Analyse croisée de modèles UML et textuels produits par des IA génératives

TER 2025-2026

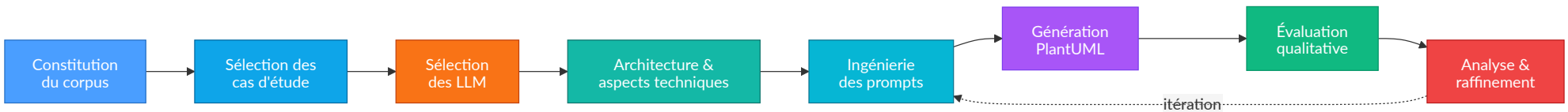
Kossi KODO · Aurivel ROGER · Diana HAPPY NYA TCHEUNKOU

Master Informatique · Université de Nantes

16 avril 2026

Présentation générale

Démarche itérative en 8 phases :



Environnement d'expérimentation

AnythingLLM

Ollama

Python

VS Code

Résultats préliminaires

Constats observés

- Cardinalités souvent erronées
- Confusion modèle / contexte
- Ajouts hors énoncé
- Meilleure qualité sur énoncés cadrés

Pistes d'amélioration

- Prétraitement des énoncés
- Post-traitement des modèles générés pour vérification et correction finale
- Approche agentique avec des SLM spécialisés

Analyse d'un run d'expérimentation LLM → UML

Expérience template — Cas demo

Run : 2026-04-16T00-00-00Z Modèle : modele-demo Évaluateur : E1

PlantUML : Valide ✓ Contenu : Non vide ✓

1. Contexte du cas

Extrait de rendu du template d'analyse de run.

2. Prompt envoyé au modèle

```
Prompt de demonstration...
```

3. Métriques de génération

Génération LLM	
Tokens prompt	1200
Tokens réponse	850
Durée (ms)	2300

Structure UML générée	
Classes	8
Interfaces	0
Relations	12
Attributs	18
PlantUML valide	Oui
Schéma vide	Non

4. Diagrammes UML

(Diagrammes non disponibles)

5. Réponse brute du modèle

```
Reponse de demonstration...
```

6. Évaluation

Chaque critère est noté sur 5. Les notes sont à remplir manuellement.

Critère	Note /5
Complétude	
Cohérence	
Qualité des relations	
Couverture des concepts UML	
Redondance (absence de)	

Critère	Note /5
Note globale *(calculée auto)*	*— en attente des 5 critères*

6.1 Complétude

Déterminer s'il manque des éléments ou s'il y en a trop.

A completer

6.2 Cohérence

Vérifier que le résultat correspond bien au sujet (périmètre).

A completer

6.3 Qualité des relations

Examiner héritage, dépendance, agrégation, association, composition — point central de l'évaluation.

A completer

6.4 Couverture des concepts UML

Mesurer la proportion des concepts UML utilisés : classes abstraites, métaclasses, notation de dérivation, etc.

A completer

6.5 Redondance

Vérifier l'absence de relations redondantes (piège courant pour les LLM).

A completer

7. Commentaire libre

A completer

8. Conclusion et pistes d'amélioration

A completer

Grille d'évaluation des diagrammes UML

Tâche : génération automatique LLM ⟶ diagramme de classes UML (PlantUML)

1. Principes généraux

Chaque run est évalué sur **cinq critères**, chacun noté de 1 à 5. Une **note globale** (/5) synthétise l'appréciation générale du diagramme.

> **Objet évalué** : le diagramme de classes UML (rendu en PlantUML) produit par le LLM en réponse à un énoncé textuel décrivant un domaine métier. Le référentiel est le corrigé fourni par les concepteurs du sujet.
>
> **Notation** : chaque note est un entier de 1 (insuffisant) à 5 (excellent). Une note de 0 n'est pas utilisée. Si le diagramme est vide ou syntaxiquement invalide, un 1 s'impose sur tous les critères.
>
> **Accord inter-évaluateurs** : deux évaluateurs notent indépendamment. L'écart absolu moyen sert de mesure de fiabilité (objectif : ≤ 0,5 point).

