



**HAL**  
open science

## Adaptation de contenus multimédia, composition de web services et pair-à-pair

Jean-Claude Moissinac, Isabelle Demeure, Zakia Kazi-Aoul

### ► To cite this version:

Jean-Claude Moissinac, Isabelle Demeure, Zakia Kazi-Aoul. Adaptation de contenus multimédia, composition de web services et pair-à-pair. Convergence des Réseaux, de l'Informatique, et du Multimédia pour les E-Services (CRIMES), Nov 2009, Saint Denis de La Réunion, France. pp.21. hal-00423824

**HAL Id: hal-00423824**

**<https://hal.science/hal-00423824v1>**

Submitted on 12 Oct 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# *Adaptation de contenus multimédia, composition de web services et pair-à-pair*

Jean-Claude Moissinac<sup>(1)</sup>, Isabelle Demeure<sup>(1)</sup>, Zakia Kazi-Aoul<sup>(2)</sup>

(1) *Telecom ParisTech, 46 rue Barrault, 75634 Paris, France.*

(2) *ISEP, France.*

---

## Résumé

Cet article présente une architecture orientée services qui adapte, par rapport à un contexte utilisateur, des documents multimédia composites. Dans cette approche, les ressources d'adaptation (ou adaptateurs) sont partagées par des nœuds distribués dans le réseau, dans une logique P2P. Ces nœuds peuvent aussi bien être des utilisateurs que des prestataires de services. La mise en œuvre de cette plateforme d'adaptation est basée sur les services Web. Le langage de description des services Web est étendu par des informations sémantiques qui permettent de décrire les services d'adaptation afin qu'ils puissent être déclarés, recherchés et instanciés automatiquement. Une composition d'adaptateurs est décrite comme un graphe qui peut être dynamiquement réorganisé. Nous définissons une chaîne complète d'adaptation en utilisant les Web Services auxquels ont été ajoutés des éléments de description sémantique. Cette approche, s'appuyant sur un effort pour limiter le couplage entre éléments du système, est prometteuse en terme de performances, de versatilité et d'évolutivité.

**Keywords:** multimédia, web services sémantiques, pair à pair, adaptation, composition de services

---

## 1 Logique pair à pair et adaptation distribuée de documents multimédia

De plus en plus d'internautes, caractérisés par des contextes différents, souhaitent s'échanger des contenus multimédia par le biais d'applications d'échange Pair-à-Pair (P2P). Un contexte utilisateur est caractérisé par les préférences de l'utilisateur, les capacités de son terminal ainsi que les caractéristiques de son réseau d'accès. Une partie de ces éléments de contexte peuvent être dynamiques et évoluer en fonction de l'instant ou du lieu. On aura, par exemple, un utilisateur préférant l'anglais et utilisant un terminal avec un petit écran (5cm x 5cm), mais une plutôt bonne résolution (320 x 320), avec un accès Wifi tandis qu'un autre sera sur un réseau filaire avec un grand écran, préférera l'espagnol...

Compte tenu de la variété des documents multimédia que ces usagers s'échangent, il est difficile d'exiger d'un producteur de contenus multimédia de fournir autant de versions d'un document que de contextes possibles. Définir les moyens permettant l'adaptation des documents variés à des contextes variés devient alors une nécessité.

Nous considérons qu'il est souhaitable d'offrir aux utilisateurs la possibilité de proposer des ressources de traitement permettant l'adaptation de ces documents. La communauté des utilisateurs du dispositif bénéficiera ainsi des ressources inutilisées des nœuds du réseau en étendant le modèle architectural Client/intermédiaire(s)/serveur au modèle P2P. Notons que la notion de P2P fait référence au fait que chaque nœud peut être client ou serveur, offrir des ressources aux autres nœuds et lui-même bénéficier en retour des ressources offertes par les autres nœuds. Plus particulièrement, chaque nœud du réseau peut être à la fois consommateur de contenus multimédia, producteur de contenus multimédia et/ou fournisseur d'adaptateurs. Nous verrons plus loin que certains nœuds (planificateurs) doivent avoir un rôle spécial dans le dispositif (un cas particulier de super-pair dans un réseau pair à pair).

