

[Main.Interview de Emmanuel Bernard](#)

Entretien avec Emmanuel Bernard

Cyberspace, le 10/11/2005.



Bonjour Emmanuel, une premi re question classique : qui es-tu, o  vis tu ?

Bonjour, je suis... Emmanuel Bernard, j'ai 28 ans, je bosse en ce moment de chez moi   Courbevoie pour JBoss Europe au sein de l' quipe Hibernate. Je suis aussi expert dans le groupe d'expertise EJB3 (JSR 220).

Pourquoi l'open source ?

Pour le fun et la gloire ;-) J'ai commenc    utiliser l'open source chez mon ancien employeur. Puis j'ai eu l'occasion de participer au projet Hibernate de mani re plus active (aide sur le forum, patchs etc...). Quand tu bosses sans contrainte, c'est que tu aimes vraiment ce que tu fais. C' tait un moyen de m'amuser tout en travaillant et c' tait b n fique et pour moi et pour ma boite (on utilisait abondamment Hibernate et d'autres projets open source), je trouvais que c' tait une juste r tribution par rapport   ce que l'open source nous apportait. Un autre point, on te juge sur ce que tu fais, un point c'est tout, c'est assez agr able. Et j'avoue que c'est assez sympa quand des utilisateurs inconnus appr cient ton travail et te remercient :-)

Pourquoi l'open source professionnel ?

Il y a plusieurs raisons entrem l es. En voici quelques unes :

- parce que je peux vivre de ce qui me pla t
- parce que  a a du sens, offrir aux entreprises qui en ont besoin le support   la source
- parce que mon travail aide certains   gagner de l'argent, pourquoi pas moi
- parce que c'est un des seuls moyens de p renniser son projet OSS sur le long terme

PJBUG

- parce que le contact avec les utilisateurs est direct, ce qui nous aide à garder les pieds sur terre en ce qui concerne les fonctionnalités et la qualité du code, ce qui n'est pas le cas pour un éditeur closed source
- parce qu'aujourd'hui en France, le management reconnaît très mal le travail technique (intellectuellement et financièrement) : qui n'a jamais écrit une ressource dans une grille MS Project ! Dans l'Open source professionnel, en tous cas les pure players, la reconnaissance du travail est celle

Quelles sont tes fonctions chez JBoss Inc. ?

Je suis core developer dans l'équipe Hibernate. J'ai plus spécifiquement en charge le lead développement d'Hibernate Annotations et d'Hibernate EntityManager, les deux composants qui implémentent la spec EJB3 en s'appuyant sur le moteur Hibernate. Ces deux composants sont également l'implémentation de la persistance dans JBoss EJB3. Je passe également 20% de mon temps soit à répondre des cas de support, soit à donner des formations sur Hibernate soit à faire du consulting sur Hibernate. C'est assez agréable, ça permet de rester sur le terrain.

Depuis quand travailles tu chez JBoss Inc. ?

Depuis quelques mois mais j'étais en relation étroite avec l'équipe Hibernate depuis fin 2003.

Que faisais tu avant ?

J'étais architecte technique nouvelles technologies et lead developer de quelques applications à la Fnac. On y a constitué un framework technique Java/J2EE à la Spring avant l'heure (c'est-à-dire un framework technique de bonnes pratiques et d'intégrations de briques open sources). Mais j'ai toujours gardé un pied dans le développement quotidien d'applications de la vraie vie(tm), quelles soient nouvelles ou des évolutions d'applications antérieures. C'est d'ailleurs le cas de tous les membres de l'équipe Hibernate, on a tous développé des applicationselles avant de travailler à temps plein sur Hibernate. Dans une vie antérieure, j'ai également pas mal travaillé sur les technologies MS (ASP, VB, procédures stockées SQLServer)

Quelles sont tes contributions sur Hibernate ?

Par ordre chronologique :

- aide sur le forum / documentations
- développements de quelques unes des fonctionnalités d'Hibernate3 comme les statistiques
- développement et lead d'Hibernate Annotations
- développement et lead d'Hibernate EntityManager

Qu'est-ce qui selon toi différencie Hibernate des autres ORM ?

Premièrement le nombre d'utilisateurs : aucune solution d'ORM n'a une base d'utilisateurs aussi grande, ça a contribué à la maturité et la richesse fonctionnelle d'Hibernate et à la maturation du marché de l'ORM dans le monde Java. Deuxièmement la cause et la conséquence de la première, les fonctions ont été développées pour répondre des besoins basés sur des cas d'utilisations réels. Depuis Hibernate 3, le "retard" fonctionnel avec les solutions ORM du marché a été comblé, on est maintenant dans une phase d'innovation par rapport aux autres ORM et dans d'autres domaines. Ensuite, la qualité de la documentation a toujours été très bonne pour un projet open source et même dans le monde de l'ORM en général.

Quelles sont tes contributions sur EJB3 ?

