

Oto : Un outil générique et évolutif d'aide à la correction de programmes — Cas d'utilisation

Note interne de recherche

Département d'informatique, UQAM

G. Tremblay et F. Guérin

Juin 2004

1 Introduction

Ce document vise à définir les principaux cas d'utilisation du système SysOto, un outil générique et évolutif d'aide à la correction de programmes. L'objectif premier est de mieux définir la portée du système — qu'est-ce qui fera partie du système, qu'est-ce qui n'en fera pas partie. L'autre objectif est de nous permettre ensuite de mieux définir les principales opérations système, c'est-à-dire spécifier l'interface (au sens de Jackson [?]) avec le *noyau* de SysOto.

Signalons que la forme du présent document a été largement influencée par l'approche suggérée par Cockburn dans son excellent manuel sur les cas d'utilisation [?, ?]. Pour représenter les différents niveaux et portées, les symboles suivants ont été utilisés :

Portée	Entreprise	Système	Composant
	$\hat{\square}$	\square	n/a
Niveau	Nuage	Mer	Poisson
	∞	\approx	$\triangleright\odot$

2 Portée du système

Item	In	Out
L'utilisateur n'a pas de compte sur le système d'exploitation		Out
Remise du listing ou de la documentation		Out

Table 1: Liste des items qui sont In/Out

Acteur	Objectif niveau tâche	Priorité
Étudiant	Vérifier une solution	1
	Remettre un travail	1
	Demander de l'information sur un travail déjà remis	2
Enseignant	Définir une évaluation	1
	Définir un nouveau groupe-cours	2
	Modifier une évaluation existante	2
	Détruire une évaluation existante	2
Correcteur	Corriger l'ensemble des travaux remis	1
	Corriger un travail spécifique	1

Table 2: Liste des acteurs et de leurs objectifs

Acteur	Objectif/tâche	Résumé
Étudiant	Vérifier une solution	L'étudiant fournit des informations identifiant le travail, puis soumet son travail. SysOto lui donne du feedback.
	Remettre un travail	L'étudiant fournit des informations identifiant le travail, puis soumet son travail. SysOto lui retourne une confirmation.
	Demander de l'information sur un travail déjà remis	L'étudiant fournit des informations identifiant son équipe. Si une telle équipe existe, SysOto lui indique les membres de l'équipe et la date de la remise précédente.
Enseignant	Définir une évaluation	L'enseignant spécifie l'évaluation. SysOto lui confirme la création de l'évaluation.
	Définir un groupe-cours	L'enseignant spécifie un groupe-cours (liste d'étudiants, correcteur, etc.). SysOto lui confirme la création du groupe-cours.
	Modifier une évaluation	L'enseignant modifie certains attributs d'une évaluation. SysOto lui confirme les modifications effectuées.
	Détruire une évaluation	L'enseignant indique une évaluation. SysOto la détruit.
Correcteur	Corriger l'ensemble des travaux remis	Le correcteur identifie l'évaluation à traiter. SysOto corrige les travaux remis puis produit un rapport global de correction.
	Corriger un travail spécifique	Le correcteur identifie l'évaluation à traiter et le travail à corriger. SysOto corrige le travail puis produit un rapport de correction.

Table 3: Cas d'utilisation résumés

3 Cas d'utilisation sommaire

Cas n° 1 [△] Définir un travail et le corriger [∞]

Portée: Université

Niveau: Sommaire

Acteurs: Enseignant, Étudiants, Correcteur

Préconditions: L'enseignant enseigne un cours et veut donner un travail aux étudiants puis le faire corriger par SysOto

Garanties de succès:

Scénario nominal :

1. L'enseignant conçoit son travail et définit les scripts qui seront utilisés pour la vérification et la correction
2. L'enseignant définit une évaluation
3. SysOto confirme la création de l'évaluation et rend disponible aux étudiants l'évaluation ainsi définie
4. Les étudiants du cours vérifient leur solution
5. Avant la date limite, les étudiant remettent leur travail et SysOto conserve les travaux remis
6. Le correcteur corrige l'ensemble des travaux remis par les étudiants d'un groupe-cours
7. L'enseigne utilise le rapport comme l'un des éléments d'évaluation des travaux remis (en plus du *listing*, de la documentation, etc., qui ont été remis en main propre)

4 Spécification des cas d'utilisation

4.1 Enseignant

Cas n° 2 □ **Définir une évaluation** ≈

Portée: SysOto

Niveau: Usager

Acteur: Enseignant

Préconditions: L'enseignant a défini un script de vérification et un script de correction

