Adaptation des IHM

Vincent Benazet

07/11/2014

Encadreur : Mme Dery

Table des matières

I. Introduction

Depuis quelques années, on constate une forte diversification des appareils mobiles.

Dans un premier temps nous étudierons les cross-platform à travers la comparaison de deux technologies, à savoir, PhoneGap et Appcelerator. Nous mettrons en avant leurs contextes d'usage (au travers des plateformes) ainsi que leurs avantages et inconvénients. Nous illustrerons chaque technologie par un exemple.

Dans un second temps nous étudierons un article de recherche visant à analyser la plasticité des interfaces. Nous expliquerons le terme de plasticité et en quoi cette notion est devenue essentielle dans la conception des IHM (Interaction Homme Machine).

Pour bien débuter, quelques notions clés des IHM.

- Quel est l'objectif d'une IHM : Le but est d'un IHM est de faire le lien entre l'utilisateur et l'application. Il s'agit de permettre à l'utilisateur d'interagir avec le système.
- Qu'est ce qui constitue une IHM : Les principaux éléments qui constituent une IHM sont le contenu (bouton, liste déroulante), des techniques d'interactions (vocale, tactile, tangible), et la disposition du contenu (les layouts).
- En quoi les IHM deviennent de plus en plus importante : C'est le point d'entrée de l'application. Son enjeu est central car l'utilisateur ne juge souvent que l'IHM dans son comportement d'achat.

II. Présentation et analyse comparative des technologies

L'étude suivante concerne les cross-platform. Les objectifs du concept de cross-platform sont de diminuer le coût et le temps de développement ainsi que de viser un marché plus large. Cela permet de développer des applications pour plusieurs plateformes.

A titre d'exemple, une application fonctionnera aussi bien sur IOS et Android avec le même code. On gagne du temps (code unique pour deux plateformes) et le public visé est plus large (App store et Play store).

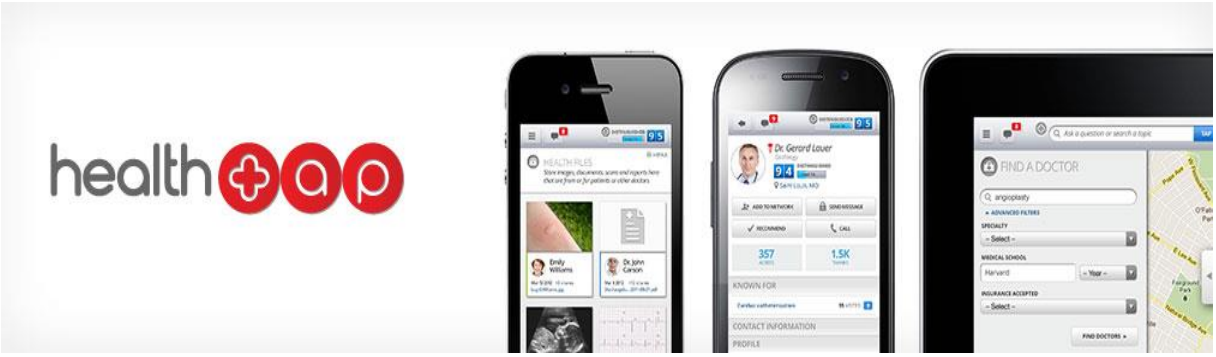
Le cross-platform possède quelques désavantages, notamment le fait que l'on n'a pas accès à toutes les fonctionnalités du mobile et on perd aussi en performance.

1) PhoneGap

PhoneGap est un framework open source qui permet la création des applications mobiles multiplateformes avec les technologies traditionnelles du web (HTML, CSS, JavaScript). Il crée un pont entre le code JavaScript et le code Natif de la plateforme visée pour accéder au matériel (appareil photo).

Idée principale de PhoneGap : On prend le langage natif de chacune des plateformes et on en crée un framework qui expose des interfaces pour le développeur JavaScript.

Prenons un exemple d'application développé grâce à PhoneGap. HealthTap est une application disponible sur les plateformes Android et IOS, permettant d'obtenir des réponses de médecin en rapport à des maladies ou des symptômes.



