.2. Composants de l'ontologie.....	2
1.3. Importance dans la gestion des connaissances.....	3
2. Language OWL.....	3
2.1. Definition et presentation.....	3
2.2. Caractéristiques.....	3
2.3. Les Sous-langages.....	4
2.4. Utilité pour la création des ontologies complexes.....	4
<b>II. Partie Pratique.....</b>	<b>5</b>
1. Documentation de l'Ontologie du TP.....	5
1.1. Classes.....	5
1.2. Propriété.....	9
1.3. Relations.....	10
2. Diagrammes illustratives.....	12
3. Détails techniques.....	18
3.1. Définition de classes supplémentaires.....	18
3.2. Vérification de la cohérence de l'ontologie.....	18
3.3. Tester l'ontologie avec un scénario concret.....	18
4. Perspectives et améliorations possibles.....	21
<b>III. References.....</b>	<b>22</b>

# I. Partie Théorique

## 1. Ontologie

### 1.1. Definition

En philosophie, l'Ontologie est une branche fondamentale de la métaphysique, qui s'intéresse à la notion d'existence, aux catégories fondamentales de l'existant et étudie les propriétés les plus générales de l'être.

Le dictionnaire d'oxford définit l'ontologie comme un ensemble de concepts et de catégories dans un domaine ou un sujet donné, qui montre leurs propriétés et les relations entre eux.

### 1.2. Composants de l'ontologie

En analysant les deux definitions precedentes, on peut déduire les composantes principales de l'ontologie :

- **Classes (ou Concepts) :** Ce sont des catégories abstraites représentant des objets ou des idées d'un domaine donné, définies par des propriétés ou caractéristiques communes. Une classe peut être spécialisée en sous-classes qui héritent de ses propriétés. Exemple : Médecin, Patient.
- **Instances (Individus) :** Ce sont les entités concrètes appartenant à une classe. Exemple : Smith, instance de la classe Médecin ; John, instance de la classe Patient.
- **Propriétés (Attributs) :** Ce sont les attributs qui décrivent les caractéristiques de cette classe. Par exemple, la classe "Patient" peut avoir les propriétés suivantes : nom, âge, poids, dateNaissance, etc.
- **Relations:** Ce sont des liens sémantiques qui associent deux classes, et donc, en pratique, deux instances. Elles expriment des interactions ou des connexions entre les entités du domaine. Exemple : John a consulté Smith (relation entre une instance de Patient et une instance de Médecin).

### 1.3. Importance dans la gestion des connaissances

- **Structuration et formalisation des connaissances** : Les ontologies permettent de représenter explicitement les connaissances d'un domaine donné. Exemple: SNOMED CT, une ontologie médicale qui standardise la terminologie utilisée dans le domaine de la santé.
- **Interopérabilité entre systèmes** : En unifiant la structure des connaissances, les ontologies facilitent l'échange d'informations de manière cohérente entre différents systèmes.
- **Inférence automatique** : Grâce à une structure de connaissances formelle, les machines peuvent raisonner sur les données en appliquant des règles logiques, ce qui permet de découvrir de nouvelles informations implicites.
- **Scalabilité et réutilisabilité**: Une ontologie bien conçue est modulaire, réutilisable et extensible, ce qui facilite son évolution et son adaptation à de nouveaux besoins.

## 2. Language OWL

### 2.1. Definition et presentation

Le langage d'ontologie du Web (OWL) du W3C est un langage du Web sémantique conçu pour représenter des connaissances riches et complexes sur des entités, des groupes d'entités, ainsi que les relations entre elles. Basé sur une logique computationnelle, OWL permet aux programmes informatiques de raisonner automatiquement sur les connaissances qu'il exprime. Les documents OWL, appelés ontologies, peuvent être publiés sur le World Wide Web et peuvent faire référence à d'autres ontologies OWL ou être référencés par elles, favorisant ainsi l'interconnexion et la réutilisation des connaissances.

### 2.2. Caractéristiques

- **Langage sémantique basé sur la logique** : OWL permet de représenter les concepts, propriétés et relations de manière exploitable par des machines.
- **Hierarchie des classes** : OWL permet la définition de classes et de sous-classes.
- **Utilisation des standards (XML/RDF)** : Compatibilité avec d'autres technologies du web sémantique.

- **Propriétés riches** : Les propriétés peuvent être enrichies par des informations telles que le domaine (domain), la portée ou co-domaine (range), le type de propriété (fonctionnelle, transitive, symétrique, etc), offrant une grande expressivité sémantique.

## 2.3. Les Sous-langages

- **OWL lite** : Utilisé pour les ontologies simples, limité dans les contextes avancés et complexes.
- **OWL Description Logic** : Basé sur la logique de description, il garantit la complétude et la décidabilité, cette garantie produit quelques restrictions sur son usage.
- **OWL Full** : Le sous-langage le plus expressif parmi les trois, utilisé pour les ontologies très complexes, avec une flexibilité syntaxique maximale, , mais ceci compromet la garantie de la décidabilité.

## 2.4. Utilité pour la création des ontologies complexes

La puissance d'OWL dans la modélisation d'ontologies complexes se manifeste à travers les aspects suivants :

- Ajout de contraintes entre les concepts. Exemple : On peut spécifier qu'un Patient ne peut être suivi que par un Médecin, à l'aide d'une propriété comme `estSuiviPar` dont le domaine est Patient et le codomaine (range) est Médecin. Cela empêche, par exemple, qu'un patient soit lié à une entité qui n'est pas un médecin.
- Support du raisonnement automatique qui permet de vérifier la cohérence du modèle et de détecter d'éventuelles incohérences logiques.
- Capacité à organiser les concepts selon des relations de spécialisation/généralisation (classes et sous-classes).
- Prise en charge de spécifications avancées comme la cardinalité, les types de propriétés et les disjonctions entre classes.

## II. Partie Pratique

### 1. Documentation de l'Ontologie du TP

À partir de l'énoncé du TP, il est possible de modéliser un système de santé intelligent en s'appuyant sur des classes, des propriétés et des relations appropriées.

#### 1.1. Classes

Les classes ont été définies dans Protégé à l'aide de l'onglet « Classes ».

