

1. Présentation

Protégé est un éditeur d'ontologies distribué en open source par l'université en informatique médicale de Stanford. Protégé n'est un outil spécialement dédié à OWL, mais un éditeur hautement extensible, capable de manipuler des formats très divers.

Le support d'OWL, comme de nombreux autres formats, est possible dans protégé grâce à un plugin dédié. Protégé est un outil employé par les développeurs et des experts de domaine pour développer des systèmes basés sur les connaissances (Ontologies). Des applications développées avec Protégé sont employées dans la résolution des problèmes et la prise de décision dans un domaine particulier. Protégé est aussi une plate-forme extensible, grâce au système de plug-ins, qui permet de gérer des contenus multimédias, interroger, évaluer et fusionner des ontologies, etc. L'outil Protégé possède une interface utilisateur graphique (GUI) lui permettant de manipuler aisément tous les éléments d'une ontologie : classe, méta-classe, propriété, instance,...etc. Protégé peut être utilisé dans n'importe quel domaine où les concepts peuvent être modélisés en une hiérarchie des classes.

Protégé permet aussi de créer ou d'importer des ontologies écrites dans les différents langages d'ontologies tel que : RDF-Schéma, OWL, DAML, OIL, ...etc. Cela est rendu possible grâce à l'utilisation de plugins qui sont disponibles en téléchargement pour la plupart de ces langages.

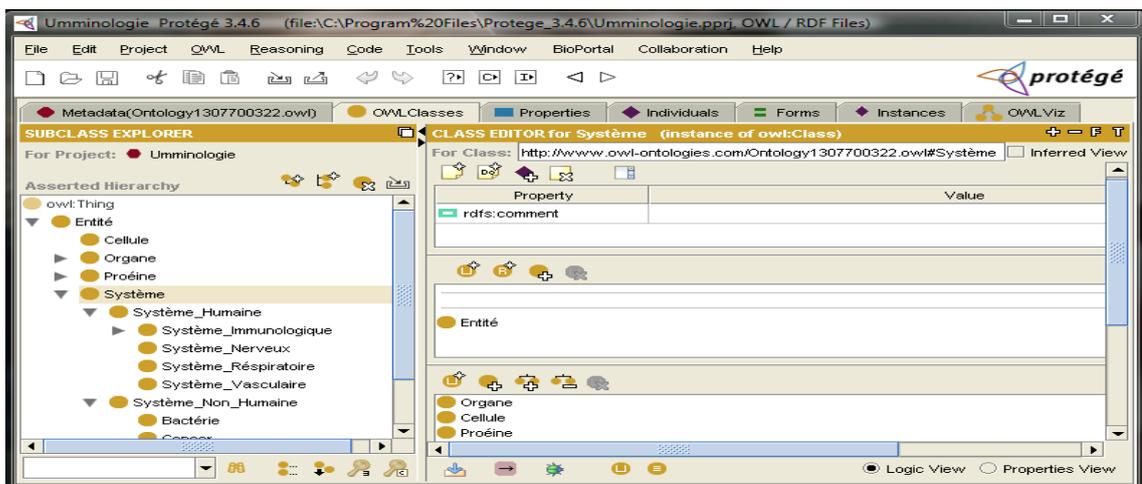


Figure (A ,1) : Interface graphique de Protégé 3.4.6

2. Création d'ontologies avec PROTEGE

2.1. Création d'un projet

Pour créer une ontologie, il faut tout d'abord créer un projet en choisissant le langage (ou format) désiré. Les langages disponibles (selon les plugins installés) sont présentés à gauche de la boîte de dialogue de création de projet.

