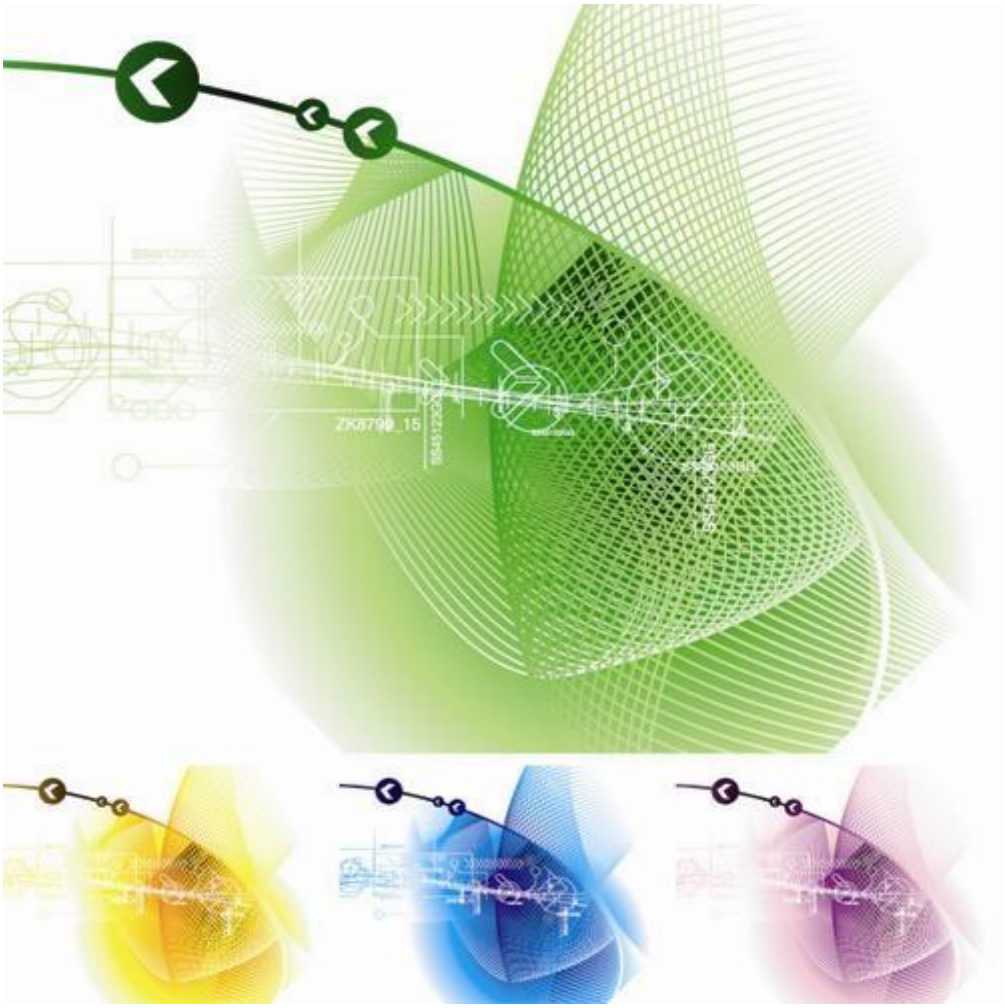
JOUINI Karim
KLEIN Eléonore
MIETLICKI Pascal
OTHMANI Jihed

MDSI - RAPPORT FINAL



Groupe 2

Conception d'une base de données semi-structurée en XML

MDSI - Rapport final

CONCEPTION D'UNE BASE DE DONNEES SEMI-STRUCTUREE EN XML - GROUPE 2

1. INTRODUCTION

Le sujet de l'APP MDSI consistait en la conception et la mise en œuvre d'une base de données permettant la gestion des heures de cours au DGEI, en collaboration avec une application Java (BE programmation orientée objet).

Cette base de donnée semi-structurée devait être réalisée en XML, nous permettant une approche structurelle de ce métalangage, ainsi que des applications variées (transformation de requêtes à l'aide de feuilles de style, génération d'une documentation au format DocBook, traitement des données via un CMS…)

A travers la réalisation de ce projet, nous avons également pu approfondir les concepts UML appris cette année et les appliquer au sein d'un travail concret.

Voici donc le détail de notre démarche, ainsi qu'un résumé des connaissances que nous avons pu acquérir durant cette APP.

2. MODELISATION

a. Cahier des charges

La première partie du projet a été consacrée à l'étude minutieuse du cahier des charges. Celui-ci a été mis à plat afin de détecter d'éventuelles incohérences, imprécisions ou manques.

Lors de cette étape, certains points importants sont ressortis :

- Le système doit-il gérer les salles où se déroulent les cours ? Est-ce qu'on gère les ressources humaines ?
- L'intervenant peut –il être optionnel ? Peut-il y avoir un cours sans intervenant ?
- Peut-il avoir plusieurs intervenants ?

Nous avons donc pris rendez-vous avec le client afin d'éclaircir ces points.
A la suite de ce rendez-vous, il est apparu que le système ne gérait pas ce qui était du domaine des ressources humaines, ni la répartition des salles (ces points étant gérés par des applications indépendantes). Un cours peut avoir zéro, un ou plusieurs intervenants, sachant que seules les heures « présentielles » seront payées.

Au vu de ces précisions, nous avons ensuite commencé la modélisation, en utilisant la méthodologie UML.

b. Modèle UML initial

Pourquoi UML ?

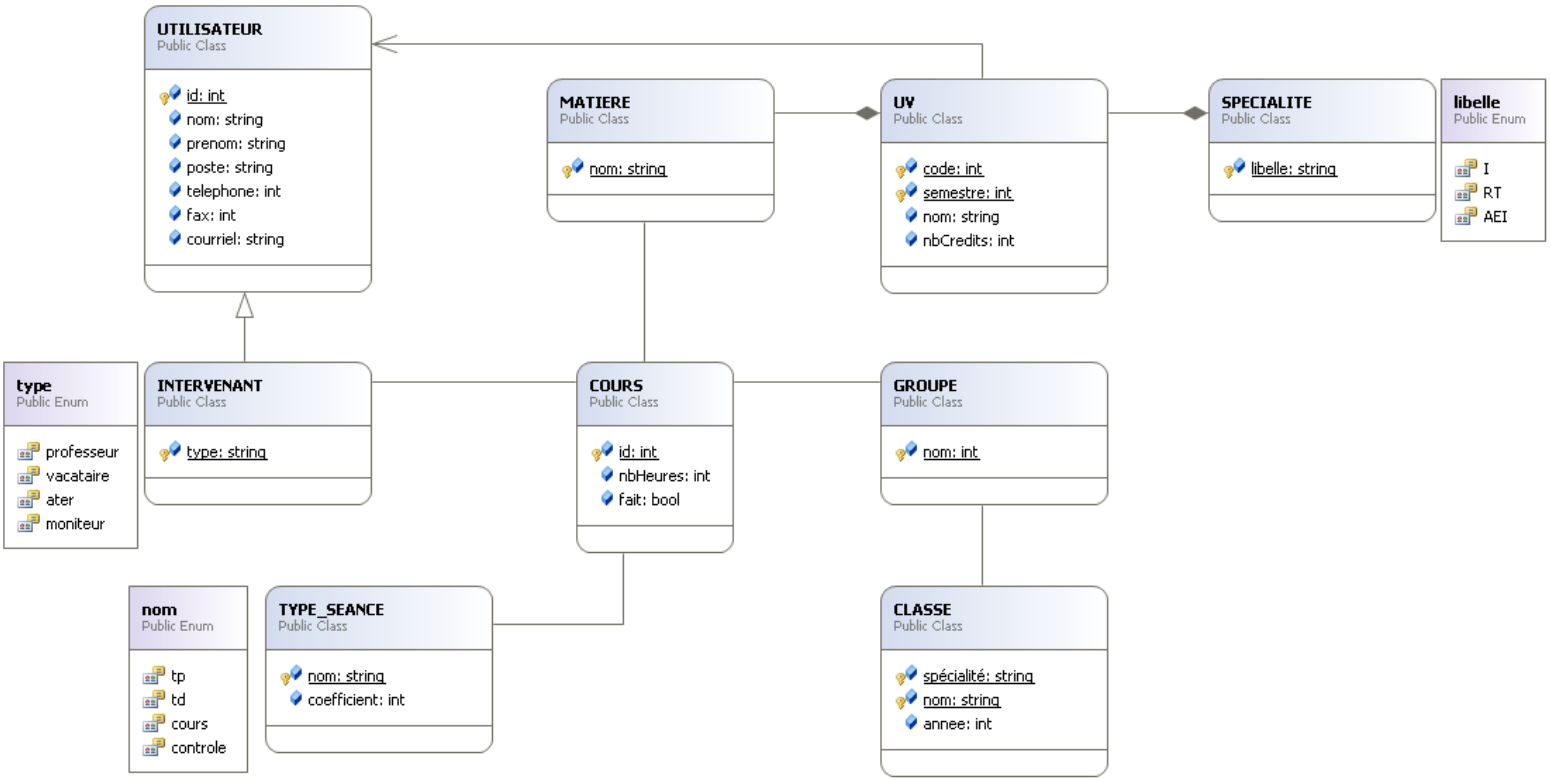
Ayant utilisé le modèle Entités/Associations à plusieurs reprises les années précédentes, nous avons souhaité aborder la modélisation UML, en parallèle avec nos cours de Conception Orientée Objet, et en anticipant sur le Bureau d'Etude COO/POO.

