



Adaptation des interfaces à l'environnement

Etude technologique sur les solutions web : **Phonegap vs Sencha**

Auteur : Paul Lavoine

Encadreur : Mme Dery

Date de première rédaction : 02/11/2014

Date de dernière modification : 12/11/2014

Quel contexte d'usage ?

- **Plateforme : téléphone mobile**
- **Environnement de développement : html5, JavaScript, CSS**
- **Environnement de qualification : iOS, Android ...**
- **Utilisateur : tous les utilisateurs**

Adobe Phonegap et Sencha Touch sont des frameworks multiplateformes permettant de créer des applications mobiles pour différentes plateformes (iOS, Android ...) en HTML, CSS et JavaScript. Les applications ainsi créées ressemblent aux applications natives créées sur appareils mobiles.

Ces frameworks ont pour but de permettre aux développeurs les plus aisés en HTML, JavaScript et CSS de développer rapidement et facilement des applications sur smartphone. Ce rapport a pour but de comparer ainsi ces deux frameworks tout en se positionnant par rapport à la plasticité.

Quel moment ?

Aussi bien Phonegap que Sencha permettent de faciliter la vie du développeur (lors de la conception).

Ces frameworks se positionnent comme des adaptateurs de modèle qui « convertissent » un langage en un autre (ils encapsulent le code html et l'affichent dans une webview). L'application créée en HTML5 se situe au niveau de l'ihm abstraite dans la phase de conception d'une application, c'est le modèle. On peut difficilement situer ces frameworks dans la pyramide à 4 niveaux dans le cas où ils vont « ré-implémenter » le code html5 pour une plateforme. Concrètement, l'objectif de ces frameworks est de réduire le fossé qui sépare les applications web de celle natives. Ils vont donc permettre d'encapsuler l'application web dans une application native et vont lui fournir des outils supplémentaires par l'intermédiaire de plugins par exemple. Cependant, l'utilisation des éléments natifs porte à dire que ces frameworks se situent au niveau de l'interface utilisateur final. Mais l'utilisation de divers plugins nous laisse penser que ces 2 outils sont situés à une IHM concrète. Pour finir, le fait que ces frameworks soient cross plates-formes nous place en haut de la pyramide au niveau de l'interface abstraite. D'après le modèle MDA, le 3^{ème} niveau de la pyramide est une PIM, « platform independant mobile ». L'utilisation de Phonegap et Sencha peut se faire sous divers plates-formes mobiles et c'est pourquoi nous situons ces frameworks à ce niveau de la pyramide de modélisation.

Quelle solution ?

Une des majoritaires différences à spécifier, est que Sencha est plus adapté pour construire l'interface web. En effet, Sencha utilise différents éléments afin de construire plus aisément une interface telle que le design pattern MVC et l'utilisation de composant pour la construction des vues. Les applications ainsi créées possèdent une architecture cliente très riche en permettant une abstraction entre les données, les traitements et l'interface utilisateur. C'est un framework complet composé de nombreux composants : carrousels, formulaires, gestion complète des événements, touch, swipe.

De l'autre côté, Phonegap nous laisse libre accès et n'oblige à rien. De plus, il permet l'accès à un plus grand nombre d'éléments natifs (caméra, GPS ...). Les applications actuelles requièrent pour la plupart beaucoup l'utilisation des différents éléments natifs des smartphones et c'est un

des gros désavantages de Sencha, qui ne peut pas gérer des fonctionnalités telles que l'accéléromètre, la localisation etc.... D'autre part, ce framework permet aussi une utilisation facile d'une grande panoplie de plugins. Ces plugins permettent la création de widget que l'on retrouve assez couramment dans nos applications actuelles, tel que la connexion via Facebook, l'utilisation du flash etc...

Enfin, il est possible de combiner Phonegap avec d'autre framework tel que jQuery. Il s'agit d'un « framework qui permet, à partir de pages HTML5/CSS3, d'obtenir des écrans et des composants graphiques adaptés au rendu d'un écran de smartphone ou tablette. »

Votre avis ?

- **Phonegap :**

Avantage :

- Les gens ont l'habitude d'html, CSS, JavaScript
- Développement & Déploiement des applications rapides
- Donne accès à une majorité de fonctionnalités natives
- Utilisation des plugins possible et facile
- Bonne intégration avec jQueryMobile

Inconvénients :

- Tous les plugins ne sont pas accessibles
- Manque de widget, transition, contrôles standards

- **Sencha :**

Avantage :

- Framework MVC complet

Inconvénients :

- Performance mauvaises
- Peu d'accès aux éléments natifs

Si on est un professionnel du JavaScript et aimant du MVC, je préconise donc Sencha qui est presque exclusivement en JavaScript. En revanche, si l'on est un développeur d'interface utilisateur habitué au html, CSS et jQuery, Phonegap serait plus conseillé.

L'utilisation d'un framework plutôt que de l'autre implique forcément des avantages et des inconvénients. C'est pourquoi la combinaison des 2 frameworks semble une idée intéressante. En effet, l'utilisation de Sencha permet d'avoir un code qui suit l'approche MVC pour un développement rapide et économique, mais ses performances sont néanmoins mauvaises. De l'autre côté Phonegap permet un accès presque complet aux éléments natifs, et donc possède un grand intérêt par ses performances et son utilisation de plugins. De plus il permet de déployer sur plusieurs plateforme (seulement iOS et Android pour Sencha).