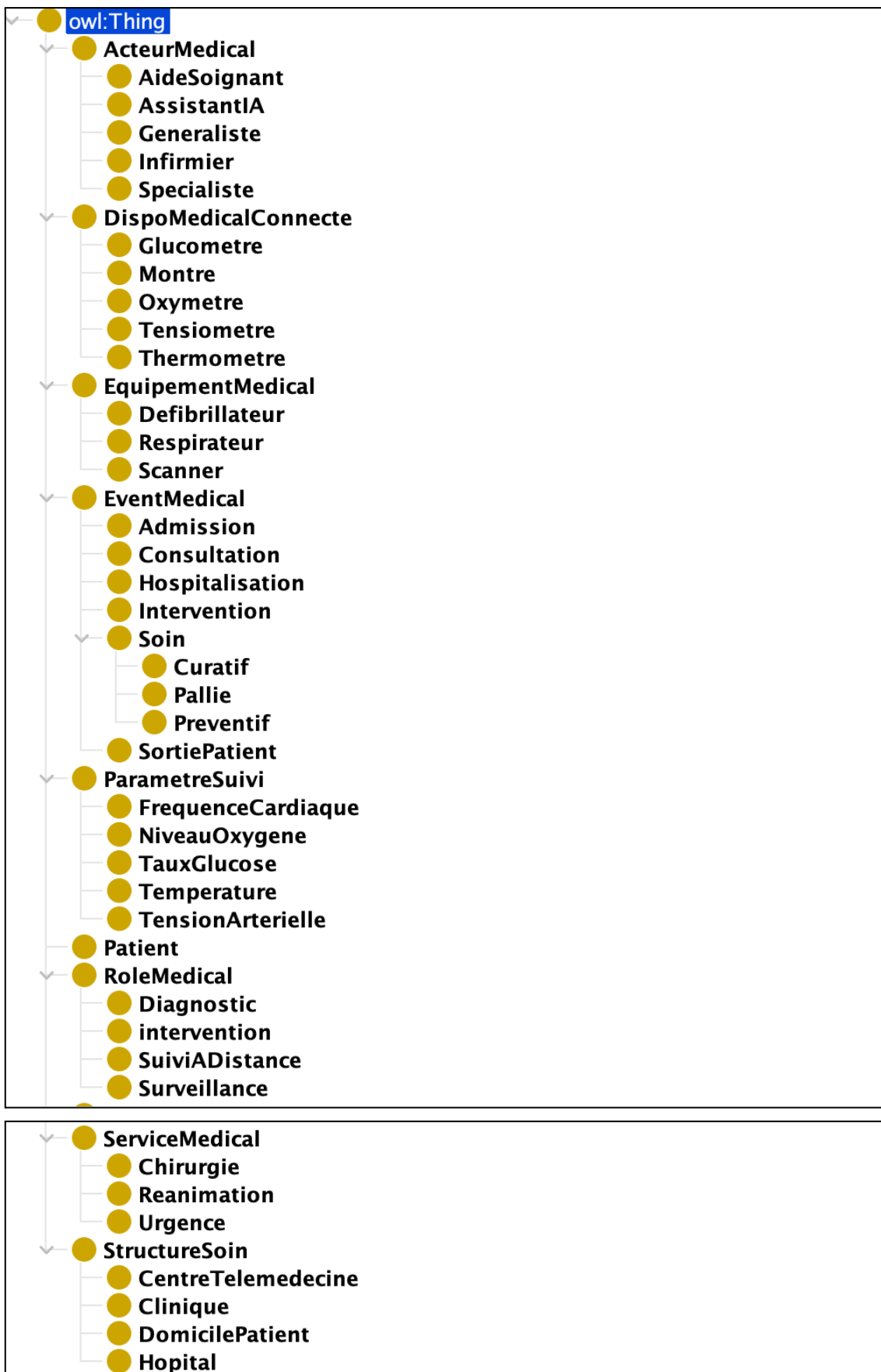
- **Patient** : Représente tout patient ayant un cas potentiel à traiter dans une structure de soins donnée.
- **StructureSoin** : Désigne le lieu ou la procédure dans lequel le traitement du patient aura lieu. Cette classe contient les sous-classes suivantes :
  - CentreTelemedecine : Permet un diagnostic à distance du patient.
  - Clinique : Un centre de traitement avec une capacité plus ou moins petite.
  - DomicilePatient : Représente le domicile du patient comme centre de traitement, notamment lorsque le patient ne peut pas se déplacer ou en cas de saturation des structures médicales.
  - Hopital : Centre de traitement disposant d'une capacité d'accueil importante.
- **ServiceMedical** : Service médical offert par une structure de soins donnée. Cette classe contient les sous classes suivantes :
  - Chirurgie : Service dédié aux interventions chirurgicales.
  - Reanimation : Service spécialisé dans les soins intensifs et la réanimation.
  - Urgence : Service chargé de la prise en charge rapide des cas critiques.
- **EquipementMedical** : Outils utilisés par certaines structures de soins. Cette classe comprend les sous-classes suivantes :
  - Defibrillateur: Appareil utilisé pour rétablir un rythme cardiaque normal.
  - Respirateur: Dispositif d'assistance respiratoire.
  - Scanner: Appareil d'imagerie médicale permettant de visualiser l'intérieur du corps.

- **DispoMedicalConnecte** : Dispositifs médicaux connectés servant à mesurer certaines données de santé du patient. Cette classe comprend les sous-classes suivantes :
  - Montre : Montre connectée mesurant en continu les données vitales.
  - Oxymetre : Évalue le taux d'oxygène dans le sang.
  - Tensiometre : Mesure la pression artérielle.
  - Thermometre : Indique la température corporelle.
  - Glucometre : Analyse le taux de glucose dans le sang.
  
- **ActeurMedical** : Personnel travaillant dans un centre de soins. Cette classe inclut les sous-classes suivantes:
  - AideSoignant : Personnel de soutien aux soins quotidiens des patients.
  - AssistantIA: Système d'intelligence artificielle assistant les professionnels de santé.
  - Generaliste : Médecin généraliste.
  - Infirmier : Personnel médical chargé des soins directs.
  - Specialiste : Médecin spécialisé dans un domaine spécifique. Exemple: cardiologue, neurologue, etc.
  
- **RoleMedical** : Rôle attribué à un acteur médical. Cette classe comprend les sous-classes suivantes:
  - Diagnostic : Rôle consistant à identifier l'état de santé du patient.
  - Intervention : Rôle impliqué dans une action médicale ou chirurgicale.
  - SuiviADistance : Suivi médical du patient à distance, souvent soutenu par des outils connectés.
  - Surveillance : Observation continue ou périodique de l'état du patient.

- **EventMedical** : Événement se déroulant dans une structure de soins concernant un patient. Cette classe contient les sous-classes suivantes :
  - Admission : Entrée officielle d'un patient dans une structure de soins.
  - Consultation : Rencontre entre un patient et un professionnel de santé.
  - Hospitalisation: Séjour prolongé dans une structure pour un traitement.
  - Intervention: Procédure médicale ou chirurgicale appliquée au patient.
  - Soins : Traitement prescrit pour le patient, avec les sous classes suivantes:
    - Curatif : Vise à guérir la pathologie du patient.
    - Palliatif: Vise à soulager les symptômes sans guérir la maladie.
    - Préventif: Vise à empêcher l'apparition d'une maladie.
  - SortiePatient : Action marquant la fin du séjour médical du patient dans la structure (ex. : retour au domicile, transfert, etc).
  
- **ParametreSuivi**: Représente les paramètres vitaux ou biologiques suivis chez un patient, généralement à l'aide de dispositifs médicaux connectés. Ces paramètres sont essentiels pour l'évaluation continue de l'état de santé du patient. Sous-classes :
  - FrequenceCardiaque : Mesure du rythme des battements cardiaques du patient.
  - NiveauOxygene : Taux de saturation en oxygène dans le sang.
  - TauxGlucose : Concentration de glucose dans le sang.
  - Temperature : Température corporelle du patient.
  - TensionArterielle : Pression exercée par le sang sur les parois des artères.

Voici ci-dessous une capture d'écran de la structure finale des classes réalisée dans Protégé.





## 1.2. Propriété

Nous avons ajouté des propriétés que nous avons jugées nécessaires pour un sous-ensemble de classes, à l'aide de l'onglet « Data Properties » du logiciel Protégé.

