



Adaptation des interfaces A l'environnement

Synthèse sur les travaux de recherche

Auteur : Paul Lavoine

Encadreur : Mme Dery

Date de première rédaction : 02/11/2014

Date de dernière modification : 12/11/2014

I. Quel contexte d'usage ?

« Avec l'informatique ambiante, c'est-à-dire diffuse et mobile, les contextes d'usage se diversifient : l'utilisateur évolue dans un environnement varié et recourt, de façon opportuniste à des plates-formes d'interaction diverses ». Afin d'exploiter au mieux les fonctionnalités technologiques de leurs applications, les utilisateurs ont recours à plusieurs types de dispositifs qui interagissent simultanément sur différentes plateformes.

Chaque application est créée à l'aide d'un environnement de développement généralement conçue pour faire fonctionner les applications sur un seul type de support, sans se soucier de l'accès à plusieurs appareils. Les développeurs d'interface utilisateur multiplateformes sont ainsi limités par les outils de développement et les interfaces utilisateurs ainsi créées ne sont généralement pas adaptées aux besoins.

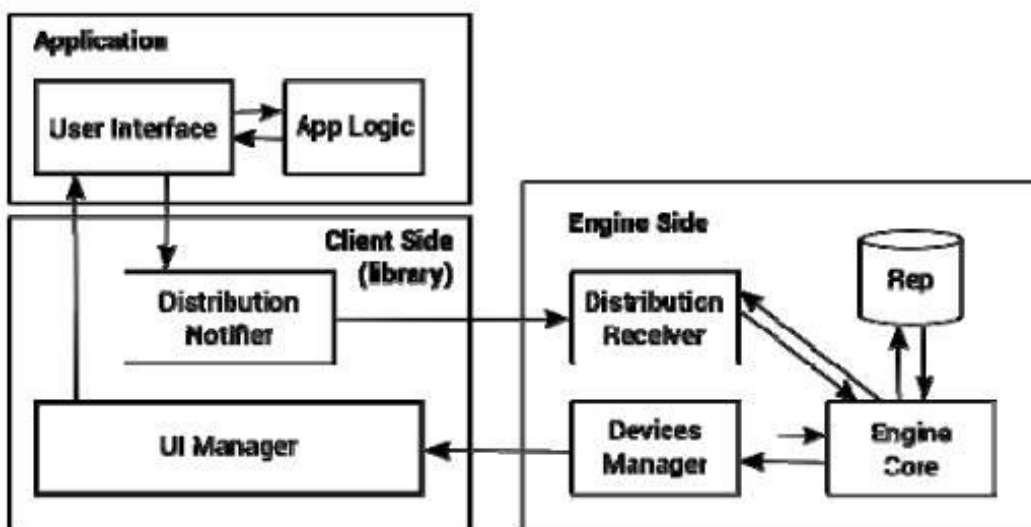
Et c'est en réponse à ce problème que Luca Frosini, et Fabio Paterno nous présentent un framework permettant le partage d'interface utilisateur entre plusieurs dispositifs et plusieurs environnements utilisateurs.

II. Quelle solution ?

Le framework présenté par nos deux auteurs, fournit aux développeurs une API qui peut être exploitée par des applications Web et Java dans le but d'obtenir plus facilement des applications pouvant être dynamiquement distribuées entre plusieurs dispositifs et plusieurs environnements.

Ce framework fonctionne par un système de sessions que des utilisateurs peuvent rejoindre (Par abus de langage, j'utiliserai le terme « groupes d'utilisateurs » au lieu de « groupe de dispositif utilisateurs »). Chaque membre d'un groupe se verra recevoir une même « interface utilisateur » (plus précisément un état), distribuée par une machine serveur, à chaque membre d'un même groupe.

Cette solution repose sur le modèle suivant :



Ce schéma représente les composants logiciels du framework. On peut **différencier deux principaux blocs** bien distincts : « Engine side » et « Client side », respectivement côté serveur et côté client.

« Engine side » va maintenir l'état actuel de la distribution. Il va aussi permettre aux utilisateurs d'accéder à la session de distribution à chaque instant et ainsi de leur fournir l'interface utilisateur avec l'état approprié.