Etablissement du modèle initial

Première étape : Répertorier l'ensemble des classes/entités intervenant dans le système

- Les principaux utilisateurs du système
  - Responsable d'UV
  - Secrétaire
  - Directeur du département
  - Directeur des études
  - Intervenants (enseignants, vacataires…)
  - …
- Les spécialités
- Les années
- Les classes
- Les UVs
- Les matières
- …

Deuxième étape : Les associer et les regrouper par héritage / association / aggrégation



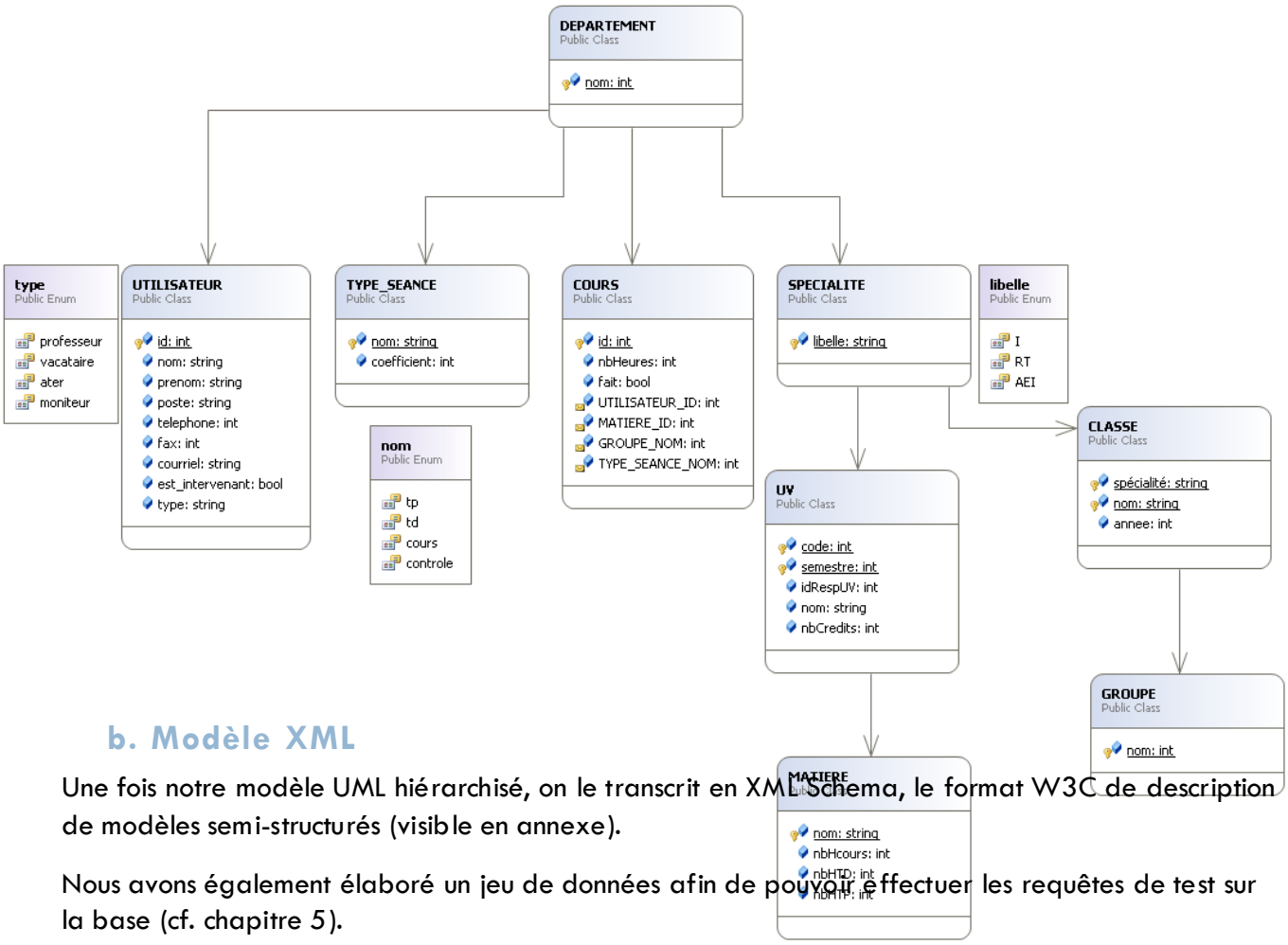
3. PASSAGE AU MODELE XML

a. Modèle UML hiérarchisé

Le XML étant semi-structuré, le modèle UML lui correspondant doit respecter un certain nombre de contraintes :

- Présence d'une racine : une classe sans parent et sans cardinalités n,1
- Absence de cycles
- Suppression des héritages multiples, qui deviennent des associations auxquelles on ajoute des contraintes d'intégrité de type XOR

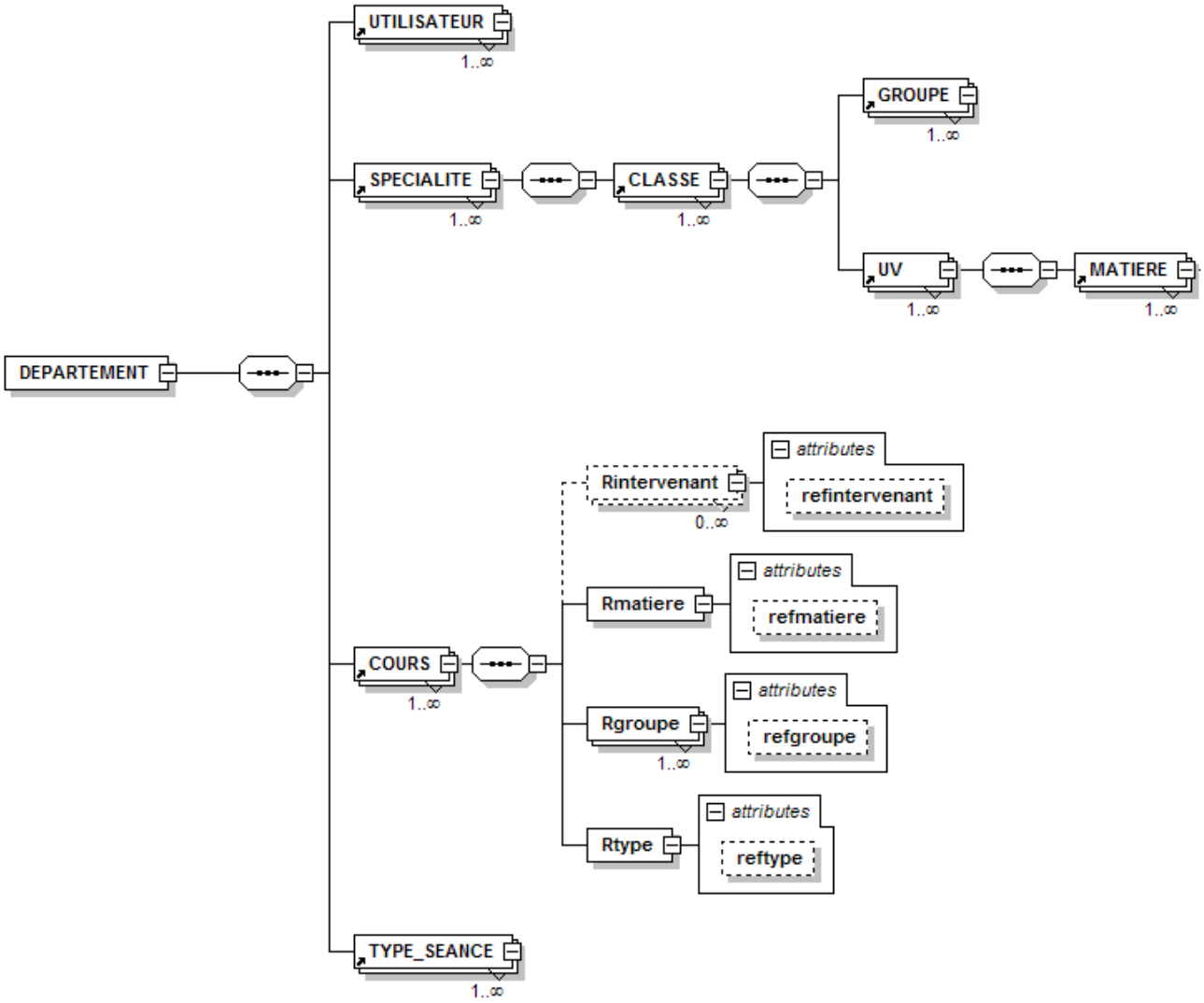
On obtient le modèle UML semi-structuré suivant :



