

Spécifications pour le module de gestion des UEs pour la forge UCBL

E. Coquery

1 Introduction

Le module UE de la forge UCBL a pour objectif de faciliter la gestion d'un ensemble de projets (au sens forge) correspondant à une série de travaux pratiques, de projets (au sens pédagogique), etc rattachés à une unité d'enseignement (UE).

2 Acteurs

Cette section définit la liste des acteurs humains et logiciels interagissant avec le module, avec les dites interactions. On y décrit également certaines interactions avec les projets de la forge entrant dans le cadre de l'UE.

2.1 UE-Semestre

Ensemble des informations rattachées à une UE pour un semestre particulier. Voir également la discussion sur le nombre de projets par UE (5.1).

Interactions

- Initialisation (3.3.1).
- Ajout d'étudiants (3.1).
- Définition des groupes (3.2).
- Définition des administrateurs (3.4.2).
- Définition des intervenants (3.4.1).

2.2 Responsable d'UE

Les responsables de l'UE tels que définis dans le SI de l'université¹. Tout responsable d'UE est également administrateur.

Interactions

- Initialisation pour un nouveau semestre (3.3.1).
- Déclaration / révocation des administrateurs de l'UE pour ce semestre (3.4.2).

2.3 Administrateur de l'UE

L'administrateur est une personne qui assure la gestion de l'UE, en ce qui concerne la forge. Cette tâche peut être déléguée par un responsable de l'UE. Les administrateurs sont également des intervenants.

1. dans Spiral ? Voir avec T.Excoffier

Interactions

- Déclarer/révoquer les intervenants (3.4.1).
- Gérer les groupes d'étudiants. Voir la section 3.2 sur la gestion des groupes d'étudiants.
- Déclencher un *push* du repository de l'UE vers ceux des groupes (3.3.3).
- Déclencher un *pull* des repositories des groupes vers celui de l'UE (3.3.4).
- Définir une date de *pull* automatisé des repositories des groupes vers celui de l'UE (3.3.5).
- Définir les permissions d'accès aux différentes pages et autres documents pour les étudiants de l'UE (3.4.3).
- Inscrire manuellement un étudiant à l'UE (3.1.1).
- Mettre à jour les étudiants nouvellement inscrits à l'UE (3.1.1).

2.4 Intervenant

Un membre de l'équipe enseignante de l'UE pour ce semestre. Tout administrateur est également intervenant. Un intervenant ne participe pas à proprement parler à la gestion de l'UE, mais dispose de droits sur les projets.

Remarque: Ce type d'acteur n'a d'intérêt que dans le cadre de grosses UEs.

Interactions

- Accéder en lecture/écriture au projet de l'UE. Lié à 3.4.1.
- Consulter la liste des groupes d'étudiants (3.1.3, 3.2.1).
- Accéder en lecture aux projets des groupes Lié à 3.4.1 et 3.3.2.
 - *Optionnellement* : laisser des commentaires dans les projets des groupes.

2.5 Étudiant

Un étudiant est un étudiant de l'université qui est inscrit à l'UE. L'inscription est normalement automatique, c'est à dire que la liste des étudiants inscrits est extraite du SI de l'université². Une inscription manuelle est néanmoins possible par un administrateur de l'UE. Un étudiant a vocation à faire partie d'un groupe d'étudiants.

Interactions

- Lire les pages "publiques" du projet de l'UE-Semestre. Lié à 3.4.3.
- Accéder en lecture/écriture aux projets de ses groupes. Lié à 3.3.2.
- Participer à la création des groupes, voir 3.2.7.

2.6 Groupe d'étudiants

Ensemble d'étudiants ayant des permissions communes pour accéder à certains projets. La définition des groupes d'étudiants peut varier (voir 3.2). Un groupe d'étudiants peut être "en cours de construction", "validé" ou bien "désactivé".

Interactions

- Groupe dans l'état "en cours de construction" :
 - Modification des membres du groupe : dépend du mode de création de groupe (3.2.7)
 - Passage dans l'état "validé" (3.2.4)
- Groupe dans l'état "validé" :

2. via apogee, LDAP, TOMUSS ?

- Initialisation d'un projet forge associé au groupe (3.3.2). Fait automatiquement à la validation, mais voir la discussion 5.1.
- Passage dans l'état "désactivé" (3.2.5).
- Groupe dans l'état "désactivé" : rien ?

2.7 Projet de l'UE-Semestre

Créé via l'initialisation d'un nouveau semestre (mais voir 5.1).

