Synthèse d'article scientifique

Hugo Martinez

11/11/2014

Table des matières

1. Introduction

Dans le cadre du module d'Adaptation des Interfaces à l'environnement il nous est demandé d'effectuer une analyse de document scientifique. Nous devrons effectuer cette analyse du point de vue de la plasticité qu'apporte la solution proposée par l'article à la conception et l'adaptation d'IHM.

Pour cette analyse j'ai choisi l'article « Beyond Responsive Design: Context-Dependent Multimodal Augmentation of Web Applications » écrit par Giuseppe Ghiani, Marco Manca, Fabio Paternò, et Claudio Porta.

2. Solution

2.1 Présentation

Dans l'article « Beyond Responsive Design: Context-Dependent Multimodal Augmentation of Web Applications » décrivent une solution permettant d'augmenter les modalités d'une application web en fonction du contexte d'utilisation de cette application, c'est-à-dire modifier les possibilités d'interactions de l'utilisateur avec l'application en fonction des modifications de l'environnement qui entoure l'utilisateur.

La multi-modalité de la solution se base sur les deux modalités que sont celle graphique, qui correspond en fait à l'interface de base mise à disposition par l'application, comme les boutons, zones de saisis ou encore les formulaires, et la celle vocale qui elle correspond à l'utilisation de fonctionnalités via la commande vocale ou encore la lecture du contenu d'une page grâce à la synthèse vocale.

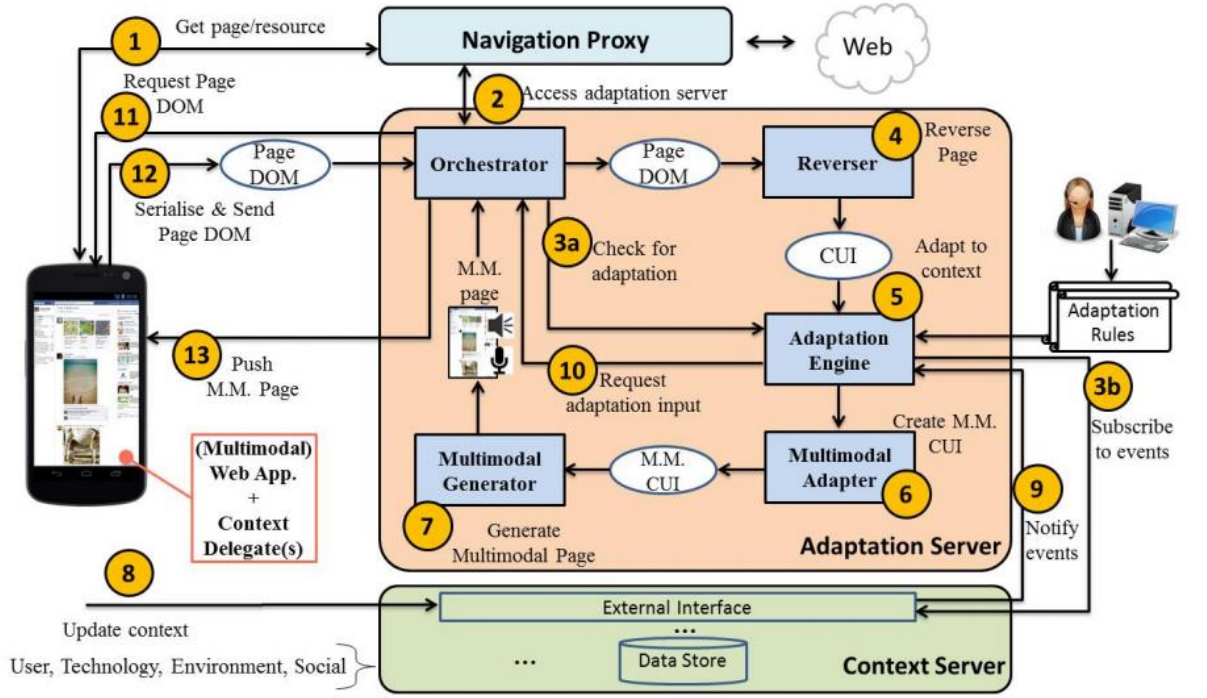
L'adaptation de l'interface se fait au travers d'une simple règle qui se forme de trois étapes, un événement, par exemple l'apparition de bruit, une condition qui est optionnelle comme par exemple le fait que l'utilisateur soit en train de se déplacer et enfin l'action est l'adaptation de l'interface en fonction de l'événement et des conditions.

