

# Licence Informatique 2008-2009

## Module Modélisation Orientée Objet (MOO)

Joel Savelli, [joel.savelli@u-bourgogne.fr](mailto:joel.savelli@u-bourgogne.fr), bureau G211

Marie-Noëlle Terrasse, [marie-noelle.terrasse@u-bourgogne.fr](mailto:marie-noelle.terrasse@u-bourgogne.fr), bureau G215

### Projet (troisième partie)

Vous devrez faire une démonstration de votre programme SMV en expliquant à l'oral la cohérence des diagrammes et du programme SMV.

---

## 1 Partie 1

Vous devez construire un modèle UML détaillé correspondant à la description ci-dessous du fonctionnement des laboratoires de CRB :

**Le cycle de vie d'un jeu d'échantillons** est construit sur les états : *creation*, *preparation*, *validationTech*, *erreur*.

- L'état *creation* correspond à la prise en charge des échantillons par le laboratoire (réception d'échantillons pour une collection donnée, attribution d'un identifiant propre au laboratoire à chaque échantillon, première phase d'enregistrement). Il ne peut pas y avoir de problème sur cet état. La seule transition possible se fait sur l'état *preparation*.
- L'état *preparation* correspond au passage des échantillons dans les robots du laboratoire. Cette préparation peut déboucher sur une validation (le jeu d'échantillon peut être versé dans la collection) ou une erreur (le délai maximum de traitement autorisé, de 6 heures, a été dépassé).
- L'état *validationTech* permet le versement du jeu d'échantillon à la collection avec enregistrement des dernières informations (comme par exemple la date d'ajout à la collection). Ce versement ne peut se faire que lorsque le robot a été vidé.
- L'état *erreur* permet la destruction des échantillons et l'enregistrement de leur destruction comme dernière information relative aux échantillons.

On suppose que les opérations sur la classe échantillon comprennent au moins :

```
reception(col:collection, d: date)
identification()
initEnregistrement()
versementCollection(c:collection, d:date)
finEnregistrement()
destruction()
```

**Le cycle de vie des dossiers patient** est construit sur les états : *extraction*, *validation*, *transfert*.

- L'état *extraction* permet de sélectionner dans le dossier patient les informations intéressantes pour le CRB dans le cadre de l'annotation des échantillons correspondants à des prélèvements sur ce patient. Cette extraction est faite en deux phases : une pré-sélection d'informations standards déterminées à partir de la nature de l'échantillon, une post-sélection (obligatoire manuelle) qui permet d'enlever ou d'ajouter des informations au résultat de la pré-sélection.
- L'état *validation* permet de vérifier 1) que toutes les informations nécessaires au suivi du patient sont bien sélectionnées et 2) que toutes les recommandations et contraintes légales relatives aux droits des patients sont respectées.
- L'état *transfert* permet d'assurer l'entrée des informations sélectionnées dans le système d'information du CRB : les informations sont tout d'abord normalisées (en fonction de la structure du système d'information du CRB) puis effectivement inscrites dans le système d'information.

On suppose que les opérations sur la classe dossier comprennent au moins :

```
preSelection (fp:fichePrelevement)
postSelection (fp:fichePrelevement)
validationPatient ()
validationlegale ()
normalisation ()
inscription ()
```

**Le cycle de vie des robots** est construit sur les états : *parametrage, chargement, preparation, annotation, vidage*.

- L'état *parametrage* permet de définir de façon précise les conditions expérimentales qui seront appliquées au jeu d'échantillons à préparer.
- L'état *chargement* permet de placer les échantillons sur les différentes zones de stockage du robot. Le chargement ne peut se faire que lorsque le jeu d'échantillon est créé.
- L'état *preparation* permet de réaliser les manipulations proprement dites sur les échantillons. Les manipulations consistent en opérations d'extraction et d'aliquotage. Une opération de vérification est exécutée en sortie de cet état préparation.
- L'état *annotation* permet d'enregistrer toutes les informations relatives aux expérimentations. Cette annotation ne peut se faire que lorsque le dossier médical patient est valide. Une opération d'annotation est utilisée si la vérification est positive. En cas d'erreur, une opération de pre-destruction permet d'indiquer les hypothèses sur les causes d'erreur.
- L'état *vidage* permet de verser les échantillons dans la collection ou de les jeter (en fonction du résultat de la vérification).