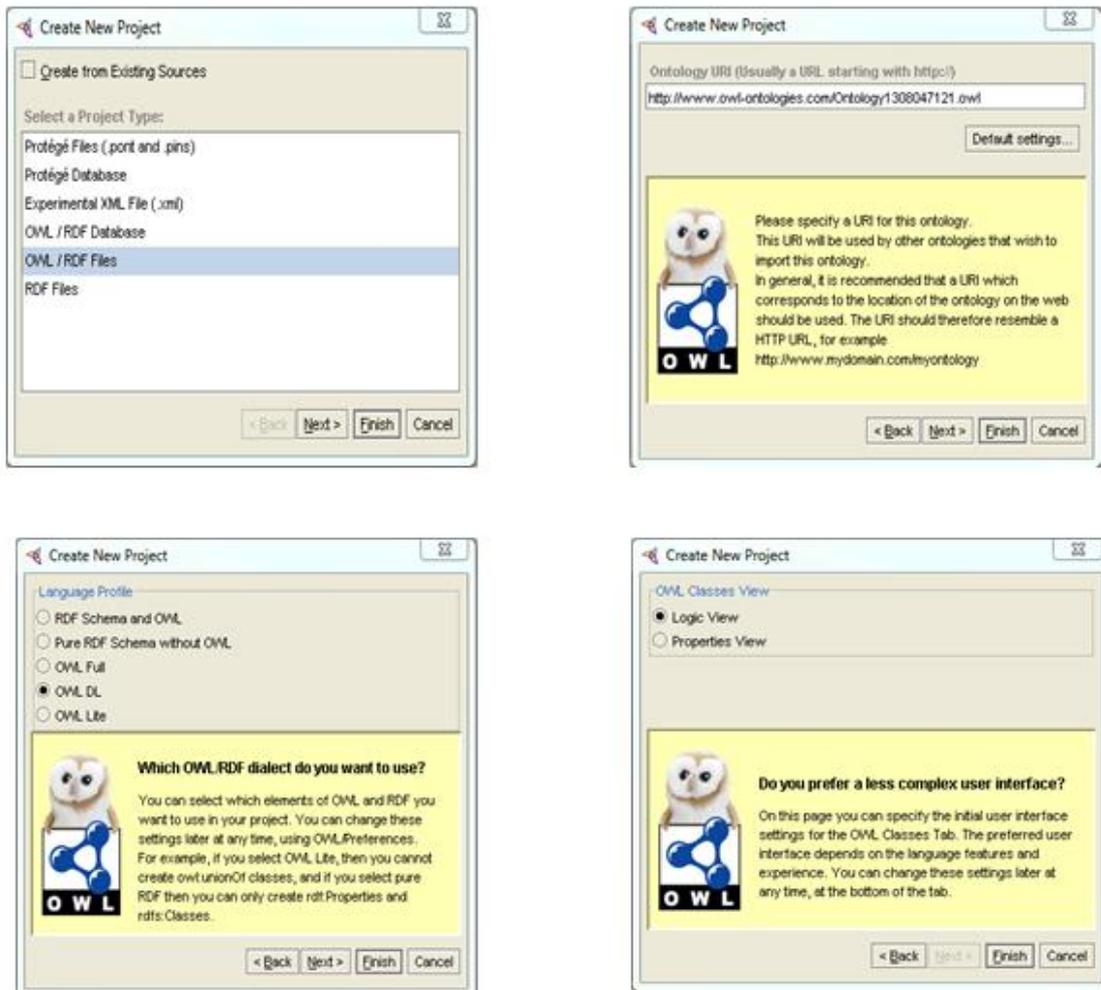


Figure (A, 2) : Création d'un nouveau projet sous Protégé 3.4.6

2.2. Création des classes

Une fois le projet créé, il faut commencer par créer les classes de l'ontologie. Pour cela, il suffit de cliquer sur le bouton et d'entrer un nom à la classe qui sera créée.