2) Appcelerator (Titanium Mobile)

Appcelerator Titanium est un framework open source destiné aux applications mobiles. Il est fondé sur des technologies web permettant aux développeurs web à utiliser leurs compétences pour créer des applications natives iOS et Android.

Idée principale d'Appcelerator Titanium : Fournir une machine virtuelle JavaScript permettant d'accéder au système natif, et ainsi de développer des applications natives en JavaScript.

Prenons un exemple d'application développé grâce à Appcelerator Titanium. Paypal est une application permettant de payer des achats en toute sécurité, gérer un compte.

3) Comparaison PhoneGap / Appcelerator

a. Les points communs

- Les deux technologies se basent sur les technologies web. Elles permettent aux développeurs web de faire des applications mobiles sans avoir la connaissance dans les langages de programmation propre aux plateformes mobiles.
- Elles permettent de déployer leurs applications sur les différents magasins (App store et Play store). Cela permet de toucher un grand public.
- Elles permettent l'utilisation d'un code unique pour différentes plateformes. Il en résulte un gain de temps et une réduction des coûts de développement.

b. Les différences

La première différence que nous pouvons observer concerne la compatibilité de PhoneGap et Appecelerator sur les différentes plateformes.

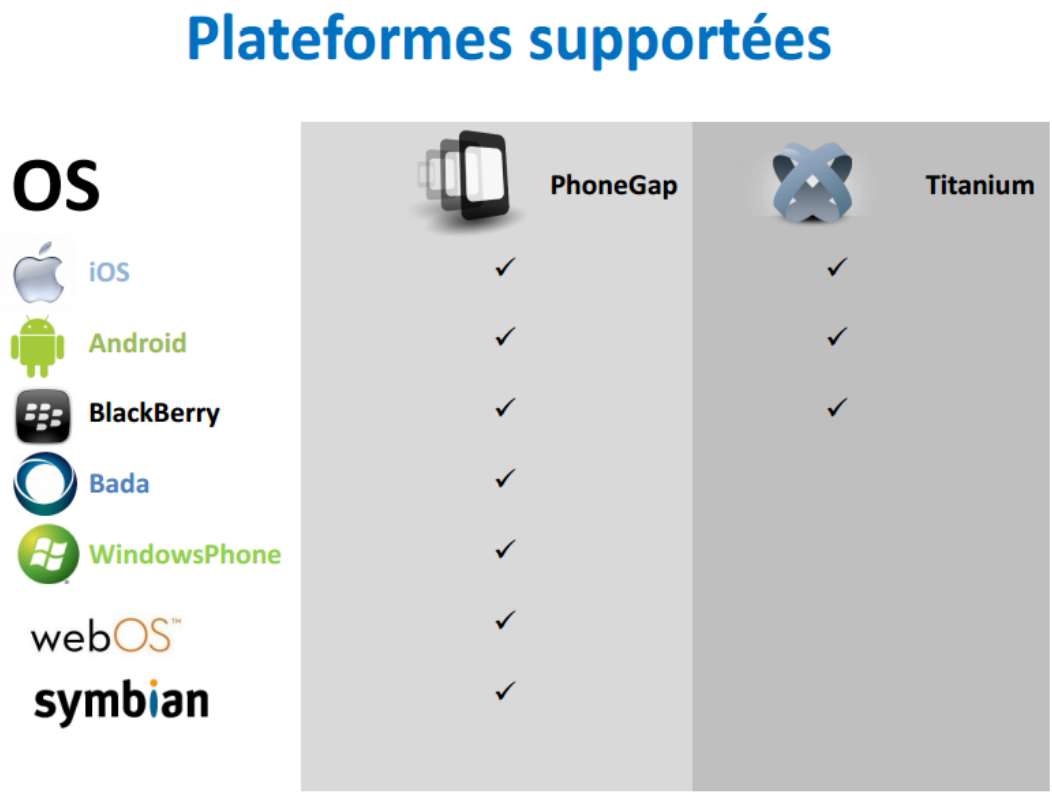


Figure 1

On remarque que PhoneGap touche un plus grand nombre de plateformes (7 contre 3pour Appcelerator). Néanmoins si on se place au niveau du nombre d'utilisateur Appcelerator cible tout de même 93% des utilisateurs de mobiles. PhoneGap s'adapte plus facilement aux différentes plateformes. Leur politique est « code once, deploy everywhere ». Cependant la large utilisation d'Android et IOS fait atténue cette différence.

