

Compte rendu de la journée : Eclipse - Now you can

par [Denis Cabasson](#)

Date de publication : 06/11/2006

Dernière mise à jour : 10/11/2006

Le 13 octobre 2006, s'est tenu à Paris la journée "Eclipse - Now you can" organisée par TNI-Software. Cette journée, suivant directement les Eclipse Summit Europe réunissait les plus grands responsables de la fondation Eclipse, et tous les acteurs de l'écosystème Eclipse français.

- I - Mike MILINKOVITCH - Overview of Eclipse : The 7 pillars of Eclipse
- II - Ralph MUELLER - Présentation de Callisto
- III - Wayne BEATON - Présentation de la plate-forme Eclipse RCP
- IV - Janet CAMPBELL - Eclipse : propriété intellectuelle et modèles économiques
- V - Philippe KRIEF - Jazz : Team collaboration platform for software development
- VI - Les ateliers

I - Mike MILINKOVITCH - Overview of Eclipse : The 7 pillars of Eclipse



La première présentation est celle de Mike MILINKOVITCH, directeur exécutif de la fondation Eclipse, qui nous présente les principes généraux de la fondation Eclipse, ainsi que ses domaines d'interventions.

La journée Eclipse - Now you can a suivi directement le Eclipse Summit Europe qui, cette année encore a été un grand succès, avec plus de 300 participants et 21 sponsors.

La fondation Eclipse peut être vue sous deux angles assez différents:

- Le premier, et le plus connu : La fondation Eclipse est une communauté Open-Source (dans la lignée d'Apache ou Mozilla) qui cherche à construire des frameworks universel pour toutes les plates-formes et l'outillage correspondant
- Le second : La fondation est un consortium des principales industries de logiciels (créant ce qui s'appelle l'écosystème d'Eclipse). Ces industriels élaborent des produits et des services qui viennent s'ajouter à l'offre de la fondation

La fondation Eclipse est une communauté Open-Source bien particulières en cela que contrairement aux autres communautés (Apache, Mozilla, ...) celle ci ne se présente pas sous la forme juridique d'une association caritative, mais bien comme une entreprise vouée à faire des bénéfices.

La fondation Eclipse gagne de plus en plus de vitesse, avec environ 65% des développeurs développant aujourd'hui sur une plate-forme Eclipse et plus de 500,000 téléchargements par mois du SDK Eclipse. La répartition internationale des utilisateurs d'Eclipse est très diversifiée, et si le premier marché reste les Etats-Unis avec une part de marché de 19%, la Chine arrive seconde avec 14% des téléchargements. La France, avec 5% des téléchargements conserve sa position dans le top 10.

Les 7 principaux piliers d'Eclipse, qui définissent autant d'écosystèmes, sont:

- Développement d'entreprise : avec en particulier PHP IDE (Zend&IBM), ou BIRT (création de bilan)
- Développement de service et embarqué : eRCP (Client Riche embarqué), CDT (C/C++)
- SOA : SOA Tools Products
- Gestion du cycle de vie des applications
- Application Internet Riches : Au sein de l'initiative openAjax, ou sur la plate-forme Flex (Adobe)
- RCP : Plate-forme client Riche
- Application Framework

II - Ralph MUELLER - Présentation de Callisto



La deuxième présentation était animée par Ralph MUELLER, responsable de l'écosystème d'Eclipse pour l'Europe, et portait sur la nouvelle approche de la Fondation Eclipse en particulier dans la nouvelle version Callisto.

Les projets Eclipse forment maintenant un ensemble complexe qui comme tous les projets de grande taille occasionne des problèmes de délais, de latence entre les releases des différents sous-projets, de promesses non tenues sur les délais de livraison. Il était donc nécessaire pour l'ensemble des entreprises voulant utiliser la plate-forme Eclipse de procéder à une longue phase d'intégration afin d'avoir des versions compatibles des multiples projets de la fondation Eclipse. Cet effort d'intégration se traduisait par des coûts d'utilisation de Eclipse plus élevés que prévus.

Lors du conseil d'expression des besoins de juin 2005, les adhérents ont donc demandé à la fondation Eclipse de prendre en compte ces coûts d'intégration.