Classe	Propriété	Description	Type
Patient	idPatient	Identifiant unique du patient	Integer
	prenom	Prénom du patient	String
	nom	Nom du patient	String
	age	Âge du patient	Integer
	groupSang	Groupe sanguin du patient	Integer
	sexe	Sexe du patient (true = homme, false = femme)	Boolean
	antecedantMed	Antécédents médicaux du patient	String
DispoMedicalConnecte	idDispositif	Identifiant unique du dispositif médical	Integer
	model	Nom ou numéro de modèle du dispositif	String
	precision	Précision de mesure du dispositif	Float
	frequenceEch	Fréquence d'échantillonnage en Hertz	Integer
StructureSoin	nombreLits	Nombre de lits disponibles dans la structure	Integer
Soin	dateDebut	Date et heure de début du traitement	DateTimeStamp
	dateFin	Date et heure de fin du traitement	DateTimeStamp
	urgence	Indique si le traitement était urgent	Boolean
	results	Résultat ou résumé du traitement	String

ParametreSuivi	mesure	Représente la valeur numérique observée ou enregistrée pour un paramètre de suivi donné à un instant donné.	Float
----------------	--------	---	-------

owl:topDataProperty
age
antecedantMed
dateDebut
dateFin
frequenceEch
groupSang
idDispositif
idPatient
mesure
model
nom
nombreLits
precision
prenom
results
sexe
urgence

### 1.3. Relations

La définition des classes nous a permis d'établir les relations suivantes, en utilisant l'onglet « Object Properties » dans Protégé.

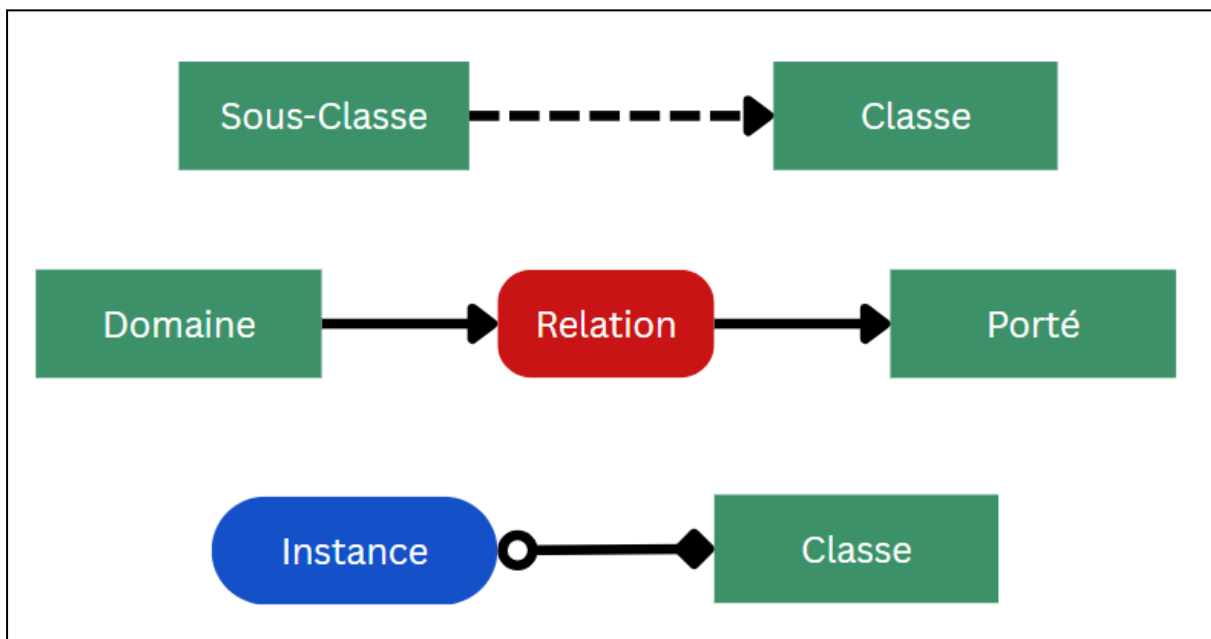
NomRelation	Domaine	Portée	Explication
EstConsulteur	ActeurMedical	Consultation	Indique qu'un acteur médical a effectué une consultation.
EvenementAEuLieuPour	EventMedical	Patient	Associe un événement médical au patient concerné.
EvenementPasseA	EventMedical	StructureSoin	Spécifie dans quelle structure de soin s'est déroulé l'événement médical.

Observe	DispoMedicalConnecte	Patient	Indique qu'un dispositif médical connecté observe ou surveille un patient donné.
PatientEstA	Patient	StructureSoin	Localise un patient dans une structure de soin.
EstRoleDe	RoleMedical	ActeurMedical	Spécifie quel rôle est attribué à quel acteur médical.
Offre	StructureSoin	ServiceMedical	Définit les services médicaux proposés par une structure.
AffecteA	ActeurMedical	StructureSoin	Relie un acteur médical à la structure où il travaille.
DisposeDe	StructureSoin	EquipementMedical	Indique qu'une structure possède un équipement médical donné.
assurePar	Intervention	ActeurMedical	Indique qu'une intervention médicale est réalisée sous la responsabilité d'un acteur médical
effectuePar	Soin	ActeurMedical	Spécifie qu'un soin est administré ou pris en charge par un professionnel de santé.
aParametreSuivi	Patient	ParametreSuivi	Indique qu'un patient a des paramètres de suivi associés pour le monitoring de son état de santé.



## 2. Diagrammes illustratives

Le schéma ci-dessus illustre trois éléments fondamentaux de la modélisation ontologique dans un outil comme Protégé.