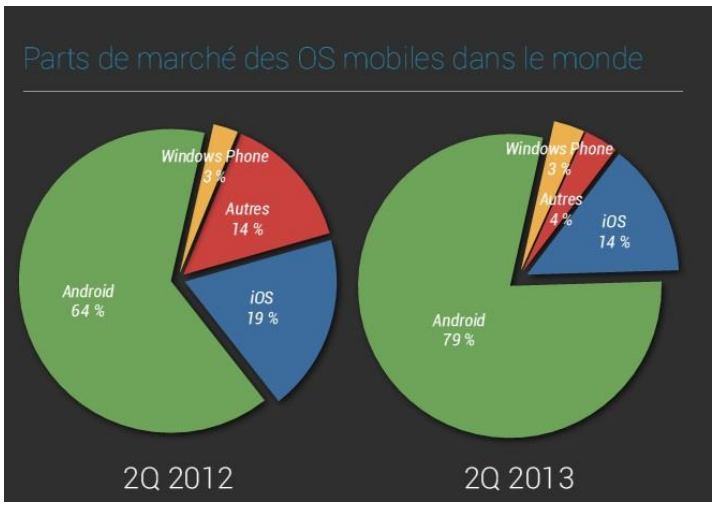


Figure 2

Maintenant comparons le framework. Pour ce qui est d'Appcelerator le point essentiel est qu'il accède aux fonctionnalités natives. Donc on aura une meilleure performance et surtout des IHM plus riches et qui ressemblerons plus à des applications natives. PhoneGap aura les inconvénients des applications Web (IHM moins riche et expérience utilisateur totalement différente). Voici un petit descriptif de qui est supporté par les deux technologies.

Figure 3

Pour ce qui est du développement, il est plus facile de développer avec Appcelerator. En effet la mise en place de PhoneGap (installation) est très contraignante et peu intuitive. De plus il n'existe pas un IDE (**Integrated Development Environment)** dédié pour PhoneGap.

Etudions maintenant un aspect important, le coût du développement. Avec PhoneGap un code unique permet de déployer une application sur sept plateformes. Pour ce qui est d'Appcelerator, étant donné qu'on développe des applications natives, il faut tout de même adapter notre code aux différentes plateformes. Ce qui est plus chronophage en comparaison de PhoneGap et donc qui engendrait d'avantage de coût de développement.

c. Synthèse

	PhoneGap	Appcelerator
Langages utilisés	✓	
Plateformes supportées	✓	
Richesse		✓
Facilité de développement		✓
Coût du développement	✓	

d. Mon point de vue

Si je me place dans l'optique d'utiliser une solution cross-platform, cela veut dire que j'ai des contraintes au niveau du coût du développement (un temps restreint). A mon niveau, je préfère développer plusieurs applications natives pour différentes plateformes par soucis de richesse d'interface et de performance quitte à perdre du temps. Donc par contrainte de temps je chercherai donc une solution me permettant de toucher le plus de plateformes en un minimum de temps de développement. Pour ces deux raisons je choisirai PhoneGap. Je m'appuie souvent lorsque j'ai le choix entre deux solutions, au taux d'utilisation des solutions (PhoneGap : 34% Appcelerator : 21% référence : Cours de Mme Dery Plasticité des interfaces). En effet, plus une solution est utilisée plus la communauté est importante, ainsi j'augmente mes chances de trouver des réponses à mes éventuelles questions grâce aux tutoriaux, documentations et aides diverses.

III. Etude d'un article de recherche

L'article que nous allons étudier est « A Reference Framework for the Development of Plastic User Interfaces » écrit par David Thevenin, Joëlle Coutaz et Gaëlle Calvary.