2. Les cinq critères et leurs niveaux

2.1. Critère 1 — Complétude

Mesure la présence de l'ensemble des classes, attributs et associations **attendus** d'après l'énoncé et le corrigé de référence. Ne punit pas les éléments supplémentaires pertinents, mais pénalise les éléments inventés sans rapport avec l'énoncé.

Note	Descripteur
1	Plus de la moitié des éléments attendus sont absents, **ou** le diagramme est vide / non compilable.
2	Au moins la moitié des éléments attendus sont présents, mais des classes ou associations importantes manquent.
3	La majorité des éléments sont présents. Quelques omissions notables (une classe secondaire ou 1–2 associations clés manquantes).
4	Quasi-complet. Seuls des éléments mineurs manquent (attributs accessoires, associations de faible importance). Aucun élément parasite significatif.
5	Tous les éléments attendus sont présents (classes, attributs essentiels, associations). Les ajouts éventuels sont pertinents et cohérents avec l'énoncé.

Tableau 1. – Critère 1 — Complétude (/5)

2.2. Critère 2 — Cohérence sémantique

Vérifie que le diagramme **correspond bien au domaine décrit** dans l'énoncé : nommage adapté, périmètre respecté (ni trop large, ni trop restreint), absence de concepts hors-sujet.

Note	Descripteur
1	Le diagramme décrit un domaine différent, ou les noms sont génériques et sans lien avec l'énoncé.
2	Le domaine est globalement reconnaissable mais de nombreux concepts sont mal nommés ou hors-sujet.
3	Les classes principales sont bien identifiées. Des incohérences de nommage ou de périmètre subsistent (ex. : classes trop génériques, termes mal adaptés).
4	Le diagramme reflète fidèlement le domaine. Les noms de classes et d'attributs sont appropriés. Quelques imprécisions mineures.
5	Le diagramme est sémantiquement très fidèle à l'énoncé. Nommage précis, périmètre exact, vocabulaire métier correctement utilisé.

Tableau 2. – Critère 2 — Cohérence sémantique (/5)

2.3. Critère 3 — Qualité des relations

Critère central. Évalue la justesse du choix et de l'orientation des relations : association, agrégation, composition, héritage, dépendance. La multiplicité (cardinalités) est également prise en compte quand elle est présente.

Note	Descripteur
1	Les relations sont absentes ou toutes incorrectes (ex. : héritages inversés, compositions confondues avec associations simples).
2	Quelques relations correctes mais confusion fréquente entre les types (agrégation/composition, héritage mal orienté). Cardinalités absentes ou fausses.
3	La moitié des relations est correctement typée et orientée. Quelques confusions (composition vs agrégation, usage abusif de l'héritage). Cardinalités partiellement correctes.
4	La majorité des relations est correcte (type, orientation, cardinalité). Erreurs ponctuelles sur les cas complexes (composition vs agrégation, généralisation multiple).
5	Toutes les relations sont correctement typées, orientées et justifiées. Les cardinalités sont cohérentes. Les choix de modélisation reflètent une compréhension fine du domaine.

Tableau 3. – Critère 3 — Qualité des relations (/5)

2.4. Critère 4 — Couverture des concepts UML

Évalue la **richesse de l'utilisation des concepts UML** : classes abstraites, interfaces, énumérations, multiplicités, visibilité des attributs, stéréotypes si pertinents. Pénalise un diagramme se limitant à des associations simples là où le domaine appelle des constructions plus riches.

Note	Descripteur
1	Aucun concept avancé utilisé. Le diagramme se réduit à des boîtes vides reliées par des traits non typés.
2	Usage minimaliste : quelques attributs, mais pas de typage, pas de visibilité, pas d'héritage ou d'interfaces là où ils s'imposent.
3	Les concepts de base sont présents (attributs typés, héritage simple). Les concepts avancés pertinents (classe abstraite, interface, énumération) sont partiellement utilisés.
4	Bonne couverture : classes abstraites, interfaces ou énumérations utilisées de façon appropriée là où le domaine le demande. Visibilité des attributs renseignée.
5	Utilisation maîtrisée et pertinente de l'ensemble des concepts UML adaptés au domaine : classes abstraites, interfaces, stéréotypes, visibilité, multiplicités précises. Aucun ajout artificiel.

