

Contexte d'usage et environnement :

Avec l'abondance des réseaux sans fil et l'accélération de la miniaturisation électronique nos objets du quotidien deviennent *smart*. Nous avons depuis un certain temps les *smartphones* puis les *smartwatches* mais aussi les *smartglasses*. Pour chaque *smart* objet vient une nouvelle interface à laquelle il faut s'adapter pour que l'on puisse voir et utiliser notre ancien web sur nos nouvelles surfaces. A notre époque la plasticité n'est plus un besoin mais une nécessité.

Cet article propose une solution nommée Mobiwis. Cette solution a pour but d'adapter automatiquement une interface à son contexte d'usage. Mobiwis n'est pas une solution de *responsive design*, en effet le *responsive design* propose d'adapter des interfaces graphiques d'un site web selon la résolution d'un écran grâce à des *media queries*. Mais le *responsive design* ne prend pas en compte les changements de contexte et d'environnement. Cette solution est un *middleware* proposé aux développeurs et designers qui veulent rendre leur site réactif aux changements de contexte et d'environnement.

Les adaptations sont définies par les personnes qui sont en charge de la *webapp* au travers de règles d'adaptations. Ces règles d'adaptations sont appelées lorsqu'un changement de contexte est détecté.

Plateforme :

Dans l'article la plateforme décrite et utilisée lors des tests sont des appareils mobiles dotés de capteurs, l'on peut donc penser aux *smartphones* dans un premier temps. Mais les possibilités d'évolutions sont d'autant plus grandes avec la popularisation de nouvelles plateformes tel que les *smartwatches*, *smartglasses*.

Utilisateurs :

Il existe au moins deux catégories d'utilisateurs pour cette solution :

- **utilisateurs niveau conception** : Les personnes responsable de la *webapp* qui doivent implémenter les règles d'adaptation dans le système.
- **utilisateurs niveau exécution** : Les personnes navigant avec leur *smartphone* sur un site web doté de la solution Mobiwis.

Phase d'action de la solution :

Conception:

Il est possible de mettre en place la solution Mobiwis sur n'importe quel *site web que se soit* au moment de la conception ou même une fois que lorsque le site web est fini et déjà mis en ligne. Les

développeurs n'ont pas à modifier le code déjà présent mais seulement à implémenter les règles d'adaptations dans le module nommé *Adaptation Engine*. L'ajout de règles se fait donc de façon incrémentale.

Exécution:

Lorsque l'utilisateur navigue sur une page web avec son téléphone celui-ci capte l'environnement présent autour de lui (luminosité, bruit ambiant, au volant). Ensuite, le navigateur prévient lorsqu'un événement de contexte intervient (démarrage de voiture, entrer dans une pièce silencieuse). Enfin, la solution Mobiwis propose d'adapter la page actuelle et de l'augmenter pour fournir les modalités présentes dans les règles d'adaptations.

Le modèle de la solution :

La solution présente 7 modules distincts.

