

Cahier des charges

Auboin Nicolas, Bendavid Olivier, Haderer Nicolas, Pallet David

6 février 2009

Encadrants du T.E.R : Huchard Marianne, Tibermacine Chouki

Table des matières

1	Sujet du T.E.R.	3
2	Contexte	3
3	Développement du projet	4
4	Organisation du travail	4
5	Solution envisagée	5
6	Contraintes du projet	6

1 Sujet du T.E.R.

Réalisation d'annuaires de composants logiciels

Résumé : Le génie logiciel à base de composants permet de construire des applications par assemblage de briques logicielles disponibles sur étagères. Les composants exposent leurs services : ceux qu'ils mettent à la disposition de leur environnement (services fournis) ceux qu'ils doivent trouver dans leur environnement pour fonctionner (services requis) Construire une application dans ce paradigme consiste à trouver et à connecter (assembler) les composants compatibles (services fournis correspondants aux services requis) en vue de réaliser l'objectif final. Lors des évolutions ultérieures de l'application, de nouveaux composants peuvent être recherchés pour offrir de nouvelles fonctionnalités ou pour remplacer des composants en panne ou devenus obsolètes. Pour faciliter ces processus, nous proposons de construire des annuaires de composants de type "Pages jaunes" en ce sens qu'ils sont basés sur les services. Ces annuaires répertorieront les composants et les classeront afin d'accélérer les recherches soit lors des opérations de construction, soit lors des opérations de remplacement. Un intérêt de la classification est de présenter des composants proches du composant recherché lorsque celui-ci n'est pas disponible.

2 Contexte

Dans le monde actuel de la programmation, une notion importante a vu le jour depuis l'apparition des langages objets : la réutilisabilité du code. Idée phare du langage objet, la réutilisation n'a pas toujours été la préoccupation majeure des développeurs. Le projet dans son ensemble a l'ambitieuse idée de mettre en avant l'aspect réutilisation des divers modules proposés par les développeurs.

Ainsi, on peut définir un composant comme un élément qui, en s'assemblant avec d'autres, forme un logiciel à part entière. Pour déterminer quel composant correspond à quelle demande, les composants sont définis par leurs services fournis, et leurs services requis (à la manière d'un composant électronique). A partir de cela, la réalisation d'un composant revient à faire un assemblage de "briques logicielles".

3 Développement du projet

L'objectif du projet vise à servir le principe d'assemblage de composants par la création d'un annuaire de composants logiciels.

L'annuaire est une hiérarchisation de modules (composants) utilisé dans les activités d'assemblage. L'objectif étant qu'à terme, l'annuaire permette le chargement d'un module de manière statique ou dynamique. Par exemple, le module doit pouvoir se substituer à un autre en cas de panne, de problème ou encore de mise à jour de ce dernier.

Pour déterminer la compatibilité entre modules, il faut regarder leurs interfaces. Plus en détails, il faut étudier chaque fonctionnalité de ces interfaces. Les fonctionnalités sont elles-mêmes décrites par une signature (un nom et les paramètres d'entrée et de sortie). Pour obtenir les diverses compatibilités possibles, il est indispensable d'avoir une hiérarchisation des types de paramètres. Un sous-type étendra le type d'origine et offrira donc des possibilités de compatibilité.

Les enseignants chercheurs avec qui nous travaillons sur ce projet ont créé une méthode de formalisation pour mettre en évidence les conditions de compatibilités entre les composants. Il nous faudra donc l'implémenter pas à pas pour obtenir le résultat souhaité.

4 Organisation du travail

Pour pouvoir faire face aux différents aspects du projet, nous essayons de coordonner au maximum nos travaux au sein du groupe. Nous faisons le point sur l'avancement de chacun tous les mardi après-midi pour comparer nos recherches, analyses et pour décider des choix importants (même si à l'heure actuelle, ceux-ci sont limités). Ainsi, d'une semaine sur l'autre, nous nous répartissons le travail et réévaluons les diverses facettes du projet.

Le projet n'étant qu'à ses débuts, les rôles au sein du groupe sont en cours de définition. La phase de compréhension du sujet est arrivée à terme lors de notre dernier rendez-vous avec nos encadrants. Nous avons pu confirmer nos recherches préliminaires, préciser notre rôle dans le projet en validant notre cahier des charges.

