Thésaurus, l'analyse

Franck Petitdemange
Nordine El Hassouni
Issam Amal
Cyril Monmouton
Marouane Denguiri
Mohamed Amine Zahir
Abdelhamid Belarbi

12 janvier 2013

Table des matières

1 Genèse 2
1.1 Le groupe et le sujet . . . 2
1.2 Définitions . . . 2
1.2.1 Définitions globales . . . 2
1.2.2 Pour les relations entre termes . . . 2
1.2.3 Pour les modes d'indexation . . . 3

2 Analyse 4
2.1 Spécifications . . . 4
2.2 Diagramme Use-Case . . . 4
2.2.1 Analyse textuelle . . . 5

3 Conception 6
3.1 Diagramme des classes . . . 6
3.2 Base de données . . . 7
3.3 Et les vues . . . 8

4 Réalisation 9
4.1 Choix des outils . . . 9
4.2 Répartition des tâches . . . 9
4.3 Details techniques . . . 10
4.3.1 L'affaire Oracle Express . . . 10
4.3.2 SQL . . . 10
4.3.3 PDO . . . 11

Chapitre 1

Genèse

1.1 Le groupe et le sujet

Le groupe est constitué des sept personnes nommées sur la page de garde et le responsable du projet est Abdelhamid Belarbi [1].
Nous avons choisi de traiter le thème de l'*astronomie* au sens large du terme. Nous inclurons donc des termes tels que *Comète*, *Galaxie*, *Étoile* ou bien *Voie Lactée*.

1.2 Définitions

Internet regorge de définitions de thésaurus et des notions qui s'y rapportent (voir [1], [2], [3] et [4]). Toutefois, beaucoup d'entre elles sont peu claires, très vagues voire obscures. En tant que concepteurs d'une application, nous nous devions de réunir les meilleures définitions, les clarifier et les spécifier sans ambiguïté.

Nous utiliserons les définitions suivantes.

1.2.1 Définitions globales

Terme : mot ou combinaison de mots significatifs.

Descripteur : terme choisi pour caractériser les informations, aide à l'indexage et la recherche (= terme préférentiel ou vedette).

Non-descripteur : terme non accepté à l'indexation, il peut s'agir de synonyme, abréviation ou variante orthographique d'un descripteur (= terme non préférentiel ou synonyme).

Microthésaurus : ensemble de descripteurs reliés au même concept. Noté *MT*.

Thésaurus : ensemble de micro-thésaurus.

1.2.2 Pour les relations entre termes

EP (Employé Pour) : relation entre un descripteur et le (ou les) non-descripteur qu'il référence.

EM (Employer) : relation entre un non-descripteur et son descripteur (symétrique de la relation précédente).

1. Élu démocratiquement à 71,42 % des voix.

TA (Terme Associé) : relation entre des descripteurs appartenants à des micro-thésaurus différents, mais entre lesquels peuvent exister des proximités sémantiques.

TS (Terme Spécifique) : relation entre un descripteur plus général et un (ou plusieurs) descripteur plus spécifique.

TG (Terme Général) : relation entre un descripteur plus spécifique et un descripteur plus général (symétrique de la relation précédente).

1.2.3 Pour les modes d'indexation

Liste alphabétique structurée : Les descripteurs sont affichés par ordre alphabétique avec leur relations sémantiques.

Liste par micro-thésaurus : Les descripteurs sont affichés hiérarchiquement, du plus général au plus spécifique.

Liste alphabétique permutée : Les termes sont affichés selon les notions dans lesquelles ils apparaissent.

Chapitre 2

Analyse

2.1 Spécifications

Les fonctionnalités que nous voulons faire figurer dans l'application sont les suivantes :
- Recherche par mot clé sur l'ensemble des descripteurs ;
- Affichage du thésaurus indexé des trois manières possibles ;
- Interface d'administration permettant d'ajouter et de manipuler des termes.

Ces différentes fonctionnalités sont décrites plus en détail dans le diagramme suivant.

