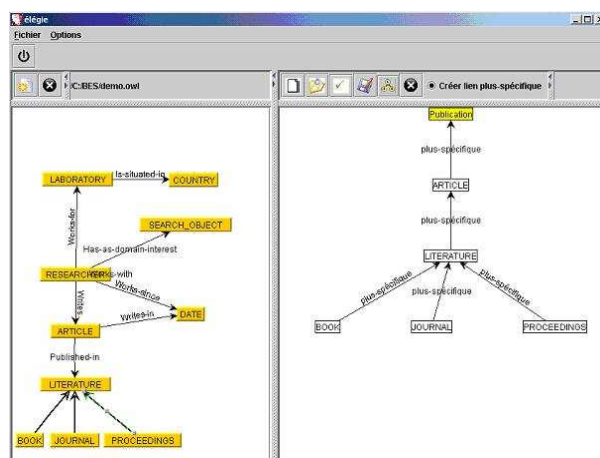


# Elégie



## Points de vues dans Elégie

Josiane Mothe, Alexandre Bès

Création d'une interface graphique en Java pour la construction de hiérarchie de concepts

pour l'Institut de Recherche en Informatique de Toulouse (IRIT)

118 Route de Narbonne



Le projet visait à permettre à une personne de créer des hiérarchies de concepts<sup>1</sup> à partir d'une ontologie<sup>2</sup> par le biais d'une interface graphique réalisée en Java. Plus spécifiquement, dans un premier temps, à partir d'un fichier texte décrivant l'ontologie au format OWL<sup>3</sup>, il s'agissait de la présenter graphiquement. Dans un deuxième temps, les éléments de l'ontologie devaient pouvoir être manipulables, déplaçables et surtout ils devaient pouvoir être importés par l'utilisateur pour construire des hiérarchies de concepts. Enfin, les hiérarchies de concepts ainsi créés devaient être sauvegarder au format OWL.

The application I had to develop should permit to a user to create concept hierarchies from ontologies through a graphical interface developed in Java language. More specifically, firstly from a text file describing an ontology in OWL format, the system had to provide a graphical representation of the ontology. Secondly, elements of the ontology should be easily to handle, move and export by a user to build a concept hierarchy. Finally, hierarchies should be saved in OWL format.

---

<sup>1</sup> Les hiérarchies de concepts représentent des concepts et les relations qu'ils ont entre eux mais à la différence des ontologies les liens sont tous du même type (lien « est-un »).

<sup>2</sup> Les ontologies sont en informatique un moyen de représenter la connaissance d'un domaine sous forme de concepts liés entre eux par des relations sémantiques.

<sup>3</sup> OWL est un langage qui permet de représenter des ontologies, il est une extension de RDF et RDFS qui sont eux même des extensions de XML.

# TABLE DES MATIERES

Introduction .....	6
1. L'Institut Universitaire de Toulouse (IRIT) <b>Erreur ! Signet non défini.</b>	
1.1 L'IRIT .....	<b>Erreur ! Signet non défini.</b>
1.2 Organisation - Organigramme .....	<b>Erreur ! Signet non défini.</b>
1.2.1 Direction .....	<b>Erreur ! Signet non défini.</b>
1.2.2 Conseils .....	<b>Erreur ! Signet non défini.</b>
1.2.3 Services techniques, logistiques et administratifs .....	<b>Erreur !</b>
<b>Signet non défini.</b>	
1.3 L'équipe SIG .....	<b>Erreur ! Signet non défini.</b>
2. Le travail à réaliser .....	7
2.1 Le sujet .....	7
3. Information sur le domaine .....	8
3.1 La recherche d'information .....	8
3.2 Le web sémantique .....	8
3.3 Les ontologies .....	8
3.4 Les hiérarchies de concepts (points de vue) .....	11
3.5 Les éditeurs d'ontologies .....	11
3.5.1 OilEd .....	12
3.5.2 Protégé 2000 .....	12
3.5.2.1 Plugin de visualisation .....	12
3.5.3 Bilan d'évaluation des éditeurs d'ontologie .....	13

4.	Le travail réalisé .....	15
4.1	Les fonctionnalités principales d'Elégie.....	16
4.1.1	Création d'un point de vue .....	16
4.1.2	Modification d'un point de vue.....	17
4.1.2.1	Transformation d'un lien de l'ontologie en « plus-spécifique ».....	17
4.1.2.2	Création d'un lien « plus-spécifique » .....	17
4.1.2.3	Suppression d'un lien du point de vue.....	18
4.1.2.4	Suppression d'un concept du point de vue.....	18
4.1.3	Mise en page d'un point de vue .....	18
4.1.4	Sauvegarde d'un point de vue.....	18
4.1.5	Ouverture d'un point de vue.....	18
4.1.6	Vérification de la cohérence d'un point de vue .....	18
4.2	Les problèmes rencontrés .....	19
5.	Bilan du stage.....	<b>Erreur ! Signet non défini.</b>
	Conclusion .....	<b>Erreur ! Signet non défini.</b>
6.	Lexique.....	21
7.	Annexes .....	23
7.1	Structure de l'application .....	23
7.1.1	Package « autres ».....	23
7.1.2	Package « interfaces.ihm ».....	23
7.1.3	Package « metier.entite.presentation » .....	24
7.1.4	Package « utilitaires.gestionnaires ».....	25
7.1.5	Package « utilitaires.graphes » .....	25
7.1.6	Package « utilitaires.preferences ».....	26
7.1.7	La classe MainElegie.....	26

# INTRODUCTION

L'application présentée ici s'inscrit dans un projet plus global appelé webOnto qui cherche à utiliser les ontologies dans des systèmes de recherche d'information. Cette application visait à transformer une ontologie en hiérarchies de concepts.