Callisto est donc le résultat de cette initiative. Callisto intègre l'ensemble des projets Eclipse qui ont accepté de rejoindre volontaire cet effort de coordination et d'intégration entre les projets de la fondation. Dans la version 2005-2006, dix projets majeurs se sont donnés rendez-vous pour produire cette version.

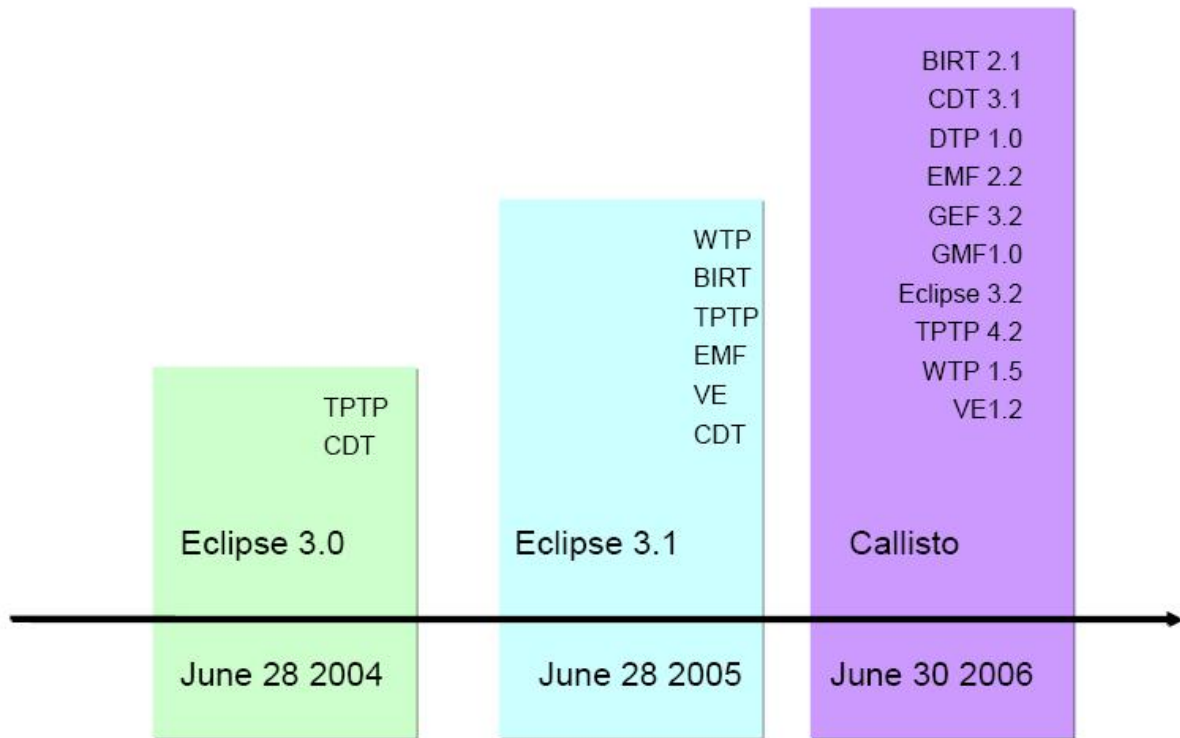
Le résultat est donc un ensemble de logiciels intégrés, représentant plus de 260 committers, issus de 15 organisations différentes, pour un total d'environ 7 millions de lignes de code. L'architecture de la plate-forme reste complètement modulaire et chaque projet conserve ses propres organes décisionnels. Il y a simplement eu un effort important fait sur la coordination des releases de ces différents projets, avec mise en concordances des plannings respectifs et des fonctionnalités attendues. Ce processus a nécessité des releases milestones fréquentes de la part des différents projets permettant de vérifier l'intégration des projets sur une base régulière.

L'avantage pour les développeurs est la possibilité de choisir les outils qui leur sont nécessaires sur une version unique et intégrée des différents outils. L'avantage est encore plus grand pour les éditeurs de logiciels qui peuvent dès lors avoir des montées de versions plus prévisibles et mieux orchestrées, réduisant ainsi les coûts de migration.