Tableau 4. – Critère 4 — Couverture des concepts UML (/5)

2.5. Critère 5 — Absence de redondance

Vérifie qu'il n'y a pas de **relations ou classes redondantes** : doublons d'association, héritage coexistant avec une association équivalente, classes fusionables, attributs répétés dans les sous-classes.

Note	Descripteur
1	Redondances massives : plusieurs relations doubles, héritage et association parallèles sur les mêmes classes, classes dupliquées.
2	Redondances notables : au moins 2–3 cas distincts de relations ou classes en double.
3	Quelques redondances présentes (1–2 cas) qui nuisent à la lisibilité mais ne rendent pas le modèle incohérent.
4	Diagramme quasi sans redondance. Une relation ou un attribut dupliqué mineur peut subsister.
5	Aucune redondance. Chaque relation, classe et attribut est unique et justifié. Le modèle est parcimonieux.

Tableau 5. – Critère 5 — Absence de redondance (/5)

3. Note globale

La **note globale** est calculée automatiquement par `evaluate_results.py`. **Il n'y a rien à saisir dans cette colonne.**

Formule (par évaluateur) :

$$\text{note\_global\_en} = \frac{\text{completude} + \text{coherence} + \text{qualite\_relations} + \text{couverture\_uml} + \text{redondance}}{5}$$

Consensus :

$$\text{note\_global} = \frac{\text{note\_global\_e1} + \text{note\_global\_e2}}{2}$$

La note consensus n'est calculée que si les deux évaluateurs ont renseigné leurs cinq critères. Si un seul évaluateur a noté, `note_global` reste vide (seul `note_global_e1` ou `note_global_e2` est renseigné).

Le tableau ci-dessous est fourni à titre indicatif pour interpréter les scores moyens obtenus.

Intervalle	Interprétation indicative
1,0 – 1,9	Diagramme inutilisable : vide, invalide, ou sans rapport avec l'énoncé.
2,0 – 2,9	Très insuffisant : lacunes majeures sur la quasi-totalité des critères.
3,0 – 3,9	Passable : domaine reconnaissable, concepts de base présents, mais erreurs importantes.
4,0 – 4,9	Bon travail : diagramme exploitable et presque complet. Défauts mineurs.
5,0	Excellent : tous les critères au maximum. Proche ou équivalent au corrigé de référence.

4. Guide d'utilisation pour l'évaluateur

1. **Consulter l'énoncé** du cas avant de regarder le diagramme.
2. **Consulter le corrigé de référence** pour identifier les éléments attendus.
3. **Évaluer le diagramme LLM** critère par critère dans l'ordre : complétude → cohérence → relations → couverture UML → redondance.

4. **Saisir les 5 notes dans `evaluation.csv`** (colonnes `note_completude`, `note_coherence`, `note_qualite_relations`, `note_couverture_uml`, `note_redondance`). **Ne pas toucher à `note_global_e1` ni `note_global` — calculées automatiquement par `evaluate_results.py`.**
5. **Rédiger les commentaires textuels** dans le fichier `e1.typ` ou `e2.typ` (champs `eval_completude`, `eval_coherence`, etc.) : quelques phrases en citant des exemples précis du diagramme.
6. **Remplir le commentaire libre et la conclusion** dans le même fichier `.typ`.

Rappel accord inter-évaluateurs. Chaque run est évalué indépendamment par deux évaluateurs (remplir `e1.typ` et `e2.typ`). Ne pas se concerter avant de noter. Une réunion de calibrage sera organisée après les 10 premiers runs pour aligner les interprétations.

Exemple metadata.json

```
{
  "experiment_id": "exp_05",
  "date": "2026-03-01",
  "model": {
    "name": "gpt-oss:120b-cloud",
    "provider": "ollama",
    "workspace": "gpt-oss-120b-cloud",
    "version": "latest"
  },
  "parameters": {
    "temperature": 0,
    "top_p": 0.9,
    "max_tokens": 8000
  },
  "prompts": {
```

```
    "system_version": "v1",
    "user_version": "v2"
  },
  "status": "completed"
}
```