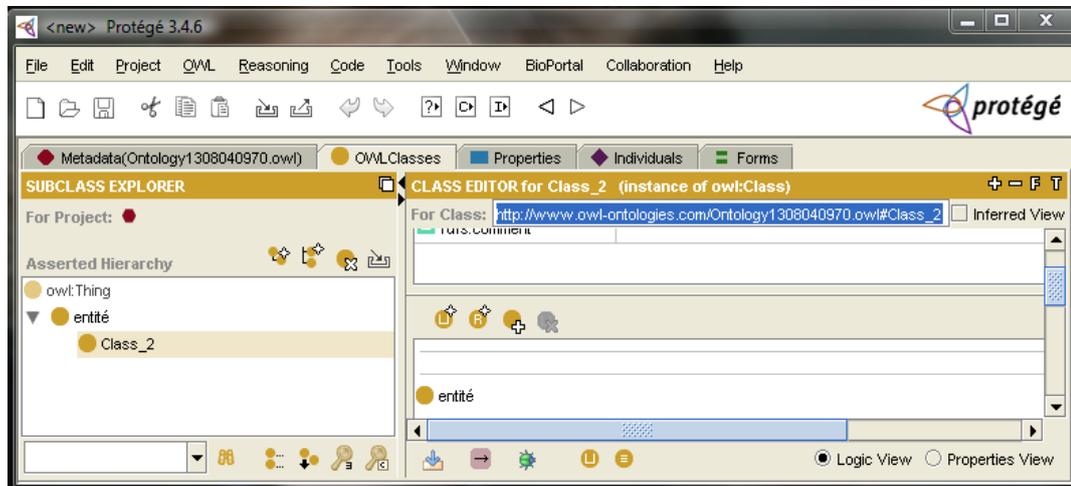


Figure (A , 3) : Création d'une classe sous Protégé 3.4.6

Pour créer une sous classe d'une classe, il suffit de sélectionner la classe mère, et de cliquer sur le bouton  .

2.3. Création des propriétés (Slots)

Une fois les classes créées, il faut définir leurs propriétés (ou Slots). Pour les définir il suffit de choisir une classe, puis de cliquer sur le bouton à droite de l'onglet de propriétés.

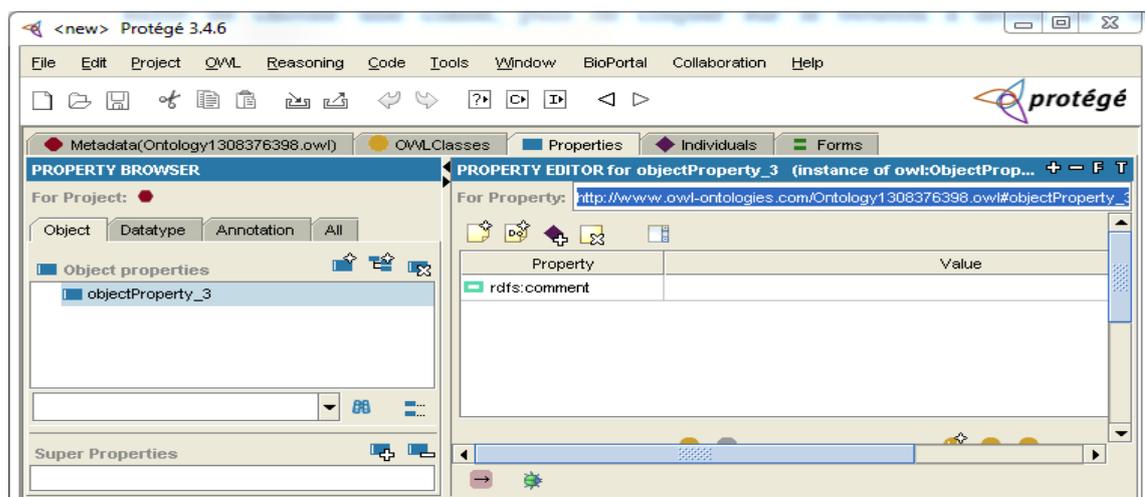


Figure (A , 4) : Onglet des propriétés sous Protégé 3.4.6

Il faut par la suite donner un nom à cette propriété, définir son type, ses cardinalités, et ses valeurs par défaut. Les types de propriétés disponibles dans Protégé sont : Chaîne, Booléen, Classe, Entier, Réel, Instance, Symbole. Il est également possible de créer la propriété inverse de cette propriété si elle existe.