Nous enregistrons dans un carnet de bord l'avancement chronologique du projet. Nous y précisons nos diverses pistes à explorer, nos objectifs pour la prochaine réunion, les rendez-vous avec nos encadrants etc...

Le groupe sera pour l'étape initiale divisé en 2 sous-groupes de 2. Le premier groupe aura charge d'analyser le travail existant fourni. Le second travaillera sur une organisation hiérarchique efficace du document XML (la ver-

sion fournie n'étant pas optimale). A partir de là, nous ferons un point avec nos encadrants pour s'assurer que cela leur convient.

Pour mettre en corrélation les tâches à réaliser et le temps imparti, un diagramme de GANTT est présent en fin de cahier des charges. Nous nous efforcerons de respecter les délais prévus dans la mesure du possible et réévaluerons notre avancement en le mettant à jour au fil du projet.

5 Solution envisagée

Dans le cadre de ce projet, les enseignants-chercheurs avec qui nous travaillons ont déjà développé plusieurs approches du problème. Ils nous chargent de résoudre un problème jusqu'alors insoluble, le cheminement pour générer l'annuaire de composants.

L'une des étapes du projet est la création d'un document recensant l'intégralité des composants logiciels avec leurs caractéristiques. Cette hiérarchisation se fera en XML, format idéal pour décrire la hiérarchisation des types. Ce document servira de base au système de recherche permettant d'effectuer des requêtes de connexion, pour déterminer les compatibilités possibles avec l'existant, puis une opération de substitution en cas de correspondance.

En effet, pour pouvoir créer cette hiérarchie de types plusieurs possibilités s'offrent à nous. La première proposée, est l'analyse (parsing) du code de l'ensemble des classes disponibles d'une bibliothèque (par exemple la librairie standard java). Avec le package reflect, il est possible de déterminer la super classe d'une classe donnée (avec la méthode `getSuperclass()`), l'opération inverse est plus problématique.

La seconde option est d'analyser la documentation complète de la librairie standard (par exemple la JAVADOC) pour en déduire une hiérarchisation des classes existantes. Cette solution est plus " lourde " mais son aboutissement donnera à coup sûr le résultat escompté contrairement à la première solution. (... parler de l'intérêt d'avoir un éditeur XML pour construire manuellement une partie de la hiérarchie des types ...).

On peut éventuellement supposer que la première option permettrait la réalisation d'une hiérarchie principale et que les développeurs puissent y greffer leurs propres éléments supplémentaires à l'aide de la seconde option. Il serait aussi intéressant d'avoir un éditeur XML pour construire manuellement une partie de la hiérarchie des types et donc proposer au développeur d'ajouter ses propres éléments.

Après cette phase initiale, il nous faudra nous approprier le travail existant fourni :

Galicja 3 : un outil développé permettant la construction de treillis de Gallois (utilisé pour la hiérarchisation des types)

CoCola : développement inachevé qui servira de base à notre projet

Après cette phase, nous y intégrerons le modèle hiérarchique créé. Nous améliorerons le logiciel sous plusieurs angles (corrections de bugs, optimisation, portabilité, extensions . . .) Contraintes fonctionnelle et logicielle Les besoins : Proposer un système de briques logicielles (annuaire de composants) qui facilitera la recherche de composants. A qui ? : tous les développeurs utilisant le paradigme de construction de logiciels par assemblage de composants. Sur quoi ? : Le but du projet est de créer une structure hiérarchique entre tous les composants d'un ensemble. A partir de là, des relations de compatibilités peuvent être établies entre ces composants, et des opérations de changement/remplacement sont réalisables.

6 Contraintes du projet

Contraintes :

- Contrainte de temps imposée par les commanditaires du projet qui souhaitent publier les résultats de recherches obtenus au terme du développement de l'outil
- Contrainte de passage à l'échelle (rapidité d'exécution et optimisation de l'espace mémoire, pertinence d'une utilisation manuelle des résultats de la recherche de composants)
- Contrainte d'utilisation d'un outil de construction de treillis
- Contrainte d'analyse d'un développement antérieur non achevé

