

# Génération d'un schéma RDFS à partir d'un fichier XML modélisant le NoDEfr-2 Rapport de projet

Alexandre Vigneron, Sipeng Zheng

Avril-Mai 2021



## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Présentation du sujet</b>	<b>2</b>
2.1	NodeFR-2 . . . . .	2
2.2	Le sujet de notre projet . . . . .	4
<b>3</b>	<b>Analyse et implémentation</b>	<b>5</b>
3.1	Phase d'analyse et de recherche . . . . .	5
3.2	Implémentation . . . . .	6
3.3	Difficultés rencontrées . . . . .	7
3.4	Collaboration avec les autres groupes . . . . .	9
<b>4</b>	<b>Conclusion</b>	<b>10</b>
<b>5</b>	<b>Annexes</b>	<b>11</b>
5.1	Annexe 1 : première version du fichier XML exemple . . . . .	11
5.2	Annexe 2 : fichier XSLT . . . . .	13
<b>6</b>	<b>Webographie</b>	<b>14</b>

# 1 Introduction

Dans le cadre du cours de Document Web Sémantique, donné lors du semestre 4.2 dans le département Informatique et Technologie de l'Information de l'INSA Rouen Normandie, nous avons à réaliser un projet comptant pour 50% dans l'évaluation de la matière.

Ce projet a pour but de nous confronter à un problème en rapport avec les domaines enseignés dans le cours, comme les technologies XML, les moteurs de recherche, le web sémantique et le web des données. Pour cette année, deux sujets différents étaient proposés, eux même divisés en plusieurs sous-sujets. Le premier sujet portait sur l'IA explicable et le deuxième sur la formalisation d'un modèle conceptuel pour la description des offres de formation. Notre avons choisi de faire notre projet sur ce deuxième sujet, et plus particulièrement sur le sous-sujet consistant à générer un schéma RDF à partir d'un fichier XML.

Ce rapport va présenter notre démarche et expliquer notre implémentation. Il comportera dans une première partie une présentation complète du sujet et de notre sous-sujet, puis se portera sur notre analyse et notre implémentation avant de terminer sur une conclusion générale.

## 2 Présentation du sujet

### 2.1 NoDEfr-2

Afin de décrire les formations, des chercheurs de l'AFNOR ont décidé de réaliser un modèle conceptuel, du nom de NoDEfr-2. Ce modèle permettra de décrire les types de formation, en suivant le méta-modèle proposé par l'ISO dans le cadre du MLR (métadonnées pour les ressources d'apprentissage), ayant pour paradigme le paradigme ensembliste reposant sur la description de classes et d'éléments de données. Ce méta-modèle a été défini dans un article datant de 2012 écrit par Yolaine Bourda, Gilles Gauthier, Rosa-Maria Gomez de Regil et Olivier Catteau (le lien vers leur article est disponible dans les ressources de ce rapport).

Les experts de l'AFNOR ont donc décidé de réaliser un tableur (.ods) suivant ce méta-modèle pour modéliser leur travail. Ce tableur définit donc un certain nombre de classes et de propriétés. Les classes sont toutes définies dans une première feuille nommée "Classes". Chaque ligne représente une classe et chaque colonne représente un attribut de la classe. Les attributs définis par le MLR pour les classes sont "Identifiant", "Nom", "Définition", "Sous-classe de" et "Note".