2.2 Diagramme Use-Case

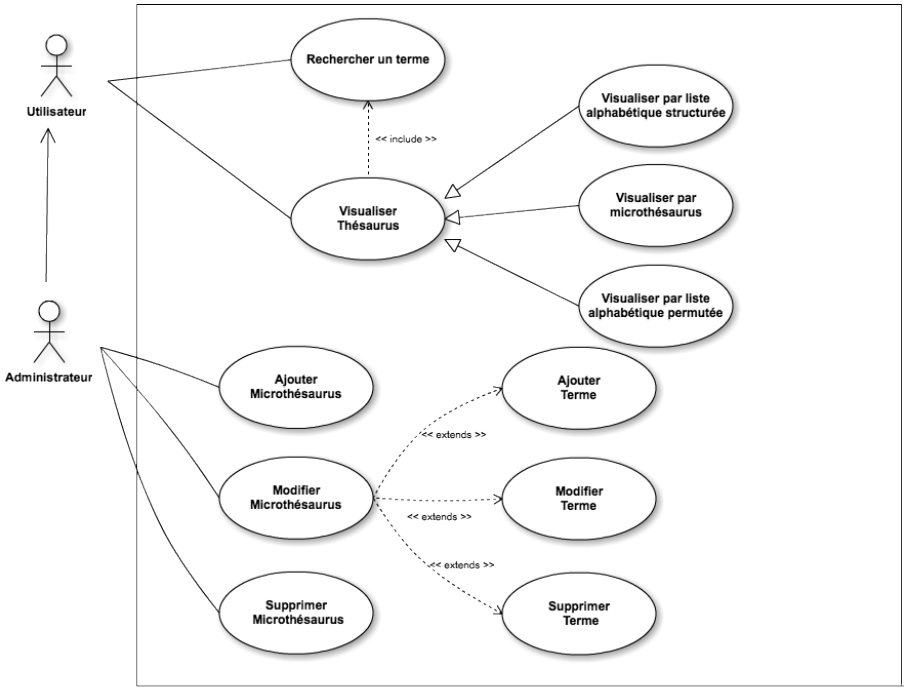

Figure 2.1 – Le diagramme des cas d'utilisation

2.2.1 Analyse textuelle

Quelques précisions s'imposent quant au diagramme précédent.

- L'utilisateur est un simple visiteur anonyme.
- La recherche s'effectue en spécifiant un mot-clé et un mode d'affichage (d'indexation).
- Visualiser un thésaurus revient soit à le voir en entier avec tout les termes de la base, soit un seul micro-thésaurus, soit un seul terme avec tous les termes auxquels il est lié.
- Lorsque l'administrateur ajoute ou modifie un terme, il peut spécifer le micro-thésaurus auquel il appartient d'où les flèches *<<extends>>*.

Chapitre 3

Conception

3.1 Diagramme des classes

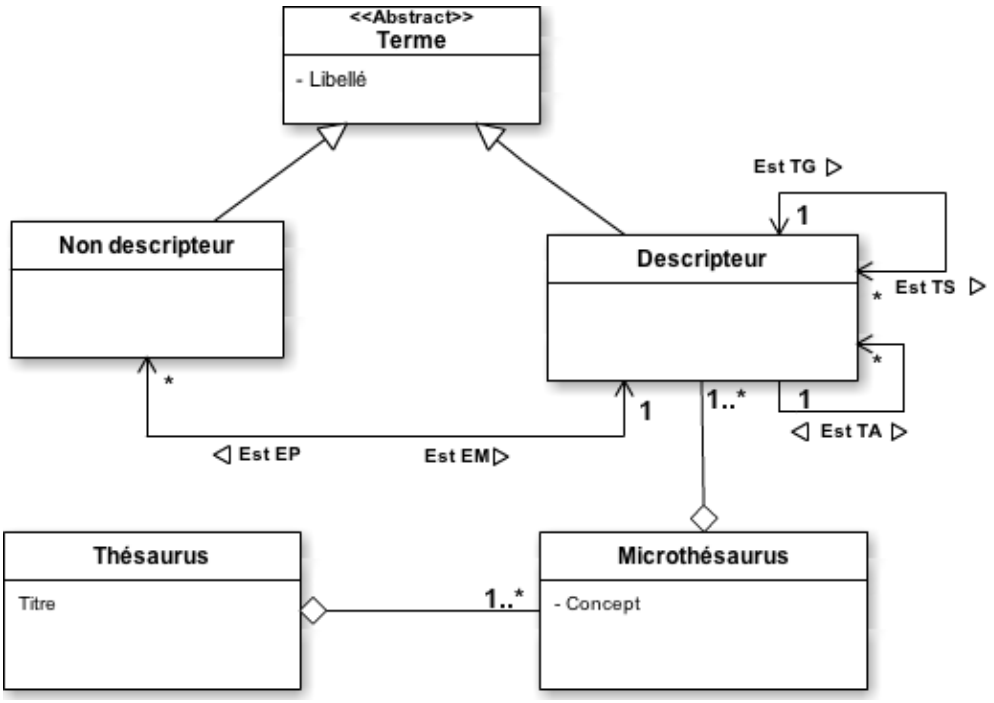

Figure 3.1 – Le diagramme des classes

Le diagramme parle de lui-même. Toutes les définitions données dans la section 1.2 sont appliquées, d'où l'utilité de la démarche de clarification effectuée.

