

ISIMA – ZZ2 F2 et F3 - Architectures Logicielles et Qualité

TD - Etude de cas – Réalisation d'un petit logiciel

La mise en œuvre d'un petit logiciel se réalise en binôme ou trinôme il donne lieu à une rédaction en 2 parties. Pour la filière 2, il s'agit de réaliser un code de simulation multi-agent (SMA : intelligence artificielle distribuée). Le temps dont vous disposez est un plus important que pour la filière 3, les attentes sont donc un peu plus grandes (vous avez un dossier d'analyse qui fait partie de votre note de simulation).

Pour la filière 3, même si un sujet de multi-agent est possible, il vous est conseillé de choisir de développer un petit code en lien avec la Business Intelligence.

Le choix spécifique du sujet est de votre ressort ainsi que le choix du langage et de l'environnement de développement. Une autonomie forte est attendue pour vous préparer au stage et augmenter votre capacité à trouver comment vous dépanner.

1) Un dossier « analyse métier » - choix de votre logiciel

Dans votre rédaction vous présenterez d'une part votre cahier des charges avec votre analyse des besoins, vos choix et spécifications, les classes et les règles retenues... Vous utiliserez UML, mais vous ne rentrerez pas encore dans la conception (la conception propose une solution le COMMENT – l'analyse n'expose que le contexte du domaine et les objectifs de votre application – le QUOI).

Vous donnerez aussi des éléments de planification des principales tâches de développement de votre développement. Comme sur un développement réel on commence petit, et on avance étape par étape en les validant au fur et à mesure. Si possible vous fournirez un diagramme de Gant final montrant les différences avec vos prévisions.

Dans la 1ère phase de rédaction (analyse), vous n'avez pas à entrer dans la conception détaillée, et encore moins dans l'implémentation. Cela fera l'objet de la rédaction de la 2ème partie de cette étude, avec une petite démonstration à mi-parcours, l'évaluation de la qualité du code source, etc.

Vous rédigerez votre analyse pour un volume de 4 à 6 pages avec diagrammes – elle est à rendre pour le jour du 1^{er} TD d'architecture logicielle et qualité pour les F2, pour le 2nd TD pour les F3 (contre élargement).

2) Partie conception et implémentation

La conception et développement concernent la deuxième partie de la rédaction. Un langage à Objet est vivement conseillé. Les contraintes pour la conception et l'implémentation sont les suivantes :

- Utilisation d'au moins 3 patrons de conception. Patrons candidats. Pour ceux qui font un SMA : le patron composite (groupe d'agent), le patron stratégie (comportement d'agents). Les patrons MVC/Observer, state et singleton peuvent être considérés ainsi que d'autres.
- Un outil de documentation automatique de type Doxygen à découvrir et à utiliser.
- Vous utiliserez au moins 2 autres outils du génie logiciel pour ce TP (cas d'un binôme), 3 en trinôme.
- La découverte de Git / GitLab est fortement conseillée (GitLab est disponible à l'ISIMA).

La rédaction finale comporte au maximum 10 pages de présentation du développement en binôme (ou trinôme).

Un complément – 5 pages est à rédiger individuellement. Il s'agit de choisir un outil du développement logiciel que vous souhaitez découvrir de façon plus spécifique. Les présentations synthétiques de ces descriptions d'outils comporteront, les points suivants :

- Résumé.
- Procédure d'installation (url, etc...).
- Présentation des fonctionnalités.
- Mini-guide d'utilisation sur les principales fonctionnalités dans le cadre du TP Multi-agents.
- Les points clés du logiciel.

Vous pourrez ensuite à la fin de cette 2ème partie présenter, dans la finale du rapport, tout ce qui a été changé par rapport à votre rendu d'analyse et de conception. Ces changements (qui sont normaux) ont induit des délais de développement, un décalage des différentes tâches que vous commenterez.

Ce petit logiciel vous initie de façon modeste à une gestion de projet de développement. Pour les binômes : une proposition vous est faite : mise en œuvre simplifiée de la méthode eXtreme Programming. Documentez-vous, il d'agit d'un cycle testeur/développeur. On parle aussi de « Test Driven Development ». Dans ce cas vous pourrez donner un pourcentage de couverture de test du logiciel. Tout comme la présentation de l'analyse, la présentation de la conception se fera avec UML.

Pour le rendu final de cette 2^{ème} partie : imprimer un dossier papier comprenant la partie conception et d'implémentation du logiciel ; l'analyse de vos résultats. Imprimer également vos présentations individuelles d'un outil du développement. Attention : éviter les copier/coller pour les présentations d'outil et citer toutes vos sources (URL). L'université dispose de l'outil « compilatio » pour l'anti-plagiat.

Pour les outils à décrire, voici une liste non exhaustive (à rafraîchir...) – un outil à décrire par personne.

Ateliers UML

Objecteering/UML Personal Edition
Poseidon for UML Community Edition
Visual Paradigm for UML Community Edition
Rational Rose
Poseidon (Argo)
Bouml
Dia
Fujaba tool suite
O mondo
...

Gestion de projets

Etude approfondie de make
Etude de cmake
Etude de Ant
Etude de Maven – (Apache Software Foundation)
Utilise un paradigme connu sous le nom de Project Object Model (POM) pour décrire un projet logiciel et ses dépendances avec des modules / bibliothèque externes
...

Gestion des versions

Git,
GitLab,
CVS (ancêtre),
Tortoise CVS
SVN, Subversion
Mercury

Etude des performances

Prof / Gprof
Jprofiler
AQtime
...

Etude des tests

CppUnit,
JUnit,
Valgrind,
Kcachegrind,
...

Logger de traces d'exécution

Log4J...

Production de documentation

Doxygen
Javadoc

Etude d'un débogueur

gdb – ddd
dbx – dbxtool
Eclipse débogueur
.Net débogueur
...