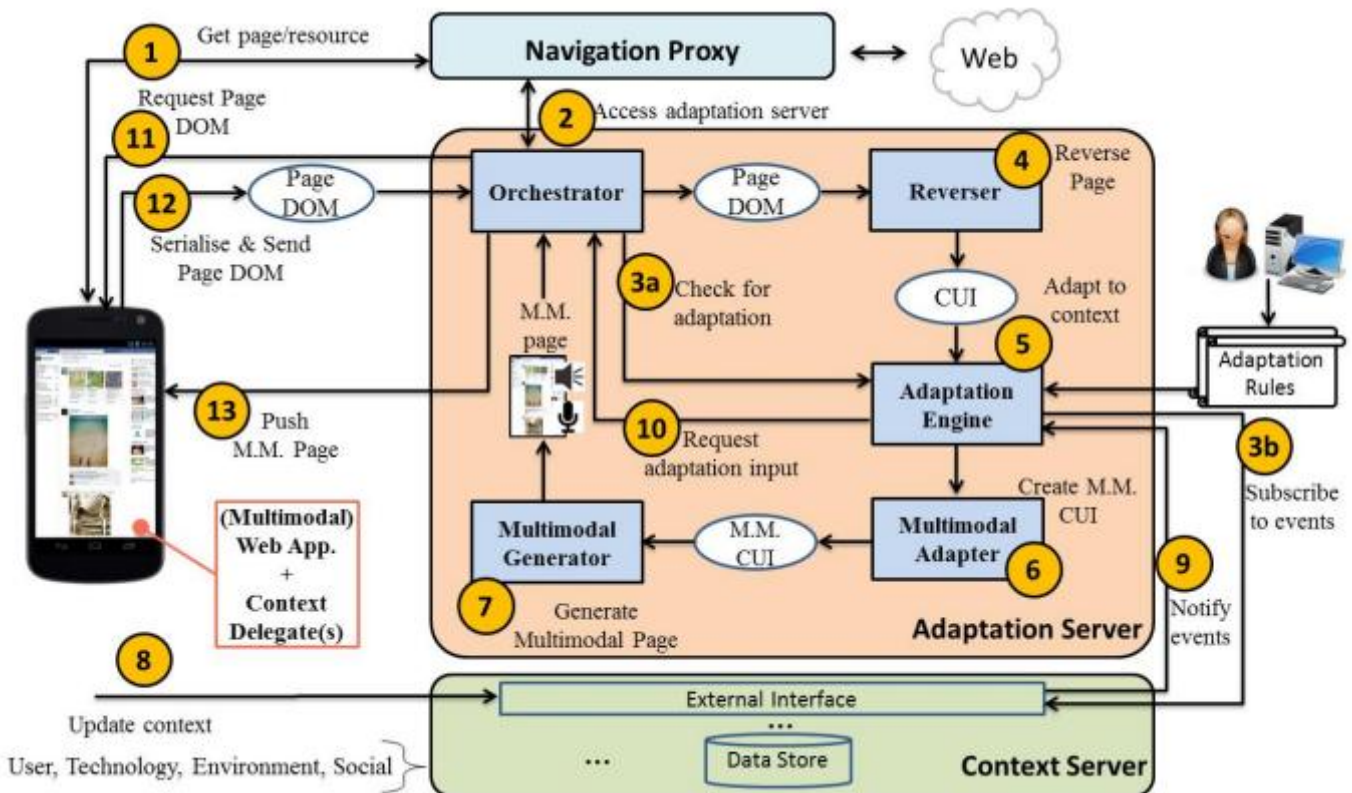


Figure 1 : Architecture de la solution Mobiwis

Orchestrator : Il sert d'interface entre le serveur d'adaptation (la solution Mobiwis) et le web. Il gère le trafic et la communication vers les autres modules du serveur d'adaptation.

Navigation Proxy : En plus de ses fonctions de proxy classique, le Navigation Proxy ajoute à la page actuellement affichée un script qui permettra à la dite page de communiquer avec le module *Orchestrator*. Le script est utilisé pour transmettre le DOM de la page actuellement affiché jusqu'à l'*Adaptation Server* puis plus tard d'*upload* la page adaptée vers le téléphone de l'utilisateur.

Reverser : C'est un outil de *reverse engineering*. Il a pour but de prendre n'importe quelle page web et de décrire son aspect graphique avec le langage MARIA, il regroupe les éléments du DOM avec leur CSS et événements associés.

Adaptation Engine : Le but de ce module est de décider quelles adaptations doivent être mises en place. Pour cela l'*Adaptation Engine* utilise les règles d'adaptations. Quand une règle est ajouté un pour un événement, l'*adaptation engine* ajoute un écouteur sur cet événement et pourra réagir lors l'événement sera déclenché.

Multimodal Adapter : Le *Multimodal Adapter* effectue la transformation de la définition graphique vers une multimodal. Il existe 4 adaptation possible : Graphical-to-Graphical, Graphical-to-Multimodal, Multimodal-to-Graphical, Multimodal-to-Multimodal. La partie action des règles d'adaptations sont exploité dans ce module.

Il est possible d'avoir une grande précision dans l'adaptation multimodale, comme par exemple prendre en compte les images dans les annotations vocales ou non. Mais aussi de demander une confirmation lors d'une entrée vocale, d'énoncer tous les éléments d'une liste déroulante et autres. La classification des éléments interactifs fourni par le langage MARIA couplé aux propriétés de CARE offre aux développeurs une grande liberté de configuration des éléments pour chaque modalités. Dans les cas les plus commun le *Multimodal Adapter* extrait des informations vocales de la page graphique, par exemple pour un élément *text input* la valeur de son label sera vocalisée, s'il ne possède pas de label une valeur par défaut pour ce type d'élément sera vocalisée.

Contexte Manager : Il permet de détecter tout changement de contexte dans l'environnement où se situe l'utilisateur. Ce dernier possède une architecture client/serveur dans laquelle les clients sont des *context delegates* qui ont pour mission de détecter les différents type d'événement de contexte et d'informer le serveur. Alors le serveur communiquera avec le *Adaptation Server* si ce dernier s'était possédait un écouteur sur type d'événement.

