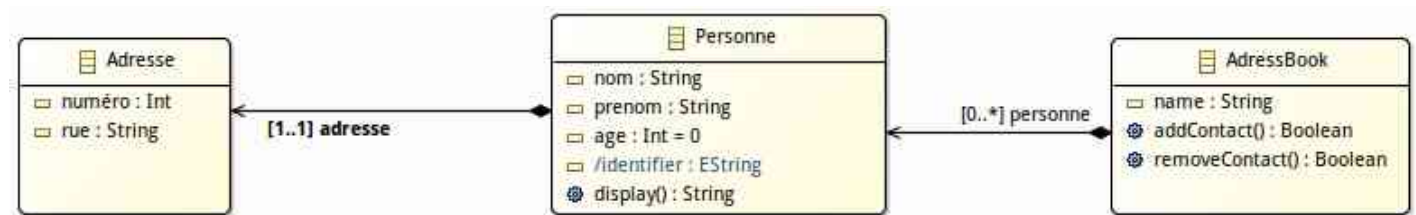
 S. Hérauvillle	Développement Collaboratif - UML	M1 GIL
	Implémentation d'un projet ECORE	TP n° 2
	Contraintes OCL	page 1 / 6

## I - Présentation

### Question I.1 - Définition des objectifs

- Créer des instances du modèle défini par son schéma UML (instanciation des classes générées, réflexivité),
- Manipuler les données (objets) produits par le modèle (persistance des données),
- Manipuler la structure du modèle Ecore.

Le schéma UML du modèle de l'application créée lors du TP précédent est le suivant :



Référence : [help.eclipse.org/mars/index.jsp?nav=%2F20](http://help.eclipse.org/mars/index.jsp?nav=%2F20)

### Question I.2 - Projet carnet d'adresse

Au démarrage de ce TP, vous devez disposer d'une application conforme au diagramme ci-dessus, ainsi que de classes de test permettant de gérer la persistance du carnet d'adresse.

## II - Insertion de contrainte OCL

Vous allez insérer des contraintes OCL dans le modèle selon les consignes suivantes :


### Question II.1 - Paramétrage du modèle

Apportez les modifications suivantes au projet :

- Accédez aux propriétés de "addressbook.genmodel" (fenêtre centrale) et positionnez les paramètres suivants :

Model	
Model Plug-in Variables	OCL_ECORE=org.eclipse.ocl.ecore
Template & Merge	
Dynamic Templates	true
Template Directory	{workspace}/{package-projet}/templates

- Modifiez les propriétés du projet : menu window → preferences → ocl  
realisation of OCL embedded within ECORE models  
generate Java Code in \*Impl classes

 S. Hérauvillle	Développement Collaboratif - UML	M1 GIL
	Implémentation d'un projet ECORE	TP n° 2
	Contraintes OCL	page 2 / 6

### Question II.2 - Insertion des contraintes OCL

- Ouvrez le modèle "addressbook.ecore" avec OCLinEcore Editor

Ajoutez les contraintes suivantes à la classe Personne :

La propriété age doit être supérieur à 16

La propriété nom doit être écrite en majuscules

Remarques :

- Les contraintes seront définies comme invariant (nommé) dans la classe.

- Utiliser la syntaxe définie dans la norme OCL

- Rechargez le modèle "addressbook.genmodel" et re-générez le code. Identifiez ensuite les modifications apportées au code.

### Question II.3 - Simulation des instances

- Ouvrez le modèle "addressbook.ecore" avec OCLinEcore Editor

- Sélectionnez la classe de base du carnet d'adresse (double-clic)