Avant de commencer nous allons rappeler quelques notions clés :

- Plasticité : Par analogie à la plasticité d'un matériau, la plasticité d'une IHM dénote sa capacité à s'adapter aux contraintes matérielles et environnementales dans le respect de son utilisabilité.
- IHM multicibles : Une cible se définit par le triplet <classe d'utilisateurs, plateforme, environnement >. Donc une IHM multicible va sélectionner plusieurs cibles.
- Plateforme : Correspond au support matériel et au support logiciel à la base de l'interaction.
- Environnement : Correspond au milieu dans lequel est utilisé l'IHM (lieu, jour/nuit)
- Un modèle est une description, une spécification partielle du système.
- Un méta-modèle est un modèle qui définit le langage pour définir des modèles.

Cet article traite de la plasticité des interfaces utilisateurs. Les auteurs mettent en avant un framework permettant de structurer le développement des interfaces utilisateurs adaptative. La structure mise en évidence par les auteurs est résumé par le modèle suivant.

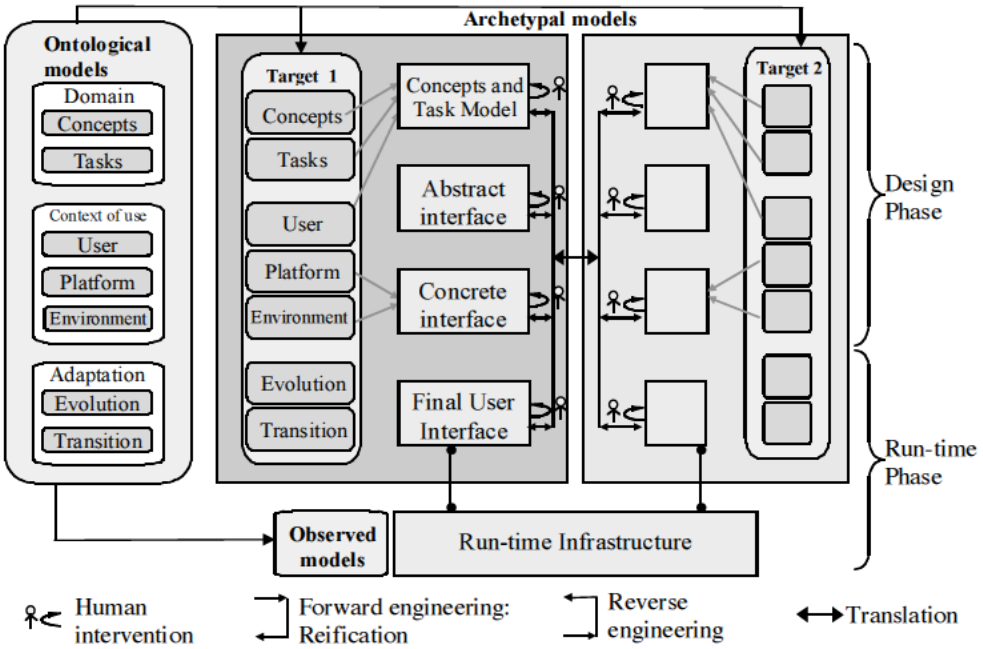


Figure 4

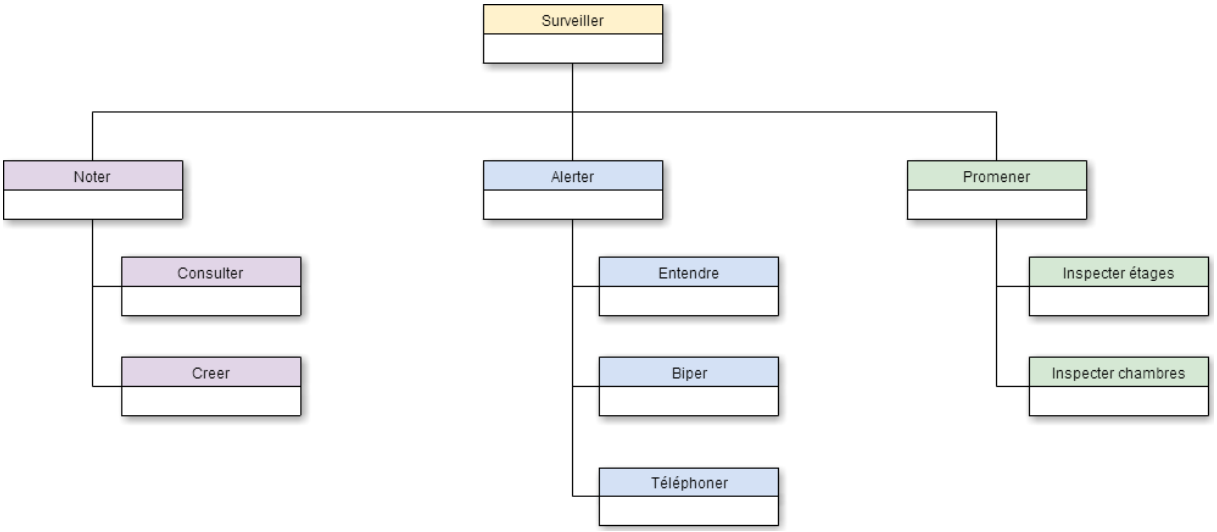
Leur solution est donc une approche basée sur les modèles. Il est composé de trois types de modèles qui correspondent à une étape dans la génération d'une IHM.