b. Modèle XML

Une fois notre modèle UML hiérarchisé, on le transcrit en XML Schema, le format W3C de description de modèles semi-structurés (visible en annexe).

Nous avons également élaboré un jeu de données afin de pouvoir effectuer les requêtes de test sur la base (cf. chapitre 5).



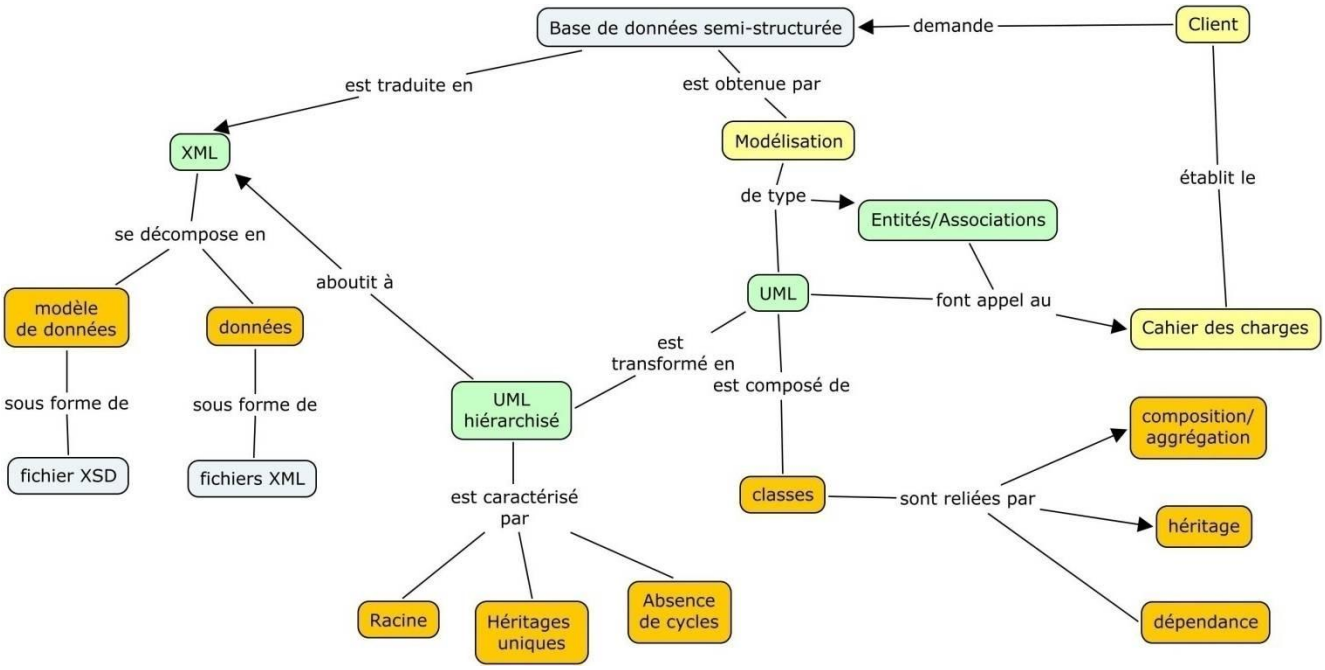
La racine de notre schéma est le département dans lequel plusieurs utilisateurs travaillent, des cours de différents types sont donnés, et il est composé par plusieurs spécialités elles mêmes constituées par des classes divisées en groupes.

Pour chaque classe, un ensemble d'UVs - qui peuvent être composées d'une ou plusieurs matières- est enseignée.

Un cours est identifié par son attribut « id » et fait référence à un intervenant, une matière, un groupe et un type. Ces références qui pointent sur les clés primaires des autres nœuds (clés étrangères) sont utilisé afin de réduire la redondance dans le jeu de données.

4. PREMIER BILAN

Carte des concepts



5. LES REQUETES

a. XQuery et XPath

XQuery est le langage de requête permettant d'extraire des informations d'un fichier XML. Il utilise la syntaxe XPath pour adresser des parties spécifiques d'un document XML. (Wikipedia)

Après avoir installé eXist et chargé notre base de données ainsi que notre jeu de données, nous avons donc élaboré les requêtes XQuery demandées.

1/Donner le tableau des U.V. de 4eme année I de l'année universitaire en cours, ordonnée suivant le code des UV.

Code :

```
for $uv in /DEPARTEMENT//CLASSE[@annee = "4" and @specialite="I"]//UV
order by $uv/@code ascending
return $uv
```

Résultat :

```
1 <UV semestre ="1" code ="1" nom ="PROGRAMMATION" idRespUV ="1">
    <MATIERE id ="1" nom ="JAVA" nbHcours ="20" nbHTD ="15" nbHTP ="15"/>
    <MATIERE id ="2" nom ="C" nbHcours ="20" nbHTD ="15" nbHTP ="15"/>
    <MATIERE id ="3" nom ="CAML" nbHcours ="20" nbHTD ="15" nbHTP ="15"/>
  </UV>
2 <UV semestre ="1" code ="2" nom ="APP" idRespUV ="2">
    <MATIERE id ="4" nom ="SDI" nbHTP ="60"/>
  </UV>
```

2/Quelles sont les UV dont Mr X, Y est responsable (code et intitulée)?

Principe : D'abord on sélectionne toutes les Uvs. Puis on affiche celles dont le responsable est Mr X Y.

Code :

```
for $uv in //UV
for $user in //UTILISATEUR[@nom="MARRE" and @prenom="Daniel" and @id= $uv/@idRespUV]
return $uv
```

Résultat :

Exemple, les Uvs dont Mr Daniel Marre est responsable :

```
1 <UV semestre ="1" code ="2" nom ="APP" idRespUV ="2">
    <MATIERE id ="4" nom ="SDI" nbHTP ="60"/>
  </UV>
```

3/Donner la liste des enseignants qui interviennent dans les UV d'une classe donnée (année/spécialité) sans, puis avec le nombre total d'heures de chacun.

Principe : On sélectionne la classe voulue (ici 5eme année I) puis les cours de cette classe et enfin on récupère tous les intervenants pour les cours pour cette même classe. Finalement, on affiche toutes les valeurs distinctes.

Code :

```
for $interv in distinct-values(
        for $classe in //CLASSE[@annee = "5" and @specialite = "I"]
        for $cours in//COURS[Rgroupe/@refgroupe = $classe/GROUPE/@nom]
        for $user in //UTILISATEUR[@id = $cours/Rintervenant/@refintervenant]
        return $user/@nom
)
return $interv
```

Résultat :

```
1 MIETLICKI
2 DILHAC
```

Principe : le même que le précédent avec en plus une sous fonction qui va faire la somme pour chaque intervenant du nombre d'heures de cours données. Puis on l'affiche de manière structurée avec le nom, prénom de chaque intervenant ainsi que le nombre d'heures total.