De ces règles découlentquatre types d'adaptation de l'interface :

- Graphique à graphique : cela consiste en une modification de l'apparence de l'interface, en modifiant par exemple la taille de ses éléments pour permettre une utilisation plus facile lorsque l'on marche ou conduit
- Graphique à modale : celle-là consiste à ajouter des modalités lors d'un changement de contexte d'utilisation, en permettant par exemple d'utiliser la commande vocale pour effectuer une recherche en plus de pouvoir l'écrire avec le clavier.
- Modale à graphique : On retire des modalités qui ne sont plus nécessaire dû à un retour à la «normale » du contexte d'utilisation.
- Modale à modale : On ajoute des modalités à une application déjà augmentée, par exemple en ajoutant la diffusion vocale des résultats de recherche alors que la recherche par commande vocale a été ajoutée au préalable.

2.2 Modèle

2.2.1 Présentation du modèle



Le modèle ci-dessus nous permet d'illustrer en quoi la solution proposée dans cet article contribue à la plasticité des webapps sur lesquelles elle est appliquée. D'une manière général, le système est composé d'un ensemble de modules qui ont une tâche bien précise et qui communiquent les uns avec les autres pour permettre l'adaptation des différentes pages.

Orchestrator : permet la communication entre le monde réel et le serveur d'adaptation, il permet aussi de coordonner l'accès et les interactions entre les différents modules.

Navigation proxy : permet la communication entre les pages et l'ochestrator, il permet l'envoie du DOM des pages au serveur d'adaptation et le chargement dynamique des pages adaptées

Reverser : ce module permet de parser le DOM de la page pour faire un regroupement par éléments significatifs associés à leur CSS et aux événements qui leurs sont attribués.

Adaptation Engine : c'est le module qui décide quelle adaptation doit être mise en place en fonction des règles d'adaptation qu'aura mis en place le développeur.

Multimodal Adapter : permet le passage de la description graphique de la page à celle multimodale (il rajoute la synthèse vocale du contenu de la page par exemple).

Context Manager : il gère les possibilités de changement de contexte.

Multimodal generator : c'est lui qui va s'occuper de générer les multi modalités associées à la page.

Context Delegate (Context Server) : Sert à détecter tout changement de contexte et le transmet à l'Adaptation Engine

2.2.2 Illustration par un exemple

L'utilisateur lance l'application :

1. L'utilisateur accède à l'application au travers du **Navigation proxy** qui va modifier la page pour en faciliter le parsing mais cela ne sera pas visible pour l'utilisateur.

2. Le **Navigation proxy** transmet le DOM à l'**Orchestrator**

3.a L'**Orchestrator** envoie donc une requête vers l'**Adaptation Engine**, qui va répondre qu'une adaptation est requise ou non et le DOM de la page est envoyé au **Reverser** s'il y a un changement à effectuer**.**

4. Le **Reverser** va ensuite parser le DOM

5. L'**Adaptation Engine** va ensuite déterminer quelles sont les adaptations à faire sur la page en fonction des règles d'adaptation qui ont été définie. Dans le cas présent cela pourrait être par exemple la désactivation des commandes vocales.

6. S'il y a besoin on fait appel au **Multimodal Adapter.** Ce n'est pas le cas ici car il s'agit d'un retour à la normale de l'interface et donc une transformation de Modale à graphique.

Maintenant que l'utilisateur est sur l'application web, un changement de contexte intervient :

8. L'utilisateur est sorti de sa voiture et s'assoie dans son canapé, l'interface doit donc s'adapter. Le **Context Delegate** informe donc le **Context Server**

9. Si ce type de changement de contexte a déjà été géré alors l'**Adaptation Engine** est averti du changement de contexte.

10. L'**Adaptation Engine** va ensuite faire la demande du DOM de la page à l'**Orchestrator**.

11. L'**Orchestrator** demande ensuite ce DOM au navigateur du client.

12. Le DOM de la page est envoyé par le navigateur à l'**Orchestrator** puis les étapes 4 à 7 se reproduisent.

13. L'**Orchestrator** envoie la page avec la modalité de synthèse vocale retirée.

3. Contexte d'usage

3.1 Plateformes

De par la solution proposée par cet article, qui consiste à adapter l'interface d'une web application

en fonction du contexte d'utilisation, les plateformes visées semble être toutes des devices mobiles, comme les smartphones, les tablettes et tout autre système permettant l'usage de web application dans différents contextes.

3.2 Environnements

Le nombre de contextes gérés pouvant être illimité, alors le nombre d'environnement dans lesquels cette solution peut s'appliquée l'est lui aussi. Par exemple le système d'adaptation permet d'adapter l'interface à un environnement qui passe de calme à bruyant, et cette situation peut se présenter dans un grand nombre de lieux. Par exemple cette situation peut se produire chez soi, en entrant dans une pièce où des enfants jouent bruyamment, ou par exemple en entrant dans une boite de nuit par une rue calme. Cela veut dire que la solution proposée par l'article peut en fait s'appliquer à tous les environnements.

Malgré cela, la solution proposée connait des limites d'utilisation en termes de modification de l'environnement. Les devices mobiles ne possèdent qu'un nombre limité de capteurs et ne pourront jamais détecter certains changements de contextes d'utilisation.