Les « Ontological models » sont des méta-modèles auxquels sont associé les « Archetypal models » qui donnent des spécifications sur le système et aux « observed models » qui sont des modèles exécutables à l'exécution. On spécifie une fois les « Ontological models » et cela permet de générer de nombreuses IHM.

Les auteurs mettent en place des étapes dans la conception d'IHM. Ces étapes correspondent à différents niveaux d'abstractions de description. Elles sont liées entre elles par la réification. La réification consiste à transformer par étapes successives une description abstraite en une interface homme-machine concrète.

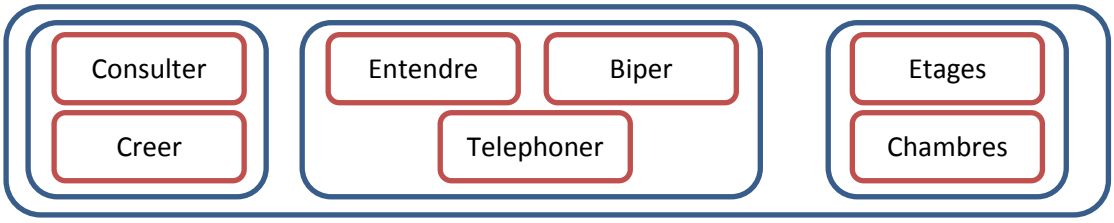
Le framework présenté dans cet article comporte quatre étapes de réification. Pour illustrer les différentes étapes d'abstraction j'ai pris l'exemple de notre projet en CEIHM (Conception et Evaluation des IHM).

- Le niveau le plus haut est le modèle de tâche ainsi que les concepts du domaine. La description du domaine est modélisée en UML (Unified Modeling Language)

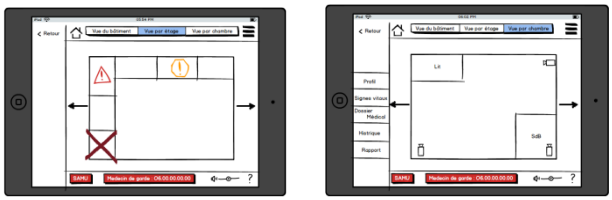


- On a ensuite l'interface Abstraite. Elle définit des espaces d'interactions et spécifie les tâches élémentaires d'interaction. Nous utilisons l'arbre des tâches pour obtenir les espaces de travail de l'IHM abstraite. Un espace de travail est une unité de présentation qui regroupe tout ce qui est nécessaire à la manipulation d'un ensemble de concepts.

.