Code :

```
for $classe in //CLASSE[@annee = "4" and @specialite = "I"]
for $interv in distinct-values(
        for $cours in//COURS[Rgroupe/@refgroupe = $classe/GROUPE/@nom]
        for $user in //UTILISATEUR[@id = $cours/Rintervenant/@refintervenant]
        return $user/@id
```

```
)
let $intervenant := //UTILISATEUR[@id=$interv]
let $nbht:=sum(
        for $cours in//COURS[Rgroupe/@refgroupe = $classe/GROUPE/@nom and
Rintervenant/@refintervenant=$interv]
        return $cours/@nbHeures
)
return
<INTERVENANT>
        <nom>{$intervenant/@nom}</nom>
        <prenom>{$intervenant/@prenom}</prenom>
        <Nbr_heures>{$nbht}</Nbr_heures>
</INTERVENANT>
```

Résultat:

```
1 <INTERVENANT>
    <nom nom ="DILHAC"/>
    <prenom prenom ="Jean-Marie"/>
    <Nbr_heures>6</Nbr_heures>
  </INTERVENANT>
2 <INTERVENANT>
    <nom nom ="MARRE"/>
    <prenom prenom ="Daniel"/>
    <Nbr_heures>2</Nbr_heures>
  </INTERVENANT>
```

4/Donner le tableau des programmes d'enseignements d'une classe avec pour chaque UV: le code apogée, le nom de l'UV, celui du responsable et les évaluations prévues.

Principe : On sélectionne la classe 4I puis on récupère toutes les Uvs de cette classe ainsi que le responsable de chaque UV. Après, on exécute une sous-requête qui va chercher si des évaluations sont prévues. Puis afficher le nombre total d'évaluations prévues, la liste des matières concernées et les dates des évaluations.

Code :

```
for $classe in //CLASSE[@annee = "4" and @specialite = "I"]
for $uv in //$classe/UV
for $resp in //UTILISATEUR[@id = $uv/@idRespUV]
return
<RESULTAT>
        <CLASSE>
                <annee>{$classe/@annee}</annee>
                <specialite>{$classe/@specialite}</specialite>
```

```
        </CLASSE>
        <UV>
            <code_apogee>{$uv/@code}</code_apogee>
            <nom>{$uv/@nom}</nom>
        </UV>
        <RESPONSABLE>
            <nom>{$resp/@nom}</nom>
        </RESPONSABLE>
        {
            for $controle in //COURS[Rtype/@reftype="Controle" and
Rgroupe/@refgroupe=$classe/GROUPE/@nom and
            Rmatiere/@refmatiere=$uv/MATIERE/@id]
            for $mat in //MATIERE[@id = $controle/Rmatiere/@refmatiere]
            return
            <EVALUATIONS>
                <nb_evaluation>{count($controle)}</nb_evaluation>
                <date>{$controle/@date}</date>
                <matiere>{$mat/@nom}</matiere>
            </EVALUATIONS>
        }
    </RESULTAT>
```

Résultat :

```
1 <RESULTAT>
    <CLASSE>
      <annee annee ="4"/>
      <specialite specialite ="I"/>
    </CLASSE>
    <UV>
      <code_apogee code ="1"/>
      <nom nom ="PROGRAMMATION"/>
    </UV>
    <RESPONSABLE>
      <nom nom ="DILHAC"/>
    </RESPONSABLE>
  </RESULTAT>
2 <RESULTAT>
    <CLASSE>
      <annee annee ="4"/>
      <specialite specialite ="I"/>
    </CLASSE>
    <UV>
      <code_apogee code ="2"/>
      <nom nom ="APP"/>
    </UV>
    <RESPONSABLE>
      <nom nom ="MARRE"/>
    </RESPONSABLE>
    <EVALUATIONS>
      <nombre_evaluation>1</nombre_evaluation>
      <date date ="2007-12-31"/>
      <matiere nom ="SDI"/>
    </EVALUATIONS>
  </RESULTAT>
```

5/Donner la liste des UV dans lesquelles Mr X, Y intervient en précisant pour chacune la classe.

Principe : On récupère les cours effectués par Mr X,Y. A partir de ces cours, on détermine le groupe, la classe, la matière ainsi que l'UV. Ensuite, nous récupérons les valeurs distinctes des codes des Uvs. A partir de ces code, on détermine les classes associés aux UVs qu'on affiche de manière structurée.

Code :

```
for $uv2 in distinct-values(
        for $user in //UTILISATEUR[@nom="DILHAC" and @prenom="Jean-Marie"]
        for $cours in//COURS[Rintervenant/@refintervenant= $user/@id]
        for $groupe in //GROUPE[@nom=$cours/Rgroupe/@refgroupe]
        for $classe in //CLASSE[GROUPE/@nom = $groupe/@nom]
        for $mat in //MATIERE[@id = $cours/Rmatiere/@refmatiere]
        for $uv in //UV[MATIERE/@id = $mat/@id]
        return $uv/@code
)
for $classe in //CLASSE[UV/@code = $uv2]
for $uv in //UV[@code = $uv2]
return
<RESULTAT>
```

```
        <UV>
            <nom>{$uv/@nom}</nom>
        </UV>
        <CLASSE>
            <nom>{$classe/@nom}</nom>
        </CLASSE>
    </RESULTAT>
```

Résultat :

```
1 <RESULTAT>
    <UV>
      <nom nom ="PROGRAMMATION"/>
    </UV>
    <CLASSE>
      <nom nom ="4 Informatique "/>
    </CLASSE>
  </RESULTAT>
2 <RESULTAT>
    <UV>
      <nom nom ="APP"/>
    </UV>
    <CLASSE>
      <nom nom ="4 Informatique "/>
    </CLASSE>
  </RESULTAT>
3 <RESULTAT>
    <UV>
      <nom nom ="TEMPS REEL"/>
    </UV>
    <CLASSE>
      <nom nom ="5 Informatique "/>
    </CLASSE>
  </RESULTAT>
```

6/Donner pour chaque classe, le taux d'encadrement par des enseignants permanents.

Principe : on divise le nombre d'heures de cours donnés par des intervenants permanents par le nombre total d'heures de cours d'une classe donnée. On affiche ensuite le résultat de manière structurée en commençant par le nombre total d'heures données par des intervenants permanents, puis le nombre total d'heures pour chaque classe et enfin le taux d'encadrements calculé en fonction des deux valeurs précédentes sous forme de pourcentage.

Code :

```
for $classe in //CLASSE
let $nbh:= sum(
        for $cours in //COURS[Rmatiere/@refmatiere=$classe/UV/MATIERE/@id and R groupe/@refgroupe=
$classe/GROUPE/@nom]
        for $user in //UTILISATEUR[@id = $cours/Rintervenant/@refintervenant]
        return
```

```
            $cours/@nbHeures
        )
        let $nbhp:= sum(
            for $cours in //COURS[Rmatiere/@refmatiere=$classe/UV/MATIERE/@id and Rgroupe/@refgroupe=
        $classe/GROUPE/@nom]
            for $user in //UTILISATEUR[@id = $cours/Rintervenant/@refintervenant]
            return
            if ($user/@type="PERMANENT") then $cours/@nbHeures
            else 0
        )
        return
        <RESULTAT>
            <CLASSE>
                <nom>{$classe/@nom}</nom>
            </CLASSE>
            <total>{$nbh}</total>
            <total_p>{$nbhp}</total_p>
            <taux>{($nbhp div $nbh)*100} %</taux>
        </RESULTAT>
```

Résultat :

```
1 <RESULTAT>
    <CLASSE>
      <nom nom ="4 Informatique "/>
    </CLASSE>
    <total>8</total>
    <total_p>8</total_p>
    <taux>100 %</taux>
  </RESULTAT>
2 <RESULTAT>
    <CLASSE>
      <nom nom ="5 Informatique "/>
    </CLASSE>
    <total>7</total>
    <total_p>2</total_p>
    <taux>28.57142857142857 %</taux>
  </RESULTAT>
```

b. Transformation de requêtes

XSLT

XSL est un langage permettant d'effectuer requêtes et mise en forme à partir d'un fichier de données XML.

La plupart des navigateurs intègrent un processeur XSL qui va permettre d´effectuer la transformation. Il suffit d´indiquer dans le fichier XML, la feuille de style XSL à appliquer et, ensuite, de l´ouvrir dans un navigateur.