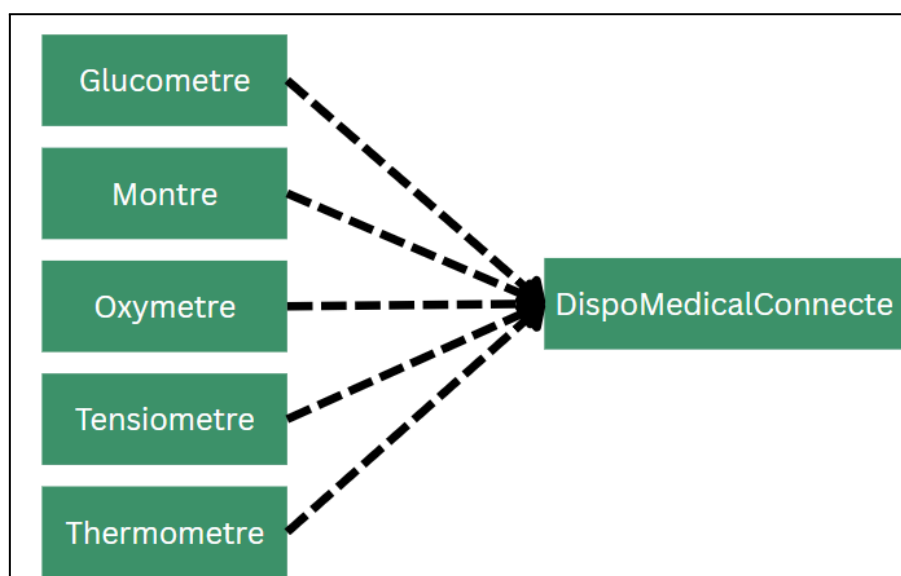
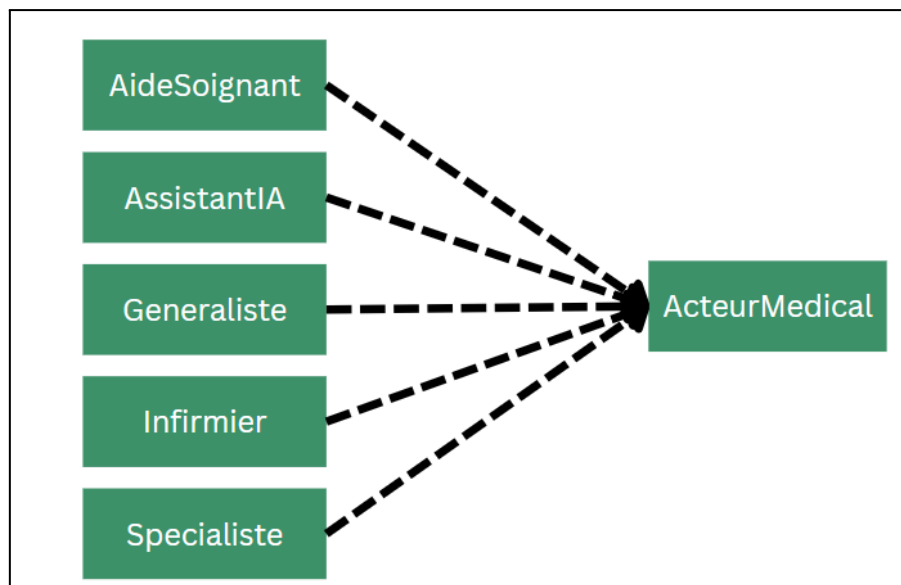


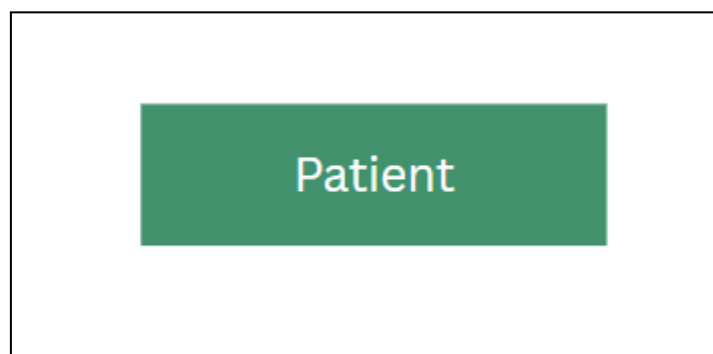
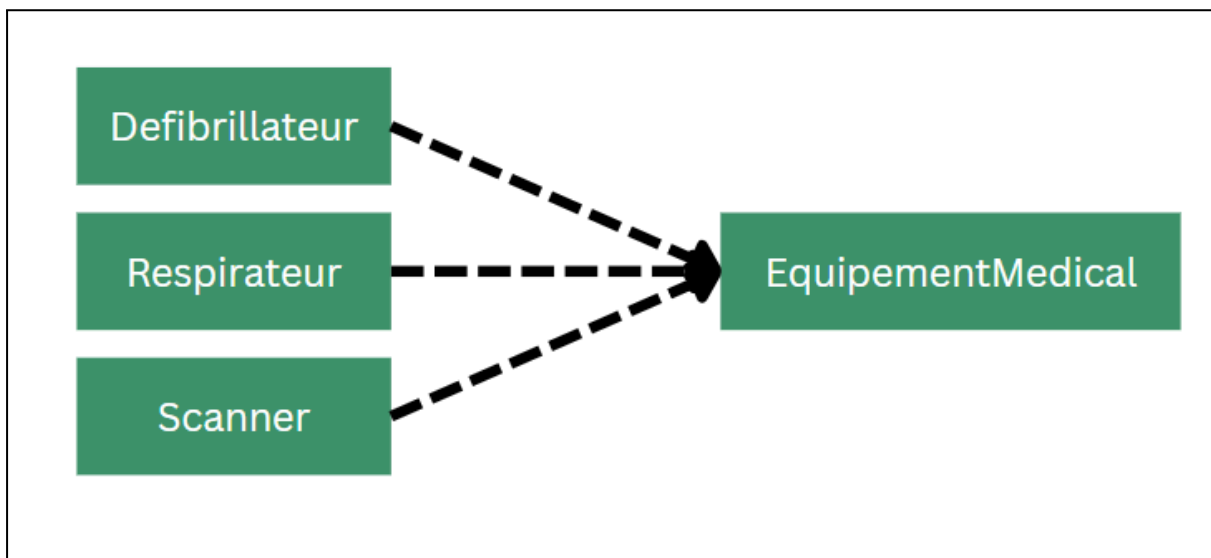
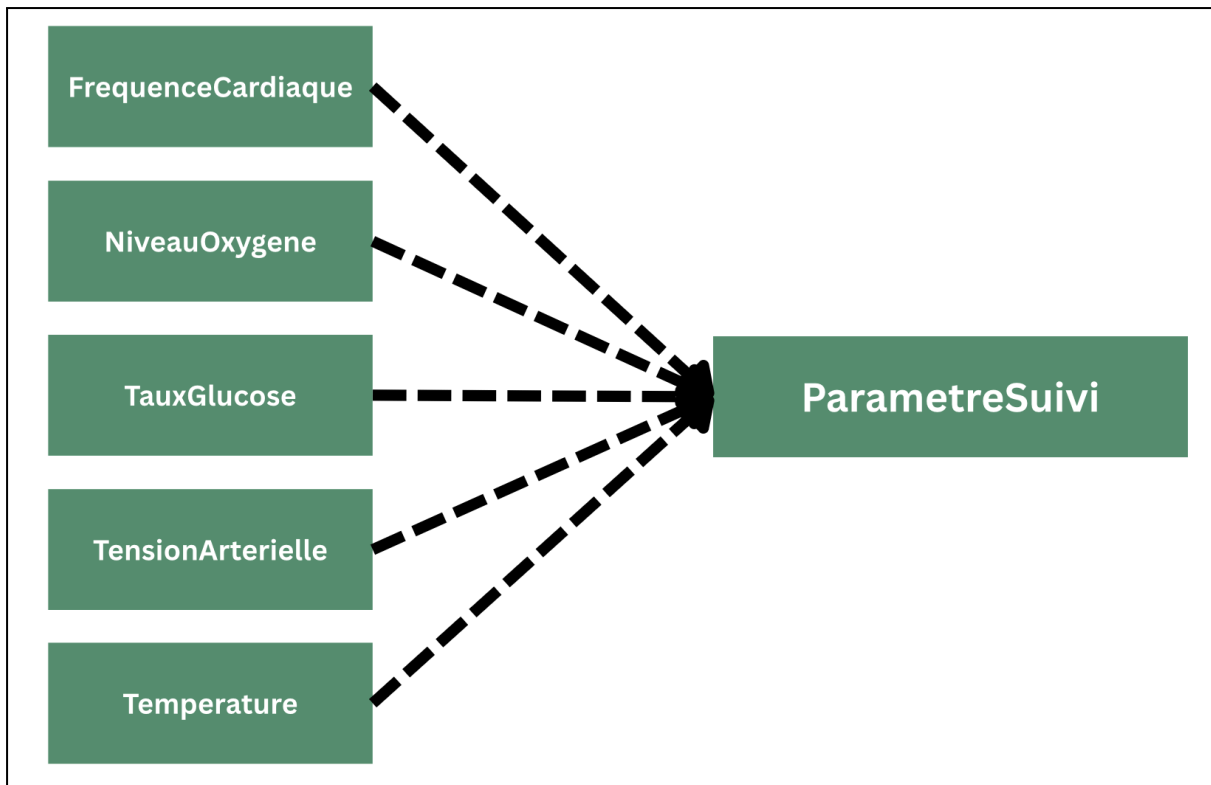
**Spécialisation des classes :** La première ligne (flèche en pointillé) représente la relation d'héritage ou spécialisation. Elle indique qu'une sous-classe hérite des propriétés de sa classe parente, selon le principe d'inclusion hiérarchique : tout individu appartenant à la sous-classe est également un membre de la classe parente.