La principale nouveauté de la version Callisto de cette année est la version stand-alone de la plateforme OSGi (nom de projet : equinox), ainsi que le support de nouveaux systèmes comme OSX ou Vista.

Les projets ayant participé à l'initiative Callisto pour l'année 2005-2006 sont:

- BIRT 2.1 (reporting)
- CDT 3.1 (C/C++)
- DTP 1.0 (Data Tools Plateform)
- EMF 2.2 (Eclipse Modeling Framework)
- GEF 3.2 (Graphical Editor Framework)
- GMF 1.0 (Graphical Modeling Framework)
- Eclipse platform 3.2
- TPTP 4.2 (Test and Performance Tools Platform)
- WTP 1.5 (Web Tools Platform)
- VE 1.2 (Visual Editor)




Les projets participants à l'initiative Callisto

Un certain nombre de leçons peuvent être tirées de cette expérience, pour améliorer les choses les années suivantes:

- 1 Les features non documentés : L'intégration a été mise à mal lorsque certains projets ont modifié des features, non documentées, mais déjà utilisées par d'autres projets, ce qui a provoqué des ruptures de build
- 2 Le propriétaire des projets sont multiples : Il a donc été nécessaire de fournir un effort important pour mettre en place des plannings et des dates butoirs communes
- 3 La taille du projet était telle que le site d'update d'Eclipse a dû être revu pour pouvoir fournir les mises à jour nécessaires
- 4 Des problèmes de synchronisations des sites miroirs ont entraînés des erreurs dans le déploiement chez certaines personnes
- 5 La nécessité d'utilisation d'une interface commune de communication entre les projets
- 6 Le besoin de test dans l'espace entre projet. La question reste ouverte sur le fait de savoir qui est responsable de cette étape d'intégration

La prochaine version de cette initiative, à paraître en 2007 sera nommée Europa, non pas d'après le continent, mais d'après les lunes de Jupiter, qui fournissent leurs noms aux différentes releases. Les 10 projets déjà présents cette année seront de nouveau au rendez-vous l'année prochaine avec, espérons-le, encore plus de candidats pour cette initiative.

III - Wayne BEATON - Présentation de la plate-forme Eclipse RCP

 Wayne BEATON est "évangéliste" à la fondation Eclipse. Son travail est donc de nous apprendre à nous faire connaître les outils de la fondation, et on peut dire que sa présentation de Eclipse RCP est très alléchante.

Eclipse RCP (Rich Client Platform) est une plate-forme de développement de clients riches. Par clients riches, on entend des clients proposant aux utilisateurs des interfaces riches et faciles d'accès, en particulier via l'utilisation de nombreux Widgets.

Cette application peut tout aussi bien être une application en stand-alone. Eclipse RCP apporte alors des atouts comme la portabilité entre différents systèmes d'exploitation, tout en conservant le look&feel natif, les différents composants disponibles, ou le mécanisme d'update intégré. Si l'application est de type 2-tiers, il s'agit alors d'un client riche se connectant à une base de données distantes, qui lui sert de référentiel de données. Enfin, l'architecture 3-tiers permet de déporter une partie de la logique métier vers un serveur applicatif. L'application peut communiquer avec ce serveur via des web services. Généralement, les applications sont d'un type croisé entre les trois types ci-dessus, utilisant si nécessaire des mash-ups de web services, faisant appel à des bases de données distantes, tout en conservant une copie locale de données permettant un travail en mode déconnecté.

