

Modalités de partenariat avec la promotion 2014-2015 de la spécialité STL

Résumé

Le parcours professionnalisant de la spécialité STL est une formation de Master 2 Informatique spécialisée dans le développement logiciel. Les étudiants sont formés à la maîtrise de l'ensemble des processus de développement et sont capables d'utiliser efficacement les plates-formes Microsoft .Net, IBM/Rational, ainsi que les principaux logiciels libres du domaine.

Ce parcours propose à des industriels plusieurs types de partenariat :

- Projets d'ingénierie, où l'entreprise joue le rôle de client et un groupe d'étudiants le rôle de prestataire,
- Stages de fin d'études, d'une durée de cinq mois au minimum,
- Conférences sur les métiers et technologies, avec présentation de l'entreprise,

Calendrier

Propositions de projets d'ingénierie (M2)

date limite : septembre 2014

Dépôt des propositions de partenariat

- par courriel au secrétariat : Emilie.Auger@upmc.fr

Proposition de projet d'ingénierie

Au cours de la deuxième année du master, les étudiants doivent réaliser un projet répondant à la demande d'un « client » externe à l'équipe pédagogique. Ils le choisissent parmi les projets proposés par nos partenaires industriels et ils doivent le réaliser dans des conditions proches de celles du monde professionnel.

La soumission des projets d'ingénierie doit être effectuée par courriel; pour garantir les délais de réalisation, les propositions de projet d'ingénierie doivent être soumises au plus tard début septembre.

Si le projet est compatible avec les objectifs et contraintes du parcours, il sera validé et homologué par les responsables pédagogiques - peut-être après avoir négocié quelques modifications avec le donneur d'ordre - et donc proposé aux étudiants.

Le projet d'ingénierie doit comporter une forte composante technologique et une phase de réalisation correspondant à une charge de travail de l'ordre de trois homme/mois. Il est prévu que les étudiants consacrent à la réalisation du projet qu'ils ont choisi, une demi-journée par semaine, jusqu'à fin novembre, puis une à deux journées durant les mois de décembre et janvier. Chaque projet, selon son importance, est pris en charge par un groupe de 2 à 4 étudiants encadrés par au moins un enseignant du Master.

Le projet consiste à mener à bien une réalisation complète qui implique toutes les phases d'un projet industriel : analyse de la demande, étude de faisabilité et choix de solutions comprenant la recherche bibliographique et des composants disponibles, réponse à l'appel d'offre, spécifications contractuelles, conception, codage, tests, documentation du logiciel et toutes les tâches annexes de gestion d'une équipe (planification, audit, ...).

Voici quelques exemples de projets réalisés ces dernières années, certains de ces projets ayant pu être développés et améliorés au cours des années :

- Réalisation d'une interface d'édition graphique et de génération d'un dialogue décrit en XML, sous Palm-OS pour la SEMA.
- Réalisation d'un éditeur de workflow pour la construction d'un système multi-agents d'analyse et de sélection de messages dans le flux émis par les agences de presse, pour la Caisse des Dépôts et Consignation (CDC).
- Définition et implémentation d'un plugin Eclipse d'édition de méta-données, pour gérer l'« *enhancer* » de l'implémentation JDO de Versant Technology (édité sur sourceforge.net).