Interactions

- Définition/modification des permissions d'accès aux éléments du projet (3.4.3).
- Déclenchement des actions de push/pull entre repository de l'UE-Semestre et repositories des groupes associés 3.3.3, 3.3.4, 3.3.5.
- Duplication partielle pour initialisation d'un nouveau projet (5.1).

2.8 Projet d'un groupe

Projet de la forge rattaché à un groupe d'étudiants.

Interactions

- Définition/modification des permissions d'accès aux éléments du projet 3.3.2, 3.4.1.
- Déclenchement des actions de push/pull entre repository de l'UE-Semestre et repositories des groupes associés 3.3.3, 3.3.4, 3.3.5.
- Passage en lecture seule suite à désactivation d'un groupe 3.2.5.

3 Interactions

Cette section détaille les différentes interactions autour du module.

3.1 Gestion des étudiants

Il s'agit essentiellement d'ajouter des étudiants à la liste des étudiants rattachés à une UE-Semestre.

Remarque: La liste des étudiants inscrits peut différer de celle provenant du SI de l'université (Apogee) car on peut vouloir gérer des étudiants qui ne sont pas inscrits pédagogiquement à l'UE-Semestre. On peut évoquer d'éventuel auditeurs libres', ou encore des étudiants dont l'inscription pédagogique n'est pas finalisée.

3.1.1 Import automatique d'étudiants inscrits

Il s'agit d'ajouter à l'UE-Semestre les étudiants qui y sont inscrits pédagogiquement dans le SI. On peut réfléchir à automatiser ces inscription (comme dans TOMUSS).

3.1.2 Import manuel d'étudiants

Il s'agit d'ajouter à l'UE-Semestre un ou plusieurs étudiants. On peut imaginer fournir une liste de numéros d'étudiants. On peut également imaginer un sélecteur "à la Spiral" avec un système de requêtes pour sélectionner les étudiants à ajouter. Voir également la discussion autour de Spiral (5.2).

3.1.3 Consultation de la liste des étudiants

Liste des étudiants, avec login, nom/prénom et liste des groupes auxquels ils appartiennent.

3.2 Gestion des groupes d'étudiants

Paramétrisation

- Plugins de génération de groupes → 3.2.7.
- Plugins de contraintes sur les groupes → 3.2.2.

3.2.1 Consultation de la liste des groupes

Liste des groupes, avec leur statut, leur identifiant/tag mercurial (*c.f.* 3.3.4), la liste des membres, un lien vers leur projet.

3.2.2 Définition de contraintes sur les groupes

Permet de définir des contraintes sur la formation des groupes. La première contrainte est la taille min/max. Mais on peut en imaginer d'autres. Dans ce cadre, un système de plugins serait le bienvenu.

3.2.3 Crédation d'un groupe par l'administrateur

L'administrateur fourni un ensemble de numéros d'étudiants (rattachés à l'UE). Un groupe est constitué et validé (3.2.4).

Les numéros peuvent être fournis dans un champ texte dans un formulaire : un numéro par ligne permet un copier/coller depuis un tableau.

On peut également imaginer mettre des cases à cocher dans la liste des étudiants (en filtrant optionnellement sur l'appartenance à un groupe) avec une action “créer un groupe”.

3.2.4 Validation d'un groupe

Passe le groupe dans l'état “validé” et initialise un projet associé (3.3.2).

3.2.5 Désactivation d'un groupe

Passe le groupe dans l'état “désactivé”. Les (ex) membres du groupes ne sont plus considérés comme faisant partie du groupe en question³. Le projet lié au groupe passe en lecture seule.

3.2.6 Modification d'un groupe

Ajoute ou supprime un étudiant dans un groupe. Le groupe peut être “validé” auquel cas les permissions sur le projet associé sont mises à jour. Cette action est surtout destinée à traiter quelques cas problématiques et ne devrait pas être utilisée pour constituer les groupes.

3.2.7 Modes de création de groupes

On peut envisager différents modes de création de groupes.

- À la main par les administrateurs de l'UE, voir 3.2.3.
- Import d'un fichier CSV. Chaque ligne contient par exemple un numéro d'étudiant et un nom de groupe.

3. ni d'aucun groupe d'ailleurs

- Par enregistrement des étudiants. Plusieurs possibilités :
 - Chaque étudiant forme un groupe à lui seul. Un groupe peut décider de fusionner avec un autre (modulo les contraintes). La fusion peut être soit entérinée par un membre de chacun des deux groupes, soit par tous les membres des deux groupes. A la fin, validation par tous les membres du groupe.
 - Un étudiant peut créer un nouveau groupe ou rejoindre un groupe existant. Une fois le groupe finalisé il est validé par tous les membres.
- Génération aléatoire.