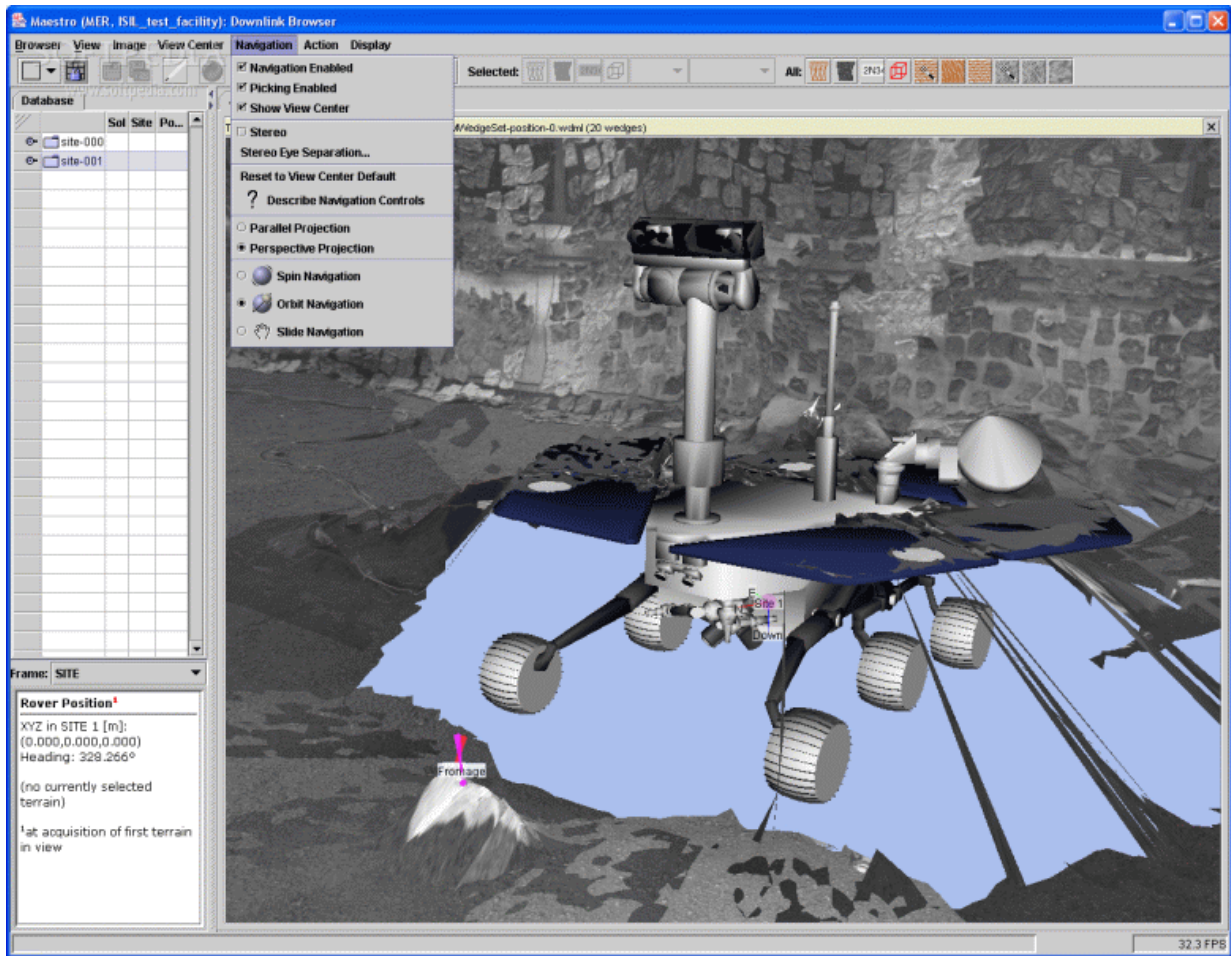
Un des gros reproche actuellement fait aux applications utilisant la plate-forme Eclipse RCP est leur look, proche de celui de l'EDI Eclipse. Mais si cette solution est celle par défaut, et la plus utilisée, ce n'est pas du tout une fatalité, et Lotus Notes "Hannover" est un très bon exemple de customisation du look&feel d'une application Eclipse RCP.

Les deux formes de clients disponibles aujourd'hui (léger ou lourd) sont la plupart du temps complémentaire. Ainsi, les utilisateurs occasionnels préfèrent ne rien avoir à installer sur leur poste (par exemple un utilisateur ayant à accéder à une banque en ligne) alors que les utilisateurs avec pouvoir (par exemple les gestionnaires de cette même banque) ont un besoin d'une interface ergonomique et riche qu'il est encore difficile d'atteindre avec un client léger.

