

Eclipse Time 2008



29 mai 2008, Toulouse

par Xavier Pucel (xpucel.developpez.com) (Blog) Benjamin Cabé (Blog)

Date de publication : 13/06/2008

Dernière mise à jour : 13/06/2008

La conférence Eclipse Time 2008 s'est tenue le 29 Mai 2008 à la Cité de l'Espace de Toulouse. Une centaine de participants a suivi les présentations sur le présent et le futur d'Eclipse. Au menu, un avant goût de Ganymède (Eclipse 3.4) qui sortira très prochainement, une feuille de route pour Eclipse 4.0, et des retours d'expérience sur le développement d'applications basées sur Eclipse.

Introduction.....	3
II - Eclipse 3.4 (Ganymede) : le présent.....	5
II-A - Les plug-ins de Ganymede.....	5
II-B - Evolution de la plateforme.....	5
III - Eclipse 4 : le futur d'Eclipse.....	6
III-A - Ouvrir Eclipse aux autres langages.....	6
III-B - Eclipse, une plateforme d'applications.....	6
IV - Conclusion.....	7

Introduction

La conférence Eclipse Time 2008 s'est tenue le 29 Mai 2008 à la Cité de l'Espace à Toulouse. Elle a été organisée par la société Anyware Technologies.



La journée était divisée en 2 parties : le matin, des présentations sur Ganymede, Eclipse 4 ou les licences logicielles ont resitué l'environnement du développeur d'applications sur Eclipse. L'après-midi, des ateliers ont permis de faire des retours d'expérience sur des projets développés sur la plateforme Eclipse.

Parmi les protagonistes rencontrés durant la journée :

- Benjamin Cabé, expert Eclipse chez Anyware Technologies, est contributeur sur plusieurs projets Eclipse, dont PDE (Plug-in Development Environment) et committer du projet uDig (User-friendly Desktop Internet GIS). Il est auteur d'une présentation de la conférence, et animateur d'un atelier de Eclipse Time.
- Ralph Mueller, Directeur de l'écosystème Europe de la fondation Eclipse

Sommaire des présentations

- **Eclipse 4.0 : Quel futur pour Eclipse ?** par Benjamin Cabé
- **Ganymede : l'état du Pays Eclipse** par Ralph Mueller
- **Licences logicielles** par Vincent Pollard et Cendrine Claviez
- **Projets collaboratifs dans l'industrie verticale** par Ralph Mueller

Les ateliers, quant à eux, couvraient des applications telles que :

- **JViews** : outils de positionnement des diagrammes dans GEF et GMF
- **Momentics IDE** : Outils pour le développement C, C++ et C++ embarqué (**annulé**)
- Application de recherche et visualisation de métadonnées d'expérimentations scientifiques basée sur uDig
- **Outils de modélisation graphique** : EMF, GMF, MDA
- **RAP** : Rich Ajax Platform. Un conteneur de plug-ins Eclipse qui en fait des applications web. Ce projet a une importance stratégique dans l'écosystème Eclipse, nous y reviendrons plus tard.
- **Du logiciel propriétaire au logiciel Open Source** : le cas d'une entreprise dans le secteur de l'aéronautique
- **BIRT** : outil de Business Intelligence et de génération de rapports.
- **Aquisition de données** : développement d'un logiciel de supervision dans le secteur de la production d'énergie.

Plutôt que de faire un résumé de chaque présentation et atelier, ce compte-rendu est concentré sur les informations qui intéresseront le plus grand nombre.



Ralph Mueller, Directeur de l'écosystème Europe de la fondation Eclipse

II - Eclipse 3.4 (Ganymede) : le présent

Après avoir été un simple environnement de développement en Java, Eclipse est devenu, depuis la version 3.0 et l'apparition de l'architecture RCP, une véritable plateforme d'applications. Si beaucoup de projets sont encore liés au développement (en C/C++, PHP, langages dynamiques, etc), on voit également apparaître une grande quantité d'applications contenues dans la plateforme Eclipse.

Eclipse est distribué sous une licence Open Source héréditaire non virale, de type LGPL. Toute modification du logiciel Eclipse est Open Source, mais la plateforme peut être intégrée avec un logiciel propriétaire (par exemple un plug-in), sans avoir de conséquences sur le mode de distribution du plug-in. Ainsi, la plateforme sera toujours Open Source, sans toutefois empêcher quelqu'un de s'en servir pour développer un logiciel propriétaire. Ce type de licence est utilisé entre autres par Mozilla Firefox.

II-A - Les plug-ins de Ganymede

Côté outils de développement, la version 3.4 d'Eclipse s'accompagne d'un ensemble de projets à brancher dans la plateforme selon la nécessité : Web Tools 3.0, CDT 5.0, etc. Les projets sont des plugins qui ont été intégrés dans le cycle de développement d'Eclipse. Cela facilite énormément leur recherche et leur installation, garantit leur intercompatibilité et leur non redondance grâce à des releases synchronisées. Au total, la release Ganymède contient 24 projets, dont le noyau Eclipse 3.4 (Callisto contenait 10 projets, et Europa 21).

A noter également la sortie de l'Eclipse Packaging Project 1.0, qui est utilisé pour créer les distributions officielles d'Eclipse (Java SE, Java EE, C/C++, RCP, ...). Ce projet facilite la préparation d'applications basées sur la plateforme Eclipse en choisissant les plug-ins à inclure dans la distribution.

