

UE Projet de Génie Logiciel

15 février 2007

Généralités

Ce document décrit le projet donné aux étudiants dans l'UE « Projet Génie Logiciel », en L3 en 2006/2007. Les caractéristiques de ce projet sont :

- **PUBLIC** : étudiants de L3 parcours **MIAGE**, parcours **MS** (Modélisation et Simulation), M1 parcours **ISV**.
- **SPECIFIER UN SUJET (APPAREMMENT) LIBRE PUIS LE PROGRAMMER** : les étudiants spécifient un logiciel, dédié à un utilisateur de leur choix, dans un domaine de leur choix. La **liberté laissée aux étudiants sur le choix du sujet** est une caractéristique importante de ce projet. Jusque là, dans les filières suivies, les projets imposent le sujet et les étudiants programment. Dans ce projet, au contraire, les étudiants expriment une spécification, ce qui n'est pas habituel pour eux, et se mettent dans la situation d'un maître d'ouvrage devant spécifier une logiciel répondant aux besoins d'un utilisateur. Ensuite, ils programment leur spécification.
- **PLAN-TYPE DE DOCUMENT IMPOSE** : Si le sujet est libre, en revanche, la méthode est imposée. Concernant les spécifications, les étudiants doivent rendre un **document de Spécifications des Besoins du Logiciel (SBL)** suivant les plans type [2] définis dans le cours de GL.
- **ARRIERE-PLAN FONCTIONNEL IMPOSE** : Il est demandé aux étudiants de faire un logiciel qui permette à un utilisateur de réaliser des fonctions imposées: **créer, détruire, associer, dissocier, sauver, charger sur fichier, des instances de classes différentes.**
- **METHODE ORIENTEE OBJET**: UML + POO (Java ou C++) La partie de la spécification objet du SBL est faite suivant le formalisme graphique UML [5]. La programmation du projet est faite en langage orienté objet, Java ou C++, au choix des étudiants.
- **2 PHASES** : Une caractéristique de ce projet est le découpage chronologique en deux phases : les étudiants spécifie un besoin avant de le programmer. La programmation ne commence pas tant que la spécification n'est pas validée. A rendre: 1 document de spécifications (2 exemplaires), 1 programme correct (démonstration) + 1 document de bilan.
- **GROUPE DE 2 ETUDIANTS**: les étudiants s'organisent à leur manière au sein du groupe qui, à priori, est évalué globalement. Le nombre d'étudiants par groupe est 2. Si un groupe comprenait 3 étudiants, le groupe sera noté sur 14 seulement. A priori, la note est commune aux étudiants d'un même groupe. Cependant, si un dysfonctionnement est observé dans le travail du groupe, ou si le travail n'a pas été fait équitablement entre les membres du groupe, la note ne sera pas commune, mais individuelle.
- **INTERFACE HOMME-MACHINE, RESEAU, BASES DE DONNEES**: le but de ce projet **n'est pas la mise en oeuvre de techniques informatiques particulières** (réseaux, bases de données, IHM graphique). La partie fonctionnelle imposée (créer, détruire, associer, dissocier, sauver, charger sur fichier, des instances de classes différentes) doit être impérativement programmée en orienté objet **sans outil de gestion de base de données**. Il est conseillé de programmer une **IHM textuelle style « console »** et pas une IHM graphique. Néanmoins, pour ne pas nuire à la liberté des spécifications, il n'est pas totalement interdit de spécifier un logiciel utilisant des techniques informatiques particulières.

Organisation pratique

Le projet se déroule sur l'ensemble de l'UE de février à mai 2007. La réunion de présentation a lieu le **vendredi 16 février 2007 à 18h30**.

Spécification

Pour la phase de spécification, 4 sous-phases sont prévues. Chaque sous-phase se traduit par un document à

rendre à Bruno Bouzy (soit dans le casier courrier au 4ème étage soit en main propre). L'encadrement est fait, dans le bureau de l'enseignant, soit par Bruno Bouzy, soit par un enseignant désigné par Bruno Bouzy.

