

Une expérience de notation en masse

Christian Queinnec* & Emmanuel Chailloux**

Université Paris 6 — Pierre et Marie Curie

4, place Jussieu 75252 Paris Cedex 05

*LIP6: Christian.Queinnec@lip6.fr

**PPS: Emmanuel.Chailloux@pps.jussieu.fr

Résumé

Une expérience d'examen en informatique mené sur ordinateur et assorti d'une notation automatique (environ 1800 copies corrigées) est décrite dans cet article. L'élaboration des sujets, le processus de notation et l'architecture logicielle de notation ainsi que les techniques d'affichage des notes et de consultation des copies corrigées sont tour à tour exposées et commentées.

Abstract

This paper describes an examination – performed on computers then, – entirely marked with computers (1800 marked deliveries). We discuss the most salient points we discovered.

Mots clés : Expérimentation et retour d'usage, Evaluation des compétences. **Keywords :** Mass-marking, Computerized exams.

1 Introduction

La licence d'informatique de l'université Pierre et Marie Curie (UPMC) compte plus de 400 inscrits administratifs. Le contenu scientifique et l'organisation de cette licence ont été rénovés en septembre 2001 et incorporent notamment un *contrôle continu transversal* indépendant des quatre modules que les étudiants ont à suivre par semestre. Ce contrôle continu transversal a pour but de proposer aux étudiants des travaux sur machine pendant tout le semestre afin — de créer une pression de travail légère mais constante et — de leur permettre de maîtriser plus rapidement le savoir-faire de développeur informatique. À la fin du semestre, un projet informatique (dit CFS pour *Contrôle Final Semestriel*) de quatre heures, individuel et sur machine, clôt ce contrôle continu.

Le nombre conséquent d'étudiants impose un recours important à l'informatique pour pouvoir traiter un tel nombre de copies à corriger. Cet article relate l'expérience acquise lors de ces épreuves où tout a été intégralement corrigé automatiquement.

2 Déroulement

Nous disposons de 105 machines ce qui nous a imposé de faire passer les étudiants en quatre fournées sur deux jours.

Nous avons donc été contraints de fabriquer deux sujets et de légères variantes sur le sujet commun aux deux fournées de la journée. Sur la base des notes des partiels, quatre groupes d'étudiants statistiquement comparables (même moyenne et écart-type) ont été déterminés afin de pouvoir comparer l'impact des différents sujets.

Les sujets n'étaient proposés qu'en HTML sur un serveur de l'UFR accompagné du règlement général du projet. Aucun document écrit n'a été autorisé pendant l'épreuve afin de mettre à égalité les étudiants mais aussi d'éviter des fuites entre matin et après-midi. Tous les sites pédagogiques sur les serveurs de l'UFR restaient accessibles ainsi que toute la documentation en ligne des différents outils.

Les corrigés ont été publiés trois jours après les épreuves. Du fait de notre inexpérience (la notation des 1800 soumissions prenait environ 12 heures), il nous a fallu 15 jours pour achever la correction automatique de ces 1800 soumissions (pour 253 étudiants) et publier les copies notées. Deux contestations ont été enregistrées concernant cette correction.

3 Préparation des sujets

Nous avons résolu que le langage de programmation ne serait pas imposé. Outre que cela permet à ceux qui n'ont pas suivi le module de programmation (utilisant principalement Ada) d'utiliser un autre langage, cela permet aux étudiants ayant quelques connaissances autres (C, C++, Caml, Java, Pascal, Perl, Prolog, Scheme, sh, etc.) de pouvoir choisir le langage le plus adapté aux questions. Cette décision a de nombreuses conséquences :

- Seul le comportement d'entrées/sorties du programme est testé.
- Le texte du programme n'est pas pris en compte.
- L'étudiant doit fournir le script construisant son programme exécutable.

Ces trois conséquences sont assez proches de certaines pratiques industrielles où l'on doit écrire des programmes conformes à leur spécification et où les programmes sont soumis à des batteries de tests automatiques. En ce sens, elles sont beaucoup plus exigeantes pour les étudiants car il n'est pas aussi simple d'amadouer un compilateur qu'un correcteur humain. En revanche, les étudiants disposent de toute la puissance informatique pour tester leurs programmes.

Puisque seul le comportement du programme importe, il est nécessaire de spécifier très finement le comportement at-

tendu. Nous avons également fourni des programmes solutions pour que, outre les exemples que nous donnions dans l'énoncé, les étudiants puissent les mettre en œuvre dans des contextes différents.

Pour chaque question, nous avons indiqué quels étaient les fichiers à soumettre et quels étaient les critères de notation que nous appliquerions sur ces fichiers. Ce dernier point garantit que les programmes seront testés pour leur comportement nominal (entrées et options sont réputées correctes) et non pour leur robustesse quant aux comportements limites.