- Ingénierie de la mise à disposition, sur un intranet sécurisé, d'une bibliothèque de fonctions financières pour le groupe BNP Paribas.
- Réalisation d'un Framework de normalisation des actions d'interaction au sein d'un Générateur d'IHM pour la société ADEX-M2i
- Réalisation d'un prototype de gestion d'un référentiel de tâches à travers l'interprétation répartie d'un workflow sur une plate-forme multi-agents pour le centre de recherche en informatique de Montréal au Canada. Ce projet a été suivi d'un stage (à Montréal) de quatre mois permettant l'industrialisation du prototype.
- Définition et réalisation d'une architecture d'interconnexion de postes de travail MS Office et d'un workgroup Lotus Notes pour gérer le flux de la gestion documentaire d'un service SAV d'Alcatel téléphonie
- Reprise et finalisation de composants du framework ObjectWeb pour le compte de France Telecom.
- Dans le cadre du projet Apache Directory Studio de la fondation Apache, mise en œuvre d'un système de patrons définissables par l'utilisateur permettant de personnaliser l'affichage des données et la saisie de celles-ci pour le plugin Eclipse LDAP browser.
- Automatisation des développements dans l'atelier RSA de la compagnie IBM par transformations de modèle UML profilé pour les Lab Rational de Marne La vallée de cette compagnie.
- Réalisation, en utilisant les technologies DSL dans dot.net de Microsoft, d'une Software Factory permettant la conception et la génération de l'application Event Catalyst spécifique à un événement pour la société SEAL.

Les étudiants choisissent, à la mi-septembre, un projet parmi ceux déposés. Ils élaborent, en collaboration avec les donneurs d'ordre, un dossier de spécifications des besoins et des exigences qui constituera le cahier des charges de leur projet. Ce cahier des charges doit être rédigé pour la fin du mois d'octobre. Il fait l'objet d'une présentation orale sous forme de "Review" devant le responsable du parcours et le prescripteur qui a fourni le sujet qui désigne, à cette occasion, un référent en charge, en tant que maître d'ouvrage, du suivi du projet. Ces derniers peuvent être accompagnés des experts qu'ils voudront bien convoquer.

Après négociation, ce cahier des charges devient le contrat qui lie les trois parties et qui servira de base pour le jugement porté sur le projet fin janvier. Les étudiants peuvent demander de façon circonstanciée la révision du contrat avant la fin du mois de novembre. Le projet est jugé à travers le produit rendu (note de réalisation) en fonction de sa cohérence avec le cahier des charges. Cette évaluation est conduite par le prescripteur du projet et le responsable du parcours ; elle comprend l'analyse de l'ensemble des documents produits, le suivi de plusieurs exposés auprès tant de l'encadrant universitaire que du référent industriel et de deux soutenances orales devant différents jurys, jalonnant la réalisation du projet. L'ensemble de ces activités servent à évaluer les capacités de communication et la maîtrise technique des étudiants.

Un donneur d'ordre s'engage à fournir toutes informations techniques nécessaires pour la bonne marche du projet. Pour cela il s'engage à participer, au moins une fois par mois, à une réunion d'avancement du projet d'au plus une heure. Cette réunion pourra se tenir soit dans l'entreprise, soit à l'université. Il fournira un moyen (courrier électronique, fax) de communication permettant aux étudiants de lui poser des questions urgentes entre les réunions. Il s'engage à répondre à ces questions sous 48 heures. Le temps à consacrer à ce suivi ne devra pas dépasser une heure par semaine.

≥ *Dépôt des propositions de projet d'ingénierie*

- par courriel au secrétariat : Emilie.Auger@upmc.fr

Master mention informatique
Spécialité Science et Technologie du Logiciel
Parcours Professionnalisant

Proposition de projet d'ingénierie

à la promotion 2014-2015 de la spécialité STL

≥ *Dépôt des propositions de projet d'ingénierie*

- par courriel au secrétariat : Emilie.Auger@upmc.fr

Organisation émettrice : *

Responsable(s) technique(s) qui encadrera (ont) l'équipe

Nom(s): **Lécharny**

Prénom(s): **Emmanuel**

Fonction(s) :

Email(s): **elecharny@apache.org**

Intitulé du projet : Construction d'une plate-forme de test répartie LDAP

Définition technique du projet

Objectifs :

L'un des problèmes que nous rencontrons est de pouvoir lancer un test de charge configurable en disposant de suffisamment d'injecteurs. L'utilisation de serveurs pour jouer le rôle de clients est problématique, principalement pour des raisons de coût (il peut être nécessaire de disposer de plusieurs dizaines de serveur d'injection, par exemple).