La plupart des solutions d'adaptation existantes sont basées sur le modèle client / intermédiaires / serveur (par exemple, ADMIT [BOSZ03], NAC [LEML04] et APPAT [LAPA05]). Le système que nous proposons dans le présent document est appelé PAAM pour «a service-oriented architecture for the Provision of AdAptable Multimedia composed documents ».

Le reste de ce document est organisé comme suit : à la section 2, nous présentons quelques travaux connexes afin de déterminer les défis auxquels nous devons faire face; nous déterminons ensuite les rôles des composants nécessaires pour permettre la réalisation d'un système dynamique d'adaptation distribuée; nous finissons cette section en présentant les informations sémantiques nécessaires pour assurer cette distribution. Dans la section 3, nous introduisons le système PAAM qui constitue une implémentation partielle des concepts présentés à la section 2. Enfin, pour conclure, nous revenons sur les principaux atouts de cette approche.

## 2 Composantes d'un système d'adaptation distribuée basée sur les Web Services Sémantiques

### 2.1 Systèmes d'adaptation antérieurs

Ces dernières années de nombreux travaux de recherche se sont attaqués au problème de l'adaptation de documents multimédia à un contexte d'utilisation.. Nous allons évoquer ceux qui traitent de l'adaptation distribuée et/ou de l'utilisation des services Web.

APPAT (Adaptation Proxies PIATform) est une architecture basée sur des intermédiaires d'adaptation et introduit la notion de la distribution de l'adaptation sur des nœuds appelés « Adaptation Proxies » [LAPA05]. Cette architecture est spécialement conçue pour des applications où les données sont transmises entre plusieurs participants. Cependant, elle n'est pas conçue pour être extensible. En outre, elle traite seulement des médias élémentaires tels que des vidéos ou des images.

DCAF (Distributed Content Adaptation Framework) est une architecture orientée services où les adaptateurs de contenu sont implémentés à l'aide des Web Services[BERH05]. Les auteurs proposent un algorithme de composition de service s'appuyant sur la technique de planification IA (AI planning technique). Les services d'adaptation sont décrits à l'aide d'ontologie.

KoMMa (knowledge-based MultiMedia adaptation) [JANN06] est une plateforme visant à planifier et à exécuter des opérations d'adaptation à travers un moteur de prise de décision pour l'adaptation. Ce moteur possède l'intelligence nécessaire pour construire les séquences d'adaptation qui réalisent effectivement les transformations sur la base d'une requête client. Cette logique d'adaptation est également basée sur les techniques de planification IA qu'elle réalise à l'aide de Prolog; un plan d'adaptation est calculé pour adapter la ressource originale afin de satisfaire les contraintes et les préférences du client en s'appuyant sur un certain nombre de règles.

DCAF ainsi que koMMa ne gèrent pas dynamiquement des adaptateurs, notamment leurs éventuelles disparitions.

Le passage à l'échelle, l'évolutivité, la versatilité et la robustesse représentent les limitations des architectures étudiées. L'objectif de PAAM est de concevoir et réaliser un système qui adapte, au contexte d'un utilisateur, des documents multimédia composites qu'il échange avec d'autres utilisateurs, où les adaptateurs sont fournis par des nœuds du réseau découverts dynamiquement en fonction de l'adaptation nécessaire.

Les avantages d'une telle solution sont la robustesse, l'évolutivité, les bases pour une gestion de l'équilibrage de charge, la tolérance aux pannes et la possibilité d'avoir un large spectre, évolutif, de traitements d'adaptation. Dans le reste de ce document, nous appelons « document multimédia composé » un document dans lequel plusieurs médias élémentaires (vidéo, audio, texte ou image) sont spatialement organisé et/ou temporellement synchronisés. Notez que, pour l'instant, nous ne supportons pas le streaming : l'adaptation des documents multimédia composites doit être terminée avant le début de la livraison à l'utilisateur final. Le streaming sera introduit ultérieurement.

