

# Thésaurus Rex

Baptiste LE-BAIL    Thibaut MARMIN  
Namrata PATEL    Clément SIPIETER  
Steeve TUVÉE

Projet BDD GMIN103 : réalisation d'un Thésaurus  
[https://github.com/thibautmarmin/bdd\\_projet/](https://github.com/thibautmarmin/bdd_projet/)

20 Janvier 2012



Besoins

○

Modélisation

○  
○  
○  
○

Choix de conception

○○○  
○○  
○

Implémentation

○○○  
○○○○○○○  
○  
○○○○

Conclusion & Démonstration

# Thésaurus Rex

Besoins

Implémentation

Modélisation

Conclusion & Démonstration

Choix de conception



Besoins

○

Modélisation

○  
○  
○  
○

Choix de conception

○○○  
○○  
○

Implémentation

○○○  
○○○○○○○○  
○  
○○○○

Conclusion & Démonstration

# Thésaurus Rex

Besoins

Fonctionnalités

Modélisation

Choix de conception

Implémentation

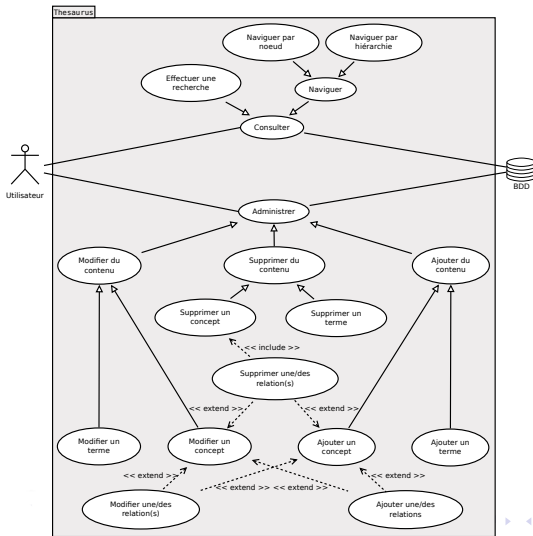
Conclusion & Démonstration





# Fonctionnalités

## Diagramme de cas d'utilisations



# Thésaurus Rex

Besoins

Implémentation

**Modélisation**

Une première piste

Évolution

Décision finale

Conclusion & Démonstration

Choix de conception



# Modélisation

Deux type d'entités :

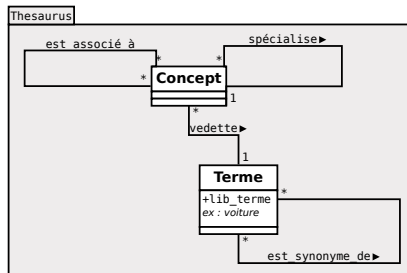
- Termes
- Concepts





# Une première piste

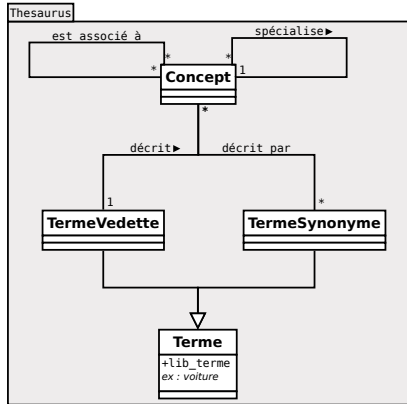
## Diagramme de classes





# Évolution

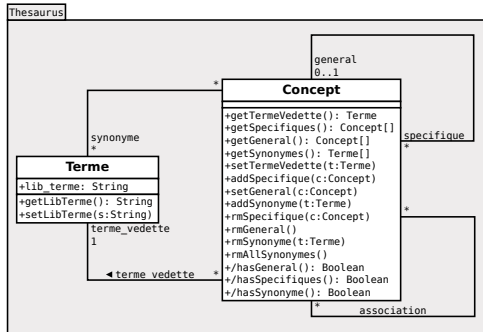
## Diagramme de classes





# Décision finale

## Diagramme de classes



Besoins

○

Modélisation

○  
○  
○  
○

Choix de conception

○○○  
○○  
○

Implémentation

○○○  
○○○○○○○○  
○  
○○○○

Conclusion & Démonstration

# Thésaurus Rex

Besoins

Implémentation

Modélisation

Conclusion & Démonstration

Choix de conception

Paradigmes

ORM

Décision



# Paradigmes

## Vision objet

### Avantages

- abstraction
- réutilisation
- maintenance

### Inconvénients

- peu de ODBMS



# Paradigmes

## Vision objet-relationnel

### Avantages

- norme SQL3

### Inconvénients

- implémentation partielle de la norme



# Paradigmes

## Vision relationnel pur

### Avantages

- mature
- performant
- compatibilité

### Inconvénients

- schéma relationnel à adapter à une programmation souvent objet



Besoins

○

Modélisation

○  
○  
○  
○

Choix de conception

○○○  
●○  
○

Implémentation

○○○  
○○○○○○○○  
○  
○○○○

Conclusion & Démonstration

# ORM

Qu'est ce qu'un ORM ?