# 1. Le travail à réaliser

## 1.1 Le sujet

Le développement de l'interface demandée dans le cadre de ce stage s'inscrit dans un projet plus global de systèmes de recherche d'information basés sur des ontologies. Les ontologies sont des structures complexes, difficilement utilisables par les utilisateurs grand-public des systèmes de recherche d'information. Pour ces derniers, un système permettant de visualiser la connaissance du domaine à travers une hiérarchie de concepts (concepts liés par une relation de type « est-un » ou « est-plus-spécifique-que ») dans laquelle ils peuvent naviguer, est plus adapté.

L'objectif visé est donc de permettre à un concepteur de transformer une ontologie en une hiérarchie de concepts. Plus spécifiquement, dans un premier temps, à partir d'un fichier texte décrivant l'ontologie au format OWL, il fallait la représenter graphiquement. Dans un deuxième temps, les éléments de l'ontologie devaient pouvoir être manipulables par l'utilisateur pour construire la hiérarchie de concepts. Enfin, cette hiérarchie de concepts devait être sauvegardée au format OWL.

## **2. Information sur le domaine**

Mise en forme : Puces et numéros

### **2.1 La recherche d'information**

La recherche d'information est une science qui vise à rendre le résultat d'une recherche documentaire le plus pertinent possible.

### **2.2 Le web sémantique**

Mise en forme : Puces et numéros

Pour comprendre la notion de « web sémantique », il faut analyser le web d'aujourd'hui et les nouveaux besoins de ses utilisateurs. Cela ne fait aucun doute, le web est aujourd'hui une gigantesque source d'informations. C'est en tant que telle qu'il a été conçu. Pourtant, si ces informations sont compréhensibles par l'homme, elles ne le sont pas forcément par des logiciels parcourant le web. En effet le format de données le plus courant actuellement est le HTML. Ce standard a le mérite d'être simple, mais en contrepartie il est quasi inefficace pour donner une dimension informationnelle supplémentaire aux données, exprimer des « méta-données » et ne supporte pas l'apparition de nouveaux services plus évolués que la simple recherche textuelle.

Le web sémantique se veut un web dont le contenu peut être appréhendé et exploité par des machines. Ainsi, le web sémantique pourra fournir des services plus aboutis à ses utilisateurs (trouver l'information pertinente, sélectionner, localiser et activer le service nécessaire...). Il peut être vu comme une infrastructure complétant le contenu informel du web actuel avec de la connaissance formalisée. Il peut conduire à faire cohabiter plusieurs degrés de formalisations allant de schémas de méta-données figés (comme celui du Dublin core) à des langages de représentation plus complexes (comme DAML+OIL, OWL et différentes logiques). Source : Simon CORBI, Sophie MORENO, Johan PIC dans « Le web sémantique - Catégorisation et recherche d'information ».

### **2.3 Les ontologies**

Mise en forme : Puces et numéros

Extrait de [Hernandez, Aussenac-Gilles dans « OntoExplo : Ontologie pour l'aide à une activité de veille ou d'exploration d'un domaine ». Etant donné que c'est une définition qui reprend des définitions données par des

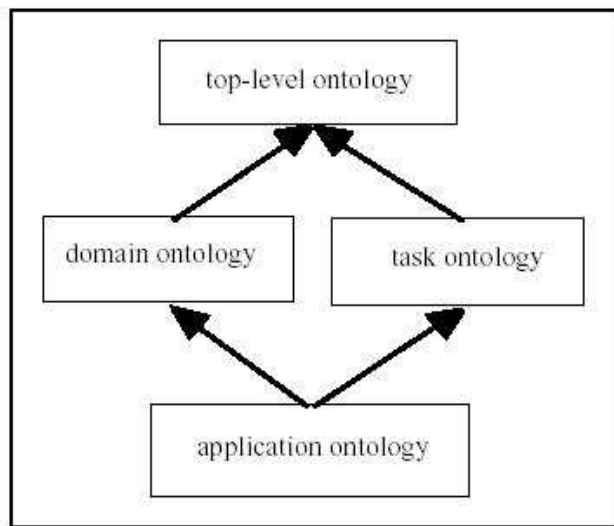


personnes plus compétentes que moi je pense que je vais laisser comme ceci.

« Le terme « ontologie » a été emprunté au domaine de la philosophie dans lequel il signifie « l'essence de l'essentiel ». Dans le domaine de la gestion de connaissance, le sens de ce mot est différent. Gruber [Gruber 1993] introduit la notion d'ontologie comme « une spécification explicite d'une conceptualisation ». Cette définition a été légèrement modifiée par Borst [Borst 1997]. Une combinaison des deux définitions peut être résumée ainsi : « une spécification explicite et formelle d'une conceptualisation partagée ». Cette définition s'explique ainsi [Studer 1998] : *explicite* signifie que le « type des concepts et les contraintes sur leurs utilisations sont explicitement définies », *formelle* se réfère au fait que la spécification doit être lisible par une machine, *partagée* se rapporte à la notion selon laquelle une ontologie « capture la connaissance consensuelle, qui n'est pas propre à un individu mais validée par un groupe », *conceptualisation* se réfère à « un modèle abstrait d'un certain phénomène du monde basé sur l'identification des concepts pertinents de ce phénomène ». Une ontologie fournit une base solide pour la communication entre les machines mais aussi entre humains et machines en définissant le sens des objets tout d'abord à travers les symboles (mots ou expressions) qui désignent et les caractérisent et ensuite à travers une représentation structurée ou formelle de leur rôle dans le domaine.