clic-droit → create dynamic instance

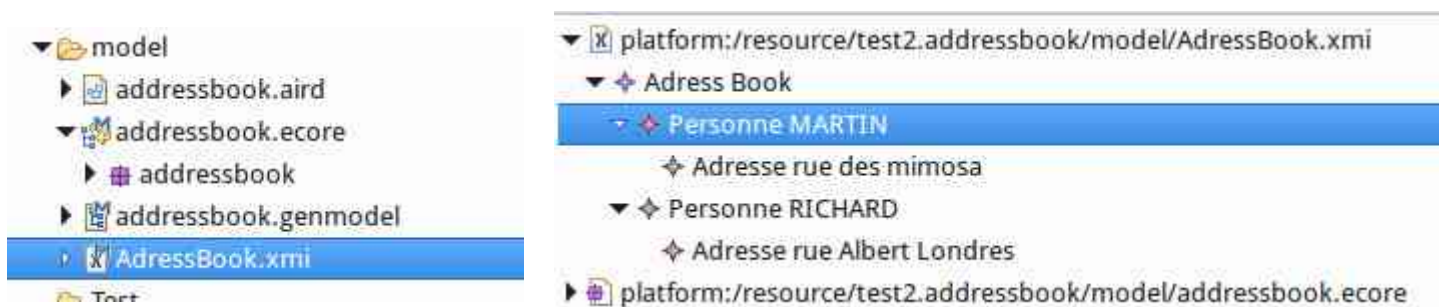
un fichier xmi est ajouté aux modèle du projet

- Sélectionnez le fichier xmi créé → ouvrir avec → other... → Sample Ecore Model Editor

- Dans la fenêtre centrale, sélectionnez Adressbook → clic-droit → new child → Personne

Complétez le contenu dans la fenêtre des propriétés

Associez une Adresse à Personne



- Sur ce principe, complétez le carnet d'adresse avec 3 personnes, afin d'obtenir au final :

Une personne dont toutes les propriétés respectent les contraintes énoncées précédemment

Une personne dont l'age est incorrect

Une personne dont le nom est incorrect

### Question II.4 - Validation OCL


- Sélectionner la classe de base du carnet d'adresse

(Adress Book sur l'illustration) → clic-droit → validate

Une fenêtre doit s'ouvrir confirmant la validation du contenu : Affichez le détail

- Corrigez les erreurs affichées, puis relancez la validation du modèle

- Sauvegardez votre fichier, puis affichez à nouveau le contenu du fichier AdressBook.xmi avec un éditeur xml.

 S. Hérauvill	Développement Collaboratif - UML	M1 GIL
	Implémentation d'un projet ECORE	TP n° 2
	Contraintes OCL	page 3 / 6

### III - Méta-Model Ecore

#### Question III.1 - Analyse du méta-modèle Ecore

Créez une nouvelle classe "TestEcore" qui permettra d'analyser le contenu du modèle Ecore de votre projet  
Créez une méthode "queryAddressbookStructure" qui permettra d'interroger le modèle.  
Annoter cette méthode avec @Test et complétez là avec le code ci-dessous :

```
public class TestEcore {
    @Test
    public void queryAddressbookStructure() {
        AddressbookPackage abPackage = AddressbookPackage.eINSTANCE;
        EList<EClassifier> eClass = abPackage.getEClassifiers();

        for (EClassifier eClassifier : eClass) {
            system.out.println(eClassifier.getName());
        }
    }
}
```

- Validez le code et exécutez le.


#### Question III.2 - Affichage du contenu

- Vous devez maintenant afficher le contenu complet du projet sous la forme suivante :

```
AdressBook
    name(EString)
    Références : personne(Personne[0..-1])
    Opérations : Boolean addContact
                Boolean removeContact
Personne
    nom(EString) prenom(EString) age(EInt) identifier(EString)
    Références : adresse(Adresse[1..1])
    Opérations : EString display
Adresse
    numero(EInt) rue(EString)
```

- Complétez le code précédent afin d'afficher le détail de chaque élément (format, attributs, opérations, ...)