« Client side » est responsable de l'envoi de requête à l' « Engine side » et recevra les mises à jour qui gère l'interface utilisateur.

Le dernier bloc représente l'interface utilisateur qui reçoit les données du « Client side ».

III. Présentation de la solution

La première phase consiste à une authentification du client vers le serveur. N'importe quels dispositifs peuvent souscrire à une machine serveur. En fonction de la catégorie du dispositif, cette machine peut choisir ou non d'accepter le dispositif et le place ou non dans le groupe qui lui convient. La catégorie d'un dispositif se définit par rapport à son type (smartphone, tablette...), sa capacité telle que sa taille d'écran par exemple et permet de classifier les dispositifs en groupe de catégorie similaire.

La seconde phase du framework consiste à associer l'état de distribution à un dispositif. L'« Engine side » contient toutes les informations concernant les états des éléments de l'interface utilisateur. Il existe **trois états différents** :

- « Enabled » éléments visible et réactifs aux éléments
- « Disabled » éléments visible mais pas réactifs aux éléments
- « Invisible » éléments invisible sur l'interface utilisateur

Au moment où la machine serveur reçoit des mises à jour, différentes opérations sont exécutées :

- Validation de la requête
- Calcul du nouvel état de distribution
- Calcul des dispositifs à mettre à jour
- Informer les dispositifs concernés d'un changement de leur interface

Le framework fonctionne à l'aide de deux commandes :

- Assign Command est utilisée afin de modifier les dispositifs qui peuvent afficher ou permettre de manipuler les éléments de l'interface utilisateur.
- Feedback Command informe les dispositifs d'un changement de valeur d'un élément de l'interface par notifications aux clients concernés.

IV. Illustration sur un exemple

La solution présentée peut s'illustrer par l'utilisation d'un jeu vidéo en ligne. L'« Engine side » est représenté par un monde regroupant tous les clients. Chaque client peut demander de rejoindre le monde en envoyant les informations concernant le dispositif qu'il utilise pour jouer. Si l'appareil est assez rapide et suffit aux besoins que nécessite le monde, le serveur peut accepter le client en le faisant rejoindre un groupe définit selon les critères indiqués précédemment.

Chaque client peut interagir avec les autres. Supposons qu'un client désire changer de monde, il peut envoyer une requête aux serveurs. Le serveur communiquera ainsi le changement de monde à tous les autres clients.

Un autre exemple plus classique pourrait être un système de messagerie instantanée. Les utilisateurs se connectent à une session de tchat. Le serveur les accepte ou non. Si oui, il peut envoyer un état courant tel que les différentes connections sur le tchat en fonction de la catégorie du dispositif. Chaque client qui désire envoyer un message l'envoie ensuite au serveur qui notifiera les autres clients du message.

V. Votre avis ?

Ce framework permet l'interaction entre plusieurs types de dispositif et dans différents environnements. Il permet de passer outre les systèmes d'exploitation, les différentes caractéristiques etc... Contrairement aux autres framework déjà proposé, celui-ci à l'avantage de ne pas se limiter à des contextes et applications spécifiques.

De plus, les tests de performance de ce framework ont permis de conclure qu'il est assez rapide pour éviter de créer un problème d'utilisation.

Mais il possède tout de même des inconvénients. Le mécanisme dédié à la sélection des éléments de l'interface utilisateur n'est pas assez souple. De plus, l'allocation de dispositifs dynamiques sur plusieurs sessions par l'« Engine side » n'est pas encore possible. Ceci restreint les clients à avoir une même session.

Ce framework est donc intéressant à utiliser dans le cadre de plusieurs dispositifs différents. Cependant, quelques améliorations doivent y être apportées sans quoi il n'aurait pas d'avantage dans le cas où tous les dispositifs seraient semblables. Il y perdrait tous ses principaux avantages.

Référence :

Lyytinen, K. & Yoo, Y. (2002). Issues and Challenges in Ubiquitous Computing. Communications of the ACM, 45(12), 62-65

Jean-Sébastien Sottet, Gaëlle Calvary, Jean-Marie Favre. (2005) Ingénierie de l'Interaction Homme-Machine Dirigée par les Modèles