De nombreux types de structures de connaissance se cachent derrière le mot ontologie (taxonomie, thesaurus, ...). Ces structures de données peuvent être à la fois terminologiques (elles contiennent un ensemble de termes) et conceptuelles (elles définissent généralement des concepts). Cependant ces structures peuvent différer par leur contenu (connaissances générales ou connaissance d'un domaine), par le type des relations sémantiques entre les concepts (relations taxonomiques, méronymiques, ...), et par le niveau de formalisation (représentation logique, représentation dans un langage dédié aux ontologies tels que DAML+OIL, OWL, ...). La méthode choisie lors de la conception de l'ontologie, la sélection des concepts et leur organisation influencent également sa nature [Guarino 2000]. »

Il existe différents types d'ontologies :



1. Ontologie de haut niveau ou générale [Guarino, 1998]

Elles décrivent des concepts très généraux tel que le temps, l'espace, les événements, etc ... elles sont indépendantes d'un problème ou d'un domaine particulier.

2. Ontologie de domaine

Contient la représentation de la connaissance liée au domaine.

3. Ontologie de tâche

Organise les méta-données<sup>4</sup>, à savoir le rôle des connaissances dans la réalisation d'une tâche.

4. Ontologie d'application [Guarino, 1998]

---

<sup>4</sup> Une méta-donnée est littéralement une donnée sur une donnée. Plus précisément, c'est un ensemble structuré d'informations décrivant une ressource quelconque.

Elles décrivent des concepts dépendants d'un domaine et d'une tâche particulière. Ces concepts correspondent souvent au rôle joué par les entités d'un domaine dans une tâche précise.

Une ontologie est constituée d'un ensemble de concept qui ont des relations de différents types entre eux.

## **2.4 Les hiérarchies de concepts (points de vue)**

La notion de hiérarchie de concepts existe en dehors des ontologies, elles représentent des concepts et les relations qu'ils ont entre eux mais à la différence des ontologies les relations ne peuvent être que d'un seul type : « est-un ».

Dans le cadre de la recherche documentaire l'utilisation des ontologies est quelque peu délicate car elles sont riches de sens et leur utilisation à travers une interface est très complexe, c'est pour cela que Mme Aussenac-Gilles et Mme Mothe ont décidé d'utiliser des hiérarchies de concepts. (cf « Ontologies as Background Knowledge to Explore Document Collections »)

Elles ont adapté la notion de hiérarchie de concepts à leurs besoins. Une hiérarchie de concepts correspond à un point de vue ou une dimension. La création d'une hiérarchie de concepts se fait de façon incrémentale en ajoutant un concept générique qui sera le point de départ de la hiérarchie, des concepts et des relations issues de l'ontologie ; ces dernières donnant lieu toujours à des relations de type « plus spécifique » entre les concepts. Le point de départ d'un point de vue est toujours le concept le plus générique, et c'est le seul concept qui n'existe pas dans l'ontologie. Les relations sont d'un unique type : « plus spécifique ».

## **2.5 Les éditeurs d'ontologies**

Durant ces 4 semaines de documentation j'ai également dû manipuler et évaluer différents éditeurs d'ontologies. Les deux éditeurs les plus intéressants sur le plan des fonctionnalités sont OilEd <http://oiled.man.ac.uk/> et Protégé <http://protege.stanford.edu/>. C'est pour cela qu'ils sont les deux éditeurs les plus utilisés dans le monde de la recherche.

### **2.5.1 OilEd**

Mise en forme : Puces et numéros

Voici les différentes fonctionnalités de OilEd :

- Construit interactivement des ontologies
- Edite des ontologies DAML+OIL
- Quelques exemples d'ontologies sont fournis sur le site officiel
- Système de raisonnement <sup>5</sup>fournit par le système FACT

### **2.5.2 Protégé 2000**

Mise en forme : Puces et numéros

Voici les différentes fonctionnalités de Protégé 2000 :

- Construit interactivement et importe des ontologies
- Une bibliothèque d'ontologies est disponible sur le site officiel
- Programmer et incorporer des plugins<sup>6</sup>, ils sont en téléchargement sur le site officiel
- Fonction d'export vers différent format de description d'ontologie (RDF(S), OIL, DAML+OIL, OWL) via plugin pour certains formats
- Compatible OKBC (Open Knowledge Base Connectivity)  
<http://www.ai.sri.com/~okbc/>

#### **2.5.2.1 Plugin de visualisation**

Mise en forme : Puces et numéros

Dans le cadre de ce stage j'ai dû évaluer les différents plugins de visualisation d'ontologie disponibles sur le site de Protégé 2000  
<http://protege.stanford.edu/>

---

<sup>5</sup> Un système de raisonnement permet de faire des inférences c'est à dire de déduire de la connaissance à partir d'autre connaissance.

<sup>6</sup> Plugin : Un plugin est un module qui étend les fonctionnalités d'une application.