II-B - Evolution de la plateforme

Faire évoluer Eclipse, c'est avant tout faire évoluer le mécanisme de la plateforme. Et on peut dire qu'avec Ganymede, on est servis ! Les projets qui m'ont le plus marqué sont surtout les projets RAP et eRCP : des technologies de runtime de plug-ins Eclipse, tout comme la plateforme RCP classique, à la différence près que RAP est destinée aux serveurs web, et eRCP à l'embarqué ! Le même plug-in peut donc être exécuté sur un poste client, mais aussi sur un serveur web, ou sur un terminal mobile, sans changer une seule ligne de code, sans même le recompiler !

Le projet RAP m'a particulièrement intrigué. Il est basé sur une implémentation "parallèle" de SWT/JFace, qui fait appel à des mécanismes autant sur le serveur que sur le navigateur web. Côté serveur on retrouve l'implémentation de la norme OSGI par Eclipse : Equinox. Côté client, c'est qooxdoo, une librairie JavaScript et AJAX, qui implémente l'interface graphique et les communications.

III - Eclipse 4 : le futur d'Eclipse

Autant vous l'annoncer tout de suite, le projet pour Eclipse 4 est ambitieux, très prometteur et annoncé pour 2010, soit dans très peu de temps. La stratégie de développement est articulée autour de deux axes principaux :

- Ouvrir Eclipse aux autres langages
- Faire d'Eclipse une plateforme d'applications plus qu'un environnement de programmation.

Les deux axes reposent sur une évolution importante de la plateforme qui a déjà commencé.

Le modèle Open Source n'est pas remis en cause, bien au contraire, avec le succès grandissant d'Eclipse et le nombre toujours accru de contributeurs, Eclipse est parti pour rester sous la même licence.

III-A - Ouvrir Eclipse aux autres langages

Dans le but simple d'augmenter le nombre de développeurs sous Eclipse, beaucoup d'initiatives, quelques unes évidentes, d'autres surprenantes, ambitieuses, voire même déroutantes sont au programme.

Faciliter la prise en main : l'API de développement de plug-ins est difficile à prendre en main, et plusieurs axes d'amélioration ont été identifiés : l'accès aux APIs et la documentation doivent être relevés, et il faut éviter les APIs redondantes.

Séparer la présentation du contenu : on pourrait être satisfait de SWT et JFace, les APIs graphiques utilisées par Eclipse. Mais il est prévu d'inclure la gestion de scripts et de styles dans la plateforme, de manière à fournir des interfaces plus facilement modifiables (à la fois en phase de développement et au runtime). Ceci ferait intervenir un (des?) langage(s) de description d'IHM, probablement basé(s) sur du XML (XUL, XAML, ...) voire du CSS.

Faire évoluer OSGi : les évolutions les plus radicales concerneront la plateforme elle-même. Dans le but de rallier plus de monde à la communauté, il est prévu de permettre le développement de plug-ins dans d'autres langages que le Java ! Dans l'absolu, une fois ce but atteint, on peut même imaginer que le noyau de la plateforme soit lui-même écrit dans un autre langage. Cette évolution est très ambitieuse, et laisse entrevoir une plateforme où des plug-ins écrits dans des langages divers communiqueraient entre eux. A suivre, donc !

III-B - Eclipse, une plateforme d'applications

Pour faire d'Eclipse une véritable plateforme d'applications, un exemple des mesures qui seront prises est de ne plus nécessairement inclure les concepts de projets et de perspectives dans le noyau de la plateforme. En effet, ce sont des notions qui ne sont pas nécessaires à toute application. Aussi, le noyau de la plateforme sera le plus générique possible, tout en continuant à offrir les mécanismes qui rendent cette même plateforme attrayante.

L'intégration des plug-ins a déjà évolué afin de gérer des dépendances complexes entre les plug-ins. Elle doit encore muter afin de proposer une approche par services. Cela permet de ne pas s'intéresser au contenu du plug-in, mais simplement à ses fonctionnalités. Outre le gain pour l'utilisateur qui n'a pas à s'occuper de pures problématiques d'implémentation qui ne l'intéressent pas directement; à terme, cela offre aux développeurs de plug-ins les moyens de mettre en place des interactions plus fortes et plus complexes entre les plug-ins.

Le développement de plug-ins doit être facilité par l'utilisation du framework EMF pour mettre en oeuvre un véritable modèle de la plateforme Eclipse.

Les implémentations Web et mobile de la plateforme Eclipse (respectivement, RAP et eRCP) sont promises à un développement rapide. Pour RAP, des supports différents de AJAX sont envisagés : Dojo, Flex, Silverlight, etc. On attend SWT 4, qui harmonisera les interfaces graphiques sur tous les supports, et facilitera la portabilité des applications.

eRCP est lui aussi un élément important, puisqu'il permet d'embarquer une application Eclipse de manière totalement transparente. Le secteur de la production industrielle est ouvertement visé par la stratégie de développement.

IV - Conclusion

Eclipse a connu jusqu'à présent un nombre de contributeurs toujours grandissant, et n'a pas l'air de vouloir s'arrêter. Les perspectives d'évolution sont à la hauteur d'une entreprise de rang mondial, et sont très prometteuses sur les possibilités de l'outil de demain.