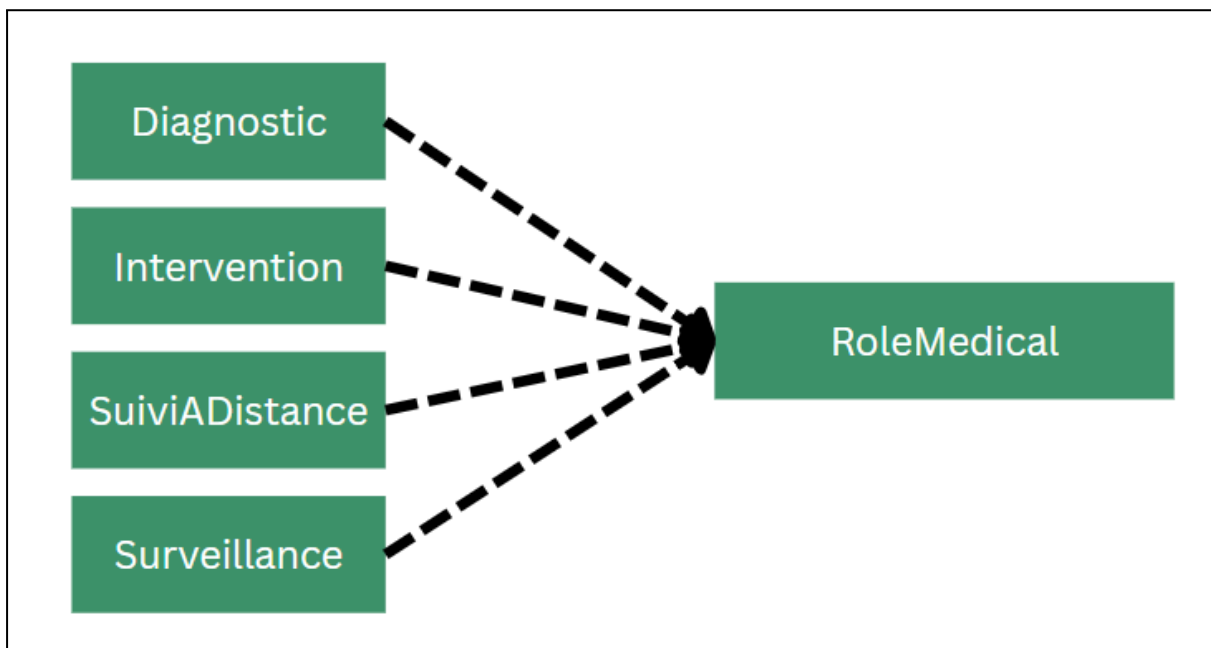
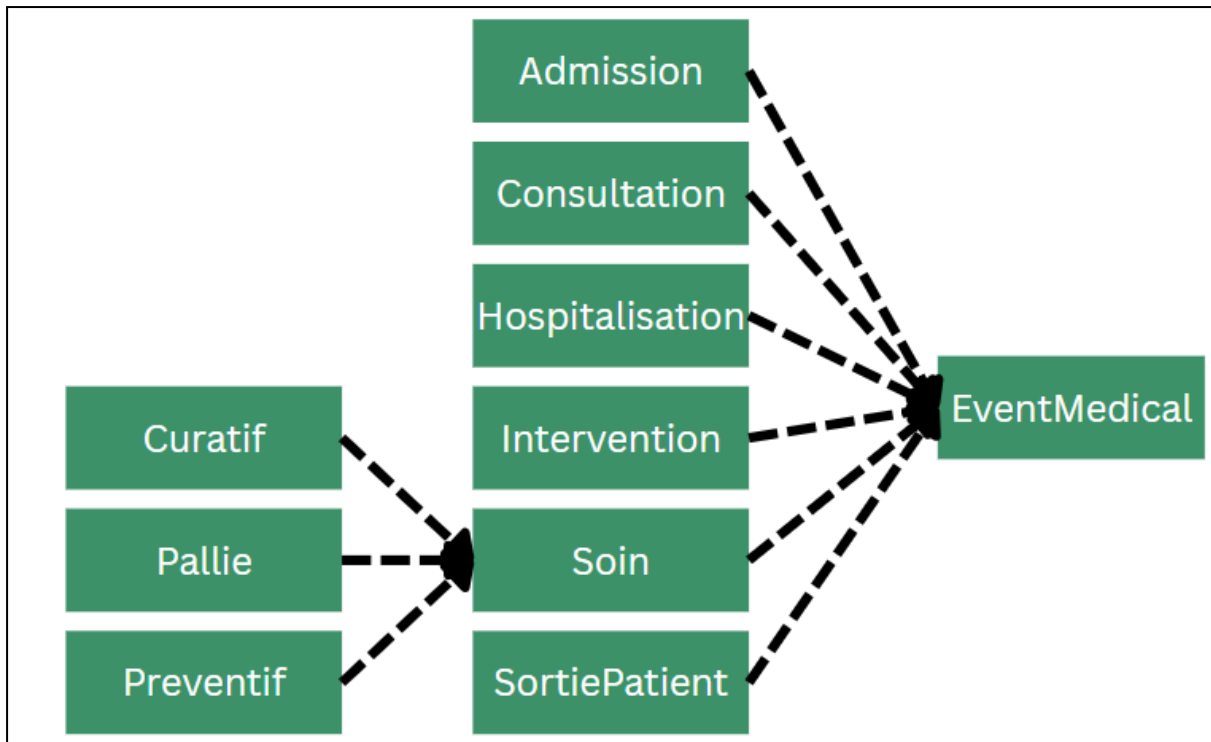
**Relations entre classes (propriétés d'objet) :** La seconde ligne met en évidence la structure d'une relation sémantique entre deux classes. Le domaine (à gauche) est la classe source de la relation, tandis que le portée (range, à droite) est la classe cible. La flèche dirigée à travers la relation rouge montre le sens du lien sémantique établi (ex. : DispositifConnecte Observe Patient).

**Appartenance d'une instance à une classe :** La dernière ligne illustre la relation entre une instance (élément concret) et la classe à laquelle elle appartient. Cette représentation schématise la notion d'instanciation, c'est-à-dire que chaque objet particulier du domaine (par exemple, John) est rattaché à une classe générique (comme Patient).