<i>Nom</i>	<i>Affichage d'un élément</i>	<i>Zoom</i>	<i>Automatisme</i>	<i>Affichage d'un graphe</i>	<i>Préférences</i>	<i>Autres</i>
ezowl	Par double-clic Se positionne sur l'élément correspondant dans la fenêtre	Oui Si on recule trop, on ne voit plus rien	Les flèches entre chaque élément sont gérées toutes seules mais avec possibilité de les déplacer manuellement	Affichage pyramidal, hiérarchique	Les couleurs des éléments peuvent être modifiées Le contenu des éléments peut être modifié	Simple d'utilisation Très bon affichage
tgviz	Assez compliqué Pas intuitif	Oui Le fait de reculer ne modifie pas la taille des éléments, ils sont donc toujours parfaitement visibles	L'affichage d'un graphe se fait automatiquement au centre L'emplacement des éléments dépendants d'un autre est automatique, mais il n'est pas possible de les mettre comme on veut	Affichage en étoile	Tout ce qui concerne l'affichage des éléments (couleur, police...) peut être modifié Choix des liens à afficher	Assez compliqué à utiliser, car peu intuitif Excellent affichage, très joli
ontoviz	Peu intuitif Par double-clic, on affiche l'élément correspondant dans la fenêtre	Oui Si on recule trop, on ne voit plus rien		Affichage hiérarchique On ne peut pas bouger les éléments dans le graphe		Nécessite l'installation de GraphViz
owlviz	Simple Clic sur le bouton correspondant dans la barre de menu Se positionne sur l'élément correspondant dans la fenêtre	Oui Non testé	Non testé	Affichage hiérarchique Possibilité d'afficher ou pas les descendants d'une classe ou ses parents	Non testé	Nécessite l'installation de GraphViz Il faut Protege 2.1 (bêta) Semble être le plus simple

### **2.5.3 Bilan d'évaluation des éditeurs d'ontologie**

Mise en forme : Puces et numéros

Je pense que Protégé est plus intéressant car il offre la possibilité de créer des plugins. Il y a déjà un grand nombre de plugins disponibles en

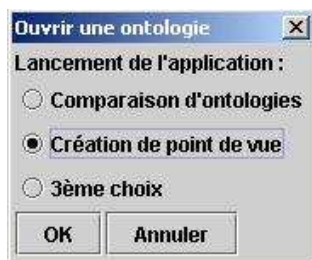
téléchargement sur le site de Protégé. De plus, il possède un plugin qui permet de manipuler et sauvegarder des ontologies au format OWL. Ce plugin est très intéressant car actuellement la représentation des ontologies tant à ce faire dans ce format qui est en cours de validation par le W3C. Ce format a été développé dans l'optique de rendre le web sémantique, il s'appuie sur les formats RDF et RDFS. Pour toute information complémentaire sur ces langages cf. : [www.w3c.org](http://www.w3c.org).

Dans le but de m'éviter d'analyser le fichier OWL contenant l'ontologie mon maître de stage a eu l'idée de créer un plugin pour Protégé. Donc je me suis documenté sur la façon de créer un plugin pour Protégé et sur OWL-plugin. J'ai également installé un certain nombre de plugins de visualisation d'ontologie dans le but d'en reprendre un pour l'affichage des ontologies (l'affichage de l'ontologie ne faisait pas partie de mon stage). Mon maître de stage et moi-même avons cherché à obtenir les sources du plugin de visualisation le plus proche de nos besoins mais la personne n'a pas répondu. C'est pour cela que finalement nous avons décidé de faire un logiciel à part entière ; un autre stagiaire ayant en charge le développement du module de visualisation des ontologies. De plus une nouvelle version de Protégé est en version bêta et développer à partir d'une version bêta n'est pas forcément un bonne chose car lors des tests ou de l'utilisation du plugin il y aurait pu avoir des bugs qui proviennent de la version bêta de Protégé et non du plugin que j'aurai développé.

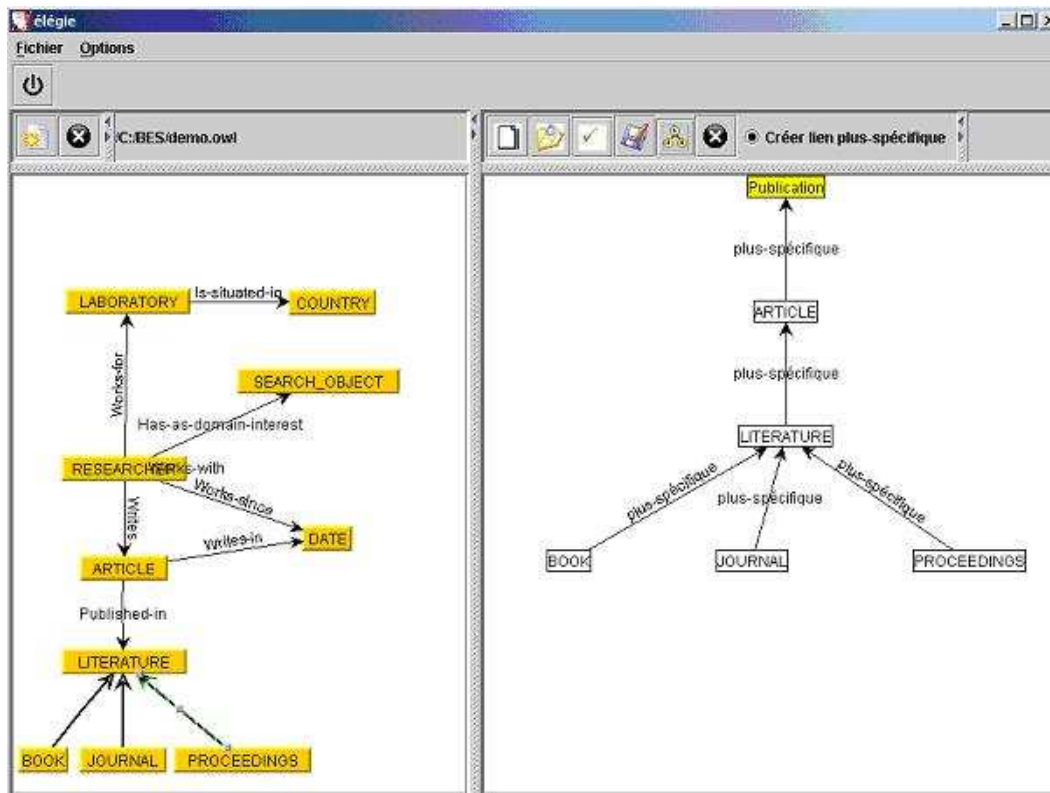
### 3. Le travail réalisé

L'application Elégie contient deux modules permettant de travailler sur des ontologies, un permet de comparer des ontologies et l'autre permet de créer des hiérarchies de concepts (aussi appelées points de vue) à partir d'une ontologie. Nous avons essayé de bien séparer l'interface graphique des traitements et également de bien séparer notre code afin qu'il soit le plus réutilisable possible. Elégie a été conçu de manière à ce que l'ajout de nouvelles tâches soit simplifiée.

