Celerio

Accélérateur de développements Java

Décembre 2007
Version 2.0

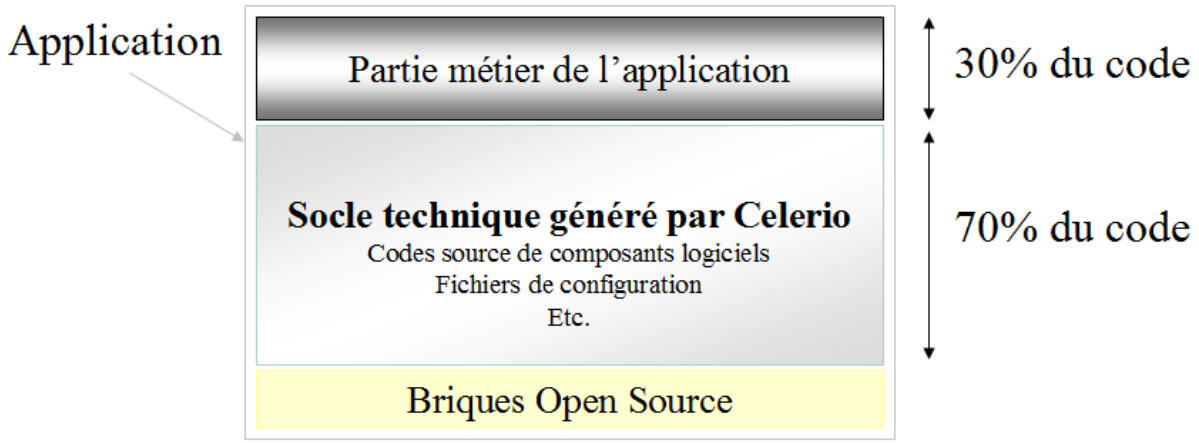
Contact
info@jaxio.com

Préambule

Celerio de Jaxio permet d'**injecter** dans tout nouveau projet d'application web java, **savoir-faire et expérience** en générant automatiquement, à partir d'un modèle de base de données, le socle technique de votre application.

Le socle technique, généré **en quelques secondes**, correspond à ce qu'une équipe d'ingénieurs expérimentés mettrait au point au bout d'un temps se chiffrant en plusieurs semaines voire plusieurs mois selon la complexité du modèle.

Ce socle est écrit suivant **les règles de l'art** et utilise les **meilleures technologies** Open Source. Votre applicatif métier utilise les composants générés prêts à l'emploi.



Enjeux et réalités

Les équipes de développement logiciel travaillent dans un contexte de plus en plus dynamique :

- Les technologies évoluent rapidement et il faut se former en permanence
- Il faut livrer de nouvelles applications ou leurs évolutions suivant des cycles courts.
- Les logiciels à développer sont de plus en plus riches fonctionnellement
- Le périmètre technique est vaste: montée en charge, internationalisation, etc.

Le directeur technique, pour maitriser les coûts et les délais, doit :

- Faire avec son équipe les bons choix technologiques
- Mettre en œuvre une méthodologie « Agile »
- Garantir la qualité : code développé suivant les règles de l'art, test unitaires, etc.
- Garantir la survie du projet lors de turnover dans l'équipe technique

Outre la complexité intrinsèque à tout projet de développement informatique – comprendre, définir, modéliser, planifier, organiser, développer, tester, packager, déployer, maintenir- les développeurs, architectes et directeurs techniques doivent aussi faire face aux réalités du terrain :

- Les négociations commerciales ou décisions prennent du temps, les projets ne démarrent jamais à l'heure prévue. Une fois le contrat signé ou la décision prise, une pression immense retombe sur les épaules des développeurs, il faut aller vite et rattraper le temps perdu en phase d'avant vente.

- Rien n'est figé, les spécifications changent en cours de développement, il faut savoir s'adapter immédiatement sans remettre en cause la qualité du projet et les délais.

- Les équipes tournent, et il faut certes s'en réjouir car cela permet aux ingénieurs de vivre de nouvelles expériences et aux dirigeants de profiter de ces expériences. Mais le challenge pour le directeur technique est que le code développé soit repris avec succès par les nouveaux venus.

- Le code source du socle technique représente la majorité du code source de l'application. C'est la clé de voute de votre application. Légitimement, les équipes de développement y consacrent souvent plus de la moitié du temps imparti et sont du coup contraintes de développer la partie métier après, très souvent trop tard ! Le client final intéressé par la partie fonctionnelle ne comprend pas que cela prenne autant de temps à être développé.

Solutions actuelles

Pour répondre aux enjeux et aux réalités, nous avons identifié quatre approches. Chacune de ces approches a un intérêt mais aussi des contreparties qui selon vos besoins et objectifs pourraient s'avérer rédhibitoires.

UML et la génération de code

Des outils dits MDA (Model Driven Architecture) s'appuyant sur le formalisme UML ou UML 2 définit par l'OMG existent. Ils permettent de définir un modèle métier puis de générer une partie du code correspondant automatiquement.
Leur utilisation est adaptée pour des projets d'envergure, comme par exemple des projets militaires.