Un moyen simple de voir le résultat d'une transformation est de lier la transformation au document XML à l'aide d'une déclaration de feuille de style comme celle ci :

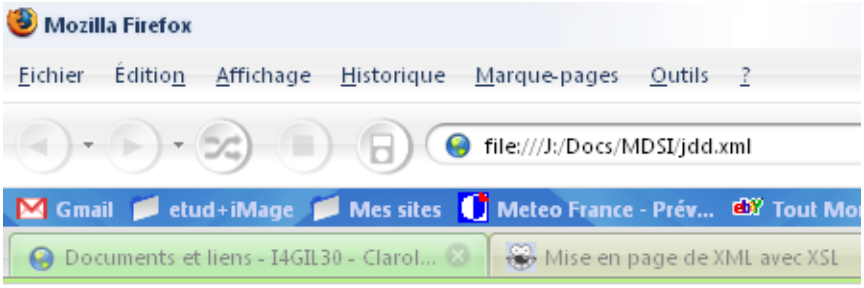
```
<?xml-stylesheet type="text/xsl" href="transform.xsl"?>
```

Ensuite, il suffit d'ouvrir le fichier XML à l'aide d'un navigateur Web comme Firefox ou Internet Explorer pour afficher le résultat.

Si la transformation fournit un fichier XML, le navigateur affiche le nouveau document XML et on peut vérifier que la transformation est correcte.

Nous avons travaillé sur deux transformations différentes de notre fichier XML :

Fiche ECTS d'une UV



FICHE ECTS PROGRAMMATION

Code : 1

RESPONSABLE UV :

Nom : Jean-Marie DILHAC

LISTE DE MATIERES :

JAVA

Intervenants :
Jean-Marie DILHAC

Cours	20
TD	15
TP	15

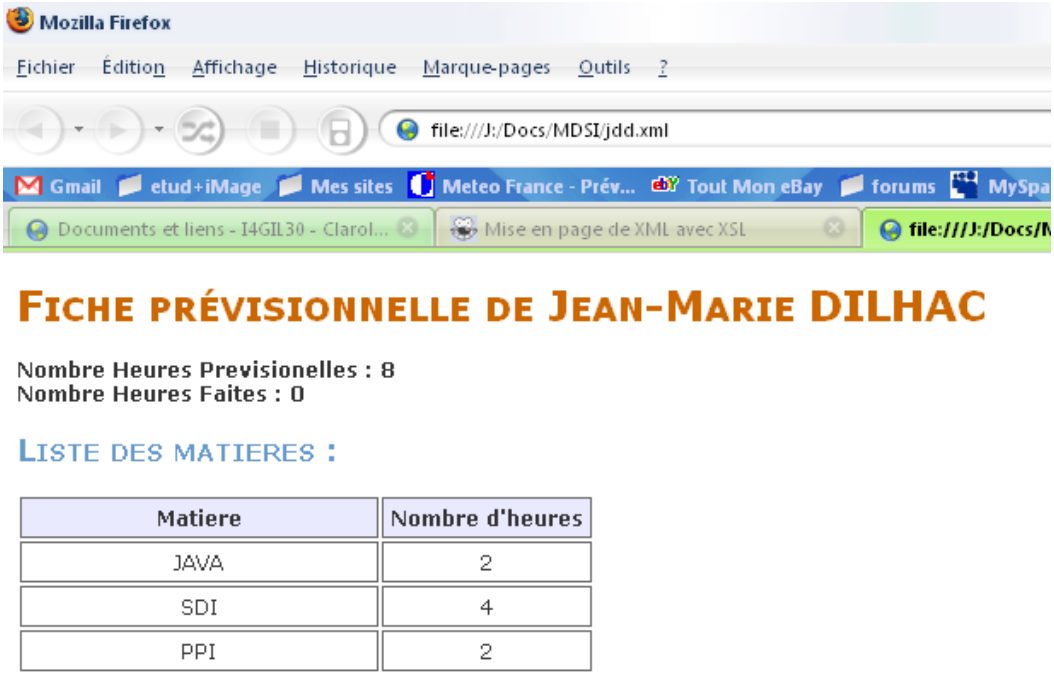
C

Cours	20
TD	15
TP	15

CAML

Cours	20
TD	15
TP	15

Fiche prévisionnelle d'un enseignant



DocBook

DocBook est une autre DTD (Définition de Types de Documents) qui permet de mettre en forme un fichier XML de manière à ce qu'il puisse par la suite être généré dans différents formats (HTML, pdf, rtf, javadoc…)

Nous avons tout d'abord utilisé une transformation XSL pour mettre les données au format DocBook (extrait) :

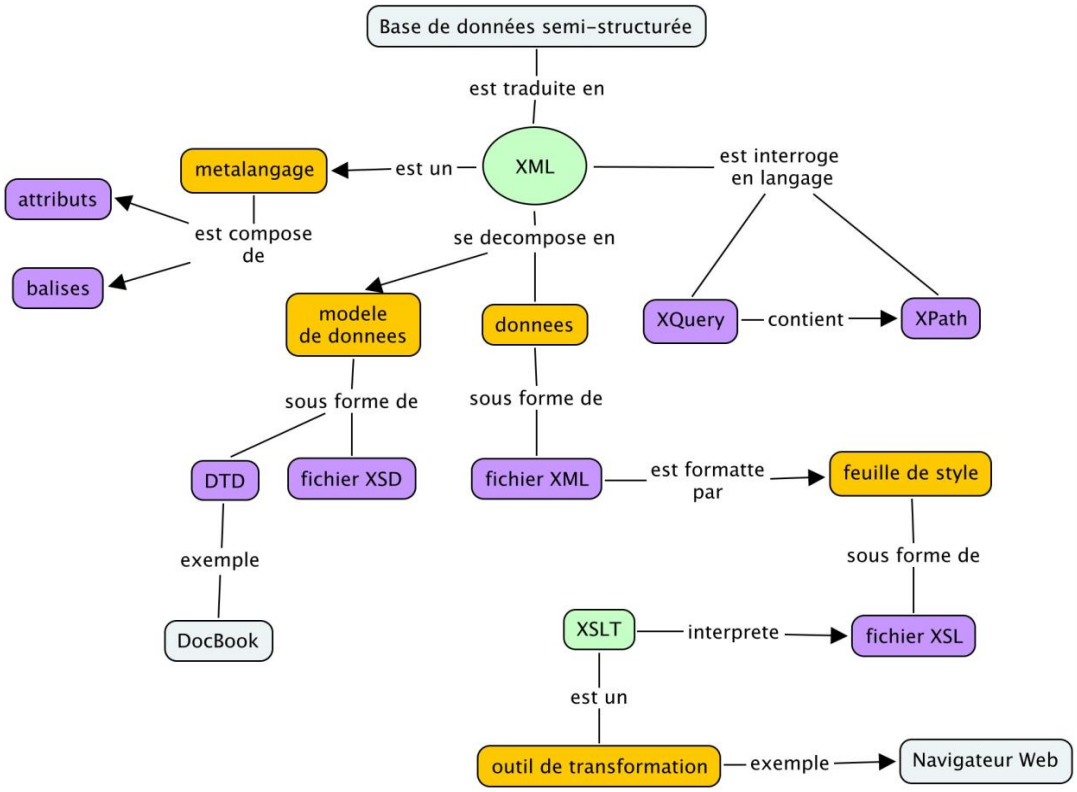
```
<xsl:template match="/">
<book>
<bookinfo>
<abstract>
<title><xsl:value-of select="@nom" /></title>
<xsl:for-each select="/DEPARTEMENT/SPECIALITE/CLASSE">
	<chapter>
	<title><xsl:value-of select="@nom" /></title>
		<xsl:for-each select="UV">
			<xsl:variable name="idRespUV" select="@idRespUV"/>
			<title><xsl:value-of select="@nom" /></title>
			<sect2>
			<title>Responsable UV</title>
			<para>
				<title><xsl:apply-templates
select="/DEPARTEMENT/UTILISATEUR[@id=$idRespUV]"/></title>
			</para>
			</sect2>
			<sect1>
			<xsl:for-each select="MATIERE">
				<para>
				<title><xsl:value-of select="@nom" /></title>
				</para>
```

```
                    </xsl:for-each>
                    </sect1>
                </xsl:for-each>
        </chapter>
</xsl:for-each>
</abstract>
</bookinfo>
</book>
</xsl:template>
<xsl:template match="/DEPARTEMENT/UTILISATEUR">
        <xsl:value-of select="@prenom" />
        <xsl:text disable-output-escaping="yes"> </xsl:text>
        <xsl:value-of select="@nom" />
</xsl:template>
```

Ensuite, grâce à xsltproc, nous avons pu **générer** le DocBook correspondant.