## 2.2 Les rôles dans un système dynamique d'adaptation distribuée

Nous allons voir ici quels sont les fonctions essentielles pour adapter automatiquement des documents multimédia très variés à des contextes d'utilisation très variés. Nous voulons aussi que l'approche permette de traiter des situations nouvelles, non prévues initialement, avec peu d'impact sur l'existant.

Une conséquence immédiate de ces hypothèses est qu'il est nécessaire d'avoir un système qui se configure dynamiquement pour réaliser une adaptation convenable d'un document multimédia donné dans un contexte connu. Le système ne doit pas être un dispositif figé, réalisant un ensemble de tâches d'adaptation fixées pour tenir compte d'une liste d'éléments de contexte pré-définie.

Les rôles importants sont :

- le **planificateur**: il prend en entrée la description du document multimédia et la description du contexte et produit un graphe d'adaptation, c'est-à-dire la description d'une série d'étapes élémentaires, éventuellement soumises à des conditions, exécutées en parallèle ou en séquence, qui permettent d'obtenir le document adapté; il est au centre du dispositif; le planificateur peut être implémenté de différentes façons allant d'une simple heuristique limitée telle que celle proposée dans [KIMI05] à un système sophistiqué de résolution de problème comme proposé dans [VITV07] pour d'autres contextes;
- le **fournisseur de contexte**: un composant du dispositif doit fournir le contexte; la façon dont il le fait n'a pas d'importance (il peut interroger une banque de donnée ou un ensemble de capteurs, il peut recevoir des informations fournies avec la requête de l'utilisateur...); en fait, comme de nombreux travaux traitent de la collecte et de la mise à disposition du contexte, nous laissons cette question hors de notre champs d'investigation et considérons qu'une 'boîte noire' est disponible et fournit le contexte,
- la **source de document multimédia** composé: un composant doit permettre d'obtenir une description du document multimédia à adapter; dans certains cas, cette description est le document lui-même (ex: document SMIL), dans d'autres, il s'agira d'un meta-document obtenu auprès du fournisseur de service ou généré par un module externe à partir du document à adapter,
- le **composeur**: ce composant doit instancier le plan en trouvant pour chaque étape élémentaire un service capable de la réaliser; pour cela, il fait appel au module de recherche de services, choisit parmi les services trouvés, puis établit la façon dont vont être exploités les fonctions des services trouvés; si un adaptateur n'existe pas, il pourrait être remplacé par la composition de plusieurs adaptateurs; par exemple, si un traducteur du français vers le mandarin n'existe pas, il sera remplacé par l'enchaînement en séquence de deux traducteurs : du français vers l'anglais puis de l'anglais vers le mandarin. [ZENG03] présente une façon de construire de tels graphes, mais dans un autre contexte.
- le module de **recherche de service** : ce module traite des demandes de services et répond par une liste, éventuellement vide, de services répondant à la demande; diverses implémentations peuvent être acceptées dans divers modules de recherche; par exemple, un module pourra faire la recherche sur Internet comme dans [ALMA08] ou dans [MOIS07], tandis qu'un autre se limitera à une liste locale de services accessibles,
- le **moteur d'exécution** du plan: ce composant prend le plan instancié et supervise l'exécution des services retenus; à la fin de l'exécution, il fournit le résultat de l'adaptation ou un lien vers ce résultat,
- les **adaptateurs élémentaires**: ces composants assurent une adaptation spécifique; par exemple, il s'agira d'un composant de transcodage de vidéo, ou d'un composant de transformation de la mise en page du document; nous verrons plus les types d'adaptateurs sur lesquels un tel dispositif peut s'appuyer.