Cependant, l'utilisation de tels outils MDA et d'UML au quotidien par des équipes techniques n'est pas adaptée pour des projets qui doivent être « Agiles », en effet:

- UML reste complexe et peu maitrisé par les développeurs
- Le code des composants générés n'est pas prêt à l'emploi, il s'agit souvent de squelette à complémenter.

Développement d'une application vierge

Pour accélérer et cadrer les lancements de projets, les SSII créent une « application vierge », c'est-à-dire une application minimale de référence servant au démarrage des nouveaux projets.

Cependant le temps gagné reste minime car sans modèle on ne peut générer la majorité du code source qu'il faudra du coup écrire à la main. Pour aller plus loin, une équipe d'ingénieurs expérimentés, dédiée en interne, devient rapidement nécessaire.

Délocalisation

Pour réduire drastiquement les coûts il est tentant de délocaliser les développements informatiques. Il faut cependant être conscient de certaines contreparties qui peuvent dans la durée s'avérer rédhibitoires :

- On s'expose à une perte progressive de la maitrise technologique et l'on risque d'être progressivement pris en otage par le prestataire offshore
- Le coût unitaire est certes plus bas mais le rendement n'est pas aussi bon que celui d'un ingénieur avec qui l'on peut communiquer efficacement. On y perd souvent en réactivité et en agilité.
- Le choc culturel, en particulier pour la France qui n'est pas de culture anglophone est un frein lorsqu'il faut travailler à distance et en anglais.

Changement de technologie

Alors que l'on observe une pénurie de développeurs et d'architectes Java, apparaissent sur le marché des nouveaux langages, faciles à appréhender, tels que Ruby, qui permettent de réaliser des applications web rapidement.

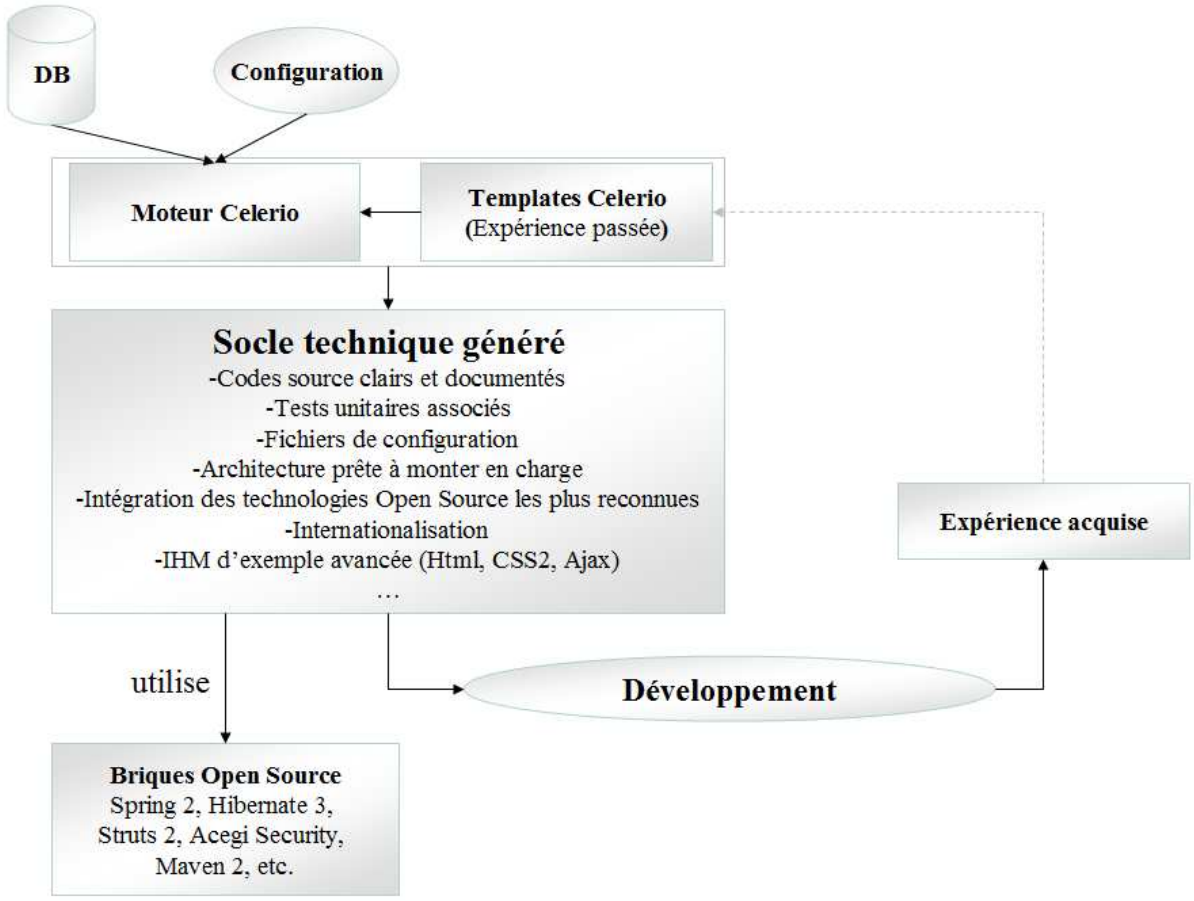
Ces langages sont appréciés pour leur capacité à accélérer grandement le démarrage des projets, cependant le temps gagné au début est souvent perdu lorsqu'il faut traiter des sujets plus complexes, déjà résolus en Java et donc plus simples à traiter en Java.