PJBUG

Je fais partie du comité d'expertise EJB3 (JSR-220), et j'ai pris de lead du développement de l'implémentation pour Hibernate. La part significative du travail a été finalement le développement du configurateur d'annotations. Les versions alpha et beta sont apparues très vite après la publication des premiers brouillons publics de la spec, ce qui a permis au comité d'avoir des retours d'expérience concrets des implémenteurs mais surtout des utilisateurs.

Comment positionnes tu EJB3 vis à vis des frameworks de type Spring ou HiveMind ?

Vaste sujet, vous l'aurez voulu ! Le premier est un standard, les deux autres sont des produits. EJB3 est en fait constitué de plusieurs choses :

- la persistance
- les "composants" (les EJB sessions, ...)
- son rôle de glue et d'intégration aux autres specs comme JTA

Les "implémentations" EJB3 et Java EE 5 en général fournissent beaucoup de choses :

- la persistance
- le pool de connections à la base de données
- un gestionnaire de transaction
- un système de clustering
- ...

Les frameworks comme Spring et HiveMind fournissent en fait une glue et un modèle de programmation pertinent pour utiliser ces technologies mais ne fournissent pas d'implémentation en temps que telle, vous aurez donc besoin de vous appuyer sur une implémentation sous-jacente pour chacune des technologies que vous devez utiliser. Ces frameworks facilitent en plus l'implémentation de tests unitaires en réduisant le cycle code - test spécialement lorsque l'on utilise des méthodologies agiles. Le succès vient en fait de la rapidité de développement et de l'augmentation de productivité. EJB3 a grandement simplifié le modèle de programmation, bien sûr au niveau de la persistance mais également dans le développement de composants EJB3. D'abord par l'injection de dépendances pilotée par les annotations (ou optionnellement par un descripteur de déploiement). Également par l'intégration du cycle de vie des contextes de persistance (session dans le langage Hibernate) dans les composants EJB3. Dans un EJB3 session, l'utilisation d'un entity manager (une session) se limite à une annotation : l'ouverture de l'entity manager, son partage au sein d'une même transaction JTA, son flush, sa fermeture sont automatiquement prises en compte par le container. Le nombre de lignes de code et d'erreurs est **beaucoup** plus petit. Quand on compare donc la productivité en nombre de lignes de code (et de clarté) du développeur EJB3 et du développeur utilisant Spring, on est probablement au pire comparable (l'utilisation des annotations est un vrai rafraîchissement même comparé aux fichiers de config de Spring, sans parler des descripteurs de déploiement J2EE 1.4).

Deuxième point les tests unitaires hors du container : tout le problème vient du fait que les tests dans le container sont long à lancer. Bonne nouvelle encore une fois, le règne des containers monolithiques a sonné. JBoss propose un container EJB3 embarquable dans votre application (main, swing ou un autre serveur d'application), et même dans vos tests unitaires. On peut donc tester son code en utilisant les EJB3 sans les désavantages d'un container classique. EJB3 a appris sa leçon en terme de productivité de développement et de test unitaire en s'inspirant des succès de Spring et des autres conteneur à injection de dépendances. EJB3 apporte également une vraie intégration pensée entre les composants et les outils de persistance. Comparer Spring ou HiveMind à JBoss Embeddable EJB3 est injuste dans la mesure où à Spring vous devez ajouter une datasource, le gestionnaire de transaction JTA, éventuellement l'implémentation MDB, etc...

Bon, j'arrête. Ne me croyez pas, téléchargez et testez par vous même.

Comment travailles tu avec tes collègues de JBoss Inc. qui sont dans d'autres pays ?

Beaucoup par mail, et par Yahoo IM, un peu par Skype. En ce qui concerne l'équipe Hibernate chez JBoss, on ne s'est retrouvé qu'une seule fois tous ensemble je crois. Mais ça fonctionne bien. CVS est le

PJBUG

point central, pour le reste, on est focalisé sur des fonctionnalités un peu différentes donc pas de clash. Evidemment quand on se retrouve ensemble, on est obligé de faire la fête ;-)

Quels sont tes outils de développement habituels ?

- Windows XP
- IntelliJ IDEA 5, c'était le seul IDE qui supportait les annotations correctement quand j'ai travaillé sur Hibernate Annotations. J'utilisais Eclipse avant, mais je n'y suis jamais revenu
- Tortoise CVS
- HSQLDB, MySQL, PostgreSQL

Sur quoi penses tu travailler à l'avenir ?

Finir Hibernate Annotations et Hibernate EntityManager, aider sur les Hibernate Tools EJB3 et finaliser la spec EJB3. On a aussi quelques idées sur des produits plus haut niveaux qui s'appuieraient sur JEMS...

Le mot de la fin ?

C'est cool de voir un groupe utilisateur se former, bonne initiative. A quand un jbug Grenoble dans un resto d'altitude !

[Main.Interview de Emmanuel Bernard](#) (fr)

Créateur: XWiki.Admin Date: 2006/10/06 14:16

Dernier Auteur: XWiki.Admin Date: 2006/10/06 14:21

[Creative Commons](#)