Décomposer les questions en préparatifs de tests et programmes a permis de pouvoir expliciter des critères de notation tels que

- nous appliquons *notre* programme sur *votre* test,
- nous appliquons *votre* programme sur *votre* test,
- nous appliquons *votre* programme sur *notre* test.

Pour chaque question, nous nous avons indiqué, en exemple, une mise en œuvre interactive et deux mises en œuvre programmatiques permettant d'automatiser les tests.

La plus grande difficulté (déjà mentionnée en [JCL00]) est que, comme les programmes sont jugés sur leurs entrées/sorties et que tous les langages ne les manipulent pas avec la même précision, il faut comparer ces sorties en prenant garde aux blancs superflus ou aux écritures numériques non canoniques.

4 Notation et consultation des copies

La notation des copies doit être entièrement automatique. Cette section détaille les buts de ce processus de notation.

Des expériences de notation automatique ont eu lieu dans le semestre précédent dont nous avons tenu compte. En particulier, une demande fréquente des étudiants étaient de voir leur « copie » corrigée. Or leur copie n'est techniquement qu'un *.tar.gz* c'est-à-dire un instantané du contenu de leur répertoire de travail. Il nous a donc paru important de verbaliser tous les tests opérés sur la copie ainsi que de commenter toutes les observations menant à des fractions de notes. Une copie corrigée est donc produite qui incorpore les fichiers de l'étudiant et les commentaires associés. Cette copie mène à un document HTML visualisable. Un jeu de style permet de montrer en couleur les différents aspects de la copie : les notes, les commandes, les entrées ou sorties (normales ou d'erreur), les annotations, les tests, les questions, etc.

Les copies annotées ont une taille variant de 10 K à 100 K environ. La consultation des copies s'effectue aisément, en parallèle et sur le réseau. Nous n'avons enregistré que deux contestations sur les 1800 copies ainsi corrigées. L'une portait sur une copie ayant systématiquement introduit des lignes blanches avant les réponses, ces lignes blanches étaient difficiles à remarquer dans la présentation HTML. L'autre portait sur la compréhension des tests mêmes.

En revanche, nous avons été amené à passer d'un barème classique à un barème beaucoup plus progressif visant à donner des morceaux de point à des faits précis dont l'accumulation totale permettait de retrouver le barème classique. Nous avons donc décomposé les notes en des myriades de notes parcellaires mais relativement indépendantes, de manière à

ne pas présenter d'effet tout ou rien. Tous ces points sont bien sûr verbalisés comme indiqué plus haut.

Nous avons résolu d'afficher les notes et les copies annotées sur le réseau. Une page PHP présente la note de chaque étudiant qui peut, à partir de là, obtenir les annotations propres à chacune de ses copies, question par question.

Outre cet affichage, ces pages nous ont aussi permis, avant affichage pour les étudiants, de vérifier le bon comportement de la procédure de notation automatique en aidant au parcours des copies annotées.

5 Perspectives et conclusions

Dans cet article, nous relatons une expérience d'examen d'informatique sur ordinateur assorti d'une notation entièrement automatique. Le site¹ contient plus de détails et notamment les sujets de CFS. Comme la correction est automatique, les CFS actuels sont aussi disponibles pour entraînement sur le réseau : il est donc possible, à cette même adresse, de soumettre ses programmes et de recevoir, par courriel, sa correction personnalisée. C'est un excellent moyen de révision que nous avons intitulé « les annales dynamiques ».

Nous envisageons de corriger, à la volée, pendant l'épreuve même, en indiquant la note minimale que la copie obtiendra. Il faut dans ce dernier cas, être particulièrement sûr de sa procédure de notation mais cela peut rassurer les étudiants.

Pour résumer, les principaux enseignements de cette expérience sont :

- la production d'une copie annotée verbalisant à la fois les tests effectués et les résultats obtenus,
- les précautions à prendre pour l'écriture de tests requérant l'exécution de programmes d'étudiants.

Nous démontrons que cette expérience est viable. Les outils sont en cours de réutilisation pour d'autres épreuves sur ordinateur. Ils peuvent être demandés aux auteurs.

Références

- [JCL00] M.S. Joy, P.-S. Chan, and M. Luck. Networked submission and assessment. In *Proceedings of the 1st Annual Conference of the LTSN Centre for Information and Computer Sciences, LTSN-ICS, 2000*, 2000.
- [QC02] Christian Queinnec and Emmanuel Chailloux. Une expérience de notation en masse (rapport complet). Mars 2002. <http://www-spi.lip6.fr/~queinnec/Papers/cfsreport.ps.gz>
- [Q02b] Christian Queinnec. Site de la licence d'informatique de l'UPMC. <http://www.infop6.jussieu.fr/licence/>

¹<http://www.infop6.jussieu.fr/licence/2001/cct/annaes/>