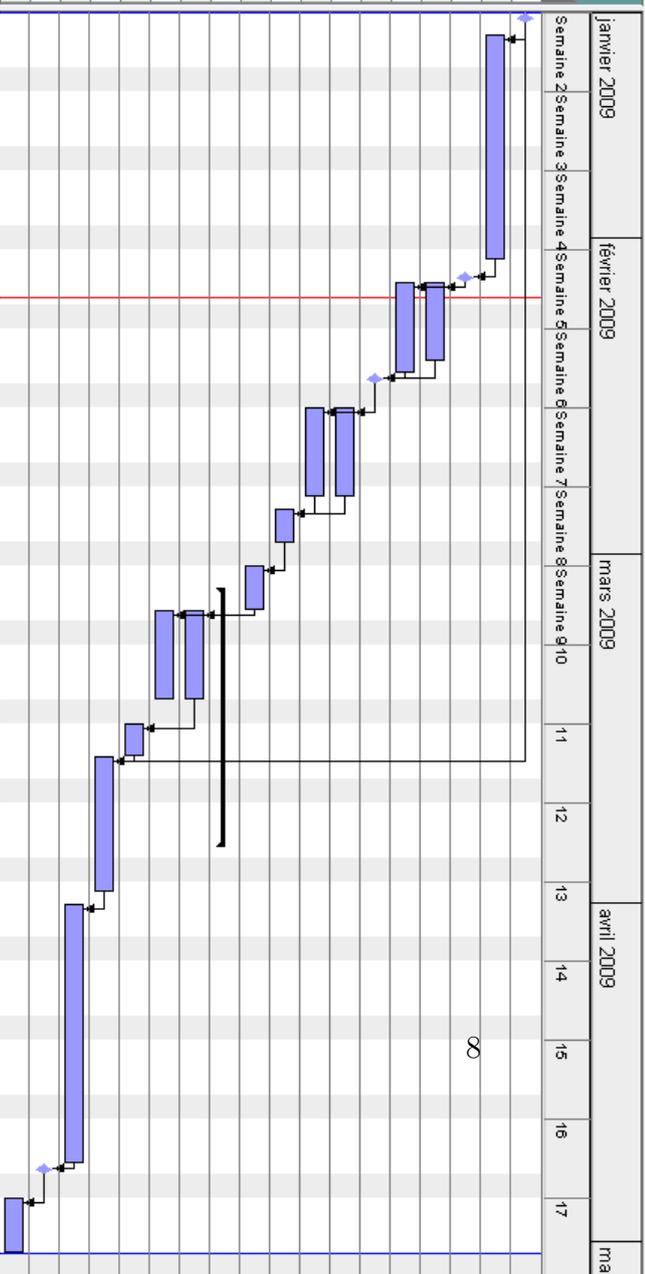
Tâches :

- Proposition d'un schéma XML correspondant à la grammaire des documents qui vont contenir les hiérarchies de types
- Construction de la hiérarchie de types à partir de la documentation JAVADOC de la librairie standard du JDK
- Développement du module permettant l'ajout de nouveaux types dans

la hiérarchie de types existante (possibilité d'introduire des JAR, des répertoires avec des .class, ...)

- Intégration de l'éditeur XML et connexion avec l'outil actuel (CoCoLa)
- Amélioration de l'outil actuel (stabiliser, améliorer de l'interface graphique, améliorer la portabilité de l'outil, compléter avec la description des exceptions dans les signatures, ...)
- Test et validation

Nom	Date de début	Date de fin
1er RDV : Appréhension du sujet	12/01/09	13/01/09
Analyse des documents fournis + Recherches	14/01/09	03/02/09
RDV : Validation CC + Récupération existant	04/02/09	05/02/09
Création schéma XML de hiérarchie de types	05/02/09	12/02/09
Analyse des programmes existants fournis	05/02/09	13/02/09
RDV : Rapport schéma XML + Bilan analyse	13/02/09	14/02/09
Tests de construction hiérarchie XML	16/02/09	24/02/09
Développement module ajout de nouveau types	16/02/09	28/02/09
Intégration éditeur XML	25/02/09	28/02/09
Connexion éditeur avec CoCola	02/03/09	06/03/09
Amélioration du logiciel	04/03/09	27/03/09
Stabilisation (débogage, gestion erreurs ...)	06/03/09	14/03/09
Interface graphique	06/03/09	14/03/09
Portabilité	16/03/09	19/03/09
Extension du logiciel	19/03/09	31/03/09
Tests	01/04/09	24/04/09
RDV final : Validation	24/04/09	27/04/09
Dernières corrections	27/04/09	02/05/09



Légende du diagramme de GANTT :

Début et fin du projet : traits bleus

Date actuelle : trait rouge

Losange bleu : rendez-vous encadrants

powered by L^AT_EX