Au lancement de l'application il faut choisir si l'on veut comparer des ontologies ou créer des points de vue.



En fonction du choix de l'utilisateur les boutons, les barres de menus et les barres d'outils sont différentes.




La partie gauche de la fenêtre principale permet d'afficher l'ontologie tant dis que la partie droite permet de créer, sauvegarder, visualiser et de modifier un point de vue. « Publication » est en couleur car c'est le nom du point de vue et donc le concept le plus générique. Comme on peut le voir sur la capture d'écran le point de vue ne possède que des liens de type « plus-spécifique ». Le seul nouveau concept par rapport à l'ontologie est le nom du point de vue « Publication ».

### 3.1 Les fonctionnalités principales d'Elégie

Dans cette rubrique seules les fonctions principales qui sont le cœur du module de création de point de vue d'Elégie sont présentées.

#### 3.1.1 Création d'un point de vue

Pour créer un point de vue à partir d'une ontologie il suffit de cliquer sur  lorsqu'une ontologie est ouverte. Ensuite il faut entrer le nom du point de vue et le valider. Cela a pour effet de faire apparaître un concept ayant pour nom celui du point de vue. Il est aisément distinguable des autres concepts car il est le seul à être de couleur jaune.



### **3.1.2** Modification d'un point de vue

Mise en forme : Puces et numéros

La modification d'un point de vue peut s'effectuer de deux manières différentes :

- En transformant un lien déjà existant dans l'ontologie en « plus-spécifique »
- En créant un lien qui n'existait pas dans l'ontologie
- En supprimant un lien du point de vue
- En supprimant un concept du point de vue

#### **3.1.2.1** Transformation d'un lien de l'ontologie en « plus-spécifique »

Mise en forme : Puces et numéros

Elle se fait de manière simple en faisant un clic droit sur le lien de l'ontologie que l'utilisateur souhaite transformer en lien « plus-spécifique ». Un menu avec deux options apparaît. Il y a deux options à ce menu afin de permettre à l'utilisateur d'indiquer quel est le concept le plus spécifique des deux. Par exemple les deux options pourraient être :

- Article -> Littérature
- Littérature -> Article

L'option « Littérature -> Article » signifie que Littérature est « plus-spécifique » que Article.

#### **3.1.2.2** Création d'un lien « plus-spécifique »

Mise en forme : Puces et numéros

Pour créer un lien « plus-spécifique » entre deux concepts il suffit d'activer le bouton radio ayant pour libellé « créer lien plus-spécifique ». Puis de sélectionner le concept le plus spécifique (par exemple « Littérature ») et le concept le plus générique (par exemple « Article »), un lien sera créé qui va de « Littérature » vers « Article ».

### 3.1.2.3 Suppression d'un lien du point de vue

Mise en forme : Puces et numéros

Pour supprimer un lien du point de vue il suffit de le sélectionner en cliquant dessus et d'appuyer sur la touche « Suppr » du clavier.


### 3.1.2.4 Suppression d'un concept du point de vue

Mise en forme : Puces et numéros

La suppression d'un concept du point de vue se fait de la même manière que celle d'un lien. La suppression d'un concept entraîne aussi celle des liens « plus-spécifique » qui lui sont liés. Seul le concept ayant pour nom celui du point de vue (le seul concept en jaune) ne peut pas être supprimé.


### 3.1.3 Mise en page d'un point de vue

Mise en forme : Puces et numéros

Pour mettre en page un point de vue il suffit de cliquer sur . La mise en page du point de vue s'effectue sous forme d'un arbre avec le concept le plus spécifique en haut (celui en jaune). La mise en page n'est effectuée que lorsque le point de vue est cohérent.


### 3.1.4 Sauvegarde d'un point de vue

Mise en forme : Puces et numéros

La sauvegarde d'un point de vue s'effectue en cliquant sur . Elle a pour effet de créer deux fichiers, un fichier au format OWL contenant le point de vue et un fichier d'index servant à retrouver à partir d'une ontologie tous les points de vue qui ont été créés avec cette ontologie. Comme pour la mise en page, la sauvegarde ne peut être réalisée que pour un point de vue cohérent.


### 3.1.5 Ouverture d'un point de vue

Mise en forme : Puces et numéros

L'ouverture d'un point de vue s'effectue en cliquant sur . Il suffit de lui indiquer ou trouver le fichier d'index et de choisir le point de vue que l'on souhaite visionner.

