Adaptation et multimédia mobile sur le Web

Nabil Layaïda, Tayeb Lemlouma, Vincent Quint

Inria Rhône−Alpes, 655 avenue de l'Europe, 38334 Montbonnot, France
{Nabil.Layaida, Tayeb.Lemlouma, Vincent.Quint}@inria.fr

Résumé. Le Web évolue vers des contenus intégrant des média de plus en plus riches et variés accédés à l'aide d'appareils et de réseaux très divers. L'information multimédia doit être adaptée à cet environnement mobile et changeant. Dans ce but, un ensemble de méthodes, langages, formats et protocoles sont développés, notamment par le W3C. L'architecture NAC a été définie et mise en oeuvre sur la base de ces techniques, en mettant l'accent sur la répartition des traitements d'adaptation, sur les modèles de description de l'environnement et sur les transformations de contenu.
Mots clés : Adaptation, Multimédia, Terminaux mobiles.

1 Introduction

Le Web a profondément évolué depuis ses origines. On n'en est plus à partager de simples pages de texte et des images à partir de stations de travail et de PC connectés à l'Internet filaire. Deux évolutions particulièrement marquantes ont eu lieu ces dernières années, dans les domaines des contenus et des moyens d'accès.

1.1 Contenus multimédia

Les contenus du Web comprennent maintenant des média continus, comme la vidéo, le son ou les animations. Ces contenus sont souvent utilisés isolément, comme un morceau de musique ou un clip vidéo. Il s'agit alors de ressources mono−média. Mais on trouve aussi de véritables documents multimédia, où plusieurs ressources de différents média sont intégrées, se complètent mutuellement et dépendent les unes des autres.

Ces documents multimédia combinent des média numériques discrets, comme le texte ou les images, avec des média continus. Les différents fragments d'information sont reliés par des relations temporelles, spatiales et navigationnelles qui déterminent leur enchaînement dans le temps, leurs positions respectives dans l'espace d'affichage, les différents parcours que l'utilisateur peut effectuer, ainsi que les autres moyens d'interaction dont il dispose. Ce type de document est typiquement mis en oeuvre dans le Web à l'aide du langage SMIL (Synchronized Multimedia Integration Language) [1], qui permet de décrire les multiples aspects des documents multimédia.

1.2 Réseaux

Les moyens d'accès ont également beaucoup changé depuis les débuts du Web, qu'il s'agisse des terminaux ou des moyens de communication. Les réseaux sans fil se sont développé et se sont intégré à l'Internet. Ce sont à la fois des réseaux de téléphonie mobile et des réseaux de transmission de données du type WiFi (IEEE 802.11). Tous font maintenant partie de l'infrastructure de communication du Web.

Dans le même temps, des appareils de plus en plus variés se sont connecté sur ces réseaux : téléphones cellulaires, assistants personnels, téléviseurs, consoles de jeux, appareils embarqués. Tous sont des terminaux multimédia, dans le sens où il gèrent plusieurs média, et notamment des média continus. De plus, la plupart accèdent à des réseaux mobiles. On est donc en plein dans le monde du multimédia mobile.

1.3 Adaptation

Cette évolution des contenus et des moyens d'accès ne va pas sans poser de problèmes. Le premier est sans doute l'adéquation entre contenus et moyens d'accès. En d'autres termes, il s'agit de permettre l'accès dans de bonnes conditions aux mêmes ressources d'information multimédia, à partir de divers appareils, à travers des réseaux différents. C'est le problème de l'adaptation des contenus que nous abordons dans cet article.

Dans la section suivante, nous présentons les différents aspects de l'adaptation tels qu'ils sont abordés aujourd'hui dans le Web, en particulier à travers les travaux du W3C. C'est un travail collectif mené par des dizaines de chercheurs et d'ingénieurs à travers le monde et auquel les auteurs de cet article participent. Dans la troisième section, nous présentons les résultats d'un travail plus focalisé mené dans notre équipe et qui se place dans ce cadre plus vaste. Il s'agit d'une architecture d'adaptation répartie appelée NAC (Negotiation and Adaptation Core) qui fait intervenir des *proxies d'adaptation*, en plus des habituels clients et serveurs.

