



Université de Yaoundé I
Département Informatique
INF5039 - Ingénierie Dirigée par les Modèles

Rapport de projet :
Modélisation de portails pédagogiques et
génération de code

Groupe 8

Réalisé par :

NDALI NABIA Guy Serge 18Z2231

Mahamat DORSOUMA 18Z2251

Sous la Supervision de:

Dr. Valéry MONTHE

Table des matières

I. INTRODUCTION.....	3
II. Outils nécessaires.....	4
III. métamodèle.....	4
1. Métamodèle du portail pédagogique PPUY.....	4
2. Le Métamodèle HTML.....	7
3. Métamodèle Bootstrap.....	8
IV. Les Transformations.....	9
1. Transformation M2M (modèle à modèle) :.....	9
2. Transformation M2T(modèle à texte).....	9
V. Structure finale du dossier du projet (livrable).....	10
VI. Quelques captures d'écran des pages Pages générées.....	10
VII. Conclusion.....	11

I. INTRODUCTION

- contexte

Ce projet s'inscrit dans le cadre du cours de l'ingénierie dirigée par le modèle (IDM) en Master 2 option Génie logiciel. Le but est de mettre en pratique les concepts IDM acquis en cours.

◆ Objectifs

Pouvoir créer un métamodèle et de générer à partir de ce métamodèle, un portail pédagogique en prenant comme modèle le portail de l'université de Lille (<https://www.fil.univ-lille1.fr/portail/>).

◆ Problématique

Générer les sites web des différents départements de l'Université de Yaoundé 1 à partir d'une description complète des informations à présenter. Nous voulons plus tard être capable de générer le contenu dans un site web avec un agencement ou une présentation différente. Mais nous voulons aussi pouvoir adapter cette présentation à des formats encore plus différents comme

◆ Méthode adoptées (option 2 du projet):

la méthode consiste à créer un métalangage du portail ensuite créer un deuxième métamodèle pivot (HTML) qui nous permettra d'avoir une génération uniforme et standardiser de notre portail. Ensuite passer du modèle pivot à un troisième métamodèle Bootstrap. La génération du code final nécessite 3 transformations: deux règles de transformation de modèle à modèle et une règle de transformation de modèle à texte. Nous aurons au final les livrables suivants :

- Trois métamodèles Ecore
- deux transformations QVT modèle à modèle : métalangage vers HTML et HTML vers Bootstrap.
- une transformation avec Aceleo : modèle à texte.

La suite de ce document permettra essentiellement d'explicitier les différents points ci-haut, c'est à dire, l'explication des différents métamodèles, des différentes règles de transformation et aussi rappeler les outils nécessaires dans Eclipse modeling. Nous présenterons des captures d'écran des différentes pages générées.

II. Outils nécessaires

Pour utiliser ce projet, les outils listés ci-dessous doivent être installés :

■ EMF:

Selon Eclipse, EMF est un cadre de modélisation et un outil de génération de code pour créer des outils et d'autres applications basés sur un modèle de donnée structurée. Il se base sur une spécification de modèle pour fournir des outils et une prise en charge de l'exécution pour produire un ensemble de classes java pour le modèle ainsi qu'un éditeur de base.

Pour ce projet, EMF permet d'avoir un aperçu du métamodèle et éventuellement de le faire évoluer. Il permet également de générer le code de manipulation des modèles (dont l'éditeur arborescent).

■ Qvto:

Le composant opérationnel Eclipse QVT est une implémentation du langage de mappage opérationnel défini par MOF (Meta Object Facility). Il assure la transformation d'un modèle A vers un nouveau modèle B.

■ Acceleo :

Basé sur des modèles comprenant des outils de création pour créer des générateurs de code personnalisés, Acceleo permet de produire automatiquement tout type de code source à partir de n'importe quelle source de données disponible au format EMF. Ainsi, il nous permet de générer du code pour Android en java à partir des modèles conforme à notre métamodèle de départ.

Éclipse:

Nous utilisons cet IDE pour permettre de programmer et compiler EMF, QVTO, Acceleo et ATL.

III. métamodèle

1. Métamodèle du portail pédagogique PPUY

le métalangage que nous avons défini permet de décrire les différentes structure des pages du portail pédagogique de l'université de Yaoundé.