### 3.1.6 Vérification de la cohérence d'un point de vue

Mise en forme : Puces et numéros

Elle s'effectue en cliquant sur . Un point de vue cohérent possède les propriétés suivantes :

- Il possède une **unique** racine (le point de vue) ce qui implique que tous les concepts sont liés

- Un seul père par concept sauf pour la racine qui n'a pas de père

### 3.2 Les problèmes rencontrés

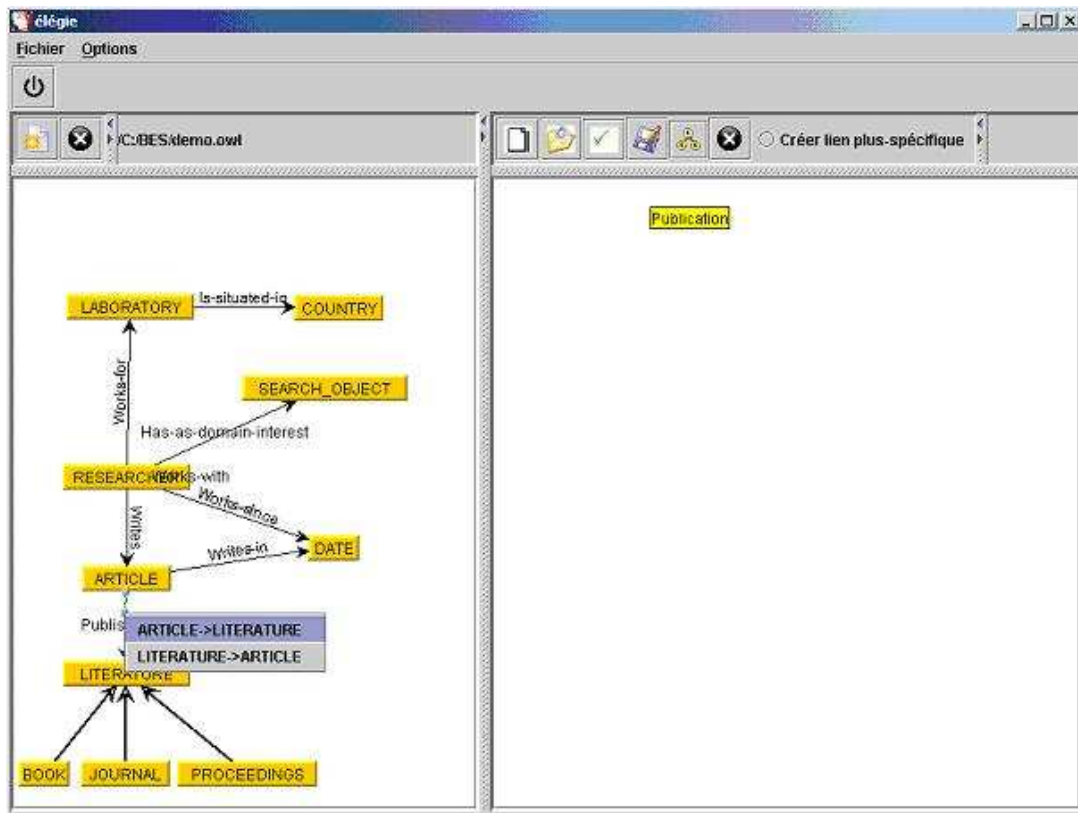
La bibliothèque Open Source du nom de JGraph [www.jgraph.com](http://www.jgraph.com) a été utilisée.

J'ai également dû réfléchir à comment faire le lien entre l'ontologie et le point de vue en cours de création car, dans le point de vue à part la racine tous les concepts proviennent de l'ontologie, et les relations « plus spécifique » de la hiérarchie de concepts sont issues de l'ontologie. En fait pour faire le lien entre l'ontologie et le point de vue, j'ai mis sur écoute<sup>7</sup> au sens Java ( grâce à la fonction `addMouseListener`<sup>8</sup> de JGraph) le JGraph représentant l'ontologie et lorsque l'utilisateur fait un clic droit sur un lien, un JPopupMenu apparaît avec deux options pour indiquer quel est le concept le plus spécifique des deux. Lorsque l'utilisateur clique sur l'option les deux sommets sont ajoutés au point de vue s'ils n'existent pas déjà dans la hiérarchie.

---

<sup>7</sup> Mettre sur écoute au sens Java : Lorsque l'on met sur écoute un objet c'est pour être averti des différents événements qui peuvent se produire sur l'objet qui a été mis sur écoute. Les événements peuvent être de plusieurs types, les plus connus sont les événements souris (clic droit, gauche ...) et les événements liés au clavier.

<sup>8</sup> `AddMouseListener` : Est une fonction qui permet de mettre sur écoute un objet c'est à dire de pouvoir gérer les événements liés à la souris lorsqu'ils se produisent.



Si l'utilisateur clique sur la première option du menu « ARTICLE -> LITERATURE », les deux concepts seront ajoutés dans le point de vue « Publication » et il y aura un lien « plus spécifique » qui va de « ARTICLE » vers « LITERATURE ». La seconde option signifie que « LITERATURE » est « plus spécifique » que « ARTICLE ».

Pour développer l'application Elégie Eclipse (<http://www.eclipse.org/>) a été utilisée. Eclipse est une plate-forme de développement écrite en Java, fruit du travail d'un consortium de grandes entreprises (Borland, IBM, ...). Il en résulte un EDI performant et Open Source qui a su trouver sa place parmi les peintures du marché que sont JBuilder et NetBeans. Elle intègre pour cela la prise en charge des outils comme Ant, CVS, JUnit. Je n'ai malheureusement pas eu le temps d'utiliser ces trois outils mais j'y compte bien dans un prochain développement en Java. Ant (<http://ant.apache.org/>) est un outil d'automatisation principalement utilisé dans la construction et le déploiement d'applications Java. Cet outil Open Source, développé en Java, est issu des projets Apache. Ant est souvent comparé à l'utilitaire « make » en C/C++. Tout comme « make », le principe de Ant consiste à effectuer une série de tâches dépendantes les unes des autres en évitant de refaire celles qui sont déjà à jour. CVS (Concurrent Versions System) est un outil libre de gestion des versions très utile pour des projets développés en équipe. Junit est un framework<sup>9</sup> de test.