	A	B	C	D	E
1	Identifiant	Nom	Définition	Sous classe	Note
2	ISO-IEC22602:RC0007	Institution	un élément de cette classe est une entité composée d'êtres humains, organisée dans la durée, en vue d'une finalité et dotée d'une personnalité juridique	RC0700	Exemple : Le lycée Flaubert de Rouen
3	RC0000	Entité du monde de la formation	classe abstraite qui regroupe les propriétés communes à toutes les situations ou à l'ensemble de situations visant un ou plusieurs objectifs de développement		Super classe de toutes les classes. Elle regroupe toutes les propriétés des classes présentes dans ce modèle
4	RC0015	Certificat, badge, diplôme	un élément de cette classe est la reconnaissance des compétences, des connaissances, des savoir-faire d'une personne	RC0000	Exemple : reconnaissance des compétences, des connaissances, des savoir-faire d'une personne
5	RC0016	Certification	un élément de cette classe est la reconnaissance de la qualité de l'offre d'un organisme donne selon un référentiel spécifique	RC0000	Exemple : Certification ISO 9001, QUALIOP1
6	RC0017	Niveau de certification	un élément de cette classe représente un niveau de certification dans un système de codification	RC0000	Exemple : Niveau 5 du CEC
7	RC0020	Offre de formation	un élément de cette classe est une formation décrite au niveau national, cette formation pouvant donner lieu à une attestation ou à préparer à un diplôme, une qualification, un certificat	RC0050, RC0030	Exemple : description d'un BTS services informatiques aux organisations (BTS SIO) au niveau national
8	RC0030	Condition d'accès	un élément de cette classe indique les conditions d'admission pour pouvoir postuler à une offre de formation	RC0000	Ces conditions peuvent être exprimées de façon formelle ou informelle, il peut y avoir plusieurs instances des modalités de sélection
9	RC0031	Prérequis	un élément de cette classe indique un ou plusieurs prérequis pour pouvoir postuler à une offre de formation	RC0000	Ces prérequis sont exprimés de manière informelle
10	RC0032	Modalité de sélection	un élément de cette classe indique une ou plusieurs modalités de sélection pour pouvoir être admis à une offre de formation	RC0000	Ces modalités de sélection sont décrites de manière informelle
11	RC0040	Frais de formation	un élément de cette classe est une dépense pour l'apprenant liée à la participation à la formation		Exemple : Les frais d'inscription au BTS SIO ou les frais pédagogiques du même BTS en formation continue
12	RC0050	Caractéristiques de la formation	classe abstraite qui regroupe les propriétés communes aux offres de formation et aux cursus décrivant l'organisation pédagogique d'une formation	RC0000	Super classe des classes "composition" obligatoire, optionnelle et à choix
13	RC0060	Composition	classe abstraite qui définit la composition d'une action de formation, d'un parcours ou d'un module en termes d'éléments de formation		Exemple : dans le BTS SIO pour le module "bloc 2" du parcours "SLAM", l'élément de formation "Conception et développement d'applications est obligatoire
14	RC0061	Composition obligatoire	classe qui définit le caractère obligatoire de la composition d'une action de formation, d'un parcours ou d'un module en termes d'éléments de formation	RC0060	
15	RC0062	Composition optionnelle	classe qui définit le caractère optionnel de la composition d'une action de formation, d'un parcours ou d'un module en termes d'éléments de formation	RC0060	Exemple : dans le BTS SIO pour le parcours "SLAM", l'élément de formation "Mathématiques approfondies" est optionnel
16	RC0063	Composition à choix	classe qui définit le caractère "à choix" de la composition d'une action de formation, d'un parcours ou d'un module en termes d'éléments de formation	RC0060	Exemple : dans le module M d'un parcours P, un étudiant peut choisir entre 2 et 3 éléments de formation parmi 5
17	RC0070	Crédits ECTS	classe qui définit le nombre de crédits ECTS qui peuvent être contextuels à une offre de formation	RC0000	
18	RC0080	Session d'information	classe qui présente une réalisation lors d'une session d'information	RC0000	Exemple : le BTS SIO option SLAM pour l'année 2020-2021 sera présentée aux étudiants le 4 avril 2020 à 14h

FIGURE 1 – Exemples de classes dans le fichier .ods modélisant le NoDefr-2

Les propriétés sont définies dans les autres feuilles. Chaque feuille porte le nom de la classe dont les propriétés font références. Chaque ligne représente une propriété et chaque colonne un attribut de la propriété. Les attributs définis par le MLR pour les propriétés sont "Identifiant", "Nom", "Definition", "Indicateur linguistique", "Co domaine", "Règle de contenu", "Raffine", "Exemples", "Notes", "Cardinalité min", "Cardinalité max" et "Raison de l'ordonnancement".

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Identifiant	Nom	Définition	Indicateur linguistique	Co domaine	Règles de contenu	Raffine	Exemples	Notes	Cardinalité min	Cardinalité max	Raison de l'ordonnancement		
2	DES000001	libellé	nommage humainement compréhensible de l'élément	oui	Littéral	MLR String		Pour une instance de la classe RC0020 Offre de formation, le libellé pourrait être "BTS services informatiques aux organisations"		1	1			
3	DES000002	date de mise à jour	dernière date de modification d'une information d'un élément	non	Littéral	MLR Date		Pour une instance de la Classe RC0105 Activité pédagogique, le changement de modalité pédagogique sera date		0	1			
4	DES000003	code	code "humain" permettant non de désigner de manière non ambiguë un élément		Littéral	MLR String		Au CNED le BTS SIO est identifié par le code "BTS11DIX20"	Ce code a du sens au niveau local	0	1			
5	DES000003001	contributeur	contributeur de la dernière modification de l'information	non	ISO IEC 19788-1:2010:RC0003				Ce contributeur peut être une personne physique ou morale	0	1			