# ORM

Pourquoi ?



# Décision





Besoins

○

Modélisation

○  
○  
○  
○

Choix de conception

○○○  
○○  
○

Implémentation

○○○  
○○○○○○○○  
○  
○○○○

Conclusion & Démonstration

# Thésaurus Rex

Besoins

Modélisation

Choix de conception

Implémentation

Framework

Structure de l'application

Schéma relationnel généré

Templates finaux

Conclusion & Démonstration



# Framework

## Symfony2

### Histoire

- Créé en 1998
- SensioLabs
- Framework aujourd'hui mature

### Caractéristiques

- MVC (Mojavi)
- ORM (Doctrine2)
- Template (Twig)



# Framework

## Structure des applications

### Bundle

- module ou plugin
- portable
- facilement installable
- architecture MVC

### Entity

- classe
- paramétrable avec l'ORM
- contrôleur
- formulaires



# Framework

## ORM Doctrine2

### Doctrine2

- ORM populaire
- GNU LGPL

### Intégration

- ORM par défaut de Symfony
- tag @ORM



# Structure

## Entité *Terme*

```
<?php
```

```
namespace ProjetBDD\Thesaurus\Bundle\Entity;
```

```
use Doctrine\ORM\Mapping as ORM;
```

```
/**
```

```
 * ProjetBDD\Thesaurus\Bundle\Entity\Terme
```

```
 * @ORM\Table()
```

```
 * @ORM\Entity(repositoryClass="ProjetBDD\Thesaurus\Bundle\Entity\
    TermeRepository")
```

```
 */
```

```
class Terme
```



# Structure

## Entité *Terme*

```
/**
 * @var string $id
 * @ORM\OneToMany(targetEntity="Concept",mappedBy="
    terme vedette")
 * @ORM\Column(name="id", type="string", length="255")
 * @ORM\Id
 */
private $id;
```





# Structure

## Entité *Concept*

```
<?php
```

```
namespace ProjetBDD\Thesaurus\Bundle\Entity;
```

```
use Doctrine\ORM\Mapping as ORM;
```

```
use Symfony\Component\Validator\Constraints as Assert;
```

```
/**
```

```
 * ProjetBDD\Thesaurus\Bundle\Entity\Concept
```

```
 *
```

```
 * @ORM\Table()
```

```
 * @ORM\Entity(repositoryClass="ProjetBDD\Thesaurus\Bundle\Entity\
    ConceptRepository")
```

```
 */
```

```
class Concept
```

```
{
```





# Structure

## Entité *Concept*

```
/**
 * @var integer $id
 *
 * @ORM\Column(name="id", type="integer")
 * @ORM\Id
 * @ORM\GeneratedValue(strategy="AUTO")
 */
private $id;
```







# Structure

## Entité *Concept*

```
/**
 * @ORM\ManyToOne(targetEntity="Terme", inversedBy="id")
 * @Assert\NotBlank()
 */
private $terme_vedette;
```





# Structure

## Entité *Concept*

```
/**
 * @ORM\ManyToOne(targetEntity=" Concept",inversedBy="
    concepts_specifiques" )
 */
private $concept_general;

/**
 * @ORM\OneToMany(targetEntity=" Concept",mappedBy="
    concept_general" )
 */
private $concepts_specifiques;
```



# Structure

## Entité *Concept*

```
/**  
 * @ORM\ManyToOne(targetEntity="Terme")  
 */  
private $synonymes;
```





# Structure

## Entité *Concept*

```

/**
 * @ORM\ManyToMany(targetEntity=" Concept", mappedBy="
    associations")
 */
private $associes_avec_moi;
/**
 * @ORM\ManyToMany(targetEntity=" Concept", inversedBy="
    associes_avec_moi")
 * @ORM\JoinTable(name=" concept_concept",
 * joinColumns={@ORM\JoinColumn(name=" concept1_id",
    referencedColumnName=" id" )},
 * inverseJoinColumns={@ORM\JoinColumn(name=" concept2_id",
    referencedColumnName=" id" )})
 */
private $associations;

```



# Schéma relationnel généré



# Templates finaux

## Accueil



# Templates finaux

Formulaire ajout



Besoins

○

Modélisation

○  
○  
○  
○

Choix de conception

○○○  
○○  
○

Implémentation

○○○  
○○○○○○○  
○  
○○●○

Conclusion & Démonstration

# Templates finaux

bla bla bla





# Templates finaux

bla bla



Besoins

○

Modélisation

○  
○  
○  
○

Choix de conception

○○○  
○○  
○

Implémentation

○○○  
○○○○○○○○  
○  
○○○○

Conclusion & Démonstration

# Thésaurus Rex

Besoins

Implémentation

Modélisation

Conclusion & Démonstration

Choix de conception



Besoins

○

Modélisation

○  
○  
○  
○

Choix de conception

○○○  
○○  
○

Implémentation

○○○  
○○○○○○○  
○  
○○○

Conclusion & Démonstration

## Conclusion & Démonstration