Puis, à partir de ce fichier, nous avons testé la génération d'un pdf, sachant que beaucoup d'autres possibilités existent. Pour cela nous avons d'abord généré un fichier .tex, converti en pdf grâce à la commande jade.

c. Carte des concepts



6. LE CMS COCOON

a. Présentation

Cocoon est un framework web Open Source développé par la fondation Apache, dont le fonctionnement est basé sur la notion de pipelines de composants, chacun d'entre eux étant destiné à un usage particulier.

L'avantage de cette architecture réside dans la possibilité de moduler facilement son application web en ajoutant ou en supprimant diverses fonctionnalités. Cocoon est codé en Java, nécessite l'utilisation d'un moteur de servlet (comme Tomcat ou Jetty) pour fonctionner et s'appuie massivement sur XML pour le traitement des données. De plus, il bénéficie d'une forte communauté et est en constante évolution.

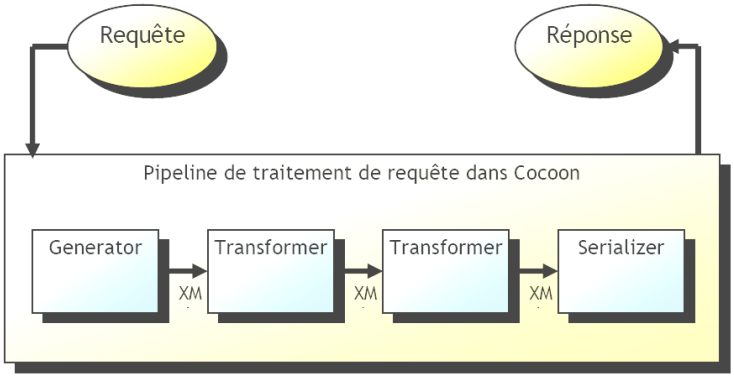
Cocoon est utilisé pour publier du contenu sur Internet, en séparant la forme des données : certains CMS (dont le CMS développé par Anyware) l'utilisent comme base. Cependant, ses applications peuvent être destinées à d'autres usages, tels que la génération de documents PDF. En effet, grâce à son modèle basé sur l'architecture MVC (Model-View-Controller), les différentes couches applicatives peuvent collaborer ensemble et peuvent être développées de façon indépendante. Ainsi, les infographistes, les développeurs Web et les rédacteurs peuvent travailler séparément pour un maximum d'efficacité.

Le Pipeline

La structure de base de l'application Cocoon est le pipeline. Elle permet de juxtaposer plusieurs étapes de traitement d'un document ou de (quasi)n'importe quelle source de données. Pour cela elle est composée de 3 types de composants :

- Le Generator
- Les Transformers.
- Le Serializer.

La structure d'un pipeline cocoon est définie par un fichier sitemap.



Le Generator

Cet élément reçoit une requête venant de l'utilisateur et charge la donnée depuis la ressource spécifiée. Le générateur est un programme qui génère du XML qui sera le point de départ du pipeline. Souvent (par défaut) c'est un programme qui lit un fichier sur le disque et retourne son contenu mais il peut aussi charger des données à partir d'une base de données (relationnelle, objet, semi structurée) ou tout autre forme de source de données.

Les Transformers

Les transformations consomment et produisent des événements SAX et s'enchaînent entre elles (on peut avoir de 1 à n transformations dans un pipeline contrairement au générateur et au sérialiseur

qui sont seuls). Le transformer par défaut est le processeur Xslt mais il existe d'autres transformeurs, par exemple pour aider à l'internationalisation (i18n).

Le serializer

Les derniers événements SAX générés par la dernière transformation sont capturés par le sérialiseur pour terminer le pipeline et produire un document.

Le sérialiseur par défaut produit un fichier HTML mais là encore il en existe pour de nombreux autres formats (XHTML, SVG, PS, PDF, ...).

La Sitemap

Pour mettre en place un service Cocoon sous forme d'un pipeline il faut rédiger un fichier qui va décrire les différentes étapes du pipeline : c'est le fichier sitemap.xmap. C'est un document XML qui décrit les différentes étapes : quel générateur utiliser, quelle(s) transformation(s) appliquer au document et enfin quel sérialiseur utiliser pour générer un fichier binaire.

b. Travail réalisé

On a commencé par créer un projet java sous eclipse basé sur les sources du CMS ametys fournies par l'enseignant lors de la première séance d'APP consacré aux systèmes de gestion de contenu.

Nous avons créé par la suite un nouveau service permettant d'afficher le résultat d'une requête écrite en XQuery. Pour cela, nous avons d'abord créé le sitemap de notre module qui contient un générateur XML vers XML, un transformeur XML vers HTML et un sérialiseur standard.

Pour la génération, on a utilisé l'API d'eXist afin d'interroger notre base de données XML et récupérer le résultat qui a dû être parsé avec SAX pour générer le flux XML de sortie.

Le transformeur dont le chemin d'accès doit être mentionné dans le sitemap, récupère le flux xml généré par SAX et le transforme en HTML grâce à une feuille de style XSLT.

Enfin, le sérialiseur génère le flux d'octets destiné au client. Une fois ce module créé, nous avons conçu une page HTML basée sur ce service dans le CMS permettant d'afficher toutes les informations concernant le personnel du DGEI.

Cette étape du projet nous a permis de mieux comprendre le principe de fonctionnement des CMS et le concept du workflow.

7. CONCLUSION

Bilan des apprentissages et du groupe

Nous venons tous de milieux différents avec des expériences professionnelles et des spécialités diverses. Cela a représenté immanquablement un point fort afin de mener à bien l'ensemble des exigences du cahier des charges et aussi d'arriver à une mise en œuvre la plus aboutie possible.

Sur le plan pédagogique, cette APP nous a permis d'approfondir les concepts UML et de les appliquer au sein d'un travail concret. Nous avons aussi pu découvrir le langage XML, comment le mettre en œuvre ainsi que les outils et points importants associés à ces nouveaux concepts.

Un autre point essentiel a été d'apprendre à gérer des délais qui se sont avérés relativement courts, au vu du sous-effectif de notre groupe (essentiellement en fin de semestre, n'étant plus que 3 au lieu de 6 personnes pour les autres groupes). N'ayant pas pris rapidement l'habitude de bien définir chaque tache importante à effectuer chaque semaine, nous avons dû rattraper le retard généré par notre manque d'expérience dans la gestion d'un projet.

Cependant, le fait que nous ayons mené à bien le projet compte tenu des difficultés rencontrées nous a procuré une certaine satisfaction. Cela a représenté un certain challenge que nous avons su relever, ainsi qu'une expérience intéressante et enrichissante.

ANNEXES

a. Fichier XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2008 sp1 (http://www.altova.com) by Jihed Othmani (PC) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
	<xs:element name="COURS">
		<xs:complexType>
			<xs:sequence>
				<xs:element name="Rintervenant" minOccurs="0" maxOccurs="unbounded">
					<xs:complexType>
						<xs:attribute name="refintervenant"/>
					</xs:complexType>
				</xs:element>
				<xs:element name="Rmatiere">
					<xs:complexType>
						<xs:attribute name="refmatiere"/>
					</xs:complexType>
				</xs:element>
				<xs:element name="Rgroupe" maxOccurs="unbounded">
					<xs:complexType>
						<xs:attribute name="refgroupe"/>
					</xs:complexType>
				</xs:element>
				<xs:element name="Rtype">
					<xs:complexType>
						<xs:attribute name="reftype"/>
					</xs:complexType>
				</xs:element>
			</xs:sequence>
			<xs:attribute name="id" type="xs:nonNegativeInteger" use="required"/>
			<xs:attribute name="nbHeures" type="xs:float" use="optional" default="0"/>
			<xs:attribute name="fait" type="xs:boolean" use="optional" default="false"/>
			<xs:attribute name="date" type="xs:date"/>
		</xs:complexType>
	</xs:element>
	<xs:element name="TYPE_SEANCE">
		<xs:complexType>
			<xs:attribute name="nom" use="required">
				<xs:simpleType>
					<xs:restriction base="xs:string">
						<xs:enumeration value="TP"/>
						<xs:enumeration value="TD"/>
						<xs:enumeration value="Cours"/>
						<xs:enumeration value="Controle"/>
					</xs:restriction>
				</xs:simpleType>
			</xs:attribute>
			<xs:attribute name="coefficient" type="xs:float"/>
		</xs:complexType>
	</xs:element>
	<xs:element name="GROUPE">
		<xs:complexType>
			<xs:attribute name="nom" type="xs:string" use="required"/>
		</xs:complexType>
	</xs:element>
	<xs:element name="CLASSE">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="GROUPE" maxOccurs="unbounded"/>
				<xs:element ref="UV" maxOccurs="unbounded"/>
			</xs:sequence>
			<xs:attribute name="id" type="xs:nonNegativeInteger" use="required"/>
			<xs:attribute name="specialite" type="xs:string" use="required"/>
			<xs:attribute name="nom" type="xs:string" use="required"/>
			<xs:attribute name="annee" use="required">
				<xs:simpleType>
					<xs:restriction base="xs:nonNegativeInteger">
```