FIGURE 2 – Exemples de propriétés dans le fichier .ods modélisant le NoDefr-2

La problématique du sujet est de formaliser automatiquement le travail réalisé par les experts de l'AFNOR en générant des schémas RDF et OWL. Pour cela, le sujet a été divisé en sous-sujet afin d'arriver à ces schémas "finaux". Tout d'abord la première étape était de développer une grammaire XSD pour vali-

der des fichiers XML qui formaliseraient les classes et éléments de données du NoDefr-2. La deuxième étape était donc de générer un fichier XML à partir du fichier .ods et qui respecte cette grammaire. Ensuite à partir de ce fichier XML il était possible de générer des schémas RDFS et OWL. 4 groupes distincts travaillaient donc sur le sujet, l'organisation était donc un point important.

## 2.2 Le sujet de notre projet

Le sous-sujet sur lequel nous avons travaillé est le 3ème, c'est à dire générer un schéma RDFS à partir du fichier XML.

Le RDF est un formalisme de représentation de métadonnées proposés le W3C. Il permet de décrire les ressources du Web, ce qui est utile notamment pour permettre le traitement automatique de ces données. Il est souvent représenté sous forme de triplet (sujet, prédicat, objet).

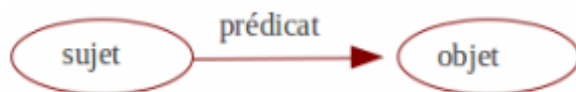


FIGURE 3 – Triplet RDF

Le sujet est la ressource sur laquelle s'applique la métadonnée (donnée sur une donnée), le prédicat représente le nom de la métadonnée et l'objet est la valeur de la métadonnée. La collection de ces triplets permettent de former des schémas RDF. RDFS est lui une extension de RDF. Langage de description de RDF, il permet de fournir des éléments de base afin de définir des ontologies et des vocabulaires RDF, en utilisant le système de classes et de propriétés.

```
teaching:Teacher
  rdf:type rdfs:Class ;
  rdfs:subClassOf foaf:Person .
```

FIGURE 4 – Définition d'une classe en RDFS

C'est à partir de ces informations (que nous avons vu dans les cours) que nous partions pour ce projet. Nous avons donc toutes les bases afin de réussir ce qui nous était demandé.

### 3 Analyse et implémentation

#### 3.1 Phase d'analyse et de recherche

Dans la première phase du projet, nous avons réalisé beaucoup de recherches, tout d'abord pour nous familiariser avec les technologies utilisées (dont nous ne connaissions que les bases) et comprendre réellement ce qui était attendu. En effet, le sujet étant court il était important de bien comprendre ce qui était réellement demandé, notamment sur le type de schéma et la précision des informations demandées. Pour bien comprendre nous avons beaucoup utilisé les exemples disponibles sur internet et la documentation du W3C sur le RDF et le RDFS. Avec ces recherches nous avons identifié quelles propriétés RDF ou RDFS nous devons utiliser. Pour cela, la documentation W3C nous a été extrêmement utile. Pour chaque attribut des classes, nous avons regardé dans la documentation quelle pourrait être la propriété RDF et RDFS la plus adaptée. Même chose pour les propriétés.

Propriétés RDFS	Description W3C	Classes	Propriétés
<code>rdfs:type</code>	used to state that a resource is an instance of a class.	<code>rdfs:Class</code>	<code>rdf:Property</code>
<code>rdfs:label</code>	used to provide a human-readable version of a resource's name	Nom de la classe	Nom de la propriété
<code>rdfs:comment</code>	used to provide a human-readable description of a resource	Description de la classe	Description de la propriété
<code>rdfs:subClassOf</code>	used to state that all the instances of one class are instances of another	Sous-classes de la classe	X
<code>rdfs:domain</code>	used to state that any resource that has a given property is an instance of one or more classes	X	Classe à laquelle se réfère la propriété
<code>rdfs:range</code>	used to state that the values of a property are instances of one or more classes	X	Règle de contenu de la propriété

Tableau présentant les propriétés RDFS que nous utilisons et leur correspondance dans le NoDEfr-2