Un cycle d'adaptation est initié par un consommateur qui envoie au **planificateur** une requête, composée d'une référence à un document multimédia et d'une référence au contexte. Le planificateur obtient auprès du **fournisseur de contexte** et de la **source de document** multimédia composés les descriptions dont il a besoin. Le planificateur peut alors appliquer ses algorithmes et ses heuristiques d'adaptation. Il décide quelles sont les opérations d'adaptation nécessaires pour fournir un document multimédia composé adapté au contexte de l'utilisateur. Cet enchaînement d'opérations est appelé graphe d'adaptation abstrait [KAZI08] car il n'y a pour l'instant aucune instanciation. Ce graphe est envoyé au **composeur** qui cherche les bons adaptateurs et constitue un graphe d'adaptation concret qui est un enchaînement, en séquence ou en parallèle, d'adaptateurs prêts à être

exécutés. Le graphe physique est communiqué au **moteur d'exécution** qui assure les tâches de base (extraction d'informations des descriptions, tests,...) et supervise le travail des **adaptateurs élémentaires**. A la fin du traitement, le moteur d'exécution renvoie au planificateur une référence permettant d'accéder au document adapté.

L'ensemble des modules ci-dessus peut-être porté par des noeuds différents du réseau. L'utilisateur ne communique qu'avec le planificateur dans la phase d'adaptation. Ensuite, il récupère le document d'une façon classique (http ou P2P).

Chaque consommateur peut choisir son planificateur; il peut alors y avoir des planificateurs spécialisés. Chaque planificateur peut avoir ses propres stratégies d'adaptation plus ou moins pertinentes pour certains documents ou certains contextes (ex: un planificateur spécialisé dans l'adaptation de documents pour les malvoyants; un planificateur spécialisé pour l'adaptation à l'iPhone...).

### 2.3 La sémantique nécessaire à chaque étape de l'adaptation

#### *Communication du planificateur avec la source de document multimédia*

Le planificateur doit disposer d'une description du document multimédia qui lui permette de comprendre l'agencement spatial et temporel des éléments qui composent ce document. En particulier, il doit identifier les médias élémentaires utilisés -textes, images, vidéo, audio-, leur mise en page, leur utilisation au cours du temps. Cette sémantique est explicite dans le langage SMIL [SMIL]. Les documents SMIL sont donc directement exploitables dans ce contexte. Nous avons cherché à représenter d'autres types de documents, et à représenter des ensembles de documents (avoir une description qui peut s'appliquer à plusieurs documents similaires). Deltour [DELT06] propose un langage dérivé de SMIL, qui présente un degré d'abstraction supplémentaire et permet d'avoir une seule description pour un ensemble de documents faits sur le même modèle. Cependant, ce langage nous paraît insuffisant pour décrire les documents d'origine dans une grande variété de langages (même en se limitant à des langages basés sur XML, tels que SMIL, SVG, XHTML, MPEG-4 XMT). Nous étudions une extension ou une variation du langage précédent afin de couvrir ce besoin. Ce langage pourrait constituer un standard de communication entre le planificateur et la source de documents multimédia; certains planificateurs pourraient accepter d'autres langages.

#### *Communication du planificateur avec le fournisseur de contexte*

Le planificateur doit aussi disposer d'une description du contexte qui contienne les informations qu'il sait traiter. De nombreux travaux portent sur la gestion du contexte de l'utilisateur. Ici, il nous suffit que le fournisseur de contexte produise une description XML du contexte compréhensible par le planificateur. Le fournisseur de contexte est donc une sorte de médiateur entre la gestion du contexte proprement dite (qui est externe à notre proposition) et le planificateur.

#### *Communication du planificateur avec le compositeur*

Le planificateur définit une sorte d'algorithme d'adaptation s'appuyant sur un ensemble d'éléments de structure de programme (affectation de variables, exécutions conditionnelle...) et un ensemble d'opérations élémentaires abstraites (ex: réduire les dimensions d'une vidéo). Pour communiquer son résultat au compositeur, il lui faut un langage de description de workflow ou de composition de services. BPEL<sup>1</sup> est un bon candidat; nous avons utilisée une solution ad-hoc inspirée de BPEL; nous devons vérifier que les capacités de BPEL à exprimer une composition de services abstraits est suffisante et cohérente avec nos descriptions de services multimédia.