2 Une approche de l'adaptation

L'approche pragmatique de l'adaptation consiste à développer des contenus spécifiquement ciblés vers un type d'appareil, éventuellement dans un format propre à cet appareil. C'est typiquement l'approche retenu par le WAP, où un format particulier, WML, permet d'organiser l'information d'une façon qui se prête bien à l'accès sur le petit écran d'un téléphone et à travers un réseau à faible bande passante. Ce modèle va à l'encontre des objectifs du Web, qui se veut un espace d'information universel, accessible à tous. Le risque de cette approche est la fragmentation, avec un Web pour les téléphones mobiles de première génération, qui ne contienne que de l'information produite pour ces appareils, mais aussi un Web pour les téléviseurs, un Web pour les assistants personnels, etc. Cela induit de fortes restrictions pour les utilisateurs de chaque type d'appareil : ils n'ont plus accès qu'à une partie de l'information. Cela induit aussi des coûts élevés pour les producteurs de contenu qui veulent toucher un large public : ils doivent créer plusieurs fois la même information pour les différents type d'appareils.

Une autre approche consiste, au contraire, à favoriser des formats universels. On produit chaque document une seule fois et on le stocke sous une forme unique. Ensuite, au moment où le document doit être envoyé à un client, on engendre une autre forme qui tient compte des spécificités de l'environnement à cet instant. L'environnement comprend l'appareil client et les moyens de communication utilisés, mais aussi l'utilisateur, ses capacités, ses handicaps et ses préférences, voire les tâches qu'il effectue.

Cette seconde approche fait intervenir quatre éléments principaux dans l'adaptation : les formats de documents, la description du contexte, la négociation et la transformation.

2.1 Formats de documents

Pour faciliter l'adaptation, il est clair que le format dans lequel un document est décrit doit faire intervenir le moins possible de caractéristiques liées à la présentation ou à l'appareil de restitution. L'approche XML va dans cette direction, en séparant le contenu et sa structure logique d'un côté et la présentation de l'autre. Contenu et structure sont représentés dans la ressource XML elle–même, alors que la présentation est décrite dans des ressources séparées, les feuilles de style. Plusieurs feuilles de style différentes peuvent être appliquées à la même ressource XML, qui prend ainsi des aspects différents. Cette propriété de XML offre ainsi une possibilité efficace d'adaptation, par le choix des feuilles de style qui conviennent le mieux à un environnement donné.

La notion de feuille de style a été introduite d'abord pour les documents textuels, mais elle s'étend très bien aux documents multimédia. Des formats XML multimédia comme SMIL [1] et SVG [2] s'appuient sur le langage de style CSS [3] pour déterminer la présentation effective. CSS de son côté ne se limite pas à définir l'aspect graphique des documents, puisqu'il comporte un module qui permet de déterminer très précisément l'aspect sonore que prend un document textuel restitué par un synthétiseur vocal. Cela permet, simplement en choisissant la bonne feuille de style, d'adapter un document à un appareil graphique ou à un appareil sonore, comme un téléphone, par exemple.

En plus des feuilles de style, les formats XML multimédia disposent de mécanismes favorisant l'adaptation. XHTML propose l'élément *object* pour inclure des ressources externes variées dans une page Web. Cet élément permet de spécifier une hiérarchie de ressources, chacune constituant une solution de repli si la précédente n'est pas utilisable. On peut ainsi inclure une image animée accompagnée d'une version discrète de l'image et d'un texte qui décrit l'image, chacun pouvant être présenté en remplacement des précédents. SVG et SMIL disposent d'un mécanisme analogue avec l'élément *switch* qui fournit une liste d'alternatives adaptées à des environnements différents.