Nous avons également durant cette phase de recherche rédigé un petit fichier RDFS afin d'avoir un exemple de sortie que nous pourrions sortir. Cela nous a permis de ne pas se lancer dans l'implémentation sans avoir une idée claire de ce que nous devons produire en sortie.

```

1  @prefix noDEfr2: <http://example.org/noDEfr2/>
2
3  //pour les classes
4  noDEfr2:identifiant
5      rdf:type rdfs:Class
6      rdfs:label Nom
7      rdfs:comment Description
8      rdfs:subClassOf Autre classe
9
10 //pour les propriétés
11 noDEfr2:identifiant
12     rdf:type rdf:Property
13     rdfs:label nom
14     rdfs:comment Description
15     rdfs:domain Classe
16     rdfs:range règle
17

```

FIGURE 5 – Fichier RDFS exemple

Nous nous sommes également familiarisé avec la syntaxe turtle car elle était imposée dans le sujet. Notre sortie devait donc être un fichier .ttl .

Nous avons également dû faire un choix concernant le langage de transformation XML que nous allions utiliser. En effet, notre sujet était de générer le schema RDFS à partir du fichier XML produit par l'équipe 2, il fallait donc trouver une technologie pour transformer le XML. Nous avons donc choisi d'utiliser le XSLT qui était la solution évidente au problème. Après des recherches sur d'autres langages de transformation XML, nous en avons conclu que XSLT était très utilisé, que nous l'avions déjà vu en cours et que donc était la solution parfaite pour notre problématique.

### 3.2 Implémentation

Après nos recherches et le choix de nos technologies fait, nous avons dû implémenter notre solution. Nous avons au départ identifié trois templates principaux (au sens XSLT) :

- un template à la racine pour mettre les préfixes au début de notre schéma
- un template pour les classes
- un template pour les relations

Nous avons identifié 3 préfixes différents pour notre schéma. Ces préfixes devaient être placés au début du schéma donc le chemin XPATH pour ce template était simple puisque nous devions juste les mettre à la racine. Les deux premiers préfixes (rdf et rdfs) sont pour les propriétés rdf et rdfs, le troisième (nodefr2) est un préfixe pour notre modèle.

Afin de générer les classes de notre schéma, nous avons utilisé un template utilisant comme sélecteur "classe". Ainsi à chaque balise <classe> rencontrée dans le fichier XML nous allions pouvoir générer nos classes RDFS. Pour quelques propriétés (au sens RDF) nous avons utilisé d'autres templates

spécialement pour les classes, essentiellement pour des raisons de propretés du code, en utilisant les "call-template". Nous avons géré dans ces templates l'affichage des valeurs des propriétés (avec les balises "xsl:value-of") mais également la vérification de l'existence des balises dans le XML (avec des balises "xsl:if"). En effet lors de nos premières versions du fichier XSLT il existait des propriétés vides (car par exemple il n'y a pas toujours de sous-classes) ce qui n'était pas possible pour avoir un schéma RDFS valide. Nous avons dû également gérer le cas où il y a plusieurs sous-classes, en utilisant les expressions XPath "substring-before" et "substring-after".

```
<xsl:template match="classe">
  nodefr2:<xsl:value-of select="@identifiant"/>
  rdf:type rdfs:Class;
  rdfs:label "<xsl:value-of select="nom"/>";
  rdfs:comment "<xsl:call-template name="comment"/><xsl:text>
</xsl:text><xsl:call-template name="subClassOf"/>
  <xsl:apply-templates select="relation"/>
</xsl:template>

<xsl:template name="comment">
  <xsl:value-of select="definition"/><xsl:choose><xsl:when test="sousClasseDe != '' and sousClasseDe">";</xsl:when><xsl:otherwise>".</xsl:otherwise></xsl:choose>
</xsl:template>

<xsl:template name="subClassOf">
  <xsl:if test="sousClasseDe != '' and sousClasseDe">
    <xsl:variable name="sousClasses" select="sousClasseDe"/>
    <xsl:choose>
      <xsl:when test="contains($sousClasses, ', ')">rdfs:subClassOf nodefr2:<xsl:value-of select="substring-before($sousClasses, ', ')" />;
      rdfs:subClassOf nodefr2:<xsl:value-of select="substring-after($sousClasses, ', ')" />.</xsl:when>
      <xsl:otherwise>rdfs:subClassOf nodefr2:<xsl:value-of select="$sousClasses" />.</xsl:otherwise>
    </xsl:choose>
  </xsl:if>
</xsl:template>
```

FIGURE 6 – Templates pour les classes dans le XSLT

Nous avons utilisé les mêmes techniques pour les propriétés de notre schéma (ce qui correspond aux relations dans le fichier XML).