- L'interface Concrète décrit l'interface. Elle permet de modéliser l'interface par une maquette (mock-up) pour donner une première représentation du rendu.



- Pour finir, l'interface Finale représente l'interface mis à disposition des utilisateurs.

Petit schéma récapitulatif :

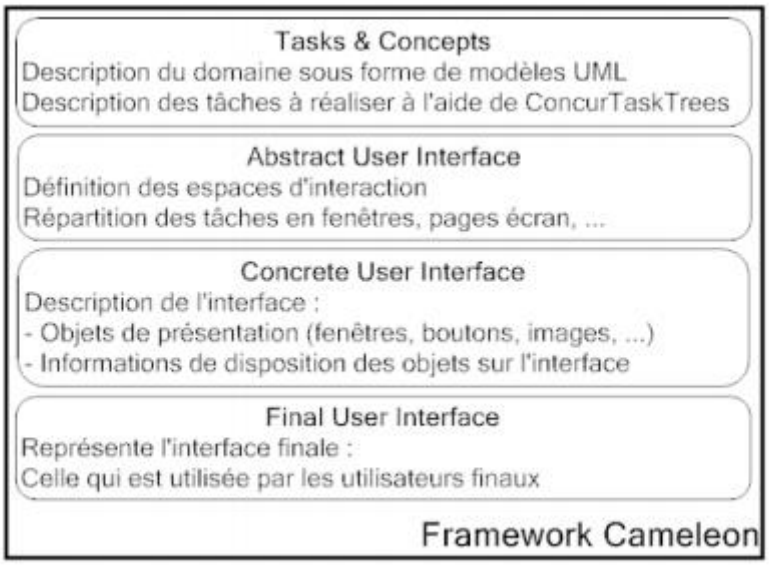
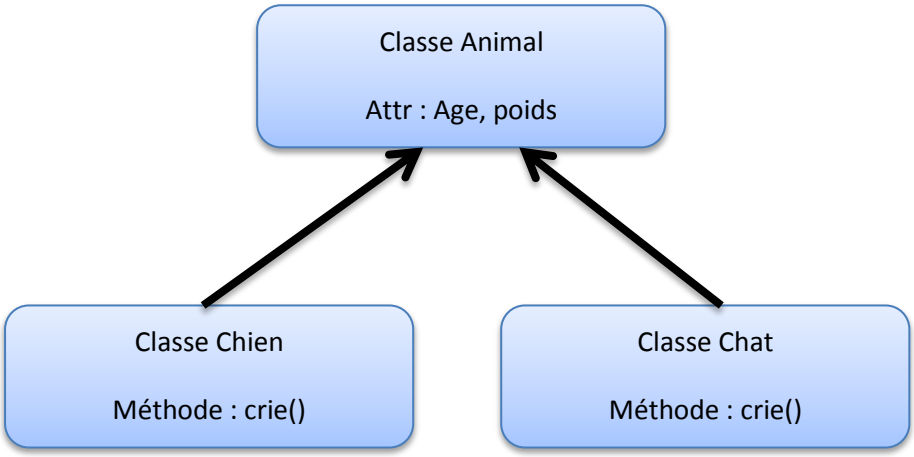


Figure 5

Un élément important dans ce modèle est l'opération de traduction (Translation sur la figure 4). Cela consiste à produire une description, à partir d'une description d'une cible, pour une nouvelle cible. Il n'y a pas de changement de niveau d'abstraction. Par exemple, dans le niveau d'abstraction d'IHM concrète, on veut à partir d'une IHM concrète pour un PC passer à une IHM concrète pour un Mac. Ce processus s'appelle la traduction.

Le processus de décoration permet de donner une information supplémentaire à une description d'éléments. Elle aide à une meilleure interprétation de la description.

Un principe mis en avant dans ce modèle est le principe de factorisation. Il permet d'identifier les parties communes à plusieurs cible (et les parties spécifiques à une cible) et permet de les rassemblées. Cet aspect est analogue à la Programmation Orientée Objet (POO). Un exemple pour illustrer la factorisation :



Les parties communes sont les attributs âge et poids, un chien ou un chat à forcément un âge et un poids. On distingue aussi un aspect spécifique à chaque classe. Le crie du chien est différent de celui d'un chat (aboiement et miaulement).