```
                                        <xs:enumeration value="1"/>
                                        <xs:enumeration value="2"/>
                                        <xs:enumeration value="3"/>
                                        <xs:enumeration value="4"/>
                                        <xs:enumeration value="5"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:attribute>
        </xs:complexType>
</xs:element>
<xs:element name="MATIERE">
        <xs:complexType>
                <xs:attribute name="id" type="xs:nonNegativeInteger" use="required"/>
                <xs:attribute name="nom" type="xs:string" use="required"/>
                <xs:attribute name="nbHcours"/>
                <xs:attribute name="nbHTD"/>
                <xs:attribute name="nbHTP"/>
        </xs:complexType>
</xs:element>
<xs:element name="UV">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="MATIERE" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="code" type="xs:nonNegativeInteger" use="required"/>
                <xs:attribute name="semestre" type="xs:nonNegativeInteger" use="required"/>
                <xs:attribute name="nom" type="xs:string" use="required"/>
                <xs:attribute name="nbCredits" type="xs:nonNegativeInteger" use="optional"
default="3"/>
                <xs:attribute name="idRespUV" use="required"/>
        </xs:complexType>
</xs:element>
<xs:element name="SPECIALITE">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="CLASSE" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="libelle" use="required">
                        <xs:simpleType>
                                <xs:restriction base="xs:string">
                                        <xs:enumeration value="I"/>
                                        <xs:enumeration value="RT"/>
                                        <xs:enumeration value="AEI"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:attribute>
        </xs:complexType>
</xs:element>
<xs:element name="UTILISATEUR">
        <xs:complexType>
                <xs:attribute name="id" type="xs:nonNegativeInteger" use="required"/>
                <xs:attribute name="nom" type="xs:string"/>
                <xs:attribute name="prenom" type="xs:string"/>
                <xs:attribute name="poste">
                        <xs:simpleType>
                                <xs:restriction base="xs:string">
                                        <xs:enumeration value="SECRETAIRE"/>
                                        <xs:enumeration value="DIRECTEUR DU DEPARTEMENT"/>
                                        <xs:enumeration value="DIRECTEUR DES ETUDES"/>
                                        <xs:enumeration value="INTERVENANT"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="telephone" type="xs:string"/>
                <xs:attribute name="fax" type="xs:string" use="optional"/>
                <xs:attribute name="courriel" type="xs:string"/>
                <xs:attribute name="est_intervenant" type="xs:boolean" use="optional"
default="false"/>
                <xs:attribute name="type" use="optional">
                        <xs:simpleType>
                                <xs:restriction base="xs:string">
                                        <xs:enumeration value="ATER"/>
                                        <xs:enumeration value="VACATAIRE"/>
                                        <xs:enumeration value="PERMANENT"/>
                                        <xs:enumeration value="MONITEUR"/>
```

```
                                        <xs:enumeration value="RATTACHE"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:attribute>
                    </xs:complexType>
            </xs:element>
            <xs:element name="DEPARTEMENT">
                    <xs:annotation>
                            <xs:documentation>Comment describing your root element</xs:documentation>
                    </xs:annotation>
                    <xs:complexType>
                            <xs:sequence>
                                    <xs:element ref="UTILISATEUR" maxOccurs="unbounded"/>
                                    <xs:element ref="SPECIALITE" maxOccurs="unbounded"/>
                                    <xs:element ref="COURS" maxOccurs="unbounded"/>
                                    <xs:element ref="TYPE_SEANCE" maxOccurs="unbounded"/>
                            </xs:sequence>
                            <xs:attribute name="nom" type="xs:string" use="required"/>
                    </xs:complexType>
                    <xs:key name="iddep">
                            <xs:selector xpath="DEPARTEMENT"/>
                            <xs:field xpath="@NOM"/>
                    </xs:key>
                    <xs:key name="idcours">
                            <xs:selector xpath=".//COURS"/>
                            <xs:field xpath="@id"/>
                    </xs:key>
                    <xs:key name="idtype_seance">
                            <xs:selector xpath=".//TYPE_SEANCE"/>
                            <xs:field xpath="@nom"/>
                    </xs:key>
                    <xs:key name="idgroupe">
                            <xs:selector xpath=".//GROUPE"/>
                            <xs:field xpath="@nom"/>
                    </xs:key>
                    <xs:key name="idutilisateur">
                            <xs:selector xpath=".//UTILISATEUR"/>
                            <xs:field xpath="@id"/>
                    </xs:key>
                    <xs:key name="idclasse">
                            <xs:selector xpath=".//CLASSE"/>
                            <xs:field xpath="@id"/>
                    </xs:key>
                    <xs:key name="idmatiere">
                            <xs:selector xpath=".//MATIERE"/>
                            <xs:field xpath="@id"/>
                    </xs:key>
                    <xs:key name="idspecialite">
                            <xs:selector xpath=".//SPECIALITE"/>
                            <xs:field xpath="@libelle"/>
                    </xs:key>
                    <xs:keyref name="idintervenant" refer="idutilisateur">
                            <xs:selector xpath=".//COURS/Rintervenant"/>
                            <xs:field xpath="@refintervenant"/>
                    </xs:keyref>
                    <xs:keyref name="refmatiere" refer="idmatiere">
                            <xs:selector xpath=".//COURS/Rmatiere"/>
                            <xs:field xpath="@refmatiere"/>
                    </xs:keyref>
                    <xs:keyref name="idRespUV" refer="idutilisateur">
                            <xs:selector xpath=".//UV"/>
                            <xs:field xpath="@idRespUV"/>
                    </xs:keyref>
                    <xs:key name="idtype">
                            <xs:selector xpath=".//TYPE_SEANCE"/>
                            <xs:field xpath="@nom"/>
                    </xs:key>
                    <xs:keyref name="reftype" refer="idtype_seance">
                            <xs:selector xpath=".//COURS/Rtype"/>
                            <xs:field xpath="@reftype"/>
                    </xs:keyref>
            </xs:element>
</xs:schema>
```

b. Extrait du jeu de données

```
	<UTILISATEUR id="1" nom="DILHAC" prenom="Jean-Marie" telephone="0512345678" fax="0512345679"
courriel="dilac@insa-toulouse.fr" poste="DIRECTEUR DU DEPARTEMENT" est_intervenant="true"
type="PERMANENT"/>

	<UTILISATEUR id="2" nom="MARRE" prenom="Daniel" telephone="0561246670" fax="0561246671"
courriel="marre@insa-toulouse.fr" poste="DIRECTEUR DES ETUDES" est_intervenant="true"
type="PERMANENT"/>

	<UTILISATEUR id="3" nom ="BRICOT" prenom ="Elsa"   telephone="0561246600"  fax="0561246601"
courriel="bricot@insa-toulouse.fr" poste="SECRETAIRE" est_intervenant="false"/>

	<UTILISATEUR id="5" nom="MAHOUT" prenom="Vincent" telephone="0561246670" fax="0561246671"
courriel="vmahout@insa-toulouse.fr" poste="INTERVENANT" est_intervenant="true" type="ATER"/>
	<SPECIALITE libelle="I">