Il faut aussi disposer d'un ensemble de description de services. En effet, plusieurs composantes du dispositif doivent avoir une compréhension partagée du rôle d'un certain nombre de services élémentaires. La description de l'interface des services (comme dans WSDL) ne suffit pas; une description 'sémantique' du rôle et des limitations du service est nécessaire. Nous pensons que la solution est de s'appuyer sur une ontologie des services multimédia élémentaires. Dans [KAZI08], nous avons abordé la question de la description des services multimédia élémentaires. Nous avons établi une description d'une quarantaine de services à l'aide d'OWL<sup>2</sup>. Cependant, d'autres candidats semblent avoir la faveur d'acteurs importants de la communauté des Web Services Sémantiques et nous pensons devoir revoir notre ontologie de services dans ce sens. [WSMO] pourrait être

---

1 <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>

2 OWL specification: <http://www.w3.org/TR/2004/REC-owlfeatures-20040210/>

retenu. A titre d'illustration, la catégorie de service 'Transcodeur' comprend des services qui permettent de changer le codage d'une vidéo (passer de WMV à MPEG-4, par exemple) ou d'une séquence audio. Dans la catégorie 'Transformeur', on aura un service qui permet de couper un morceau d'une image. Dans la catégorie 'Transmodeur' -pour changer de modalité-, on aura un service qui transforme un texte en séquence audio grâce à une lecture automatique, ou qui transforme un texte en image (par exemple, parce que le texte utilise une police non disponible sur le terminal de l'utilisateur).

#### *Communication du composeur avec la recherche de services*

Le planificateur a utilisé des services abstraits. En fait, chaque service abstrait désigne une classe de service définie dans notre ontologie dont il va falloir trouver au moins une instance. Pour chaque service nécessaire, le composeur va établir une requête auprès du module de recherche de service. Au minimum, la requête indique la classe de service cherchée; la méthode proposée dans [PDCN08] permet de désigner une classe de service par une URN<sup>3</sup>, par exemple « urn:paam:transcodeur:video:video:VideoFormatConverter » . La requête peut également contenir des contraintes complémentaires (par exemple, format de fichier d'entrée pour un service de traitement d'images).

Nous utilisons l'outil WSDL de description des services Web qui est une spécification du W3C fournissant une syntaxe XML pour décrire l'interface d'appel d'un Web Service [14]. Contrairement à nos travaux précédents, nous proposons désormais d'envelopper la description WSDL dans un document « paam » qui y apporte des compléments sémantiques. Cela permet de découpler fortement les différents niveaux d'implémentation.

Les documents paam peuvent être regroupés et exploités dans des systèmes d'indexation et de recherche efficaces, indépendants de l'emplacement des fichiers wsdl, eux-mêmes indépendants de l'emplacement des Web Services correspondants. Cela contribue à un couplage faible. La production d'un service, celle de la description WSDL de son interface et celle de sa description sémantique PAAM peuvent être réalisées par des entités complètement indépendantes et mis en ligne sur des dispositifs indépendants. Cela permet d'exploiter dans l'architecture PAAM des services créés sans aucune connaissance de PAAM.

Afin de permettre d'exploiter dans PAAM des services créés hors de PAAM, il est possible à ce niveau d'indiquer la correspondance entre les paramètres définis dans l'ontologie des adaptateurs de PAAM et les paramètres définis dans l'interface WSDL de l'adaptateur. Dans le cas d'adaptateurs basés sur cette ontologie, aucune information complémentaire n'est nécessaire à ce niveau.

```
<paam xmlns="http://www.perso.enst.fr/~kazi/paam2.xsd" xmlns:wSDL= "...">
<urn>urn:paam:transmoder:video:audio:audioExtractor </urn>
<inputTypes><mimetype>video/mpeg;video/quicktime</mimetype></inputTypes>
<outputTypes><mimetype>audio/mpeg</mimetype></outputTypes>
<import importType="http://schemas.xmlsoap.org/wsdl/"
    location="http://shadok.enst.fr/.../test_BindingInternal.wsdl"
    namespace="http://www.eclipse.org/stp/b2j/2006/02" />
</paam>
```