3.3 Utilisateurs

3.3.1 Conception

D'un point de vue conception, la solution peut servir à toutes personnes souhaitant développer une web application mais aussi à tous ceux qui en ont déjà développé car il est possible de l'adapter à n'importe quelle web application déjà existante. De plus elle pourrait sûrement s'adapter à d'autres types d'applications ou de logiciels, mais pour le moment les auteurs ont préféré se concentrer sur les webapp.

3.3.2 Utilisation

L'utilisation finale elle peut se faire par tout utilisateur possédant un device mobile et utilisant des webapps. Soit un très grand nombre d'utilisateurs, allant de l'enfant jusqu'à la personne agée.

4. Plasticité

4.1 Conception

D'un point de vue conception, cette solution semble être idéale pour tous les développeurs de webbapplication. La plasticité apportée est importante sur la plupart des plans de la conception :

- **Réutilisation** : une fois les règles d'adaptation définies par le développeur, le système peut fonctionner quasiment avec n'importe quelle autre webapp moyennant quelques modifications selon les besoins.
- **Coût de développement** : pour un développeur qui souhaite rendre son application web context-dependent cette solution semble permette une réduction des coûts drastique

puisque la seule tâche qu'a à effectuer le développeur est l'implémentation des règles d'adaptation.

4.2 Execution

Au niveau de l'exécution, la solution proposée n'a pas pour but de permettre à une application de faire le portage d'une plateforme à une autre car lors du développement d'une web application cela est censé être géré de base, grâce par exemple au design responsive.

Globalement le système proposé a pour objectif d'augmenter les facilités et habitudes d'usages en fonction du contexte d'utilisation de l'application. Elle offre la possibilité de rendre l'application utilisable parfaitement peu importe le contexte d'utilisation et facilite ainsi grandement l'interaction avec l'application de l'utilisateur final.

5. Avantages et inconvénients

5.1 Avantages

- **Une solution pour la plasticité des web applications** : même si cela englobe le sujet de cette solution, cela représente un véritable avantage. Puisqu'elle peut s'appliquer à n'importe quelle application web existante ou en cour de développement, elle est donc la solution qui semble la plus intéressante pour augmenter au maximum la plasticité de votre web application.
- **Adaptation aux personnes handicapées** : en plus de s'adapter aux contextes d'utilisation, l'adaptation peut aussi se faire en fonction de l'utilisateur, comme par exemple changer les couleurs de l'interface si l'utilisateur et daltonien ou encore grossir les éléments s'il est malvoyant.
- **Pas de travail à fournir en amont** : les développeurs n'ont pas à adapter, ni à concevoir d'une manière spéciale leur webbapp ce qui réduit fortement le travail à fournir pour appliquer la solution. Le seul travail qui est demandé est l'implémentation des règles d'adaptation.

5.2 Inconvénients

Nous pouvons discerner un certain nombre inconvénients majeurs dans l'approche qui est faite dans cet article :

- **Le fait de proposer une solution basée sur un serveur :** même s'il parait évident que sans réseau, une webapp ne peut pas fonctionner, on peut tout de même trouver un certain nombre de situation où le fait que l'adaptation se face au travers de la communication avec un serveur se révèle être un terrible inconvénient. Imaginons par exemple que la page que souhaite afficher l'utilisateur soit déjà chargée, mais il se retrouve dans le métro, qui est bondé et arrive difficilement à lire son écran. Malheureusement à cause de l'absence de connexion au réseau, l'adaptation de la page ne pourra pas se faire.

- **Le nombre de modules, et donc de traitements effectués** : de par le nombre de traitement effectués, notamment le reverse engineering sur le DOM de la page, il est possible que lors du traitement de pages avec beaucoup de contenu l'adaptation de l'interface soit lente, voire trop lente comparé à l'évolution du contexte d'utilisation de l'application.
- **La limitation de l'utilisation de la solution** : les auteurs s'étant limité à l'adaptation des interfaces aux webapp, ils ne permettent pas l'utilisation de leur solution à d'autres type d'applications (donc les applications natives), ce qui réduit l'intérêt globale de celle-ci.
- **La limitation dû à l'implémentation des règles** : comme le nombre d'adaptation possible dépend des règles d'adaptation implémentées, il se peut que ces règles ne répondent pas entièrement aux attentes de l'utilisateur. Cela demandera donc un investissement en temps pour ajouter les règles d'adaptation nécessaires.
- **Interruption de la navigation** : à chaque changement de contexte, une pop-up apparait pour demander si l'utilisateur autorise l'adaptation de l'interface. Lors de changement fréquent de contexte d'utilisation, l'apparition de ce pop-up pourrait nuire à l'expérience de l'utilisateur, ce qui est gênant pour une solution qui souhaite l'améliorer.