3.2 Base de données

Le schéma objet-relationnel est directement inspiré du diagramme des classes. Il se présente de la manière suivante (code ODL).

```
interface Terme {
        attribute string libelle;
}

class Descripteur:Terme {
        relationship list(NonDescripteur) EP      // Employé pour
                    inverse NonDescripteur::EM;
        relationship list(Descripteur) TA         // Termes associés
                    inverse Descripteur::TA;
        relationship list(Descripteur) TS         // Termes spécifiques
                    inverse Descripteur::TG;
        relationship Descripteur TG               // Terme général
                    inverse Descripteur TS;
        relationship Microthesaurus MT
                    inverse Microthesaurus::concept;
}

class NonDescripteur:Terme {
        relationship Descripteur EM;              // Employer
}

class Microthesaurus {
        attribute string concept;
        relationship list(Descripteur) descripteurs
                    inverse Descripteur::MT
}

class Thesaurus {
        attribute string titre;
        relationship list(Microthesaurus) microthesaurus;
}
```

Concernant la classe *Thesaurus*, il est à noter qu'elle est inutile puisque nous ne traitons qu'un seul thésaurus. Mais par souci de clarté et par respect des définitions et du sujet nous la mettons quand même.

Pour ce qui est des autres classes la transformation est triviale, chaque lien d'association est traduit par la relation correspondante. Nous avons choisi d'utiliser des *list* pour représenter les données multiples. Les relations inter-objets seront explicitées grâce au type *REF*.

Remarquez de plus que les opérations des objets ne sont pas représentées ici car il ne s'agit globalement que d'accesseurs.

3.3 Et les vues

Il est assez judicieux de penser à créer des vues pour représenter les différents modes d'indexation afin notamment d'optimiser le temps de travail du serveur.
On peut penser par exemple à une vue *listeAlphabetiquePermutee* qui se présenterait de la sorte :

```
view listeAlphabetiquePermutee {
        libelle string;
        TG string;                // Le libellé du terme général
        TA list(string);          // Les libellés des termes associés
        EP list(string);          // Les libellés des termes employés
}
```

En étudiant cette vue d'assez près on se rend compte qu'elle possède à peu de chose près la même structure que la table *Descripteur*. C'est sur cette table (et uniquement celle là) qu'il faut effectuer une requête pour construire cette vue. C'est donc une sorte de redondance des données.

Pour mettre à jour les vues il faut implémenter différents triggers qui se déclencheront à la mise à jour des tables concernées (ici *Descripteur*). Ces ajoutent des traitements du SGBD pas forcément utiles.

Le mieux pour représenter les 3 modes est d'effectuer une requête par mode. Ainsi nous tirons parti de la notation pointée de l'objet-relaionnel qui nous permet d'écrire

```
self.tg.libelle
```

plutôt que de faire une jointure comme en relationnel.

Ce sont toutes ces raisons qui nous dispensent d'écrire des vues dans notre base objet-relationnel.

Chapitre 4

Réalisation

4.1 Choix des outils

L'essor des technologies du web ces dernières années à fait que nous avions une large palette d'outils à notre disposition. Nous avons fait le choix, lors d'une réunion de groupe, d'utiliser les outils suivants :

- Pour le back-end, le langage PHP avec un projet organisé selon le modèle MVC.
- La base de données sera gérée par Oracle Express utilisant le langage SQL.
- Pour le front-end, une interface Html/Css tout ce qu'il y a de plus classique.

En plus des outils déjà cités nous avons décidé d'utiliser *Smarty*, un générateur de template pour PHP qui simplifie grandement l'affichage des vues.
Les versions du projet furent gérées par *Git*. D'ailleurs voici le lien vers le dépôt :

`https://github.com/AbdelBa/Thesaurus`

Nous avons aussi discuté via Google Groupes pour organiser les réunions, poser les questions publiques et synchroniser notre travail.

4.2 Répartition des tâches

La taille du groupe nous permet de travailler sur chaque partie du projet simultanément, nous avons donc divisé le groupe en sous-équipes, chacune chargée d'une partie de la besogne.

1. Équipe SQL : Issam et Abdelhamid
2. Équipe PHP : Franck, Marwan, Cyril et Mohamed
3. Équipe Graphisme : Nordine

L'équipe *SQL* fut chargée de l'implémentation de la base de données. C'est à dire la création des types objets, des tables ainsi que tout élément nécessaire au bon fonctionnement de l'application.
L'équipe PHP devait programmer les solutions à intégrer au modèle MVC. Ce qui signifie faire les classes, les scripts de connexion à la base de données, la logique metier...
L'équipe singleton de Nordine était chargée de faire une interface graphique simple et sobre et de paufiner le travail de l'équipe PHP sur les vues.