Les langages XHTML, SMIL et SVG sont organisés sous forme d'un ensemble de modules décrivant chacun une fonction particulière du langage. Cette modularisation permet d'identifier pour chaque contexte le sous–ensemble de modules supporté par le client. Dans le cadre de l'adaptation, cette connaissance est précieuse, pour la négociation comme pour l'identification précise du langage cible de la transformation.

L'expérience montre que les producteurs d'information n'utilisent pas pleinement l'ensemble de ces mécanismes. Il est donc important de développer des techniques d'écriture [4] et des environnements auteur qui aident les producteurs à utiliser au mieux les capacités d'adaptation des nouveaux formats XML. Ces environnements, comme Amaya [5] ou LimSee2 [6], mettent l'accent sur la structure logique des documents et fournissent des moyens efficaces pour manipuler le style séparément du document.

2.2 Description du contexte

Une fois que le document est dans un format favorable, il faut encore, avant de l'adapter, connaître les conditions précises de son utilisation. C'est le rôle de la description du contexte. Cette description concerne au moins le terminal et certaines préférences de l'utilisateur. Elle peut comprendre aussi les caractéristiques du réseau, les fonctionnalités du contenu et les capacités d'adaptation disponibles. Pour que l'ensemble des applications concernées à travers le Web puissent les partager, les descriptions de contexte doivent être exprimées dans un langage largement accepté et doivent être manipulables par toutes sortes de machines et de logiciels. Les formats standard de description de contexte jouent donc un rôle clé.

CC/PP [7] est un standard de ce type. Bien qu'il soit très récent, il semble être le dénominateur commun des applications mettant en jeu l'adaptation. C'est en fait un cadre de travail qui s'appuie à la fois sur XML et sur RDF (Resource Description Framework). Il permet de décrire les caractéristiques logicielles et matérielles d'un terminal, mais aussi les préférences de l'utilisateur.

2.3 Négociation

La description du contexte sert de base à une négociation entre un client et un serveur. Grâce à cette description le serveur peut connaître les contraintes qu'il doit satisfaire. Il se peut qu'il soit en mesure de les satisfaire toutes et, dans ce cas, la négociation est vite conclue. Mais il se peut aussi qu'un dialogue s'engage pour établir un compromis acceptable entre ce que le client demande et ce que le serveur peut offrir, ou simplement pour échanger davantage d'information de contexte, lorsque celle–ci est trop volumineuse pour être transmise en entier systématiquement. C'est le rôle du protocole de négociation de mener ce dialogue.

Dans le Web, le protocole le plus utilisé est HTTP. Sa première version offrait peu de possibilités de négociation. Tout au plus permettait–il au client d'énoncer quelques préférences (entête *Accept*) sur les jeux de caractères, les langues ou les média acceptés. Il laissait au serveur le soin de faire au mieux à partir de cette maigre information. HTTP 1.1 va plus loin et favorise un véritable dialogue de négociation. Cela permet d'impliquer le client et le serveur et ainsi de répartir la charge de la négociation.

Même si HTTP 1.1 apporte des progrès significatifs, il reste limité à une approche fondée sur des versions multiples préexistantes des contenus et ne permet pas vraiment une adaptation dynamique fondée sur des transformations.

2.4 Transformation

La transformation est la dernière phase d'un processus d'adaptation. Elle consiste à produire une version bien adaptée au contexte à partir d'une ressource unique. Elle peut intervenir à plusieurs niveaux différents : sur les objets média, sur la structure du document et sur certains aspects de sa sémantique.

Le premier niveau de transformation concerne les objets média. On peut transformer les images et la vidéo en appliquant une réduction de couleurs ou de niveaux de gris, en réduisant les dimensions, ou en changeant l'encodage. Une transformation plus drastique peut remplacer une vidéo par une image fixe qui en est extraite, ou une image animée par sa version discrète, ou encore une image fixe ou animée par un texte la décrivant.