Pour résumer, la plasticité occupe une place de plus en plus importante dans le domaine des IHM. En effet le nombre d'utilisateurs augmente. Les personnes sont de plus en plus « connectées ». Grâce à l'évolution de la technologie, le contexte d'usage des IHM augmente. Le nombre d'appareilles électronique (téléphones, tablettre, montre) est en expansion. Il en résulte un large nombre de plateformes et l'environnement dans lequel on peut les utilisés augmente lui aussi grâce à la mobilité des technologies. C'est un grand défi de pouvoir adapter les IHM pour les différents contextes d'usages. Ce défi est complexe en termes de réalisation et il devient un enjeu primordial dans la conception d'IHM.

Pour faciliter la mise en œuvre d'IHM adaptative, les chercheurs D. Thevenin, J. Coutaz, G. Calvary proposent des solutions. Ils veulent structurer la conception des IHM grâce à des modèles et différents niveaux d'abstractions. Ils proposent des outils permettant de faciliter le cheminement de conception. Ils offrent une méthodologie et conceptualisent des processus (réitération, traduction, factorisation, décoration)

IV. Point de vue personnel

Il est évident que les IHM adaptatives deviennent une priorité. Le nombre croissant d'appareils mobiles et de plateformes nous impose d'adapter nos IHM. On doit prendre en compte les différents utilisateurs. En effet, tous les âges adoptent la technologie, les personnes sont de plus en plus connectées. On doit donc s'adapter aux différents utilisateurs. Certains préfèrent utiliser une modalité plutôt qu'une autre. Par exemple pour la rédaction d'un message (SMS), certains utilisateurs vont préférer l'utilisation d'un clavier, d'autres vont préférer une interaction vocale. Il faut prendre l'environnement en compte. L'environnement est le moment ou le lieu dans lequel on utilise l'application (jour/nuit, à la maison, en déplacement). Une application ne s'utilise pas forcément de la même manière en fonction de l'environnement. Revenons à notre exemple d'écriture d'un message. Un utilisateur utilise le clavier pour écrire son message lorsqu'il est assis dans son fauteuil, en revanche il utilise l'interaction vocale lorsqu'il conduit sa voiture.

Tous ces aspects sont à prendre en considération et cela favorise l'expérience de l'utilisateur.

On a donc besoin d'une méthodologie et de concept propre à la plasticité qui simplifie la mise en place d'IHM adaptative. C'est exactement ce qui est proposé dans cet article (framework). Les auteurs nous offrent une structure qui nous aide à réaliser des IHM adaptatives. Le fait de diviser la conception en différentes étapes permet une meilleure compréhension des besoins et permet de simplifier une tâche. En effet, lorsque l'on a une tâche compliquée à réaliser, il est recommandé de la diviser en sous-tâche pour faciliter la résolution du problème (principe divide and conquer).

V. Bibliographie

Phonegap : http://atelierihm.unice.fr/enseignements/wp-content/plugins/pdfjs-viewer-shortcode/web/viewer.php?file=http://atelierihm.unice.fr/enseignements/wp-content/uploads/2014/10/formation-mobile-cross-platform.pdf&download=true&print=true&openfile=false

Appcelerator Titanium : http://fr.wikipedia.org/wiki/Appcelerator_Titanium

Aide à la Comparaison : http://fr.slideshare.net/Olivia2590/phone-gap-or-titanium?related=1

Aide à la definition des concepts : http://iihm.imag.fr/publs/2001/THESE2001_Thevenin.pdf

Figure1 : http://fr.slideshare.net/kcresus/phonegap-vs-appcelerator

Figure 2 : Michael Laguerre Techniques d'interaction et multimodalité

Figure 3 : http://fr.slideshare.net/Olivia2590/phone-gap-or-titanium?related=1

Figure 4 : Article de recherche

Figure 5 : http://univ-valenciennes.fr/LAMIH-intra/site/specifique/publications/203569.pdf