Le code est documenté grâce à l'outil Javadoc pour permettre à une autre personne d'utiliser les services offerts par les classes écrites sans se préoccuper de l'implantation. En plus des commentaires Javadoc, d'autres commentaires ont été inclus dans le code source afin de faciliter la maintenance.

## 4. Lexique

Mise en forme : Puces et numéros

Les mots sont classés par ordre de première apparition.

**Ontologies :** Les ontologies sont en informatique un moyen de représenter la connaissance d'un domaine sous forme de concepts liés entre eux par des relations sémantiques.

**Hiérarchie de concepts :** Les hiérarchies de concepts représentent des concepts et les relations qu'ils ont entre eux mais à la différence des ontologies, les liens sont tous du même type (lien « est-un »).

---

<sup>9</sup> *Un framework est une infrastructure logicielle qui facilite la conception des applications par l'utilisation de bibliothèques de classes ou de générateurs de programmes, soit dit en quelques mots : un cadre de développement.*

OWL : OWL est un langage qui permet de représenter des ontologies, il est une extension de RDF et RDFS qui sont eux même des extensions de XML.

Méta-donnée : Une méta-donnée est littéralement une donnée sur une donnée. Plus précisément, c'est un ensemble structuré d'informations décrivant une ressource quelconque.

Système de raisonnement : Un système de raisonnement permet de faire des inférences c'est à dire de déduire de la connaissance à partir d'autre connaissance.

Plugin : Un plugin est un module qui étend les fonctionnalités d'une application.

Point de vue : Ce que nous avons appelé point de vue est en faite une hiérarchie de concepts construite à partir d'une ontologie et qui ne possède que des liens du type « plus-spécifique », les liens « plus-spécifique » sont issus d'une ontologie.

Mettre sur écoute au sens Java : Lorsque l'on met sur écoute un objet c'est pour être averti des différents événements qui peuvent se produire sur l'objet qui a été mis sur écoute. Les événements peuvent être de plusieurs types, les plus connus sont les événements souris (clic droit, gauche ...) et les événements liés au clavier.

AddMouseListener : Est une fonction qui permet de mettre sur écoute un objet c'est à dire de pouvoir gérer les événements liés à la souris lorsqu'ils se produisent.

Framework : Un framework est une infrastructure logicielle qui facilite la conception des applications par l'utilisation de bibliothèques de classes ou de générateurs de programmes, soit dit en quelques mots : un cadre de développement.

POO : Programmation Orientée Objet

UML : Unified Modeling Language

Mise en forme : Puces et numéros

## 5. Annexes

### 5.1 Structure de l'application

L'application est composée de plusieurs packages. Afin de permettre une meilleure réutilisabilité.

#### 5.1.1 Package « autres »

Il contient deux classes « ContextElegie » et « ContextPointDeVue ». Ce sont des classes « static » qui permettent d'avoir accès de n'importe où au contexte de l'application, par exemple aux graphes représentant l'ontologie et la hiérarchie.

La classe « ContextElegie » est la classe qui permet de connaître le contexte de l'application, elle est utilisée par le module de comparaison d'ontologie développé par Laurent Cardonner.

Dans un souci de réutilisabilité, j'ai créé la classe « ContextPointDeVue » qui encapsule « ContextElegie » plutôt que d'ajouter des attributs et des méthodes à « ContextElegie ». Dans toutes les classes que j'ai écrites, je ne me sers pas directement de la classe « ContextElegie » mais de « ContextPointDeVue ». De plus « ContextPointDeVue » contient des méthodes qui sont propres au module de création de point de vue que j'ai développé. Si une personne voulait rajouter des tâches en se servant du module de comparaison d'ontologie il suffirait qu'elle crée son contexte comme je l'ai fait et si elle voulait se servir de ce que j'ai fait, il faudrait qu'elle crée aussi son contexte qui utiliserait « ContextPointDeVue ».

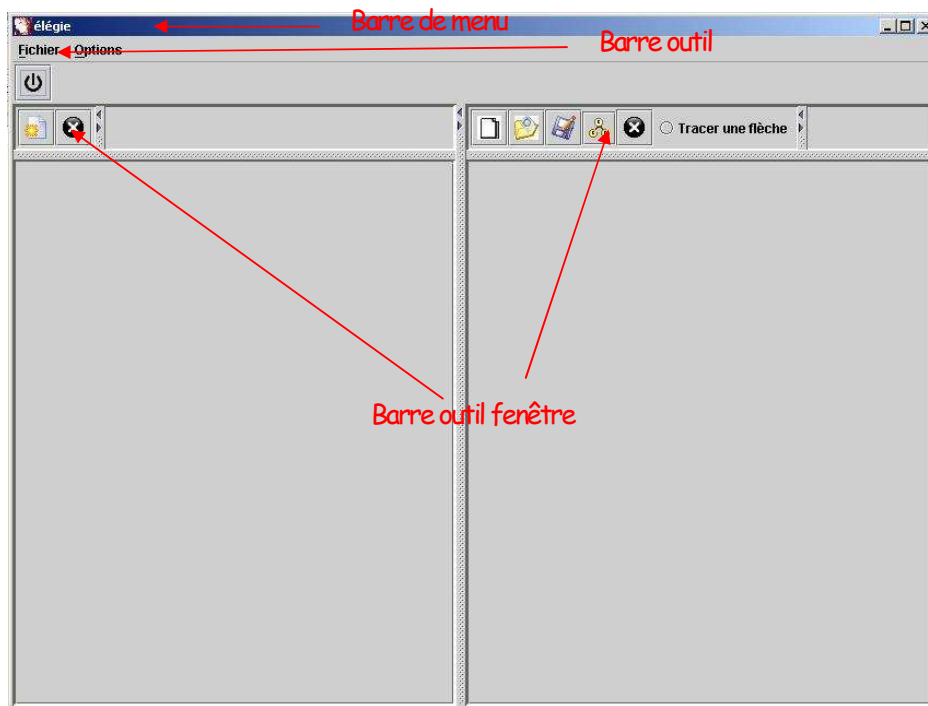
#### 5.1.2 Package « interfaces.ihm »