Figure 2. Une description d'adaptateur PAAM

Dans la description de la figure 2, on trouve un extrait de la description PAAM d'un adaptateur. Cette description fait référence à la description WSDL du service -qui comprend donc des descriptions des paramètres d'entrée et de sortie du service et des liens permettant de l'invoquer- et ajoutent à cette description quelques informations de plus haut niveau:

- la valeur urn:paam:transmoder:video:audio:audioExtractor de l'élément <urn>, nous indique que le service assure un changement de modalité en prenant en entrée une vidéo et en fournissant en sortie une séquence audio, le nom générique du service étant AudioExtractor
- l'élément <inputTypes> sert à donner des indications sur les paramètres d'entrée; ici, la seule indication est une liste de types MIME acceptés pour l'entrée vidéo de l'instance de service décrite,

---

<sup>3</sup> URN specification: <http://www.ietf.org/rfc/rfc2141.txt>

- enfin, l'élément <outputTypes> sert à donner des indications sur les paramètres de sortie; ici, la seule indication est le type MIME fourni pour la sortie audio de l'instance de service décrite,

#### Communication du composeur avec le moteur d'exécution

Le composeur a transformé le résultat du planificateur en remplaçant des services élémentaires abstraits par un ou plusieurs services concrets. A nouveau, BPEL est le langage de prédilection pour décrire une telle composition de Web Services. En fait, le composeur peut passer directement un graphe BPEL à un moteur d'exécution BPEL qui va se charger de l'exécution de l'adaptation.

Un tel graphe aura toujours les mêmes entrées et les mêmes sorties:

- en entrée, la description du document multimédia et la description du contexte,
- en sortie, la référence permettant d'accéder au document multimédia adapté.

Nous étudions les propositions d'environnement d'exécution WSMX [WSMX07]. Cependant, l'architecture proposée par WSMX, pour la composition et l'exécution automatique de Web Services Sémantiques, intègre plusieurs des composants présentés ici, notamment le planificateur, le composeur et le moteur d'exécution. Il semble qu'il n'y ait pas à ce jour d'implémentation WSMX qui nous assure la modularité et le faible couplage que nous obtenons avec les composants décrits ci-dessus.

### 3 L'exemple de l'architecture PAAM: implémentation actuelle

L'essentiel des descriptions ci-dessus ont un caractère général et peuvent donner lieu à des implémentations très diverses basées sur les principes exposés.

Nous avons mis en place une chaîne d'adaptation complète en utilisant les technologies des services Web. Nous avons développé des Web Services avec AXIS2 et avec JAX-WS.

Nous nous sommes inspirés de [MPEG-21] pour décrire les éléments du contexte. Cette description nous permet notamment de décrire des informations spécifiques telles que les mime-types supportés par chaque lecteur. Concernant le document multimédia composé, nous avons travaillé sur des documents écrits en langage SMIL.

Dans l'implémentation actuelle, le planificateur et le composeur sont implémentés au sein d'un même processus. Nous avons produit des graphes d'adaptation, d'abord dans un format spécialement créé, puis nous avons validé l'utilisation de BPEL pour la représentation des graphes concrets. Nous avons utilisé le moteur ODE pour l'exécution des graphes BPEL.

Le graphe produit une URL d'un cache où le document SMIL adapté sera accessible. Les médias élémentaires adaptés peuvent éventuellement être répartis dans les caches des adaptateurs.