### 3.3 Difficultés rencontrées

Nous avons rencontré deux principales difficultés lors de l'implémentation. La première était les points virgules et les points qui se situent en fin des lignes. En effet, le point doit être mis seulement sur la dernière ligne, or la dernière ligne de la définition des classes et des propriétés ne sont pas toujours pour la même propriété : par exemple une classe ne se termine pas forcément par la définition de la sous-classe (car certaines classes n'ont pas de sous-classes) donc dans ce cas le point doit se situer à la fin de la ligne qui définit "rdfs :comment". Pour régler ce problème nous avons utilisé des balises "xsl:choose" qui permet de faire une structure conditionnelle. Dans le cas où l'élément "sousClasseDe" existait pour cette classe, alors on place un point-virgule sur la ligne de définition de la propriété "rdfs :comment", sinon on met un point. Nous avons eu le même problème avec l'élément "regleDeContenu" pour les propriétés que nous avons réglé de la même façon. Une autre solution consistait simplement à ne pas mettre les éléments non obligatoires à la fin des définitions des classes ou

```

<xsl:template match="relation">
  nodefr2:<xsl:value-of select="@identifiant"/>
  rdf:type rdf:Property;
  rdfs:label "<xsl:value-of select="non"/>";
  rdfs:comment "<xsl:value-of select="definition"/>";
  rdfs:domain nodefr2:<xsl:call-template name="domain"/><xsl:text>
</xsl:text><xsl:call-template name="range"/>
</xsl:template>

<xsl:template name="domain">
  <xsl:value-of select="../@identifiant"/><xsl:choose><xsl:when test="regleDeContenu != '' and regleDeContenu"></xsl:when><xsl:otherwise></xsl:otherwise></xsl:choose>
</xsl:template>

<xsl:template name="range">
  <xsl:if test="regleDeContenu != '' and regleDeContenu">rdfs:range "<xsl:value-of select="regleDeContenu"/>".</xsl:if>
</xsl:template>

```

FIGURE 7 – Templates pour les propriétés dans le XSLT

des propriétés, mais nous avons déjà implémenté l'autre solution quand nous nous en sommes rendu compte et nous avons préféré garder celle-ci afin d'être raccord avec le schéma de sortie que nous avons réalisé précédemment.

La deuxième difficulté rencontrée est sur les éléments "sousClasseDe". En effet la valeur de la propriété "rdfs:subClassOf" devait être "nodefr2:idclasse", or il peut y avoir plusieurs sous-classes et donc plusieurs "rdfs:subClassOf" dans la définition d'une classe. Le problème était que dans le XML les sous-classes étaient toutes dans un même élément <sousClasseDe> séparées par des virgules. Nous avons donc dû rechercher un moyen de faire un traitement de chaînes de caractères, afin de séparer les id des classes entre les virgules. Nous avons utilisé pour cela les fonction XPath "substring-before" et "substring-after", qui permettent d'avoir la sous-chaîne avant et après un certain caractère (dans notre cas la virgule). Les éléments "sousClasseDe" contenant au maximum 2 sous-classes, ces deux fonctions étaient suffisantes pour régler notre problème (avec l'utilisation de "xsl:choose" et des variables).

D'autres petites difficultés ont également émergé, comme par exemple pour la mise en page. Nous voulions une mise en page de fichier .ttl qui soit cohérente et bien indentée. Malheureusement ce n'est pas parfait, notamment en raison de certains "rdfs:comment" dans la définition des classes, où le point et les guillemets sont mis une ligne en dessous. Cela est causé par le fichier XML, lorsque la balise </description> n'est pas directement collée à la fin de la phrase mais mis une ligne en dessous.



### 3.4 Collaboration avec les autres groupes

La collaboration avec les autres groupe étaient très importantes lors de notre travail. En effet, la sortie du travail d'un groupe est l'entrée du travail du groupe suivant.