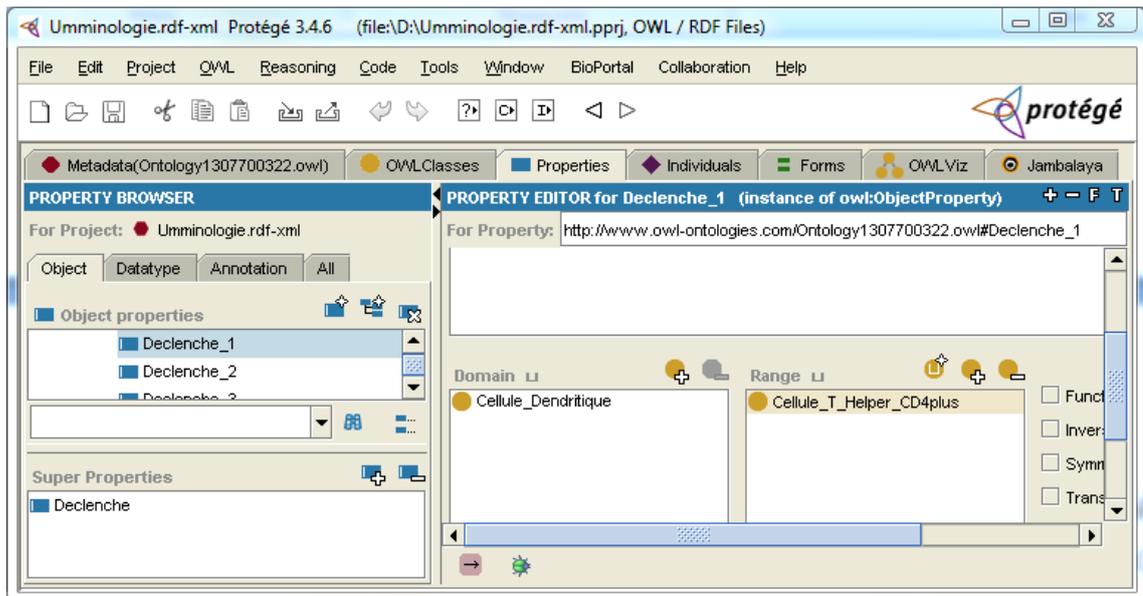


Figure (A, 5) : Création d'une propriété sous Protégé 3.4.6

2.4. Création des instances

Chaque classe peut avoir plusieurs instances. Une instance est une représentation des données concrètes de cette classe dans le monde réel. La création d'une instance se fait tout simplement en choisissant une classe dans l'onglet Instances puis en cliquant sur le bouton  , Exemple : les instances possibles pour la classe « Voiture » sont : «Mercedes », « Fiat », «Renault », « Kia »...

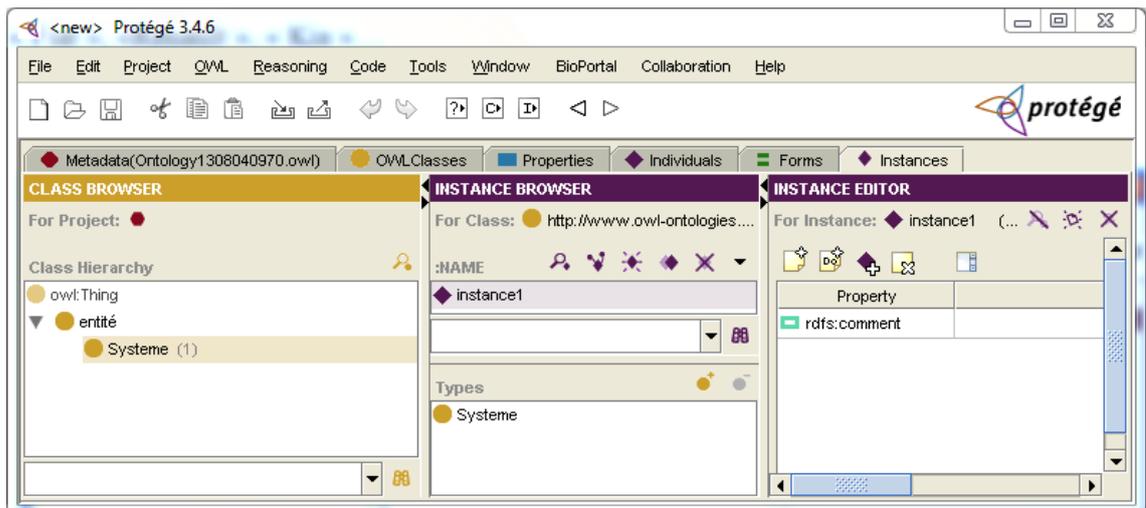


Figure (A, 6) : Création d'une instance sous Protégé 3.4.6

3. L'API Protégé pour le développement de systèmes basés sur les ontologies

Protégé fournit une interface de programmation d'application (API) écrite en JAVA. Cette API permet aux programmeurs en JAVA de développer des applications qui peuvent accéder aux bases de connaissances de Protégé. Cette API fournit des packages et des classes JAVA pouvant effectuer des opérations complexes. L'interface entre les programmes et les projets de bases de connaissances de Protégé se fait en utilisant la classe «*edu.stanford.smi.protege.model.Project*» qui se trouve dans le package «*protege.jar*» fourni avec Protégé-2000. Cette classe possède une méthode *getKnowledgeBase ()* permettant d'accéder au contenu de la base de connaissances.