Garanties de succès: Les scripts sont installés et potentiellement utilisables par les étudiants et les correcteurs (en fonction des dates spécifiées)

Déclencheurs: L'enseignant est prêt à utiliser SysOto pour définir une évaluation pour un travail dans un cours

Scénario nominal :

1. L'enseignant s'identifie.
2. SysOto vérifie que l'enseignant peut utiliser le système.
3. L'enseignant spécifie le groupe-cours.
4. SysOto valide le groupe-cours.
5. L'enseignant spécifie le nom de la nouvelle évaluation, les dates d'utilisation, les scripts associés.
6. SysOto valide que l'évaluation n'existe pas.
7. SysOto installe les scripts.
8. SysOto envoie une confirmation à l'enseignant.

Extensions :

2a L'enseignant n'est pas reconnu comme un enseignant valide :

- .1. L'enseignant doit contacter l'analyste de la machine arabica pour se faire créer un compte du groupe professeur.

4a Le groupe-cours n'existe pas :

- .1. L'enseignant définit un nouveau groupe-cours.

6a Une évaluation du même nom existe déjà :

- .1. L'enseignant fournit un nouveau nom d'évaluation.

6b Les dates spécifiées ne sont pas valides :

- .1. L'enseignant fournit de nouvelles dates.

6c Les fichiers indiqués pour les scripts ne semblent pas exister :

- .1. L'enseignant fournit de nouveaux noms de fichiers pour ses scripts.

4.2 Étudiant

Cas n° 3 □ Vérifier une solution ≈

Portée: SysOto

Niveau: Usager

Acteur: Étudiant

Préconditions: Une évaluation a été créée par l'enseignant et l'étudiant a produit un début de solution

Garanties de succès: L'étudiant est informé si sa solution satisfait minimalement aux exigences de l'évaluation définie par l'enseignant

Déclencheurs: L'étudiant décide de faire évaluer son travail

Scénario nominal :

1. L'étudiant s'identifie.
2. SysOto authentifie l'étudiant.
3. L'étudiant spécifie l'évaluation (enseignant, groupe-cours, travail).
4. SysOto valide que l'évaluation existe et qu'une vérification peut encore être effectuée.
5. L'étudiant fournit sa solution (par ex., fichier jar zippé).
6. SysOto valide l'existence du fichier, de son format, ainsi que sa taille.
7. SysOto, à l'aide du script spécifié par l'enseignant, vérifie la solution de l'étudiant et retourne un rapport (*feedback* immédiat).

Extensions :

2a L'étudiant n'a pas de compte sur arabica :

- .1. SysOto indique à l'étudiant qu'il doit contacter son enseignant.

4a Le travail indiqué n'existe pas :

- .1. SysOto demande à l'étudiant de fournir à nouveau les informations appropriées.

4b L'étudiant n'est pas inscrit dans le groupe-cours :

- .1. SysOto demande à l'étudiant de fournir à nouveau les informations appropriées.

4c Aucun script d'évaluation n'a été défini par l'enseignant pour ce travail :

- .1. SysOto informe l'étudiant et lui suggère de vérifier auprès de son enseignant.

4d La date limite pour évaluation est dépassée :

- .1. SysOto en informe l'étudiant.

6a Le fichier indiqué n'existe pas :

- .1. L'étudiant spécifie un nouveau nom de fichier.

6b La solution fournie par l'étudiant n'est pas dans un format correct :

- .1. SysOto informe l'étudiant du format requis pour cette évaluation. L'étudiant génère le fichier approprié puis resoumet à nouveau.

6c La taille du fichier dépasse nettement la taille permise par l'enseignant pour ce travail :

- .1. SysOto informe l'étudiant de la taille maximal permise pour ce travail. L'étudiant génère un nouveau fichier puis le resoumet.

Cas n° 4 [□] Remettre un travail ≈

Portée: SysOto

Niveau: Usager

Acteur: Étudiant

Préconditions: Une évaluation a été créée par l'enseignant et l'étudiant est prêt à remettre sa solution

Garanties de succès: Le travail est déposé dans l'espace Oto de l'enseignant

Déclencheurs: L'étudiant décide de remettre son travail

Scénario nominal :

1. L'étudiant s'identifie.
2. SysOto authentifie l'étudiant.
3. L'étudiant spécifie l'évaluation (enseignant, groupe-cours, travail).
4. SysOto valide que l'évaluation existe et qu'un travail peut encore être remis.
5. L'étudiant identifie les membres de l'équipe.
6. SysOto confirme que l'équipe est valide : les étudiants sont inscrits dans le groupe-cours et pour l'ensemble des membres de l'équipe, l'intersection avec les équipes déjà connues est vide (i.e., première remise) ou il existe une équipe qui est identique ($n^{ième}$ remise).
7. L'étudiant fournit sa solution (par ex., fichier jar zippé).
8. SysOto valide l'existence du fichier, de son format, ainsi que sa taille.
9. SysOto conserve la solution remise et prend en note l'équipe associée. Ensuite, il retourne une confirmation.