- PPUY: représente une page dans le portail pédagogique de l'université de Yaoundé 1 et est composé de tous les autres éléments du métamodèle.
- Département est composé de formations et des enseignants.
- Personne: modélise l'ensemble des personnel du département. D'après le modèle, enseignant peut être chef de département, il peut être responsable d'un format, il peut aussi enseigner plusieurs UE.
- Formation: a des contenus décrit sous forme de ressource, elle est départagée par niveaux
- Niveau: dans un niveau il y a deux semestres.

- UE: une matière est appartient a un semestre, est dispose de ressource ,

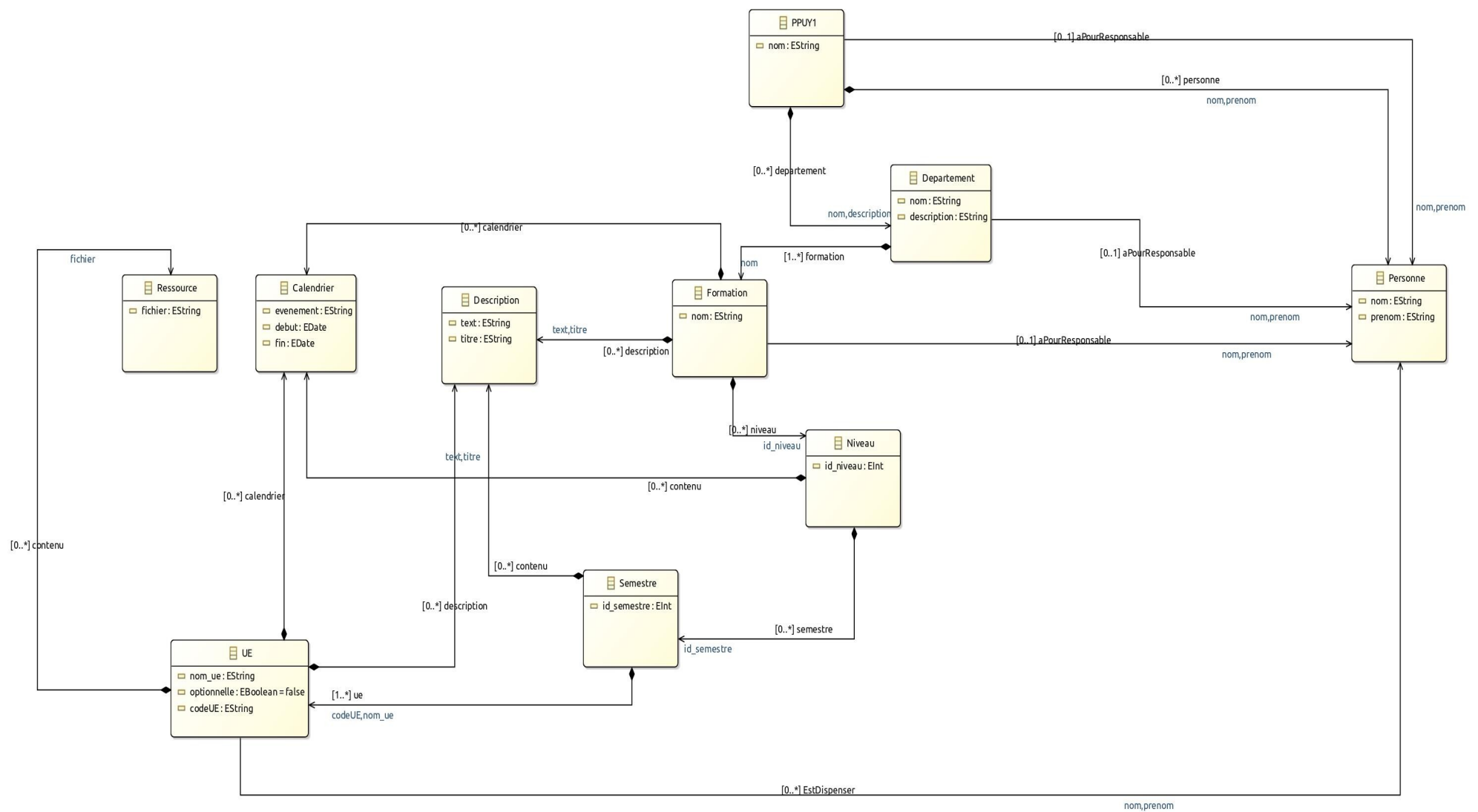


Fig1: métamodèle du portail

2. Le Métamodèle HTML

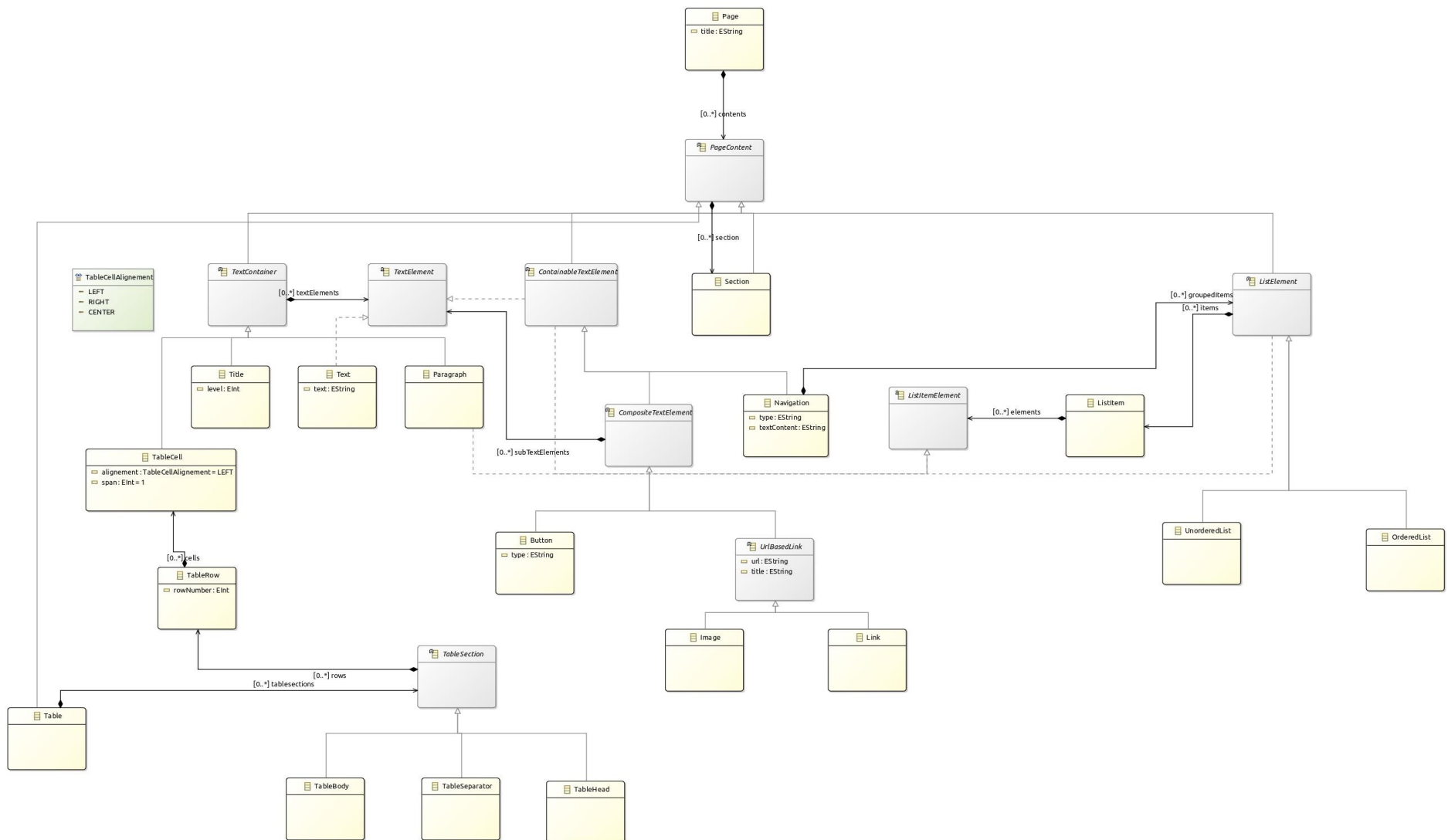


Fig.2 métamodèle d'une page HTML

Le Métamodèle Bootstrap nous permet de de décrire un fichier HTML avec les propriétés des classes Bootstrap. Les différentes classes représentent les propriétés des classes bootstrap.