Remarque: Attention aux interactions avec les plugins de contraintes sur les groupes. Restreindre éventuellement aux contraintes de taille.

3.3 Gestion des projets

3.3.1 Initialisation d'une UE-Semestre

L'initialisation d'une UE-Semestre consiste à :

1. Créer les données internes au module de gestion des UEs pour gérer cette UE-Semestre.
2. Créer au niveau de la forge un projet pour cette UE-Semestre. Préciser vis-à-vis de la discussion sur le nombre de projets par UE-Semestre (5.1).
3. Déclarer les (permissions des) responsables d'UE pour cette UE-Semestre et pour le projet associé.
4. Les étudiants inscrits à l'UE-Semestre sont importés depuis le SI.

3.3.2 Initialisation d'un projet pour un groupe

Lorsqu'un groupe est validé, un projet créé pour ce groupe. Les permissions en lecture/écriture sont donnée à tous les étudiants du groupe. Les permissions en lecture sont données à tous les intervenants.

3.3.3 Push UE → groupes

Déclenche une synchronisation du repository de l'UE-Semestre vers ceux des groupes d'étudiants validés⁴. Voir également la discussion sur les push/pull partiels (5.3). Normalement mercurial⁵ gère ça bien (de par le principe du gestionnaire de versions distribué).

3.3.4 Pull UE ← groupes

Déclenche une synchronisation du repository de chaque groupe d'étudiant vers le repository de l'UE. Il faudrait une branche par groupe étiquetée de façon à pouvoir facilement faire *checkout* du repository pour un groupe particulier.

Problèmes éventuels Si on autorise plusieurs phases de *push/pull* UE ↔ groupes, une groupe *A* peut avoir accès au code d'un groupe *B* via le deuxième *push*. Plusieurs réponses :

- N'autoriser qu'un seul *push*. Le désavantage est que le contenu pédagogique ne peut être changé après le *push* initial.
- Faire les *pull* vers un autre repository. Cela implique d'avoir plusieurs repository attachés à une UE-Semestre.

4. mais pas désactivés

5. ou git

3.3.5 Date de pull

Il s'agit de définir des dates/heures auxquelles l'action *pull* UE ← groupes (3.3.4) est déclenchée. L'utilisation principale est le rendu de devoir/projet, mais on peut également imaginer des pulls réguliers pour faire du suivi.

Ce sont les administrateurs qui définissent ces dates. En termes d'interface, une liste de dates avec action de suppression (si la date n'est pas passée) et un formulaire d'ajout devraient suffire.

3.4 Gestion des rôles et des permissions

3.4.1 Gestion des intervenants

Un administrateur doit pouvoir ajouter et supprimer des intervenants. On peut imaginer un champ texte simple dans lequel on saisit le login CAS d'un nouvel intervenant. On dispose également d'une liste des intervenants avec une action de suppression.

Propager les définitions d'intervenants au projet UE-Semestre. En particulier, les intervenants ont les droits de lecture/écriture sur le contenu du projet UE-Semestre.

Propager les définitions d'intervenants aux projets des groupes : accès en lecture + optionnellement droit de laisser des commentaires.

3.4.2 Gestion des administrateurs

Première version Un responsable d'UE doit pouvoir ajouter et supprimer des administrateurs. On peut imaginer un champ texte simple dans lequel on saisit le login CAS d'un nouvel administrateur. On dispose également d'une liste des administrateurs avec des actions de suppression ou de transformation en intervenant.

Deuxième version La notion d'administrateur est un simple rôle pour un intervenant. On prévoit une case à cocher lors de la création d'un intervenant, ainsi que dans la liste des intervenants. Il faudra bien vérifier les permissions : seul le responsable d'UE peut changer les administrateurs.

3.4.3 Permissions d'accès aux pages

Définition des permissions d'accès aux pages du projet d'une UE-Semestre. En particulier définir quelles pages sont accessibles aux étudiants. *A priori, cela devrait être pris en charge par la forge.*

4 Scénarii

Description de quelques scénarii pour aider à la conception.

4.1 Projet en binôme dans le cadre des TPs d'une UE

On considère une UE dans laquelle l'ensemble des TPs réalisés forment un projet cohérent. On imagine une UE à faible effectif (environ 20 étudiants).