La solution Celerio

Principe

Celerio est une solution pragmatique basée sur des standards et les meilleures pratiques de notre industrie. L'objectif de Celerio est d'industrialiser la phase de développement du socle technique qui est répétitive et freine tous les développements informatiques.

A partir d'un modèle de base de données, de conventions simples et/ou d'un fichier de configuration, Celerio permet de générer les différents composants logiciels constituant le socle technique de votre application.



Les composants logiciels du socle technique sont prêts à l'emploi. Codés suivant les règles de l'art, les composants peuvent être configurés, étendus, modifiés ou remplacés.

Les composants utilisent les meilleures briques Open Source. Ainsi, le socle technique bénéficie des caractéristiques intrinsèques de ces briques. Notamment le cœur de votre application est prêt à monter en charge en vue d'une mise en production à l'échelle industrielle.

Dégagé de la mise en place des briques de base récurrentes à toute architecture d'applications Java, vous développez votre application en un temps record en vous concentrant sur l'implémentation des besoins spécifiques métiers.

Vous restez libre et autonome. Le code source généré par Celerio n'introduit aucune dépendance propriétaire et le produit Celerio n'intervient pas à l'exécution de votre application.

Vous accélérez vos développements en toute tranquillité.

Ce que vous apporte l'usage de Celerio

Vitesse

- Celerio permet aux développeurs de se concentrer très rapidement sur la partie métier de l'application. Cela ne signifie certainement pas que les développeurs ne doivent plus maitriser la conception du socle technique, mais simplement qu'elle ne devient plus un frein au développement de la partie métier.

- Celerio, permet de faire retomber la pression que subissent les équipes techniques, en optimisant la phase d'initialisation du projet. Moins d'une semaine après le démarrage du projet, la première version de l'application peut être livrée.

Maitrise technique

- Celerio, cadre dès le départ le socle technique du projet en y injectant les meilleurs pratiques et une architecture de référence.

- Nous pensons que le résultat de l'automatisation doit être irréprochable dans tous les domaines. Aussi, le code source du socle généré par Celerio est-il, lisible, documenté, testé unitairement et forme une architecture nominale.

- Pour le directeur technique, l'utilisation de Celerio par ses équipes permet de définir une ligne de conduite, de maintenir une homogénéité d'un projet à l'autre, de réduire les cycles de développement, d'attirer les meilleurs développeurs en leur proposant d'utiliser les meilleures technologies Open Source et enfin d'améliorer la qualité globale des développements dont il a la responsabilité.

- Celerio n'embarrasse pas les développeurs avec un modèle abstrait intermédiaire, dont la mise à jour est souvent délaissée lorsque les spécifications changent ou que les développements prennent du retard. Le modèle de référence utilisé par Celerio est celui de la base de données, qui est de fait toujours à jour.

Autonomie

- Le code généré permet aux développeurs juniors de se former durant le projet et par mimétisme de produire un code de qualité. Ainsi, ils montent en compétence plus rapidement.

- Pour les développeurs plus expérimentés, blasés par l'aspect rébarbatif de la mise en œuvre d'un nième socle technique, le code généré est une aubaine permettant de se consacrer pleinement à la valeur ajoutée de l'application et élargir ses connaissances techniques

- Celerio par son approche consistant à n'utiliser que des briques Open Source reconnues et ayant fait leurs preuves permet aux développeurs de valoriser leur expérience lorsqu'ils quittent l'équipe et d'être très rapidement opérationnels lorsqu'ils rejoignent l'équipe.

Synthèse

Celerio permet d'accélérer vos cycles de développement, d'augmenter leur qualité, et de créer un environnement d'excellence dans lequel vos développeurs pourront s'épanouir et être encore plus efficaces.

Nous pouvons résumer les principaux bénéfices de Celerio en trois points :

Vous restez « au top » des technologies

- Standardisation de facto et grande qualité du code généré
- Réduction des coûts de développement et de maintenance
- Fonctionnalités avancées en phase avec les tendances du moment

Vous restez autonome

- Conservation de la maîtrise technique
- La qualité du code généré facilite les transferts de connaissances
- « No Celerio Inside »: Celerio n'intervient pas à l'exécution

Vous allez beaucoup plus vite !

- 100% du code source généré est déjà testé
- Focus sur la partie métier de l'application sous une semaine
- Homogénéité du code généré d'un projet à l'autre