On peut aussi intervenir au niveau de la structure logique, de l'organisation globale du document. La structure relativement riche d'un document HTML peut être ramenée à une structure XHTML Basic [8], qui est une version XMLisée de HTML conçue pour les terminaux mobiles. Un autre exemple est la transformation d'un format XML abstrait vers un format plus proche de la présentation, comme XHTML ou SVG. Cette transformation de structure peut s'accompagner d'une transformation des objets média qu'elle contient. L'outil de choix des transformations de structures est le langage XSLT, spécialement adapté aux transformations d'un langage XML vers un autre langage XML, ou vers une syntaxe textuelle d'un autre type.

Le dernier niveau de transformation considéré ici est celui de l'adaptation sémantique [9] [10]. Celle–ci consiste à considérer le sens associé aux éléments structuraux d'un document et aux relations qui les lient. L'adaptation est alors plus ciblée que par de simples transformations structurales, puisqu'elle tire partie du sens associé à la structure. À titre d'exemple, dans SMIL, les éléments *par* et *seq* ont des structures très proches du strict point de vue XML, mais ils portent des sémantiques fort différentes, celles d'objets présentés en parallèle ou en séquence. La prise en compte de cette sémantique permet de les traiter différemment lors de l'adaptation.

3 NAC, une architecture pour l'adaptation

Dans la section précédente, on a vu les principes de base de l'adaptation des documents multimédia sur le Web, ainsi que les principaux formats, protocoles et langages disponibles aujourd'hui dans ce domaine. Il reste encore à les mettre en oeuvre dans une architecture cohérente et à explorer ainsi leurs possibilités et leurs limites. C'est l'objectif de l'architecture répartie NAC [11] que nous avons définie et mise en oeuvre. NAC s'inscrit dans une ligne de travaux très actifs sur l'adaptation, tels que [12] et [13].

3.1 Architecture

NAC vise à assurer l'accès adapté aux contenus multimédia, où qu'ils se trouvent dans le réseau et quel que soit le protocole utilisé pour les transférer. Mais le serveur détenant l'information peut ne disposer d'aucun moyen d'adaptation et utiliser un protocole insuffisant en matière de négociation de contenu. Le client peut également souffrir des mêmes limitations. La solution consiste à interposer un *proxy* entre le client et le serveur. Ce proxy prend en charge l'essentiel de la négociation et de l'adaptation. En contact à la fois avec le client et le serveur, il a une vision globale de l'environnement et est ainsi bien placé pour effectuer une adaptation pertinente. Il peut s'exécuter sur une machine dédiée pour offrir des services d'adaptation à une large population de clients et de serveurs qui n'ont pas besoin d'être modifiés pour bénéficier de possibilités d'adaptation. Il peut aussi résider sur un serveur, soit pour optimiser les performances, soit pour offrir des services plus spécifiques à ce serveur.

Pour permettre une bonne adaptation, il faut un protocole de négociation. On l'a vu, les protocoles de transfert disponibles couramment sont généralement insuffisants dans ce domaine, voire inexistants. L'approche de NAC consiste à séparer la négociation et le transfert de contenu dans deux protocoles différents. Ainsi, même si un client ne supporte pas la négociation, le proxy peut s'engager pour lui, à partir de son profil, dans une négociation avec un serveur et lui retourner ensuite le contenu négocié. Pour cela, le client n'a besoin que du protocole de transfert habituel.

La description des contextes est une ressource importante que NAC stocke dans une base de profils. Celle–ci décrit les terminaux clients, les préférences des utilisateurs, l'état du réseau, les capacités des serveurs, etc. En la plaçant au plus près du proxy qui effectue l'adaptation, on rend celle–ci plus efficace. Le client n'a pas besoin d'envoyer une description complète de ses caractéristiques, puisqu'à partir de la seule indication du modèle d'appareil, on peut retrouver le détail de son profil dans la base. De même, les préférences des utilisateurs peuvent être stockées dans cette base, pour éviter de les transmettre à chaque connexion. Cette base doit évidemment être mise à jour régulièrement pour intégrer les nouveaux terminaux, changer les préférences des utilisateurs ou tenir compte des fluctuations du réseau.