- Sous-phase 1 : choix du sujet : utilisateur du L ? diagramme de classes et diagramme de généralisation (document n°1, 1 page) **à rendre le lundi 5 mars 2007** au plus tard.
- Sous-phase 2 : spécification d'interfaces : 4 à 5 cas d'utilisations, diagramme d'instances « avant » « après » (document n°2, 4 à 5 pages) **à rendre le lundi 12 mars 2007** au plus tard.
- Sous-phase 3 : spécification objet complète : description des classes : attributs + méthodes publiques (document n°3, 10 à 15 pages) **à rendre le lundi 19 mars 2007** au plus tard.
- Sous-phase 4 : remise du SBL complet (document n°4 = SBL, 20 à 25 pages) en 2 exemplaires pour le **lundi 26 mars 2007** au plus tard. Pénalité de 1 point par jour de retard.

Programmation

Les étudiants travaillent chez eux ou en salle machine. Le langage de développement est Java ou C++. L'environnement de développement est libre. La phase de programmation se termine par la soutenance sur machine.

Soutenances sur machine

Soutenance **sur machine de 20 à 25 minutes** pendant la première session d'examen. Remise d'un **document de bilan** présentant les résultats atteints effectivement et comparant ces résultats avec les spécifications prévues dans le SBL (document n°5 = SBL, 5 à 10 pages). La soutenance se passe en deux temps. D'abord (10 à 15 minutes) côté test et utilisation, l'accent est mis sur la **correction du programme** présenté en relation avec les fonctions imposées. Attention, un programme qui marchait la veille de la soutenance, mais pas le jour de celle-ci est très mal noté. Les étudiants doivent préparer ces tests afin de valoriser au mieux leur travail. Ensuite (5 à 10 minutes), des **questions précises sur la programmation** du logiciel sont posées aux membres du groupe.

Notation

Note Projet = 1/5 Spécification + 4/5 (ProgrammationBilan)

Lien avec l'UE de Génie Logiciel

Un étudiant ne peut pas faire l'UE « projet de GL » s'il n'est pas inscrit à l'UE de « programmation 6 ». L'inverse est possible : on peut, même si c'est dommage pédagogiquement dommage, faire l'UE de « programmation 6 » sans faire le projet. Par ailleurs, un étudiant doit savoir programmer en orienté objet. L'UE de « programmation 5 » est un pré-requis conseillé à l'inscription à l'UE « projet de GL ». Sans ce dernier pré-requis, l'expérience prouve que les étudiants échouent à l'UE « projet de GL ».

Lien avec les TPs

Le TP de l'UE de GL fait programmer aux étudiants un mini-projet, le projet STADE, dont la nature ressemble à ce qui est demandé dans le vrai projet : faire un logiciel qui permette à un utilisateur de créer, détruire, associer, dissocier, sauver, charger sur fichier, des instances de classes différentes [3, 4]. Le mini-projet STADE comprend 3 classes. Le vrai projet demande une spécification de 6 à 10 classes. Un étudiant ayant effectué complètement le mini-projet STADE donne la garantie de ne pas avoir de difficulté technique pour la programmation du vrai projet. Il est donc impératif de ne pas commencer la programmation du vrai projet avant d'avoir terminé le mini-projet STADE. L'encadrement des TPs du vrai projet ne doit pas correspondre à une aide déjà offerte lors des TPs du mini-projet STADE.

Lien avec les TDs

Le cours et les TDs de l'UE de GL abordent principalement UML. Le SBL utilise largement UML, en particulier les notions présentées lors des séances 3, 4, 5 et 6 du cours.

Documents associés

- [1] Emploi du temps UE GL + projet – documents Excel
- [2] Plans-type de documents : <http://www.math-info.univ-paris5.fr/~bouzy/Doc/Plan-TypesDocuments.pdf>
- [3] Sujet du TP STADE en C++ : <http://www.math-info.univ-paris5.fr/~bouzy/Doc/TPC++pdf>
- [4] Sujet du TP STADE en Java : <http://www.math-info.univ-paris5.fr/~bouzy/Doc/TPJava030527.pdf>
- [5] autres documents (cours GL, UML, etc.) accessibles à <http://www.math-info.univ-paris5.fr/~bouzy/enseign.html>