Le responsable de l'UE commence par initialiser un projet pour l'UE-Semestre. La liste des étudiants inscrits à l'UE est alors importée depuis le SI. La taille est groupes est définie comme égale à 2. Le sujet du premier TP est alors déposé parmi les documents du projet de IUE-Semestre. Il met également dans le repository de l'UE un squelette de projet (par exemple un fichier C++ et un Makefile, ou encore un archetype maven). Lors de la première séance de TP, les étudiants s'installent à deux pour une machine. Au sein d'un binôme, l'étudiant connecté se rend sur la

forge, sur la page de l'UE-Semestre. Il crée un nouveau groupe ayant pour autre membre sont binôme et valide pour sa part le groupe. Le second étudiant se connecte alors et valide à son tour le groupe, ce qui valide définitivement ce dernier et crée le projet forge associé. Lors de cette phase un étudiant se retrouve seul. Il crée alors un groupe à lui seul. Le responsable d'UE valide alors ce groupe (qui ne peut être validé par l'étudiant car il viole la contrainte de taille). Lors de la seconde séance un nouvel étudiant arrive. Cet étudiant n'étant pas encore inscrit pédagogiquement, le responsable d'UE l'inscrit. Il modifie au passage le groupe de l'étudiant seul de façon à former un binôme avec le nouvel arrivant. Entre la deuxième et la troisième séance, l'enseignant programme une bibliothèque de fonctions pour aider au projet. La bibliothèque est déposée dans le repository de l'UE et transmise aux étudiants via un *push* UE → groupes. Au long des séances de TP, les étudiants synchronisent régulièrement leur travail avec le repository du projet de leur groupe. Vers la fin du semestre, l'enseignant définit une date de rendu des TPs et déclare une date de *pull* UE ← groupes. Pour corriger les TPs, il clone le repository de l'UE et corrige le TP de chaque groupe en changeant de branche.

4.2 Série de TPs réalisés en binôme

On considère une UE d'effectif assez important (80 étudiants) comportant une série de TPs réalisés en binômes. Ce n'est pas le responsable de l'UE qui supervise directement les TPs, aussi, après avoir initialisé l'UE-Semestre, déclare-t-il un autre enseignant comme administrateur. C'est ce dernier qui va gérer de manière effective les TPs. Il commence par déclarer quelques autres enseignants comme intervenants. L'équipe ainsi constituée commence la rédaction de sujets de TPs. Ces sujets n'étant pas finalisés et certains d'entre eux correspondant à un contrôle en séance, il ne sont pas rendus publics. Les inscriptions pédagogiques ayant du retard, les étudiants sont inscrits manuellement via un copier/coller un tableur. Dans le repository de l'UE, chaque séance de TP est représentée par un répertoire. Les binômes sont créés lors de la première séance de TP comme dans le scénario 4.1. Selon les séances, les étudiants doivent soit rendre le TP en fin de séance, soit disposer d'un délai pour le rendu. Dans les deux cas le rendu est effectué via un *pull* UE ← groupes à la date voulue.

4.3 UE comportant un projet à quatre ou cinq étudiants

Dans le cadre de cette UE, les étudiants doivent réaliser un projet par groupe de 4 à 5 étudiants. On considère que les groupes sont formés lors d'une réunion de départ et saisis par le responsable via copier/coller depuis un tableur. Les étudiants de chaque groupe collaborent via la forge. Chaque groupe réalisant un projet différent, il n'y a pas de *push* UE → groupes. Le rendu est constitué par un ensemble de documents déposés sur la forge, par un wiki et par une réalisation logicielle dont les sources sont dans le repository du projet de chaque groupe.

5 Discussions

5.1 Nombre de projets par groupe

Autorise-t-on plusieurs projets par groupe, par exemple un projet par séance de TP ? Cela semble compliquer les choses en termes de gestion de la forge. Cela permet en revanche de bien séparer les différents sujets de TP dans une UE.

On peut envisager d'avoir plusieurs projets d'UE par semestre et d'avoir une fonctionnalité de duplication partielle de projet. Par exemple, on pourrait garder les groupes d'étudiants et l'équipe enseignante.

Alternativement, on peut imaginer créer des nouveaux projets pour chaque groupe valide et *invalider* automatiquement les anciens projets. Cela ne semble cependant pas très cohérent avec le fait de s'appuyer sur le même repository d'UE.

5.2 Connection à Spiral

Essaye-t-on de connecter le module UE à Spiral, en allant au delà d'une simple récupération de données (e.g. les responsables d'UE) ? On pourrait par exemple connecter une UE-Semestre à un module Spiral et synchroniser les enseignants, les étudiants. Quoi d'autre ?

5.3 Push/pull partiels

Il pourrait être intéressant de permettre de ne synchroniser qu'une partie (une branche, une répertoire ?) du repository d'une UE afin de laisser un espace d'échange pour l'équipe pédagogique, typiquement une correction type.