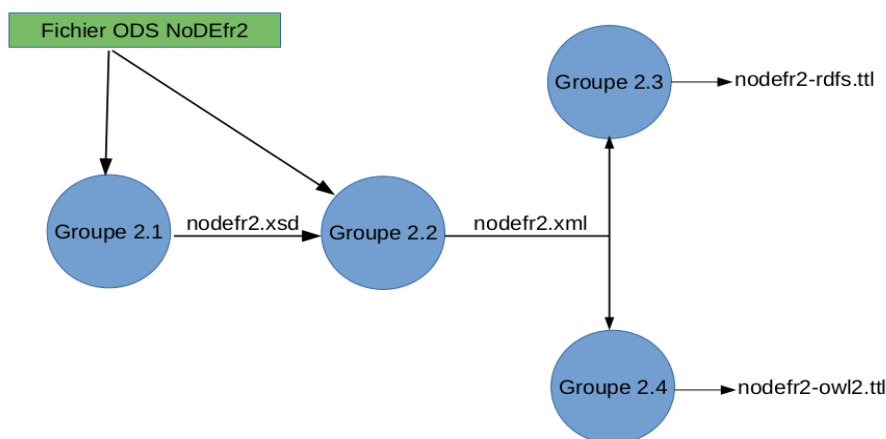


FIGURE 8 – Schéma modélisant les entrées/sorties des groupes

Nous devons donc nous organiser afin que les derniers groupes ne commencent pas leur projet une fois le travail des groupes précédents terminé. Pour cela, le premier groupe a créé un serveur Discord afin que nous puissions communiquer. Nous avons organisé régulièrement des petites réunions avec tous les groupes ou seulement certains, pour mettre au clair certains points et se coordonner.

Dans notre cas, nous avons surtout besoin de la sortie du groupe qui produisait le fichier XML. Il a donc fallu bien se mettre d'accord sur la sortie qu'ils allaient produire afin que nous puissions avancer de notre côté, en créant un fichier XML exemple reprenant ce qu'ils allaient produire mais seulement avec une partie des données (cf annexe 1).

Nous avons également beaucoup parlé avec le groupe produisant le schéma OWL, vu qu'ils travaillaient au même niveau que nous. Nous avons pu discuter de ce que nous devions produire et des différences entre nos deux schémas.

## 4 Conclusion

Ce projet aura été formateur, tant sur le plan technique que relationnel. Nous avons pu mettre en pratique sur un vrai cas concret les connaissances que nous avons apprises sur le web sémantique et le web des données. Les technologies utilisées sont très intéressantes et ce projet aura justement permis de mieux comprendre dans quel cadre elles sont utilisées. Le fonctionnement en sujet/sous-sujet nous aura permis de plus s'organiser en amont et nous pousse à ne pas commencer le projet trop tard, même si c'est parfois un peu difficile pour les groupes étant au bout de la chaîne (qui sont obligés de travailler sur des "faux" exemples amenés à être modifié).

Concernant notre travail nous sommes plutôt satisfait de ce que nous avons réalisé. Nous pensons que le schéma RDFS généré avec le fichier XML est correcte et représente correctement les informations du NoDEfr-2.