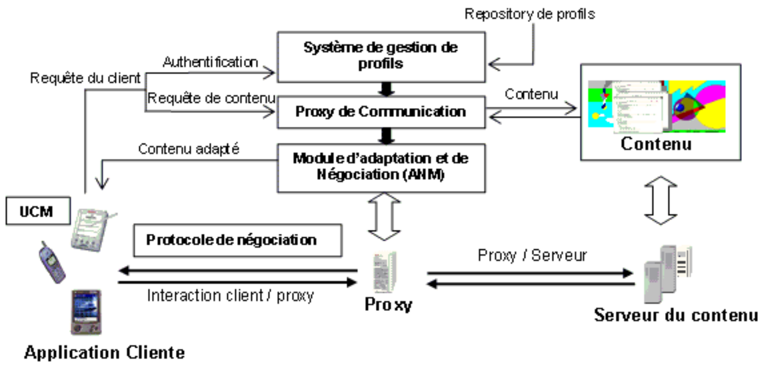

Fig 1 – Organisation générale de l'architecture NAC

Sur la base de ces principes, NAC est constitué comme le montre la figure 1. L'architecture comporte cinq composants :

Le proxy de communication assure l'accès au contenu et le transfert habituel entre serveur et client. Il reçoit les requêtes du client et les transmet au serveur. Inversement, il transmet au client les réponses du serveur, après les avoir transformées dans le module d'adaptation ANM.

Le module ANM effectue la négociation et l'adaptation. Il coopère avec les autres composants pour prendre la meilleure décision dans la négociation : choix de version du contenu, choix de méthode d'adaptation, choix de méthode de transmission du contenu final. L'adaptation est le plus souvent dynamique. Elle dépend du contexte (application cliente, formats acceptés, taille de l'écran, etc.)

Le module de contexte utilisateur UCM réside sur le client. Il prend en charge la négociation de façon indépendante de l'application cliente. Il peut travailler pour plusieurs applications différentes. C'est lui qui gère le profil client et il permet à l'utilisateur d'en changer à tout moment.

Le protocole de négociation met en oeuvre la stratégie de négociation. Il définit les interactions entre le module UCM du client et le processus de négociation (ANM) du proxy. Il permet d'interroger le terminal sur ses capacités instantanées et les préférences courantes de l'utilisateur ou de vérifier si le profil client a changé.

Le système de gestion de profils assure la gestion et l'interrogation des descriptions de contexte stockées dans la base de profils. Il travaille notamment pour le compte du module ANM.

3.2 UPS : représentation de profils

Dans les bases de profils, les descriptions de contexte utilisent un modèle appelé UPS qui se présente comme une extension de CC/PP [7]. Comme CC/PP, UPS utilise RDF pour représenter les profils sous la forme de graphes étiquetés. Les descriptions exprimées selon ce modèle concernent les profils et les ressources clients, les instances et les ressources de documents, les méthodes d'adaptation et l'état du réseau.

Les profils peuvent résider dans trois composants de l'architecture : sur le client, sur le proxy ou dans la base de profils. Dans tous les cas, les protocoles peuvent transporter l'information des profils jusqu'au composant qui en a besoin, mais ils ne transportent pas des profils entiers. Ils essaient de ne transmettre que le minimum. Parmi les informations véhiculées de cette façon figurent celles qui sont dynamiques, comme la bande passante disponible, la charge du serveur, ou la mémoire libre sur le terminal.

3.3 Méthodes d'adaptation

Le nombre et la diversité des paramètres qui interviennent dans le contenu, les caractéristiques du terminal et les autres éléments du contexte nécessitent une grande variété de méthodes d'adaptation. Toutes les méthodes envisagées dans la section 2.4 sont utilisées dans NAC, en plus des méthodes plus traditionnelles comme le choix d'une variante ou l'exploitation des alternatives offertes par les formats de documents.