On suppose que les opérations sur la classe robot comprennent au moins :

```
parametrer ()
stocker (e:echantillon)
extraire (e:echantillon)
aliquoter (e:echantillon)
verifier (e:echantillon)
annoter (e:echantillon)
pre-destruire (e:echantillon)
verser (c:collection)
jeter (e:echantillon)
```

A partir de l'énoncé ci-dessus vous devez préparer un dossier contenant :

1. **Un premier modèle des trois classes** jeux d'échantillon, robots et dossiers patient telles qu'elles sont décrites ci-dessus. Vous devez indiquer tous les attributs et opérations nécessaires. Vous devez indiquer événements échangés entre ces classes et construire le diagramme de communication.
2. **Les statecharts** de ces trois classes. N'oubliez pas d'indiquer de façon précise les événements envoyés et reçus.
3. **Un programme SMV** permettant de vérifier que tout jeu d'échantillon créé atteindra un des états *validationTech* ou *erreur*. Il n'est pas nécessaire de donner le code complet du programme mais il faut en expliquer les grandes lignes et les points délicats. Vous devez montrer la cohérence entre les diagrammes de classe, statechart et le programme SMV.

## 2 Partie 2

Le comportement des robots (en particulier le paramétrage) est décrit de façon plus précise de la façon suivante :

- L'expérimentation peut être constituée de deux à trois phases d'extraction (notées E1, E2 et E3) et de une à trois phases d'aliquotage (notées A1, A2 et A3).
- Deux vérifications (notées Ve et Va) sont faites au cours de la séquence de réception des paramètres.
- Les paramètres de chaque phase sont reçus en un bloc. L'enchaînement des réceptions de blocs et des vérifications correspond au schéma donné en Figure 1.
- Le succès de la vérification Ve permet de commencer le chargement du robot.

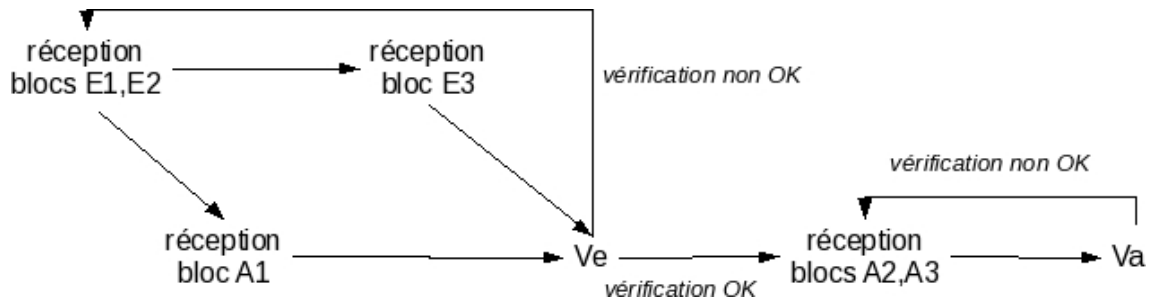


FIG. 1 – Paramétrage des robots

– Le succès de la vérification Va est nécessaire avant de commencer la préparation des échantillons.

1. **Proposer un nouveau statechart pour le paramétrage des robots**
2. **Faire un second programme SMV** avec une spécification similaire. Il n'est pas nécessaire de donner le code complet du programme mais il faut en expliquer les grandes lignes et les points délicats. Vous devez montrer la cohérence entre les diagrammes de classe, statechart et le programme SMV.