## 5 Annexes

### 5.1 Annexe 1 : première version du fichier XML exemple

```
<?xml version="1.0" encoding="UTF-8">
<Modele
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file:///home/avigner/TP/S8/DOCWEB/Projet/DocWeb/Projet_3/tests/NoDEfe-2.xsd">
  <classe>
    <identifiant>RC0000</identifiant>
    <nom>Entité du monde de la formation</nom>
    <definition>classe abstraite qui regroupe les propriétés communes à toutes les situations ou à l'ensemble
    de situations visant un ou plusieurs objectifs de développement personnel ou professionnel</definition>
    <sousClasseDe></sousClasseDe>
    <note>Super classe de toutes les classes. Elle regroupe toutes les propriétés des classes présentes
    dans ce modèle</note>
    <relation>
      <identifiant>DES000001</identifiant>
      <nom>libellé</nom>
      <definition>nommage humainement compréhensible de l'élément</definition>
      <indicateurLinguistique>oui</indicateurLinguistique>
      <coDomaine>Littéral</coDomaine>
      <regleDeContenu>MLR String</regleDeContenu>\subsection{Choix et implémentations}
      <raffine></raffine>
      <exemple>Pour une instance de la classe RC0020 Offre de formation, le libellé pourrait
      être " BTS services informatiques aux organisations"</exemple>
      <note></note>
      <cardinaliteMinimale>1</cardinaliteMinimale>
      <cardinaliteMaximale>1</cardinaliteMaximale>
      <raison></raison>
    </relation>
    <relation>
      <identifiant>DES000002</identifiant>
      <nom>date de mise à jour</nom>
      <definition>dernière date de modification d'une information d'un élément</definition>
      <indicateurLinguistique>non</indicateurLinguistique>
      <coDomaine>Littéral</coDomaine>
      <regleDeContenu>MLR Date</regleDeContenu>
      <raffine></raffine>
      <exemple>Pour une instance de la Classe RC0105 Activité pédagogique, le changement de
      modalité pédagogique sera daté</exemple>
      <note></note>
      <cardinaliteMinimale>0</cardinaliteMinimale>
      <cardinaliteMaximale>1</cardinaliteMaximale>
      <raison></raison>
    </relation>
  </classe>
```

```

<classe>
  <identifiant>RC0015</identifiant>
  <nom>Certificat, badge, diplome</nom>
  <definition>un élément de cette classe est la reconnaissance des compétences, des connaissances,
  des savoir-faire d'une personne</definition>
  <sousClasseDe>RC0000</sousClasseDe>
  <note>Exemple : reconnaissance des compétences, des connaissances, des savoir-faire d'une
  personne</note>
  <relation>
    <identifiant>DES015001</identifiant>
    <nom>description</nom>
    <definition>Description du diplôme, certificat ou badge</definition>
    <indicateurLinguistique>oui</indicateurLinguistique>
    <coDomaine>Littéral</coDomaine>
    <regleDeContenu>MLR String</regleDeContenu>
    <raffine></raffine>
    <exemple>L'ONISEP décrit le BTS SIO de la façon suivante : "Couvrant tous les secteurs professionnels,
    les BTS forment des collaborateurs d'ingénieur, des assistants commerciaux ou techniques, des
    assistants clientèle, des chefs de chantier, etc. selon les secteurs. Le BTS se prépare en 2 ans
    après le bac en lycée, en école ou par l'apprentissage. Le recrutement se fait sur dossier. Une mise
    à niveau de 1 an est obligatoire pour l'accès en BTS management en hôtellerie-restauration pour les
    élèves n'étant pas titulaires du bac sciences et technologies de l'hôtellerie et de la restauration
    (STHR). Les BTS visent une insertion professionnelle directe. Poursuite d'études possible,
    notamment en licence pro (pour 1 an)."</exemple>
    <note></note>
    <cardinaliteMinimale>0</cardinaliteMinimale>
    <cardinaliteMaximale>1</cardinaliteMaximale>
    <raison></raison>
  </relation>

  <relation>
    <identifiant>DES015017001</identifiant>
    <nom>a pour niveau</nom>
    <definition>Niveau du diplôme d'un système de codification donné</definition>
    <indicateurLinguistique>non</indicateurLinguistique>
    <coDomaine>RC0017</coDomaine>
    <regleDeContenu></regleDeContenu>
    <raffine></raffine>
    <exemple></exemple>
    <note></note>
    <cardinaliteMinimale>0</cardinaliteMinimale>
    <cardinaliteMaximale>n</cardinaliteMaximale>
    <raison></raison>
  </relation>
</classe>

</Modele>

```

## 5.2 Annexe 2 : fichier XSLT

```
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:nodefr2="http://example.org/nodefr2/">
  <xsl:output method="text" indent="yes"/>

  <xsl:template match="/">
    @prefix rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
    @prefix rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#>.
    @prefix nodefr2: &lt;http://example.org/nodefr2/>.
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="classe">
    nodefr2:<xsl:value-of select="@identifiant"/>
      rdf:type rdfs:Class;
      rdfs:label "<xsl:value-of select="nom"/>";
      rdfs:comment "<xsl:call-template name="comment"/><xsl:text>
    </xsl:text><xsl:call-template name="subClassOf"/>
      <xsl:apply-templates select="relation"/>
  </xsl:template>

  <xsl:template name="comment">
    <xsl:value-of select="definition"/><xsl:choose><xsl:when test="sousClasseDe != '' and sousClasseDe">";</xsl:when><xsl:otherwise>".</xsl:otherwise></xsl:choose>
  </xsl:template>

  <xsl:template name="subClassOf">
    <xsl:if test="sousClasseDe != '' and sousClasseDe">
      <xsl:variable name="sousClasses" select="sousClasseDe"/>
      <xsl:choose>
        <xsl:when test="contains($sousClasses, ', ')">rdfs:subClassOf nodefr2:<xsl:value-of select="substring-before($sousClasses, ', ')" />;
          rdfs:subClassOf nodefr2:<xsl:value-of select="substring-after($sousClasses, ', ')" /></xsl:when>
        <xsl:otherwise>rdfs:subClassOf nodefr2:<xsl:value-of select="$sousClasses"/></xsl:otherwise>
      </xsl:choose>
    </xsl:if>
  </xsl:template>

  <xsl:template match="relation">
    nodefr2:<xsl:value-of select="@identifiant"/>
      rdf:type rdf:Property;
      rdfs:label "<xsl:value-of select="nom"/>";
      rdfs:comment "<xsl:value-of select="definition"/>";
      rdfs:domain nodefr2:<xsl:call-template name="domain"/><xsl:text>
    </xsl:text><xsl:call-template name="range"/>
  </xsl:template>

  <xsl:template name="domain">
    <xsl:value-of select="../@identifiant"/><xsl:choose><xsl:when test="regleDeContenu != '' and regleDeContenu">";</xsl:when><xsl:otherwise>".</xsl:otherwise></xsl:choose>
  </xsl:template>

  <xsl:template name="range">
    <xsl:if test="regleDeContenu != '' and regleDeContenu">rdfs:range "<xsl:value-of select="regleDeContenu"/>".</xsl:if>
  </xsl:template>

</xsl:stylesheet>
```

