

C. Queinnec UPMC - LIP6

Correction automatisée

Un cours (avec programmation)?

- des connaissances
 - théorie, documents (livres, manuels, guides, ...), ressources (sites, forums, ...)
- un savoir-faire
 - raisonner, programmer
- des vérifications
 - exercices, devoirs, examens
- un scénario pour tout lier

La moitié environ des cours de licence utilisent un quelconque langage de programmation.

Buts

C'est en programmant que l'on devient programmeur!

- On programme au sein d'un environnement de programmation (documentation en ligne, aides diverses à la saisie, tests,...)
- On s'entraine à programmer avec ces outils
- On s'imprègne du paradigme dominant « les programmeurs aiment tester! »

Buts

C'est en programmant que l'on devient programmeur!

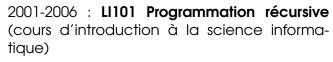
- On programme au sein d'un environnement de programmation (documentation en ligne, aides diverses à la saisie, tests,...)
- On s'entraine à programmer avec ces outils
- On s'imprègne du paradigme dominant « les programmeurs aiment tester! »

La correction doit être automatisée et ainsi être

- professionalisante
- inlassable
- uniforme



Première expérimentation





- Livre de cours, d'exercices, annales corrigées et commentées, bandes son, video du cours
- CDrom avec environnement de développement
- QCM et exercices corrigés automatiquement pour travail personnel et autonome.

Deuxième expérimentation

- 2001-2004 : CFS Contrôle final semestriel pour L3
 - Plusieurs centaines d'étudiants
 - Examen réalisé sur ordinateur
 - Examen rejouable : annales dynamique (E.Chailloux)
- depuis : 4 cours (licence et master) organisant régulièrement des examens sur ordinateur et corrigés plus ou moins automatiquement

Leçons

- notation progressive (pas de tout ou rien)
- verbalisation du processus
- pas de question oui-non
- pas d'aléa non maîtrisé
- attention aux dépendances globales (détection de plagiat, compétition, etc.) qui passent mal à l'échelle

Conclusions sur cette aventure

- Techniquement, ça marche!
- Avis mitigé des étudiants
 - pas de travail en groupe,
 - stressant,
 - machine ne corrige pas comme les enseignants
- Investissement conséquent pour les enseignants
 - 1. énoncé
 - 2. solution
 - 3. notateur
- préparation méticuleuse

Correction automatisée

- Tests unitaires (en boîte noire ou blanche)
- par assertion
- par comparaison à une solution

Squelette d'un correcteur de question

- 1. Vérification des fichiers attendus
- 2. Affichage des fichiers soumis
- Boucle sur les tests
 - 3.1 Obtenir résultat étudiant
 - 3.2 Montrer résultat étudiant
 - 3.3 Normaliser résultat étudiant
 - 3.4 Montrer résultat étudiant normalisé
 - 3.5 Appréciation résultat (ici par comparaison)
 - 3.5.1 Obtenir résultat auteur
 - 3.5.2 Montrer résultat auteur
 - 3.5.3 Normaliser résultat auteur
 - 3.6 Détermination du gain final

et ajout de la verbalisation et détermination des gains



Infrastructure de correction

Création d'une infrastructure (et d'un écosystème)

- faciliter l'écriture d'exercices
- multi-langages de programmation
- déploiement aisé
- passage à l'échelle
- robuste

Précautions techniques

Programme d'un apprenant :

- une fonction C, Scheme, Caml, ...
- une classe Java
- une page PHP
- un script Shell, Perl
- etc.

Précautions techniques

Programme d'un apprenant :

- une fonction C, Scheme, Caml, ...
- une classe Java
- une page PHP
- un script Shell, Perl
- etc.

Mais

- bogue
- boucle en temps
- sorties en excès
- bloqué
- et même malveillant

Précautions techniques

Programme d'un apprenant :

- une fonction C, Scheme, Caml, ...
- une classe Java
- une page PHP
- un script Shell, Perl
- etc.

Mais

- bogue
- boucle en temps
- sorties en excès
- bloqué
- et même malveillant

Une solution: le confinement



Autres précautions techniques (suite)

Programme d'un auteur d'exercices:

- tests unitaires
- inspection source (indentation, findbugs,...)

Autres précautions techniques (suite)

Programme d'un auteur d'exercices:

- tests unitaires
- inspection source (indentation, findbugs,...)