Mise en forme : Puces et numéros

Ce package contient tout ce qui est fenêtre, barre outil, barre de menu et ce qui permet de gérer les icônes. Toutes les classes post-fixées par « PointDeVue » sont des classes qui ne servent que pour la création et la manipulation de point de vue, c'est moi qui les ai écrites.

La barre de menu hérite de JMenuBar et contient « Fichier » et « Options ».



La barre d'outils et les deux barres outils fenêtre héritent de JToolBar.






### 5.1.3 Package « `metier.entite.presentation` »

Mise en forme : Puces et numéros

Ce package contient toutes les classes qui permettent de faire des actions. Toutes les actions héritent de `AbstractAction`. Voici la liste de celles que j'ai écrites :

- `CreerLienPlusSpécifiqueAction` : Cette classe permet de créer une option du `JPopupMenu` qui va permettre de transformer une ontologie en hiérarchies de concepts et de gérer le traitement à effectuer lorsque l'utilisateur clique sur cette option.
- `CreerPointDeVueAction`  : Cette classe permet de créer une option qui va permettre de créer un nouveau point de vue en entrant son nom, le concept sera créé dans la hiérarchie de concepts et sera de couleur jaune.
- `FermerFichierPointDeVueAction`  : Cette classe permet de créer une option qui va fermer le point de vue.



- `MettreEnPageHierarchieAction`  : Cette classe permet de mettre en page le point de vue (hiérarchie de concepts) créé. Cette action n'a d'effet que si la hiérarchie de concepts est cohérente. La mise en page consiste à mettre le concept le plus générique en haut et les plus spécifiques en bas, c'est un arbre dont la racine est le concept qui a pour nom le point de vue.
- `OuvrirPointDeVueAction`  : Cette classe permet de créer une option qui va permettre d'ouvrir un point de vue créé précédemment. L'ouverture d'un point de vue a pour effet d'afficher l'ontologie dans la partie gauche de la fenêtre principale et le point de vue dans la partie droite.
- `SauverHierarchieAction`  : Cette classe permet de sauvegarder le nouveau point de vue qui vient d'être créé au format OWL. Cette action n'aura d'effet que si la hiérarchie de concepts est cohérente. Un fichier d'index est également créé, il permet de retrouver tous les points de vue créés à partir d'une ontologie.

#### **5.1.4** Package « `utilitaires.gestionnaires` »

Mise en forme : Puces et numéros

Ce package contient toutes les classes qui me permettent de gérer le clavier et la souris sur la représentation graphique de l'ontologie et du point de vue (hiérarchie de concepts). C'est notamment l'une des classes de ce package qui me permet d'afficher un menu Popup sur la représentation de l'ontologie afin de transformer un lien en « plus spécifique ».

#### **5.1.5** Package « `utilitaires.graphes` »

Mise en forme : Puces et numéros

J'ai écrit toutes les classes de ce package sauf « `AffichageGraphe` ». Les classes « `Graph` » et « `Nœud` » me permettent de gérer une structure de donnée de type graphe pour vérifier la cohérence du point de vue. En effet un point de vue est un graphe qui a les propriétés suivantes :

- Une unique racine
- Chaque nœud n'a qu'un prédécesseur

La classe « MonJGraph » permet de représenter le point de vue de manière graphique. Cette classe utilise une bibliothèque gratuite nommée « JGraph » [www.jgraph.com](http://www.jgraph.com).

### **5.1.6** Package « utilitaires.preferences »

Mise en forme : Puces et numéros

Ce package contient les classes permettant de gérer le multi-langage. En effet tous les messages, titre de fenêtre et libellé sont dans des fichiers et non pas dans le code. Ces classes permettent de choisir la langue dans laquelle on veut l'application et aussi de récupérer à partir d'une clé spécifique qui est utilisée dans le code le libellé dans la langue choisi par l'utilisateur.

Les fichiers contenant les libellés des différentes langues pour l'application sont dans le répertoire « langues » du package « utilitaire.preference ».

Actuellement Elégie supporte deux langues, le français et l'anglais.

Le fichier « elegie\_en.properties » contient les libellés anglais.

Le fichier « elegie\_fr.properties » contient les libellés en français.

Le fichier « elegie.properties » contient les libellés en français. Ce fichier est utilisé par défaut si le fichier correspondant à la langue choisie par l'utilisateur n'est pas présent dans le répertoire « langues ».

Je ne vais pas parler des autres packages car je ne les ai pas développés moi-même.

### **5.1.7** La classe MainElegie

Mise en forme : Puces et numéros

Elle ouvre une fenêtre permettant de choisir entre la comparaison d'ontologies ou la création de points de vue. Puis après elle enregistre les actions et ouvre la fenêtre principale. C'est dans le constructeur de la fenêtre principale, qu'en fonction du choix de l'utilisateur, les barres d'outils, de fenêtre et de menu sont affichées.