4.3 Details techniques

Jusqu'à maintenant tout se passait bien, une immense plaine verdoyante s'étendait de toutes parts à perte de vue et la brise légère faisait onduler nos cheveux de manière quasi-imperceptible.
Mais lorsqu'il s'est agi de programmer, commencèrent à émerger ça et là divers tourments que voici énumérés.

4.3.1 L'affaire Oracle Express

Oracle propose sur son site une version de son célèbre SGBD dénommée Oracle Express. À notre grand désarroi cette version n'est disponible que pour Windows x32 ou Linux x64 et c'est tout.
Autant dire que les deux membres du groupe possesseurs d'un ordinateur Apple étaient hors course, eux qui fanfaronnaient si souvent avec leur machine en aluminium ne pouvaient même pas faire fonctionner Oracle Express dans une machine virtuelle.
Mais l'affaire fut vite réglée lorsque ils prirent la décision de venir travailler à l'université.

Bien sûr il y avait un outil en ligne fourni par Oracle même, Apex (voir [7]), mais la base de données qui est procurée au développeur n'est disponible qu'au sein même d'Apex sans possibilité de connexion externe.

4.3.2 SQL

Le lecteur attentif aura remarqué que dans le chapitre 3, les relations entre les différentes classes sont drôlement entremêlées (voyez la classe *Descripteur*). SQL n'offre pas la possibilité de faire des références en avant, ce qui pose un problème lorsque deux types dépendent l'un de l'autre. La solution réside dans ce qui est appelé dans la documentation les types incomplets. Voici comment nous avons mis en oeuvre cette solution (voir le fichier *src/modeles/sql/types.sql*) :

```
-- Créations de types incomplets pour les références en arrière.
CREATE OR REPLACE TYPE NonDescripteur_t UNDER Terme_t (); /
CREATE OR REPLACE TYPE Descripteur_t UNDER Terme_t (); /

-- Une table imbriquées pour stocker une liste de non descripteurs.
CREATE OR REPLACE TYPE ListeNonDescripteurs_t AS
    TABLE OF REF NonDescripteur_t; /

-- Type Descripteur_t
ALTER TYPE Descripteur_t ADD ATTRIBUTE (
    EP ListeNonDescripteurs_t
)CASCADE;

-- Type NonDescripteur_t
ALTER TYPE NonDescripteur_t ADD ATTRIBUTE (
    EM REF Descripteur_t
)CASCADE;
```

4.3.3 PDO

Nous avons d'abord un peu hésité à utiliser le driver PDO pour oracle de PHP car il est écrit dans la documentation [6], dans un grand cadre rouge que « Ce module est EXPERIMENTAL. Cela signifie que le comportement de ces fonctions, leurs noms et, concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS ! » Comme le projet ne devait pas durer trop longtemps et que l'ancien driver (OCI8) n'était pas fait en objet nous avons quand même utilisé PDO.

Les problèmes rencontrés avec de module ne se situaient pas là mais plutôt au niveau de la configuration des différents acteurs, à savoir PHP (via le fichier *.ini*) et le serveur Apache. Il faut savoir que sous Windows, le driver PHP ne fonctionne qu'avec un serveur 32 bits, étrange...

Webographie

[1] L'encyclopédie que l'on ne présente plus, utile pour les définitions.
Adresse : `fr.wikipedia.org`

[2] Un thésaurus très bien fait qui est présenté dans les trois modes d'indexation.
Alphabétique structuré : `www.inpes.sante.fr/30000/pdf/thesaurus/thes_01.pdf`
Microthésaurus : `www.inpes.sante.fr/30000/pdf/thesaurus/thes_02.pdf`
Alphabétique permuté : `www.inpes.sante.fr/30000/pdf/thesaurus/thes_03.pdf`

[3] Le (très) gros thésaurus de la Banque de données en Santé Publique, très instructif.
Adresse : `asp.bdsp.ehesp.fr/Thesaurus/`

[4] Le thésaurus de l'Unesco, une autre aide précieuse pour bien comprendre les principes.
Adresse : `databases.unesco.org/thesfr/`

[5] La documentation Oracle officielle pour faire du SQL dans les normes.
Adresse : `docs.oracle.com/cd/E11882_01/server.112/e26088/toc.htm`

[6] La documentation de PHP PDO.
Adresse : `fr2.php.net/manual/fr/ref.pdo-oci.php`

[7] Le service web qui fournit une base de données consultable via le navigateur.
Adresse : `apex.oracle.com`