Mais

bogue aussi

Une même solution : le confinement

FW4EX: Infrastructure de correction

- Définition d'une grammaire pour la description d'un exercice (XML)
- Définition d'un format d'empaquetage (.tgz)
- Définitions de protocoles REST d'interaction avec l'infrastructure

Actuellement

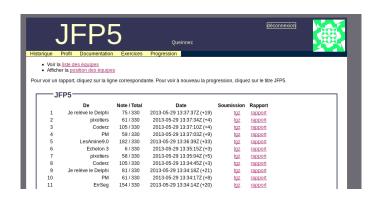
- tourne depuis 2008
- 64 exercices ou examens (et JFP) déployés
- 36000 corrections depuis 2008
- 650 étudiants concernés
- plus de 1000 réponses pour certains exercices



Exemple récent : JFP5



Exemple récent : JFP5



Apports de l'infrastructure

• la jouissance de voir les copies se corriger...

Apports de l'infrastructure

- la jouissance de voir les copies se corriger...
- Moins de maintenance pour les auteurs
 - Mémorisation de toutes les soumissions d'étudiants
 - Mémorisation des toutes les rapports de correction et notes
- Indépendance vis-à-vis
 - des outils d'ingénierie pédagogique
 - des bases de données scolaires
 - des modes d'authentification
- Exercices réutilisables
- Protection vie privée
- Topologiquement souple (pare-feux, etc.)
- Décorable (skinnable)

Apports de l'infrastructure

- la jouissance de voir les copies se corriger...
- Moins de maintenance pour les auteurs
 - Mémorisation de toutes les soumissions d'étudiants
 - Mémorisation des toutes les rapports de correction et notes
- Indépendance vis-à-vis
 - des outils d'ingénierie pédagogique
 - des bases de données scolaires
 - des modes d'authentification
- Exercices réutilisables
- Protection vie privée
- Topologiquement souple (pare-feux, etc.)
- Décorable (skinnable)
- Éternité pour les exercices

mais éco-système à bâtir au-dessus



Conclusions partielles

- La correction automatisée est mûre!
- Elle est un peu utilisée
- Elle a des aspects assez techniques
 - confinement
 - passage à l'échelle
- mais elle sort des ornières ordinaires
- et apporte des données pour une réflexion sur son efficacité

À propos des MOOC

État des lieux

- UPMC: premiers cours en ligne en 2000
- derniers en licence en 2008
- cours = plan + transparents
- + quelquefois examen, corrigé?, TD/TP (texte, corrigé?)??
- et, rarement, bandes-son, video, correction automatisée

État des lieux

- UPMC: premiers cours en ligne en 2000
- derniers en licence en 2008
- cours = plan + transparents
- + quelquefois examen, corrigé?, TD/TP (texte, corrigé?)??
- et, rarement, bandes-son, video, correction automatisée

Que manque-t-il pour en faire des MOOC?

- texte et autres ressources de référence
- chapitrage absorbable
- l'évaluation (correction automatisée, crowd marking)
- l'organisation des communautés (apprenants, enseignants)



Apports du monde académique

Et s'il n'y avait plus que des MOOC?

- Diplômation sûre et authentique
- Encadrement (coaching)
 - groupes de travail d'étudiants
 - explicitation des cours
- Stimulation pour apprendre

Corollaires:

- unification des programmes d'enseignement
- valeur universelle de l'ECTS
- classement des universités par valeur ajoutée

Où sont les 19 millions d'autres?

- 1 milliards d'apprenants
- 20 millions en L1

Il n'y a donc pas que des MOOC (au début)!

- focalisation sur L1, M1 (les plus rentables (sic))
- limitation de la concurrence
- définition d'un programme d'enseignement de l'informatique
- lutter contre le mal français : « non écrit par moi ! »
- accroissement du temps pour les recherches

Devoirs(?) du monde académique

- Reconnaissance des auteurs (réputation, rétribution, modèles d'affaire, etc.)
 - avant, pendant, après
- Régler la propriété intellectuelle (entre livres et programmes)
- et le calcul des temps de service des enseignants
- Besoin de moins de concepteurs de cours (professeurs?)
- mais de plus d'animateurs
- et de plus d'investissement pour la réalisation de cours attractifs

Pour en savoir plus

- FW4EX
- JFI A 2010 et bande-son
- CDroms sur C et Scheme