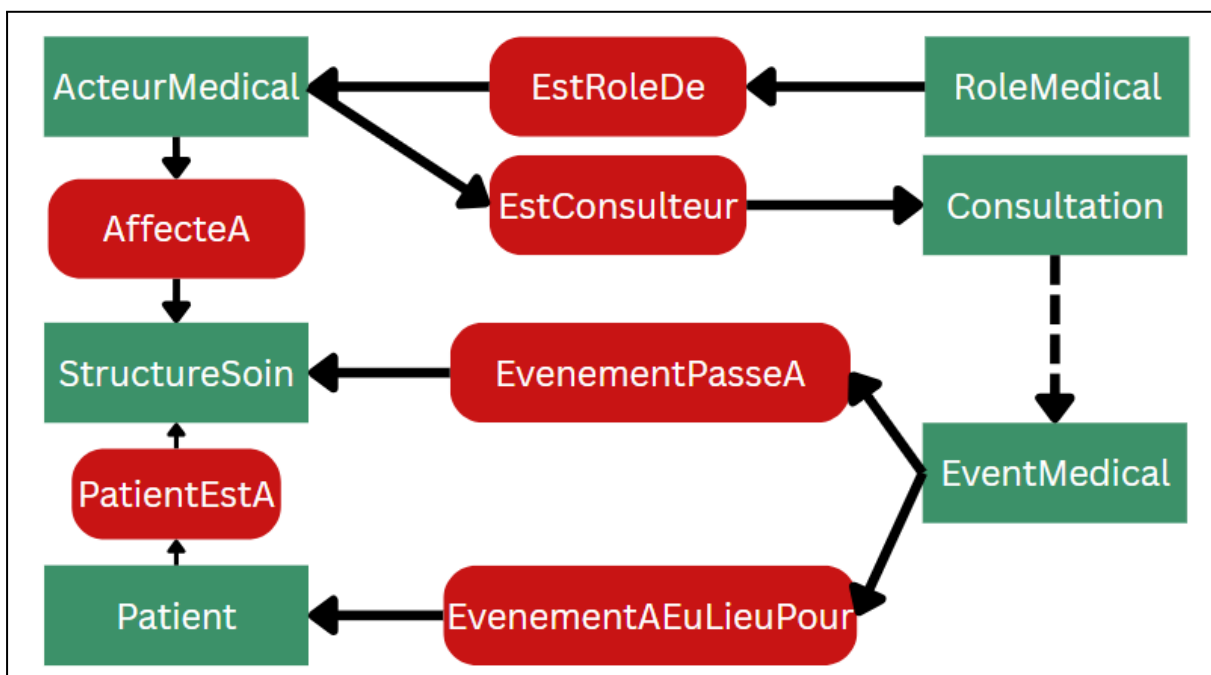
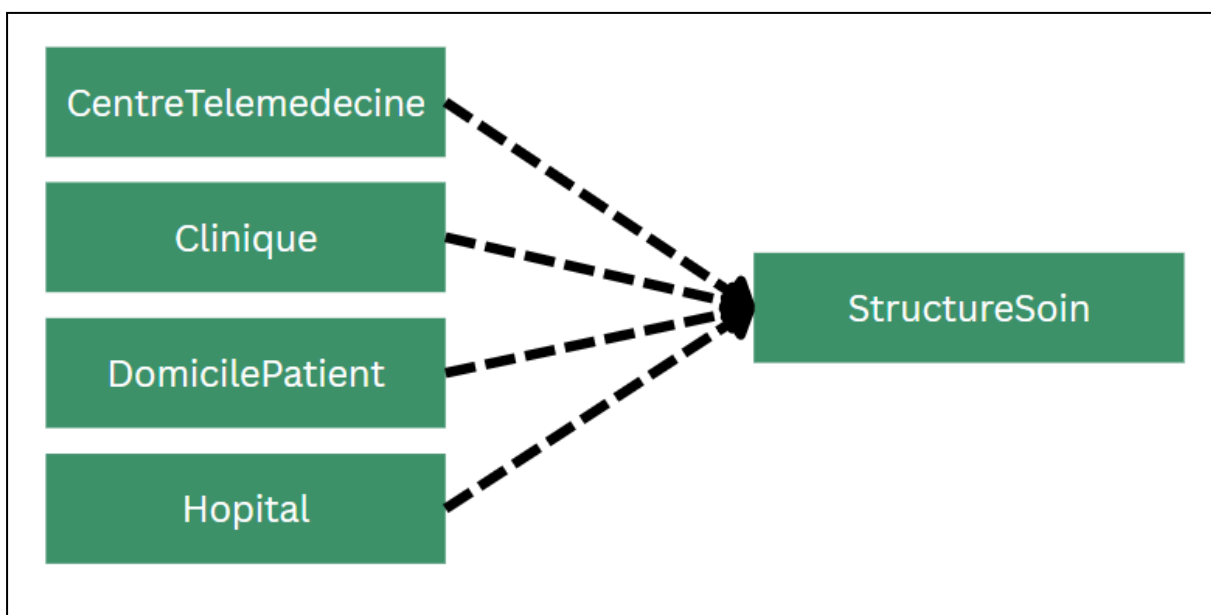
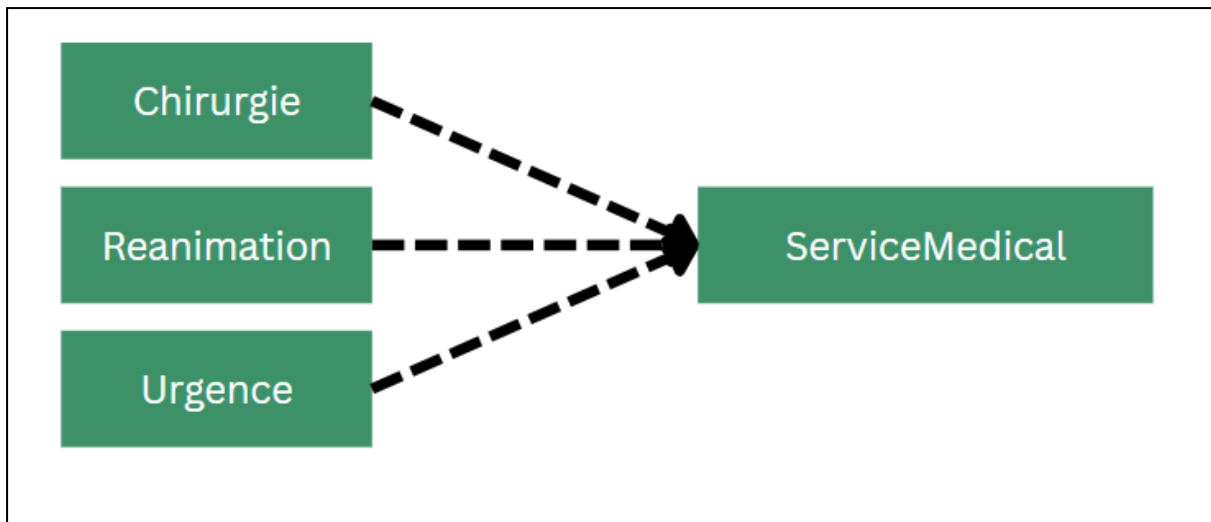
Ci-dessous figurent les diagrammes illustrant les classes définies précédemment, leurs liens d'héritage/spécialisation, ainsi que les relations entre les différentes classes.

