

Article de recherche

Towards Intelligent Migration of User Interfaces

Devant la multitude d'appareils et d'écrans auxquels nous avons à faire face aujourd'hui depuis l'avènement des ordinateurs et autres mobiles, la possibilité de passer d'un appareil à un autre est devenu une perspective intéressante. Actuellement cela requiert, d'installer les applications sur chaque dispositifs, et de migrer les interfaces vers ces nouvelles plate-formes. La synchronisation des données entre une même application présente sur plusieurs appareils est une tâche fastidieuse, c'est pourquoi le nombre de données stockées sur le Cloud ne cesse d'augmenter. Pour répondre à ce problème, cet article expose des recherches faites sur les problèmes de migrations d'interfaces. Celles-ci définissent, et implémentent une preuve de concept d'un prototype visant à automatiser la migration d'applications Web en se basant sur le contexte dans lequel l'utilisateur se trouve. Ce type précis d'application a été choisi car elles sont le parfait exemple de systèmes qui sont habituellement accédés depuis différents endroits avec des appareils hétérogènes. Des tests utilisateurs ont été effectués afin de rassembler des données et avoir un retour sur la corrélation entre le comportement du système et l'appréciation des usagers. La pertinence et l'utilisabilité du prototype sont évalué selon trois modes : manuel, assisté, et automatique. Comme nous le verrons, d'autres avantages, dû à la migration dynamique basé sur le contexte, ont été mis en avant par cette solution, notamment concernant l'amélioration des interactions entre le système et l'utilisateur.

En se basant sur de précédent travaux consistant à analyser l'expérience utilisateur des applications cross-plate-formes, trois critères d'études ont été mis en valeurs : la composition, la continuité et la cohérence. L'approche présenté est particulièrement attentive à ces aspects, premièrement en permettant à l'utilisateur de migrer tout ou une partie de l'interface ; également en préservant la continuité de l'expérience utilisateur, par la transition transparente d'un dispositif à un autre ; et enfin en visant à conserver la cohérence des fonctionnalités de l'application Web.

Un système relativement similaire existe déjà : *Deep Shot* est une solution de migration automatique d'interfaces graphique entre deux dispositifs, tout en préservant l'état de celle-ci. La migration s'opère en photographiant depuis un mobile une partie d'un site ou d'une application native.

La plate-forme développée dans le cadre de l'étude vise elle à être automatisé, ou assisté, afin de faire intervenir un minimum l'utilisateur. Pour cela un service de reconnaissance de contexte est déployé sur les appareils, et permet alors de choisir lorsque la migration doit s'effectuer.

Ce service de *monitoring* se base tout d'abord sur la découverte des autres dispositifs dans les alentours par la détection *bluetooth* de la proximité entre eux. Selon les technologies à disposition il aurait pu être choisi d'autres moyen de communication comme le *NFC*, ou le *RFID*. D'autres paramètres sur le mobile, comme la localisation *GPS* ou le réseau local, son mouvement, ou son orientation, sont pris en compte dans l'analyse contextuelle. Le moniteur étudie si il y a de l'activité virtuel sur l'appareil, comme par exemple si une application est utilisé, ou si il envoi un message. Il est également considéré que l'activité physique de l'utilisateur (en mouvement ou immobile), obtenue par l'accéléromètre, influe sur la décision de déclenchement de la migration.

Les paramètres de contexte peuvent être choisis et configurés de façon à laisser l'utilisateur quand la migration doit s'effectuer. Par exemple il peut configurer « opérer la migration depuis le téléphone vers l'ordinateur du bureau, quand le téléphone est posé à plat sur le dos, sur une table proche de l'ordinateur ».

Concernant la migration entre dispositifs, il a été identifié deux principaux avantages d'un système se basant sur le contexte, premièrement en utilisant les informations de l'environnement le système à automatiquement connaissance de l'appareil cible vers lequel il doit migrer. Deuxièmement la migration est opéré automatiquement ou suggéré et ainsi facilite l'interaction de l'utilisateur.

Dans l'étude le système mis en place est basé sur quatre composants : les appareils source/cible, un client de migration tournant sur chacun d'eux, le moniteur de contexte, et un serveur proxy.



Architecture et scénario

Lorsqu'une page Web est ouverte sur A (1), le serveur proxy exécute la requête (2-3) et renvoi les informations (4) à A en ajoutant un script *JavaScript* dans la page *HTML*. Le code injecté permet à la page d'être potentiellement migré sans modifier le contenu ou le *layout* de celle-ci. Suivant la configuration, le moniteur de contexte envoie au moment opportun une demande de migration au serveur proxy (5). Celui-ci transmet la demande au client de migration de A (6), c'est alors que le code *JavaScript* injecté permet la sérialisation de la page Web et est envoyé au serveur (7). Finalement le client de migration de B est notifié et ouvre la page Web (8).

L'utilisateur ciblé par ce papier est familier des technologies, il sait paramétrer une application selon ses besoins, il possède un ordinateur de bureau et un smartphone, et se sert de cet outil lorsqu'il est au travail ou chez lui.

On peut extrapoler cette étude, étendre la synchronisation et la migration d'un appareil à un autre, et ce, dans tout type d'environnement. On peut imaginer faire une réservation pour un billet de train sur internet en allant à la gare, et migrer l'état actuel de la réservation vers une borne de paiement une fois arrivé dans le hall. Ou bien alors, rechercher une destination sur son smartphone avant de rentrer dans sa voiture, et une fois le *GPS* allumé, directement synchronisé l'itinéraire. Le principal défi est de pouvoir sérialiser les données contenus sur une interface et de les représenter sur un autre dispositif. Cependant la plus-value apportée par cette étude est qu'elle montre l'intérêt de l'analyse du contexte pour effectuer l'action de migration. Grâce à l'étude contextuelle dans lequel se trouve l'utilisateur, dynamiquement le système lui permet de transiter entre son ordinateur, une borne, ou son *GPS*, et son smartphone, ce qui facilite sa navigation sur le Web, mais pourrait également rendre inter-opérable les différents dispositifs autour de lui. Un minimum d'interactions est nécessaire à la migration, et permet une adaptation très rapide au système, c'est ce que montre principalement les tests utilisateurs concernant la plasticité.

L'avantage d'un tel modèle est de supprimer les barrières d'interactions entre les appareils.

Aujourd'hui nous effectuons des actions individuelles sur notre smartphone, notre ordinateur personnel, ou tout types de dispositif que nous pouvons rencontrer. Ceci limite nos possibilités et nous sommes souvent amené à ré-effectuer ces actions pour revenir à un même état d'avancement. Par exemple lorsque cherche le titre d'une chanson sur *Shazam* et que je l'obtiens, je recherche ensuite la chanson sur mon ordinateur pour la jouer ou l'enregistrer dans mes favoris. Il pourrait clairement être intéressant suivant les cas d'utilisations de rendre possible la migration d'interfaces ou de données, et notamment si des paramètres contextuels peuvent être configurés.

Cependant on peut se demander, suivant les cas, si il est toujours pertinent d'effectuer automatiquement ces actions. Il est possible qu'un utilisateur veuille discrètement consulter un page Web sur son téléphone sans automatiquement faire apparaître celle-ci sur son ordinateur à la vue de tous.