*Remarque : Exploitez le schéma du méta-modèle Ecore fourni en annexe*

 <b>UNIVERSITÉ DE ROUEN</b> <i>S. Hérauville</i>	Développement Collaboratif - UML	<b>M1 GIL</b>
	<b>Implémentation d'un projet ECORE</b>	<b>TP n° 2</b>
	<b>Contraintes OCL</b>	page 4 / 6

## IV - Manipulation du modèle Ecore

### Question IV.1 - Création d'instance via le méta-modèle Ecore

- La méthode utilisée précédemment dans les classes "AddressbookTest" & "ProjectTest" pour créer des instances du projet est qualifiée de "EarlyBinding", et exploite les classes générées par Eclipse.

Maintenant, nous allons créer et modifier des instances du modèle par réflexivité (qualifié de LateBinding)

- Construire un nouveau projet EMF vide

File → Project ... → Eclipse Modeling Framework → Empty EMF Project  
nommez le projet : earlybinding.addressbook

- Ajouter au projet les dépendances suivantes : org.junit et org.eclipse.emf.ecore.xmi

- Créez le package "univrouen.adressbook.latebinding"

Créez dans ce package, la classe "AddressbookLateBinding"

- S'il n'existe pas déjà, créez un répertoire "model", dans lequel vous recopierez votre modèle Ecore : "addressbook.ecore".

*Remarque : Votre projet ne contient aucune dépendance vers les projets précédents.*

### Question IV.2 - Analyse du méta-modèle Ecore

- Créez une nouvelle classe comportant la méthode de test nommée

"queryAddressbookStructureWithoutCode" dont l'objectif est le suivant :

Charger le fichier "addressbook.ecore" afin de charger le modèle

Afficher la structure du modèle (idem question B3)


@Test

```
public void queryAddressbookStructureWithoutCode() {
    Resource.Factory.Registry reg = Resource.Factory.Registry.INSTANCE;
    Map<String, Object> m = reg.getExtensionToFactoryMap();
    m.put("ecore", new XMIResourceFactoryImpl());
    ResourceSet resourceSet = new ResourceSetImpl();
    URI fileURI = URI.createFileURI("model/addressbook.ecore");

    Resource resource = resourceSet.getResource(fileURI, true);
    EPackage ePackage = (EPackage) resource.getContents().get(0);
```

... suite du code : idem question B3

Complétez le code et validez le résultat (affichage) obtenu.

 <i>S. Hérauville</i>	Développement Collaboratif - UML	<b>M1 GIL</b>
	<b>Implémentation d'un projet ECORE</b>	<b>TP n° 2</b>
	<b>Contraintes OCL</b>	page 5 / 6

### Question IV.3 - Création d'une instance LateBinding

L'objectif est maintenant d'implémenter le modèle directement sans utiliser le code généré (voir TP n°1)

Créez une nouvelle classe comportant une méthode de test nommée

"implementsAddressbookFromModel".

- Le début du code est le suivant :

```
public void implementsAddressbookFromModel() throws IOException {
    Resource.Factory.Registry reg = Resource.Factory.Registry.INSTANCE;
    Map<String, Object> m = reg.getExtensionToFactoryMap();
    m.put("ecore", new XMIRResourceFactoryImpl());

    ResourceSet resourceSet = new ResourceSetImpl();
    URI fileURI = URI.createFileURI("model/addressbook.ecore");
    Resource resource = resourceSet.createResource(fileURI);

    resource.load(null);
    EPackage ePackage = (EPackage) resource.getContents().get(0);

    EClass eAddressbook = (EClass) ePackage.getEClassifier("Addressbook");
    EReference eContains =
        (EReference) eAddressbook.getEStructuralFeature("personne");
    EAttribute eName =
        (EAttribute) eAddressbook.getEStructuralFeature("name");
    EObject addressbookInstance =
        ePackage.getEFactoryInstance().create(eAddressbook);
    addressbookInstance.eSet(eName, "Mon Carnet d'Adresses");
    ...
}
```

Complétez le code afin d'implémenter le projet

Ajoutez 2 membres au projet et validez (Assert) le résultat

### Question IV.4 - Persistance des données

- Ajoutez à votre code la gestion de la persistance des données (instances de Personne créées)

Validez le format de sauvegarde des données

Mettez ensuite en œuvre le rechargement des données à l'initialisation