## 6 Webographie

Yolaine Bourda, Gilles Gauthier, Rosa-Maria Gomez de Regil, Olivier Cateau. **Métadonnées pour ressources d'apprentissage (MLR) : nouvelle norme ISO de description de ressources pédagogiques**. STICEF (Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation), ATIEF, 2010, 17, 11 p. fhal-00696450f

[*www.w3.org*] "**RDF Schema 1.1**", <https://www.w3.org/TR/rdf-schema/>

[*www.w3.org*] "**RDF Primer**", <https://www.w3.org/TR/rdf-primer/>

[*moodle.insa-rouen.fr*] Nicolas Delestre, Nicolas Malandain "**XSLT**", [https://moodle.insa-rouen.fr/pluginfile.php/133451/mod\\_resource/content/1/XSLT.pdf](https://moodle.insa-rouen.fr/pluginfile.php/133451/mod_resource/content/1/XSLT.pdf)

[*moodle.insa-rouen.fr*] Nicolas Delestre, Nicolas Malandain "**RDF-SPARQL**", [https://moodle.insa-rouen.fr/pluginfile.php/133458/mod\\_resource/content/1/RDF-SPARQL.pdf](https://moodle.insa-rouen.fr/pluginfile.php/133458/mod_resource/content/1/RDF-SPARQL.pdf)

[*moodle.insa-rouen.fr*] Nicolas Delestre, Nicolas Malandain "**RDFS-OWL**", [https://moodle.insa-rouen.fr/pluginfile.php/133459/mod\\_resource/content/2/RDFS-OWL.pdf](https://moodle.insa-rouen.fr/pluginfile.php/133459/mod_resource/content/2/RDFS-OWL.pdf)

[*blog.onyme.com*, 2011] Marina Soler, "**Les ontologies informatiques : l'exemple par OWL et autres**", <http://blog.onyme.com/les-ontologies-informatiques-lexemple-par-owl-et-autres/>

[*www.w3.org*] "**RDF Schema 1.1**", <https://www.w3.org/TR/rdf-schema/>

[*lms.fun-mooc.fr*] Inria, "**le modèle de données RDF**", <https://lms.fun-mooc.fr/c4x/inria/41002S02/asset/C013FG-W2.pdf>

[*edutechwiki.unige.ch*], "**Tutoriel XSLT débutant**", [http://edutechwiki.unige.ch/fr/Tutoriel\\_XSLT](http://edutechwiki.unige.ch/fr/Tutoriel_XSLT)

[*perso.univ-mlv.fr*] O.Curé, "**RDF**", <http://perso.univ-mlv.fr/ocure/RDF.pdf>

[*www.fil.univ-lille1.fr*] Anne-Cécile Caron, "**Vocabulaire RDF : RDFS**", <https://www.fil.univ-lille1.fr/caronc/WS/rdfsPar4.pdf>

[*www.info.univ-angers.fr*] "**RDF-Schema**", [http://www.info.univ-angers.fr/pub/genest/fichiers/m1\\_websem/ws\\_chap4.pdf](http://www.info.univ-angers.fr/pub/genest/fichiers/m1_websem/ws_chap4.pdf)