- Bouton : un bouton ayant sa propre classe, son propre lien et son propre texte ;
- Navbar : une barre de navigation composée de plusieurs liens ;
- Line : une ligne horizontale permettant d'effectuer une séparation ;
- Container : une div permettant de stocker d'autres block sous formes de Colonnes ;
- Header : un entête avec un niveau de taille pouvant varier ;
- Paragraphe : un paragraphe textuel ;
- Link : un lien vers un autre site ;
- Image : une image.

IV. Les Transformations

1. Transformation M2M (modèle à modèle) :

Pour les transformation, nous avons utilisé QVTO pour les deux transformations. Les règles définies dans le fichier qvto permet de faire le mapping entre les différents éléments de chaque modèles. Le module transformation usd2html.qvt prend en entrée le modèle PPUY1.xmi et en sortie le modèle htmlmm.xmi. Le deuxième module prend en entrée le modèle htmlmm.xmi produit lors de la première transformation, et en sortie le modèle bootstrap.xmi.

```

14 /**
15  * Map a ppuy element to usd Page element.
16  */
17 mapping ppuymm::PPUY1::document2Page() : usdmm::Page {
18     title := self.nom;
19 }
20
21
22 // ===== Start transformation block to pagecontent =====//
23 /**
24  * Map a Block element to usd PageContent element.
25  */
26 abstract mapping ppuymm::PPUY1::block2PageContent() : usdmm::PageContent {}
27

```

fig4:extrait de code du fichier formation/usd2html.qvto

l'extrait de ce code permet de faire le mapping du plus haut élément du métalangage qui est ici PPUY1 du portail au plus haut élément dans la hiérarchie de notre modèle HTML qui est ici Page l'équivalent de la balise HTML.

2. Transformation M2T(modèle à texte)

nous avons utilisé Aceleo pour définir les règles de transformation et de génération du code HTML et Bootstrap. Les spécification se trouvent dans le fichier usd2bootstrap.qvto

V. Structure finale du dossier du projet (livrable)

Le projet est composé des Dossiers suivants :

- ProjetIdmPorttailP : le dossier principal du projet renfermant tout le projet sauf le dossier Acceleo ;
- Acceleo.generate : dossier générer par Acceleo contenant le fichier .mtl qui décrit la transformation M2T et les pages HTML générées dans le dossier Test.
- BootstrapModel qui contient le métamodèle Bootstrap ;
- projetIdm.bootstrapmm : contient le code générer en mm ;
- projetIdm.bootstrapmm.edit : contient le code générer en edit ;
- projetIdm.bootstrapmm.editor contient le code générer en editor ;
- projetIdm.htmlmm : contient le métamodèle générer en mm ;
- projetIdm.htmlModel.edit : contient le code générer en edit ;
- projetIdm.htmlModel.editor : contient le code générer en editor ;
- projetIdm.ppuymm : contient le code générer en code mm ;
- projetIdm.ppuymm : contient le code générer en code edit ;
- projetIdm.ppuymm.edit : contient le code générer en editor ;
- Transformation : contient les différentes les fichiers .qvto contenant les règles des transformations;
- Image : contient les différents capture d'écran métamodèles respectifs au format image ;
- Glossaire_protail : une brève description de la structure du portail pédagogique.

VI. Quelques captures d'écran des pages Pages générées



Fig 5 : Page accueil



fig 6 : image page de formation Génie logiciel



fig 7 : image page de formation licence fondamentale

VII. Conclusion

Ce projet nous a permis d'acquérir et de développer des nouvelles connaissances et compétences dans l'approche ingénierie dirigés par des modèles, les différentes règles de transformations. Il nous a permis:

- De mieux comprendre l'approche IDM utilisé pour développer cette application, ce qui nous a été utile dans le cadre de notre cursus académique.
- De mieux exploiter les règles de transformations, les modèles, les métamodèles et la génération des codes.
- notamment la découverte différentes plugins comme QVTO,
- de pousser davantage notre connaissance sur Acceleo.

En définitive, ce projet nous a permis de toucher concrètement aux concepts IDM et de découvrir combien c'est fascinant. Et surtout l'impact sur notre regard par rapport au développement logiciel qui n'est, évidemment plus le même.