La question suivante est: "Pourquoi utiliser Eclipse RCP?". Eclipse RCP nous offre toutes les fonctionnalités basiques d'un client riche (gestion des fenêtres et de l'interface, mises à jours automatisées, gestion de l'aide), est accompagné d'outils de développement performants, possède un look&feel natif sur toutes les plate-formes et sert de "middleware" en permettant aux développeurs de se concentrer sur les fonctionnalités plutôt que sur les détails techniques.

La plate-forme Equinox présente l'implémentation des spécifications OSGi, version R4, utilisée par Eclipse depuis des années. La spécification de l'OSGi vise, dans le cadre des programmes Java, à combler l'absence de notion explicite de composants au sein de Java. Les composants OSGi (appelés aussi 'bundles' ou plug-in dans une typologie Eclipse) sont des entités versionnées, définies déclarativement, qui vont être chargés dynamiquement par la plate-forme, qui s'occupera également de tous les mécanismes d'installation et de mise à jour. Chacun de ces composants définit ses dépendances, ses interactions avec les autres composants (parfois appelés points d'extension), et sa visibilité à l'exécution.

Eclipse RCP offre donc une occasion unique de construire de plate-forme métier, spécifique à un domaine. L'exemple le plus marquant de telles plate-forme est certainement Maestro, la plate-forme de la NASA, destinée au développement des interfaces de contrôles des engins spatiaux.



Maestro, plate-forme Eclipse RCP de la Nasa

IV - Janet CAMPBELL - Eclipse : propriété intellectuelle et modèles économiques



La dernière présentation de la matinée a été assurée par Janet CAMPBELL, "Legal Counsel & Manager: Intellectual Property" pour la fondation Eclipse.

La propriété intellectuelle dans Eclipse joue un rôle particulièrement important en cela que le code diffusé par la fondation est appelé à être revendu par les membres de la fondation. Il est donc essentiel de s'assurer que le code introduit dans la base de code d'Eclipse est bien ou libre de droit, ou sous une licence compatible avec la EPL (comme la licence Apache).

V - Philippe KRIEF - Jazz : Team collaboration platform for software development

La dernière présentation est animée par Philippe KRIEF, d'IBM et a pour objet la présentation de la plate-forme JAZZ, plate-forme de développement collaboratif développée par IBM (la génération suivante et bien plus ambitieuse du modélisateur UML Rational Rose).

Cette plate-forme se compose à la fois d'une application coté serveur mettant en place l'ensemble des services nécessaires à une équipe de développeurs (suivi des bugs, gestion des sources, communication instantanée, ...). Cette plate-forme coté serveur est hébergé grâce au module Equinox d'Eclipse, et peut donc être complétée par toute sorte de plugins de la même façon que l'IDE Eclipse. Cette plate-forme comporte également un module d'intégration continue, pour permettre le développement de projets par des méthodes agiles, nécessitant des livrables fréquents. La plate-forme est dans sa version de base hébergée sur Tomcat/Derby mais peut s'accommoder d'équipes de grande taille via l'utilisation d'outils plus puissants tels que WebSphere/DB2.

Coté client, le poste de travail du développeur se compose d'un client riche, basé sur Eclipse RCP et reprenant les fonctionnalités de l'IDE Eclipse, complété par des menus spécifiques aux fonctionnalités de la plate-forme cités ci-dessus. En particulier un client de chat permet à l'équipe de garder un contact continu, un moniteur du serveur d'intégration continue permet de se renseigner en permanence sur l'état du build, et des outils de suivi des consommations permettent à chacun des développeurs d'imputer correctement son travail (fait ou en cours), de façon à permettre à un chef d'équipe une bonne visibilité du travail effectué ou en cours, toujours dans le principe des développements agiles. Le client peut également être un simple navigateur web, se connectant à un frontal de type client léger riche (dans la vague du web 2.0).

VI - Les ateliers

L'après-midi, les présentations étaient réparties en deux ateliers, auxquels je n'ai malheureusement pas pu assister. Le premier était animé par Eliane FOURGEAU (TNI-Software) et portait sur les systèmes embarqués, le second, animé par Didier GIRARD (Groupe SFEIR) portait sur les application buisness.

Vous pouvez retrouver toutes les présentations sur: [Le site de la journée Eclipse - Now you can](#) (attention, la plupart des présentations sont en anglais).