Extensions :

4abcd Voir les extensions 4abcd du cas d'utilisation Vérifier une solution.

5a L'étudiant, dans le cas d'une remise additionnelle, a oublié l'identité des membres de son équipe :

.1. L'étudiant demande l'information sur son travail.

6a Les membres indiqués pour l'équipe ne correspondent pas à une équipe déjà existante (certains des membres indiqués, mais pas tous, sont déjà membres d'une autre équipe qui a remis un travail)

.1. SysOto informe l'étudiant du problème.

.2. L'étudiant demande l'information sur son travail.

8abc Voir extensions 6abc du cas d'utilisation Vérifier une solution.

Cas n° 5 □ **Demander de l'information sur un travail déjà remis** ≈

Portée: SysOto

Niveau: Usager

Acteur: Étudiant

Préconditions: Une évaluation a été créée par l'enseignant et l'étudiant pense avoir déjà remis un travail

Garanties de succès: L'étudiant est informé de l'existence d'un travail préalablement remis et des membres de l'équipe qui ont remis ce travail

Déclencheurs: L'étudiant décide de remettre un travail

Scénario nominal :

1. L'étudiant s'identifie.
2. SysOto authentifie l'étudiant.
3. L'étudiant spécifie l'évaluation (enseignant, groupe-cours, travail).
4. SysOto valide que l'évaluation existe pour ce travail.
5. SysOto indique si un travail pour cette évaluation a déjà été remis par une équipe dont fait partie l'étudiant, et retourne l'identité des autres membres de l'équipe.

4.3 Correcteur

Cas n° 6 [□] Corriger l'ensemble des travaux remis \approx

Portée: SysOto

Niveau: Usager

Acteur: Correcteur

Préconditions: Une évaluation a été créée par l'enseignant, la date de remise est passée et des correcteurs ont été désignés par l'enseignant

Garanties de succès: Un rapport est produit qui comprend un compte rendu sur chacun des travaux remis

Déclencheurs: Le correcteur décide de corriger les travaux

Scénario nominal :

1. Le correcteur s'identifie et spécifie l'évaluation (enseignant, groupe-cours, travail).
2. SysOto valide que l'évaluation existe et que le correcteur a les permissions appropriées (pour le groupe-cours).
3. Pour chacun des travaux remis en lien avec cette évaluation, SysOto effectue la correction du travail à l'aide de l'évaluation appropriée et produit un rapport pour ce travail.
4. SysOto produit un rapport global portant sur l'ensemble des travaux remis.

Extensions :

- 2a L'évaluation n'existe pas :
- 2b Le correcteur n'a pas les permissions pour corriger cette évaluation.

Cas n° 7 [□] Corriger un travail spécifique \approx

Portée: SysOto

Niveau: Sous-fonction

Acteur: Correcteur

Préconditions: Une évaluation a été créée par l'enseignant, la date de remise est passée, des correcteurs ont été désignés par l'enseignant et des travaux ont été remis par les étudiants. Le correcteur a aussi été correctement identifié (authentification réussie).

Garanties de succès: Un rapport est produit qui comprend un compte rendu sur le travail indiqué

Déclencheurs: Le correcteur décide de corriger un travail spécifique (par ex., remise tardive, problème spécial avec le logiciel de remise, etc.) et s'est déjà identifié

Scénario nominal :

1. Le correcteur identifie l'étudiant dont le travail doit être corrigé.
2. SysOto valide qu'un travail existe pour cet étudiant.
3. SysOto effectue la correction du travail à l'aide de l'évaluation appropriée et produit un rapport pour ce travail.

Références

- [Coc01a] A. Cockburn. *Rdiger des cas d'utilisation efficaces*. Eyrolles, 2001. [QA76.9O35.C6314.2001].
- [Coc01b] A. Cockburn. *Writing Effective Use Cases*. Addison-Wesley, 2001.
- [Jac95] M. Jackson. *Software Requirements & Specifications — a lexicon of practice, principles and prejudices*. ACM Press & Addison-Wesley, 1995.