		<CLASSE id="1" annee="4"  specialite="I" nom="4 Informatique ">
			<GROUPE nom="4INFO_A"/>
			<GROUPE nom="4INFO_B"/>
			<GROUPE nom="4INFO_C"/>
		<UV semestre="1" code="1" nom="PROGRAMMATION" idRespUV="1">
			<MATIERE id="1" nom="JAVA" nbHcours="20" nbHTD="15" nbHTP="15"/>
			<MATIERE id="2" nom="C" nbHcours="20" nbHTD="15" nbHTP="15"/>
			<MATIERE id="3" nom="CAML" nbHcours="20" nbHTD="15" nbHTP="15"/>
		</UV>

		<UV semestre="1" code="2" nom="APP" idRespUV="2">
			<MATIERE id="4" nom="SDI"  nbHTP="60"/>
		</UV>
		</CLASSE>

		<CLASSE id="2" annee="5"  specialite="I" nom="5 Informatique ">
			<GROUPE nom="5INFO_A"/>
			<GROUPE nom="5INFO_B"/>
			<GROUPE nom="5INFO_C"/>
		<UV semestre="1" code="3" nom="APS" idRespUV="4">
			<MATIERE id="5" nom="SPORT" nbHcours="20"/>
		</UV>

		<UV semestre="1" code="4" nom="TEMPS REEL" idRespUV="5">
			<MATIERE id="6" nom="PPI" nbHcours="50" />
		</UV>
		</CLASSE>

	</SPECIALITE>
	<COURS id="1" nbHeures="2" fait="false" date="2007-12-30">
		<Rintervenant refintervenant="1"/>
		<Rmatiere refmatiere="1"/>
		<Rgroupe refgroupe="4INFO_A"/>
		<Rtype reftype="Cours"/>
		</COURS>

		<COURS id="2" nbHeures="2" fait="false" date="2007-12-31">
		<Rintervenant refintervenant="2"/>
		<Rmatiere refmatiere="4"/>
		<Rgroupe refgroupe="4INFO_A"/>
		<Rtype reftype="Controle"/>
		</COURS>

		<COURS id="3" nbHeures="2" fait="false" date="2008-01-31">
		<Rintervenant refintervenant="1"/>
		<Rmatiere refmatiere="4"/>
		<Rgroupe refgroupe="4INFO_A"/>
		<Rtype reftype="Cours"/>
		</COURS>

<TYPE_SEANCE nom="Cours" coefficient="1.5"/>
<TYPE_SEANCE nom="TP" coefficient="0.66"/>
<TYPE_SEANCE nom="TD" coefficient="1"/>
<TYPE_SEANCE nom="Controle" coefficient="1"/>
```

c. Fichiers XSL

Fiche ECTS

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:param name="codeuv" select='1'/>
<xsl:template match="/" >
        <html>
    <head>
        <link href="jdd.css" rel="stylesheet" type="text/css"/>
    </head>
        <body>
        <xsl:apply-templates select="DEPARTEMENT/SPECIALITE/CLASSE/UV[@code=$codeuv]"/>
        </body>
        </html>
</xsl:template>

<xsl:template match="DEPARTEMENT/SPECIALITE/CLASSE/UV" >
        <xsl:variable name="respUV" select="@idRespUV"/>
        <h1>Fiche ECTS <xsl:value-of select="@nom"/></h1>
        <b>Code : </b><xsl:value-of select="@code"/><br/>
        <h2>Responsable UV : </h2>

        Nom : <xsl:apply-templates select="/DEPARTEMENT/UTILISATEUR[@id=$respUV]"/>

        <h2>Liste de matieres : </h2>
        <xsl:for-each select="MATIERE">
                <xsl:variable name="idMat" select="@id"/>
                <h3><xsl:value-of select="@nom"/></h3>

         <b>Intervenants :</b><br/>
                <xsl:for-each select="/DEPARTEMENT/COURS[Rmatiere/@refmatiere=$idMat]">
                        <xsl:variable name="idInt" select="Rintervenant/@refintervenant"/>
                        <xsl:apply-templates select="/DEPARTEMENT/UTILISATEUR[@id=$idInt]"/>
                </xsl:for-each>

                <br/><br/><table width="200">
         <tr><td>Cours</td><td><xsl:value-of select="@nbHcours"/></td></tr>
                <tr><td>TD</td><td><xsl:value-of select="@nbHTD"/></td></tr>
                <tr><td>TP</td><td><xsl:value-of select="@nbHTP"/></td></tr>
         </table>
        </xsl:for-each>

</xsl:template>

<xsl:template match="DEPARTEMENT/UTILISATEUR" >
        <xsl:value-of select="@prenom" />
        <xsl:text disable-output-escaping="yes"> </xsl:text>
        <xsl:value-of select="@nom" />
</xsl:template>

</xsl:stylesheet>
```

Fiche prévisionnelle professeur

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:param name="codeuser" select='1'/>

<xsl:template match="/" >
        <html>
    <head>
        <link href="jdd.css" rel="stylesheet" type="text/css"/>
    </head>
        <body>
        <xsl:apply-templates select="DEPARTEMENT/UTILISATEUR[@id=$codeuser]"/>
        </body>
        </html>
</xsl:template>

<xsl:template match="DEPARTEMENT/UTILISATEUR" >
```

```
        <h1>Fiche prévisionnelle de <xsl:value-of select="@prenom" />
        <xsl:text disable-output-escaping="yes"> </xsl:text>
        <xsl:value-of select="@nom" /></h1>
        <strong>Nombre Heures Previsionelles : <xsl:value-of
select='sum(/DEPARTEMENT/COURS[Rintervenant/@refintervenant = $codeuser]/@nbHeures)'/></strong><br/>
        <strong>Nombre Heures Faites : <xsl:value-of
select='sum(/DEPARTEMENT/COURS[Rintervenant/@refintervenant = $codeuser and
@fait="true"]/@nbHeures)'/></strong>
        <h2>Liste des matieres : </h2>
        <table>
        <tr><td width='200' id="strong">Matiere</td><td id="strong">Nombre d'heures</td></tr>
        <xsl:variable name="nbH" select='1'/>

        <xsl:for-each select="/DEPARTEMENT/COURS">
                <xsl:if test="Rintervenant/@refintervenant = $codeuser">
                <xsl:variable name="idMat" select="Rmatiere/@refmatiere"/>
                <tr><td><xsl:value-of
select="/DEPARTEMENT/SPECIALITE/CLASSE/UV/MATIERE[@id=$idMat]/@nom" /></td><td><xsl:value-of
select='sum(/DEPARTEMENT/COURS[Rintervenant/@refintervenant = $codeuser and
Rmatiere/@refmatiere=$idMat]/@nbHeures)'/></td></tr>
                </xsl:if>
        </xsl:for-each>
        </table>
</xsl:template>
</xsl:stylesheet>
```

d. DocBook

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet
version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output
method="xml" encoding="ISO-8859-1"
media-type="text/html"
omit-xml-declaration="no"
indent="yes"
doctype-public= "-//OASIS//DTD DocBook XML V4.1.2//EN"
doctype-system="http://www.oasis-open.org/docbook/xml/4.0/docbookx.dtd" />

<xsl:template match="/">
<book>
<bookinfo>
<abstract>
<title><xsl:value-of select="@nom" /></title>
<xsl:for-each select="/DEPARTEMENT/SPECIALITE/CLASSE">
<chapter>
<title><xsl:value-of select="@nom" /></title>
<xsl:for-each select="UV">
<xsl:variable name="idRespUV" select="@idRespUV"/>
<title><xsl:value-of select="@nom" /></title>
<sect2>
<title>Responsable UV</title>
<para>
<title><xsl:apply-templates select="/DEPARTEMENT/UTILISATEUR[@id=$idRespUV]"/></title>
</para>
</sect2>
<sect1>
<xsl:for-each select="MATIERE">
<para>
<title><xsl:value-of select="@nom" /></title>
</para>
</xsl:for-each>
</sect1>
</xsl:for-each>
</chapter>
</xsl:for-each>
</abstract>
</bookinfo>
</book>
</xsl:template>
<xsl:template match="/DEPARTEMENT/UTILISATEUR">
<xsl:value-of select="@prenom" />
<xsl:text disable-output-escaping="yes"> </xsl:text>
<xsl:value-of select="@nom" />
```

```
</xsl:template>
</xsl:stylesheet>
```