

Cas IDM - étude de cas plus complexe

Yan Rabefaniraka, stagiaire de Meriem Ouederni et David Brunet

25 Juillet 2025

Contents

1	Installation des outils	1
2	CM	1
3	TPs sur Tina et Eclipse	1
3.1	TP1	1
3.2	TP2	2
3.3	TP3	2
3.3.1	Explication OCL	2
3.3.2	Écriture OCL	2
3.4	Tests à compléter avant la fin du stage	2

Remarques :

- Fichier engendré avec:

```
pandoc --toc -o ../../pdfs/resume_IDM.pdf resume_IDM.md
```

1 Installation des outils

Le changement de machine m'a obligé à réinstaller les outils du module.

- ☒ Téléchargement de Tina
- ☒ Installation de Tina
- ☒ Téléchargement de Eclipse Modeling Tools
- ☒ Installation des plugins Eclipse requis:
 - ☒ Acceleo 3.7
 - ☒ Eclipse OCL 6.17.1
 - ☒ Eclipse XTest 2.39.0
 - ☒ Sirius 7.3

-> Installation moins problématique et chronophage que sur l'ancienne machine, pourtant sous *fork* de Fedora 42 aussi.

2 CM

À compléter

3 TPs sur Tina et Eclipse

3.1 TP1

J'ai pu utiliser l'IA (Copilot avec Claude Sonnet 4) pour m'expliquer les résultats du graphe de marquage du cas d'interblocage sur *Tina*. Elle a su très bien expliquer tous les éléments pour quelqu'un comme moi qui s'y connaît

peu: Elle détaille les états, comprend correctement les transitions et explique pertinemment d'où vient l'interblocage.

Interaction avec l'IA: voir [ici](#)

3.2 TP2

Il ne semble pas y avoir de cas d'utilisation possible de l'IA; Il s'agit principalement de comprendre l'outil de travail qu'est Ecore. Qui plus est, ce dernier est graphique.

3.3 TP3

3.3.1 Explication OCL

J'ai utilisé l'IA pour comprendre le métamodèle en OCL de SimplePDL: Je lui ai demandé spécifiquement de détailler la syntaxe, dans le but de comprendre celle-ci et de pouvoir la réutiliser individuellement. Le résultat est globalement plutôt bon même si elle a du mal à comprendre l'utilité de certains invariants qui semblent pourtant relativement évident à comprendre. C'est donc impossible d'apprendre avec l'IA seule. Pour autant, il s'agit définitivement d'une bonne manière de réviser la syntaxe OCL.

Interaction avec l'IA: voir [ici](#)

3.3.2 Écriture OCL

L'une des consignes demande de créer un nouveau fichier OCL où on définit des contraintes sur *SimplePDL*. En demandant à Copilot de m'aider à ajouter de nouvelles contraintes, il analyse la structure courante du fichier SimplePDL.ocl et propose des contraintes cohérentes avec le métamodèle. L'IA comprend bien la logique des dépendances entre activités et peut suggérer des invariants pertinents, bien qu'il faille parfois corriger la syntaxe OCL spécifique ou la signification de certains **contextes**.

Interaction avec l'IA: voir [ici](#)

3.4 Tests à compléter avant la fin du stage

- ☐ Xtext - écriture et explication d'éléments avec syntaxe PDL1 et PDL2
- ☐ Sirius - syntaxe graphique
- ☐ Acceleo - Transformation de modèle à texte