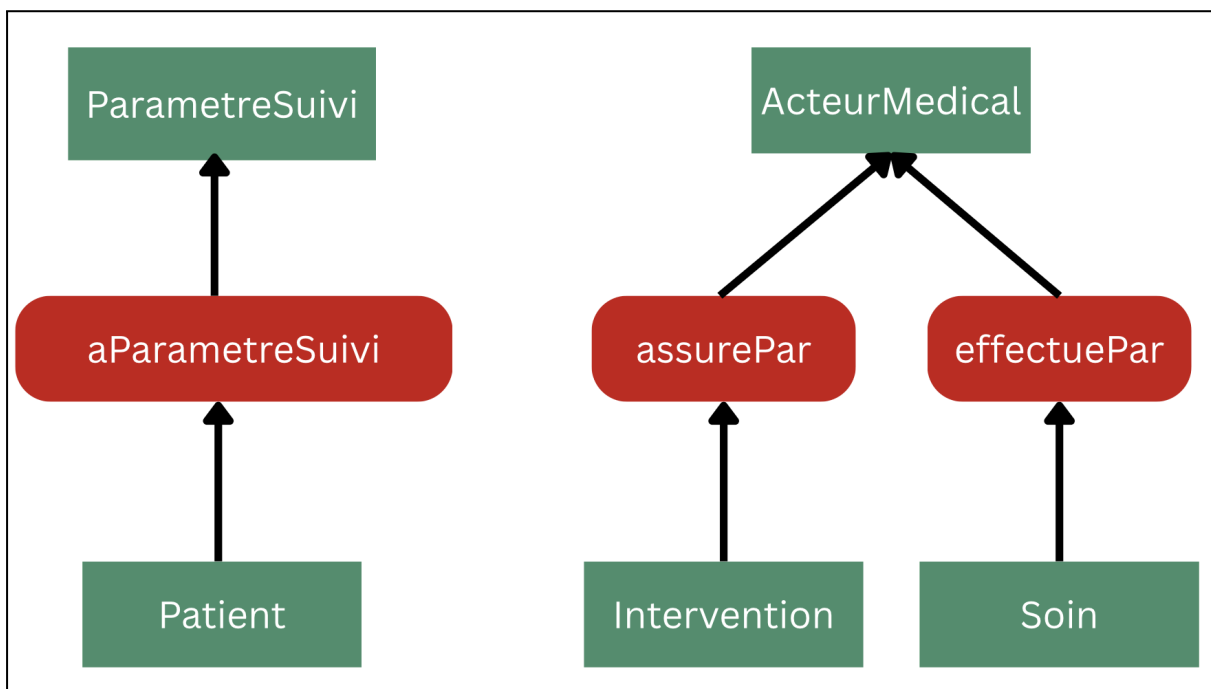
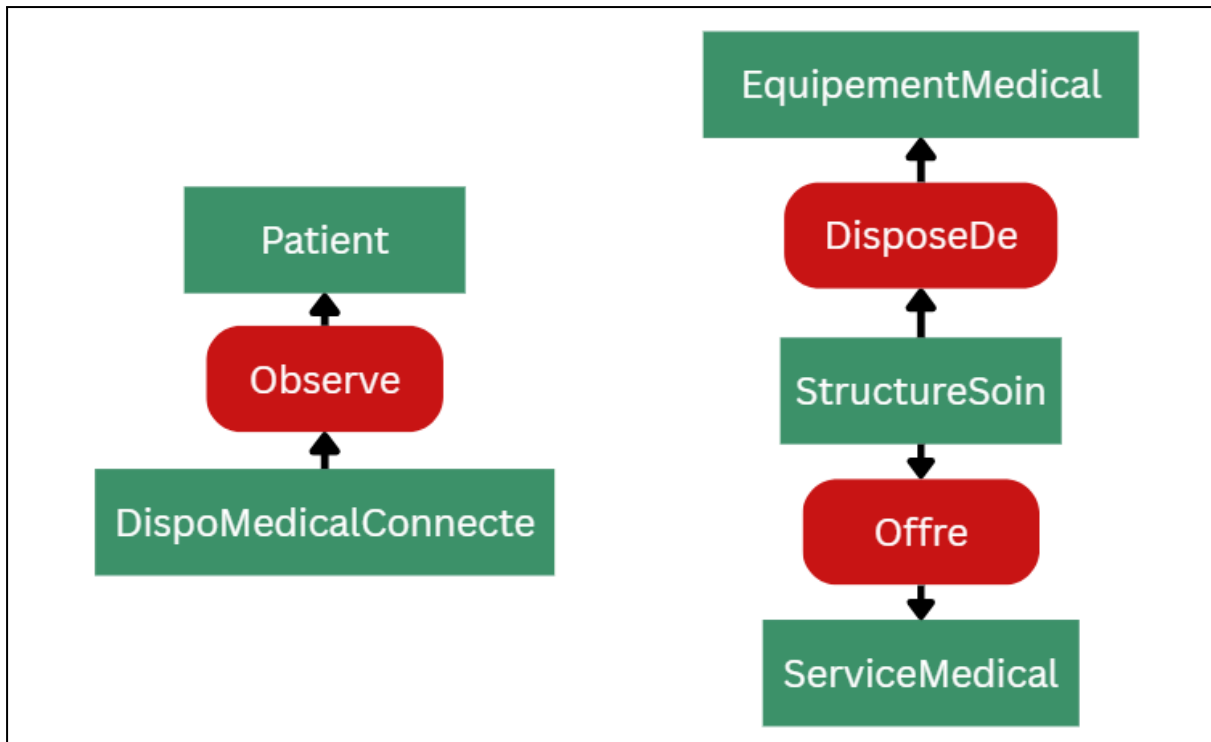












### 3. Details techniques

#### 3.1. Définition de classes supplémentaires

Lors de l'analyse de l'énoncé du TP, certains attributs se sont révélés trop complexes pour être représentés sous forme de simples valeurs. Il a donc été nécessaire de les modéliser comme des classes indépendantes, puis de les relier aux classes d'origine à l'aide de relations sémantiques appropriées. C'est notamment le cas pour les classes telles que ServiceMedical et RoleMedical.

#### 3.2. Vérification de la cohérence de l'ontologie

Protégé dispose d'un outil appelé Reasoner, dont l'une des principales utilités est de vérifier la cohérence de l'ontologie. Ainsi, après avoir défini les classes, les relations et les propriétés, nous avons sélectionné HermiT 1.4.3.456 comme moteur de raisonnement, puis cliqué sur Start Reasoner. Cette action a généré le journal (log) suivant :

```
INFO 15:18:33 ----- Running Reasoner -----
INFO 15:18:33 Pre-computing inferences:
INFO 15:18:33   - class hierarchy
INFO 15:18:33   - object property hierarchy
INFO 15:18:33   - data property hierarchy
INFO 15:18:33   - class assertions
INFO 15:18:33   - object property assertions
INFO 15:18:33   - same individuals
INFO 15:18:33 Ontologies processed in 47 ms by HermiT
INFO 15:18:33
```

Le Reasoner n'a affiché aucune erreur, ce qui indique que l'ontologie est cohérente.