Ces méthodes restent néanmoins d'un nombre raisonnable grâce à deux approches qui permettent l'utilisation de chacune d'entre elles dans plusieurs contextes différents. La première approche est fondée sur des paramètres qui déterminent le comportement précis de la méthode. La valeur de ces paramètres est prise directement dans le contexte, ou calculée à partir du contexte. C'est de cette façon par exemple que les images peuvent être adaptées à la largeur de l'écran du terminal, la méthode s'appuyant sur un paramètre qui donne la taille de l'écran. La deuxième approche utilise deux phases. Elle engendre d'abord des règles à partir du profil du client, puis elle applique ces règles au contenu à adapter [11].

4 Conclusion

L'adaptation des ressources multimédia du Web à l'environnement hétérogène constitué par les appareils et réseaux mobiles est un problème vaste aux facettes multiples.

Plusieurs techniques convergent à sa résolution, qui vont des formats de document indépendants des terminaux aux transformations de structures et de contenu, en passant par les langages de description de contexte et les protocoles de négociation.

Sur chacune de ces techniques des progrès ont été faits récemment et d'autres sont en cours, mais en même temps, il est nécessaire d'aborder les questions d'architecture : le modèle très simple du Web des débuts où un client dialogue directement avec un serveur à l'aide d'un unique protocole de transfert ne suffit plus. Des architectures réparties plus riches sont nécessaires pour permettre de nouveaux progrès dans ce domaine. C'est cette voie que nous avons voulu explorer en définissant l'architecture NAC, en la mettant en oeuvre [11] et en l'expérimentant [10] dans différents contextes.

Références

1. Ayars, J. et al. (eds): Synchronized Multimedia Integration Language (SMIL 2.0), W3C Recommendation (2001) http://www.w3.org/TR/smil20/
2. Ferraiolo, J., Fijisawa, J., Jackson, D. (eds.): Scalable Vector Graphics (SVG) 1.1 Specification, W3C Recommendation (2003) http://www.w3.org/TR/SVG/
3. Bos, B., Lie, H.W., Lilley, C., Jacobs, I. (eds.): Cascading Style Sheets level 2, CSS2 Specification, W3C Recommendation (1996 http://www.w3.org/TR/REC−CSS2/
4. Hanrahan, R., Merrick, R. (eds.): Authoring Techniques for Device Independence, W3C Working Draft (2003) http://www.w3.org/TR/di−atdi/
5. Amaya, http://www.w3.org/Amaya/
6. LimSee2, http://wam.inrialpes.fr/software/limsee2/
7. Klyne, G. et al. (eds.): Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0, W3C Recommendation (2004) http://www.w3.org/TR/CCPP−struct−vocab/
8. Baker, M., Ishikawa, M., Matsui, S., Stark, P., Wugofski, T., Yamakami, T.: XHTML Basic, W3C Recommendation (2000) http://www.w3.org/TR/xhtml−basic
9. Euzenat, J., Layaïda, N., Dias, V.: A semantic framework for multimedia document adaptation. Proceedings of the 18th International Joint Conference on Artificial Intelligence IJCAI'2003, Morgan Kauffman, San−Mateo (2003) 31−36
10. Lemlouma, T., Layaïda, N.: Context−Aware Adaptation for Mobile Devices, IEEE Int. Conf. on Mobile Data Management, Berkeley, California, USA, (2004)
11. Lemlouma, T., Layaïda, N.: Adapted Content Delivery for Different Contexts, Proc. International Symposium on Applications and the Internet (SAINT 2003), IEEE Computer Society (2003) 190−197
12. Lum, W.Y., Lau, C.M.: A Context−Aware Decision Engine for Content Adaptation, Pervasive Computing, vol. 1, no 3 IEEE (2002) 41−49
13. Mohan, R., Smith, J., Li, C.S.: Adapting Multimedia Internet Content For Universal Access, IEEE Transactions on Multimedia, March 1999, 104−114