

Contribution à l'implantation en Java d'un modèle minimal de composants

Jacques Malenfant

Le projet consiste à contribuer à l'implantation d'un modèle de composants minimal pour Java. Ce modèle est actuellement implanté et utilisé dans le cadre d'un projet de recherche où il a permis de construire des applications à grande échelle (plusieurs dizaines de milliers de composants s'exécutant sur des dizaines de machines virtuelles Java). Il s'agit d'un modèle minimal au sens où il est défini autour d'un minimum de concepts centraux (composants emboîtables, interfaces offertes et requises proposés sur des ports explicites, connexion des ports par connecteurs explicites, assemblages de composants pouvant être répartis sur plusieurs machines virtuelles et plusieurs ordinateurs), et utilisant des technologies à la fois basiques et stables de Java, comme RMI, les « *sockets* », etc. Il s'agit donc d'un modèle très dépouillé plutôt destiné à faire de la recherche et de l'enseignement, et non à concurrencer des modèles visant une exploitation dans des applications réelles, commerciales.

Le projet consiste à consolider l'implantation de ce modèle, et ce dans plusieurs directions pouvant être attaquées indépendamment les unes des autres :

- Mise en œuvre :
 - génération de scripts de déploiement et d'exécution permettant de lancer des assemblages de composants sur plusieurs machines virtuelles et sur plusieurs ordinateurs physiques ;
 - implantation d'un registre global de publication des ports passant à l'échelle en répartissant le registre centralisé actuel sur plusieurs machines et avec réplication des informations publiées ;
- Améliorations du modèle :
 - implantation d'un mode de communication par « *sockets* » pour les échanges de messages de données entre les composants ;
 - définir des ports et des connecteurs multiples permettant des relations 1 :n, voire n :n, entre les composants ;
- Améliorations à la programmation par composants :
 - extension du modèle de programmation répartie par l'implantation de variables futures permettant la synchronisation entre composants concurrents et répartis ;
 - rationalisation des modes de traces et de « *debug* » pour faciliter la mise au point des programmes ;
- Extensions au modèle :
 - étendre le modèle pour inclure des composants temps-réel avec une définition des tâches et de leurs contraintes de temps utilisées par un algorithme d'ordonnancement statique produisant un ordonnancement à exécuter par le composant ;
 - produire une extension du modèle de composants supportant les spécifications du modèle de Fractal.

Cette année, je compte prendre sur ce projet une seule équipe de 2 à 3 étudiants. Les tâches précises à réaliser parmi la liste précédente sera à négocier au moment du choix du projet.

J'utilise ce modèle dans un cours de M2 (ALASCA), et donc le code produit doit résister à une

utilisation par des étudiants dans le cadre de travaux pratiques de niveau M2. Les compétences requises pour ce projet sont :

- Une excellente connaissance de programmation en Java, et en particulier sur les aspects concurrence et répartition.
- Une grande rigueur dans la conception, la programmation et la documentation.
- Des connaissances en programmation système (scripts sous Linux et Mac Os X) et sur la programmation répartie en général sont d'excellents atouts, selon les thèmes abordés effectivement.
- Pour certains des thèmes, sur les extensions par exemple, il faudra être capable d'aller chercher les informations nécessaires (algorithmes, techniques d'implantation) par soi-même, les comprendre et les traduire en une solution opérationnelle, ce qui requiert une bonne autonomie et de la débrouillardise (en ce sens peut-être plutôt destinés à des étudiants qui pensent poursuivre en parcours recherche en M2).

Dans tous les cas, le projet débutera par une prise en main du modèle en explorant une application existante et la réalisation d'une petite application à partir d'une description courte. Cette phase passée, vous serez plus à même de comprendre les attentes que je peux avoir sur les différentes tâches énoncées précédemment.