#### 3.3. Tester l'ontologie avec un scénario concret

En utilisant l'onglet « Individuals » de Protégé, nous avons modélisé un scénario concret dans le contexte d'un système de santé intelligent.

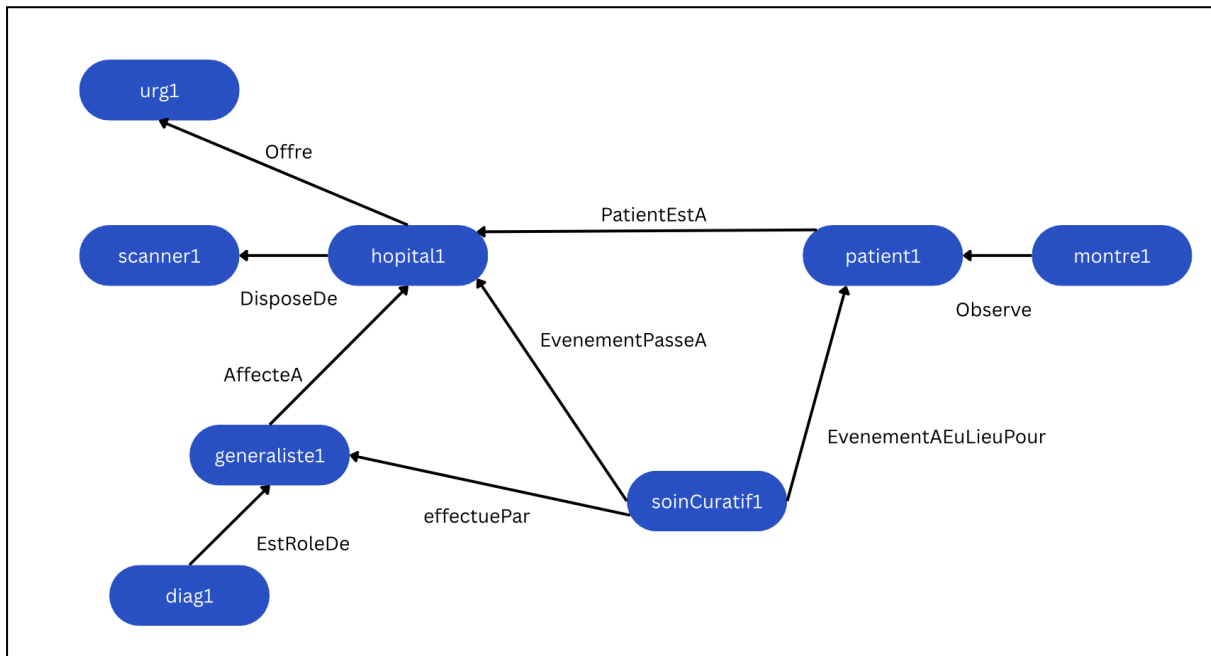
**Scénario proposé:** Il est minuit, le 1er juillet 2024. John Doe, âgé de 42 ans, se rend à l'hôpital pour un diagnostic auprès d'un médecin généraliste. Après analyse des résultats du scanner, le diagnostic conclut que le patient doit être transféré au service des urgences afin de recevoir un soin curatif d'une durée d'un jour. Durant cette période, John porte une montre connectée permettant de surveiller en continu ses données vitales.

## Individus :

- **generaliste1** : Generaliste.
- **patient1** : Patient
  - idPatient : 101
  - prenom : John
  - nom : Doe
  - age : 42
  - groupSang : --
  - sexe : True "Male"
  - antecedantMed : "aucun"
- **montre1** : Montre
  - idDispositif : 102
  - model : Garmin
  - precision : 0.95
  - frequenceEch (Hz) : 5
- **hopital1** : Hopital
  - nombreLits : 100
- **soinCuratif1** : Curatif
  - dateDebut : 2024-07-01T00:00:00Z
  - dateFin : 2024-07-02T00:00:00Z
  - urgence : True
  - results : "Positive"
- **scanner1** : Scanner
- **diag1** : Diagnostic
- **urg1** : Urgence

## Relations :

- **EvenementAEuLieuPour**(soinCuratif1, patient1)
- **EvenementPasseA**(soinCuratif1, hopital1)
- **effectuePar**(soinCuratif1, generaliste1)
- **Observe**(montre1, patient1)
- **PatientEstA**(patient1, hopital1)
- **EstRoleDe**(diag1, generaliste1)
- **Offre**(hopital1, urg1)
- **AffecteA**(generaliste1, hopital1)
- **DisposeDe**(scanner1, hopital1)
- **DisposeDe**(hopital1, scanner1)



◆ diag1
◆ generaliste1
◆ hopital1
◆ montre1
◆ patient1
◆ scanner1
◆ soinCuratif1
◆ urg1

## 4. Perspectives et améliorations possibles

- **Affiner les spécialisations d'ActeurMedical** : La classe `Specialiste` pourrait être décomposée en sous-classes plus fines comme `Cardiologue`, `Neurologue` et `Chirurgien`. Cela permettrait d'affecter des interventions spécifiques à des spécialistes adéquats.
- Enrichir les classes `EquipementMedical`, `DispoMedicalConnecte` et `ServiceMedical` en y ajoutant davantage d'équipements médicaux, de dispositifs médicaux connectés et de services médicaux.
- **Enrichir ServiceMedical** : Les services existants (`Chirurgie`, `Reanimation`, `Urgence`) pourraient être enrichis avec des propriétés comme `chefDeService` (faisant le lien avec une instance d'`ActeurMedical`) et `capaciteAccueil` (un entier pour le nombre de patients).
- **Ontologie des Médicaments** : Introduire une nouvelle classe `Medicament` avec des propriétés comme `dosage` et `formePharmaceutique` (ex: comprimé, sirop). Cette classe pourrait être liée à la classe `Soin` via une nouvelle relation objet `prescritUn`.
- **Représentation des Pathologies** : Créer une classe `Pathologie` pour modéliser formellement les maladies. Elle pourrait avoir des attributs comme `codeCIM10` (code de la Classification Internationale des Maladies), `symptomesCourants` et `traitementsRecommandes`. Cela permettrait de lier un `Patient` à un diagnostic formel.
- **Intégration de Données Administratives** : Ajouter des classes pour la gestion administrative, comme une classe `Facturation` liée aux `Soin` pour modéliser les coûts.
- **Information géographique** : intégrer une classe `Localisation` avec des propriétés telles que `adresse` et `coordonneesGPS`. Cela serait utile pour localiser un hôpital ou pour gérer les interventions à domicile d'un patient.
- **Ajout des contraintes et cardinalité** : On pourrait par exemple définir qu'une `Consultation` doit avoir lieu avant un `Soin`, Un `Patient` ne peut pas être simultanément dans deux `StructuresSoin` différentes (contrainte de cardinalité). Vérifier qu'un `DispositifConnecte` est associé à un seul `Patient` à la fois.

### III. References

Oxford Languages. (s.d.). Oxford Languages – The Home of Language Data. Retrieved from <https://languages.oup.com/>

Euzenat, J., & Gandon, F. (s.d.). Ontologies informatiques. Interstices. Retrieved from <https://interstices.info/ontologies-informatiques/>

W3C. (2012). OWL 2 Web Ontology Language Document Overview (Second Edition). World Wide Web Consortium (W3C). Retrieved from <https://www.w3.org/OWL/>