Une des solutions à ce problème serait d'utiliser des injecteurs peu coûteux, comme des Raspberry Pi.

L'idée est de valider la faisabilité technique de cette approche, qui présente néanmoins de nombreux points technique qui méritent d'être approfondis :

- o CPU limité sur les injecteurs
- o capacité à lancer N injecteurs de façon synchrone
- o capacité à diriger les injecteurs à distance
- o capacité à récupérer les données produites et à les agréger en un tableau synthétique

Globalement, un scénario de test doit comprendre les étapes suivantes :

- lancement de 1 à N injecteurs
- "chauffage" du serveur pendant un temps déterminé avec N injecteurs
- démarrage du test proprement dit avec N injecteurs
- agrégation des résultats en un tableau synthétique

Un des besoins complémentaire serait de pouvoir mesurer l'impact d'une montée en charge progressive (ie, impact de l'ajout de M injecteurs et détection du nombre d'injecteurs idéal pour un test donné). L'intérêt et de mesurer le point à partir duquel le serveur n'est plus en mesure d'offrir un service optimal.

Le protocole à tester est LDAP. cela n'implique aucunement que l'outil de test soit limité à LDAP, mais cela correspond à un besoin actuel. Il doit être possible d'utiliser le système pour monter des tests de charge HTTP ou n'importe quel protocole.

Un des problèmes des tests de charge LDAP est la lourdeur du processing des requêtes et des réponses LDAP : cela nécessite l'encodage et le décodage des requêtes en binaire, ce qui est très coûteux en CPU. Une idée serait de pré-calculer le binaire à envoyer, et le binaire reçu, pour éviter cette charge de travail sur l'injecteur. On ferait tourner un test à blanc, dont l'objectif serait de générer toutes les requêtes binaires, pour les envoyer dans un second temps directement. Le test

serait donc effectué en deux temps :

- o une phase de préparation et d'enregistrement
- o une phase d'exécution.

Dans tous les cas de figure, les tests doivent pouvoir être envoyés aux injecteurs sans avoir à les installer sur N machines, c'est à dire qu'il faut prévoir un mécanisme de transport de ceux ci sous la forme de packages dynamiquement exécutables.

Dernier point : une interface d'administration graphique doit être disponible pour lancer les tests et afficher les résultats. Cette interface peut être une application web.

Il est possible de s'inspirer de projets existant permettant de mettre en oeuvre cette proposition :

- o <http://www.jppf.org/>
- o <http://jmeter.apache.org/>
- o <http://gatling-tool.org/> (mais le langage de script utilisé est Scala...)

Tout outil jugé utile dans la mesure où cela permet de répondre au besoin peut être utilisé, à la condition expresse que la licence soit compatible AL 2.0 (Apache Licence 2.0).

Dernier point : se rappeler que la cible est un injecteur qui ne dispose ni de beaucoup de mémoire, ni de beaucoup de CPU...

Contexte technologique :

Plateforme technique désirée pour l'exploitation des résultats

Matériel : 4 Raspberry PI pour les injecteurs, un serveur sous linux

Système : Linux

Plateforme technique désirée pour le développement

Langages de spécification, programmation : Java

Environnement de développement, gestionnaire de sources et configuration :

Git ou SVN

Compétences nécessaires et désirées:

Connaissance de Java, Git ou SVN, Junit.

LDAP et Jmeter ne sont pas des prérequis, mais une connaissance à ce sujet est un plus.

Références bibliographiques et sites Web:

<http://jmeter.apache.org/>

<http://directory.apache.org/>

<https://java.net/projects/slamd>

<http://www.jppf.org/>

<http://gatling-tool.org/>

Définition de la recette :

Nous disposons de scénario d'injections exécutés sur une autre plate-forme, avec un autre outil. L'objectif est de pouvoir rejouer ces scénarios sur l'outil développé.

Documents complémentaires : à joindre au Formulaire