Multimodal Generator : Le serveur d'adaptation exploite également un générateur d'interface multimodales. Il permet littéralement le passage d'une description graphique d'une page (en utilisant l'outil de *reverse engineering*) vers une page multimodale, en rajoutant par exemple une synthèse vocal pour énoncer le contenu d'une page.

Présentation de la solution

Scénario 1 : Démarrer une voiture

L'utilisateur navigue sur son téléphone en marchant, il s'installe au volant de sa voiture et la démarre. Le téléphone capte le Bluetooth de la voiture et déclenche un événement. Mobiwis affiche une pop-up signalant que l'événement "user_driving" a été détecté et propose d'adapter la page. Maintenant la page n'est plus seulement graphique mais aussi peut prendre en entrée des commandes vocales et possède un synthétiseur vocale en sortie.

Scénario 2 : Passer d'un contexte bruyant à silencieux

L'utilisateur navigue sur son téléphone, il quitte un contexte bruyant (exemple un chantier) et rentre dans une pièce silencieuse. L'événement "low_noise" est détecté, la solution affiche une *pop-up* demandant si l'utilisateur veut obtenir une page augmentée en fonction de ce nouveau contexte. La page augmenté est chargé, l'aspect graphique ne change pas mais maintenant la solution utilise la synthèse vocale pour décrire la page. L'utilisateur choisi de saisir l'*input* graphiquement.

Avantages et inconvénients :

Avantages:

Les avantages de cette solution sont multiples mais tout d'abord elle répond à la nécessité qu'est la plasticité des interfaces.

Ensuite grâce à la mise en place de la multimodalité, en l'occurrence le vocale en entrée et sortie, Mobiwis rend les *webapp* plus accessible aux personnes mal voyante ou ayant du mal à utiliser un écran tactile (personne âgé, personne ayant un handicap moteur).

De plus la solution ne demande absolument aucune manipulation particulière à l'utilisateur, il n'a pas à installer d'application supplémentaire ou de matériel particulier à acheter. Un simple *smartphone* et l'utilisation de son navigateur internet favoris est suffisant.

Mobiwis ne demande pas de modification du code déjà existant ni d'ajout de code aux personnes en charge de la maintenance du site voulant implémenter la solution. En effet la seule tâche nécessaire est l'implémentation de règles d'adaptations pour décrire la réaction qui doit être engendrée lors de la détection d'un changement de contexte. L'avantage majeur de cela est qu'il n'est pas nécessaire que les personnes développant les règles d'adaptations soient les mêmes que les personnes ayant développé la *webapp*.

Enfin d'après les arguments précédents il est possible d'intégrer Mobiwis à n'importe quelle *webapp* et donc de rendre la dites *webapp* adaptable à son environnement.

inconvénients:

Néanmoins cette solution n'est pas parfaite. En effet il s'agit d'un *middleware* et donc il dépendant du réseau, par exemple si l'utilisateur n'a pas de connexion internet disponible au moment du

changement de contexte Mobiwis ne pourra pas percevoir ce changement et par conséquent ne pourra pas adapter la page qui est déjà chargée.

L'implémentation des règles d'adaptations sont à la charge de l'équipe qui s'occupe du site web visé. Si un contexte n'a pas été prévu alors il n'y aura pas de règle d'adaptation et donc aucun changement sur la page.

Lorsque le changement de contexte est bien détecté et géré par les règles d'adaptation du *middleware* une *pop-up* apparaît pour demander s'il veut que les règles d'adaptations soient appliquées à la page. Dans le cas où l'utilisateur change souvent de contexte la *pop-up* apparaîtra à chaque changement et interrompra la navigation de l'utilisateur à chaque détection.

Pour qu'une transformation ait lieu il est nécessaire à un moment donné transmettre le DOM à Mobiwis, si le DOM est complexe le processus de transformation peut être trop lent. Ainsi le temps que la transformation soit faite l'utilisateur peut avoir changé de contexte et n'aurait plus besoin de la transformation qui était en cours.

Il est également dommage de voir cette solution s'étendre uniquement au domaine du web, en effet cette solution pourrait être très bénéfique aux applications natives des téléphones mais également aux logiciels pour PC. Cela permettrait de rendre ces logiciels plus accessibles aux personnes mal voyantes ou souffrant d'un handicap physique.

Sur la vidéo de tests l'on peut constater que lors d'une détection de changement de contexte le téléphone affiche une *pop-up* proposant d'adapter l'interface à ce contexte mais ne précise pas quelles seront les modalités mises en place. De ce fait lors de l'entrée dans une pièce silencieuse, comme une salle d'attente, il se peut que la modalité vocale s'active et dérange les personnes autour de l'utilisateur.

(<https://www.youtube.com/watch?v=7Y670aWNUDM&feature=youtu.be>)