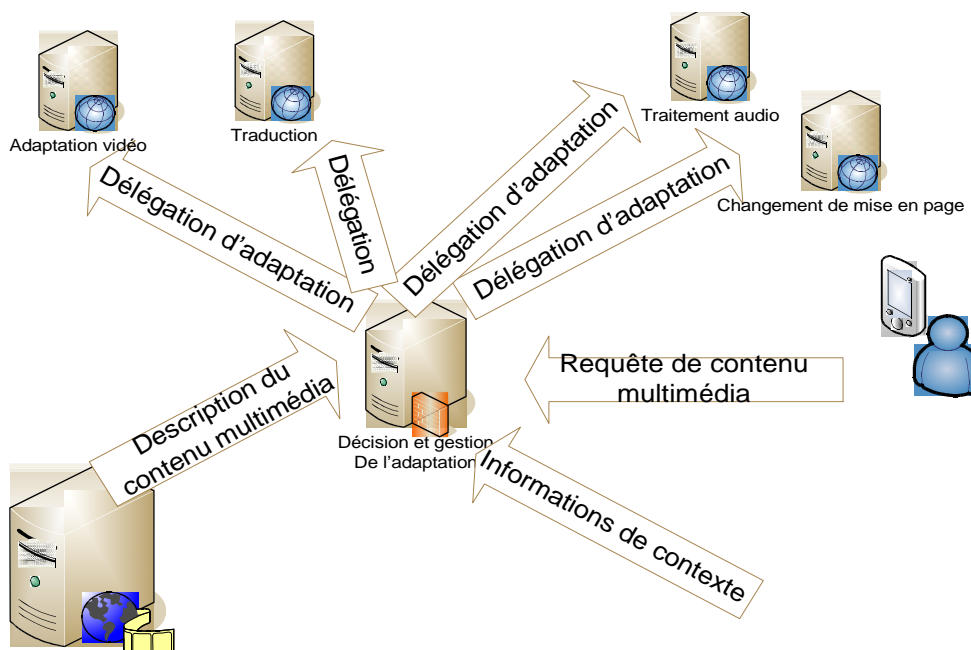


Figure 3

Dans la figure 3, le document requis par l'utilisateur pour une consultation sur son équipement mobile:

- nécessite un codage de la vidéo pour obtenir une dimension et un format acceptables par le terminal (en haut à gauche),
- nécessite la traduction de certains textes composants le document multimédia, afin de s'adapter à la langue de l'utilisateur (en haut, au centre gauche),
- nécessite un transcodage audio pour passer à un format reconnu par le terminal,
- enfin, il faut reconstituer un nouveau document multimédia qui référence correctement les médias élémentaires modifiés (en haut à droite); ici, on supposera que le superviseur permet d'assurer une redirection vers ce document multimédia re-composé stocké au niveau du cache de l'hébergeur de ce dernier service.

Cet exemple montre de façon simplifiée comment le traitement d'un document multimédia peut être assuré de façon distribuée, en gagnant éventuellement en efficacité d'une part du fait de la spécialisation de chacun des services, d'autre part de la parallélisation d'au moins une partie des traitements.

## 4 Conclusion et perspectives

Dans cet article, nous avons présenté une architecture orientée services qui adapte dynamiquement des documents multimédia composites par rapport aux contextes des usagers, dans un environnement où les ressources d'adaptations sont proposées par des participants à un réseau (utilisateurs et/ou prestataires). Cette architecture est un cas d'étude significatif pour la démonstration des bénéfices de la mise en œuvre des Web Services Sémantiques. Nous disposons avec le système PAAM d'une implémentation partielle des principes proposés.

Notre approche apporte plus de possibilités en terme de flexibilité, d'évolutivité, d'équilibrage de la charge et de robustesse que les architectures basées sur une architecture centralisée, figée et où les rôles sont prédéfinis. PAAM introduit la notion de partage des ressources et de découverte grâce à une utilisation pertinente de Web Services Sémantiques.

Le système d'adaptation PAAM prend en entrée un document multimédia composé et un contexte. Il prend alors les décisions d'adaptation, cherche les adaptateurs nécessaires, éventuellement les compose et pilote l'adaptation. Pour chaque document demandé, le système d'adaptation est en mesure de redéfinir le graphe d'adaptation en tenant compte des changements dans l'environnement des adaptateurs.

Nous avons réalisé une chaîne d'adaptation complète. Nous avons utilisé les services Web pour implémenter les adaptateurs. Notre approche conduit à des gains de performance, en exploitant la parallélisation des processus d'adaptation et des bénéfices fonctionnels, en autorisant l'exploitation de divers services disponibles sur le réseau. Elle devrait pouvoir tirer parti à l'avenir des travaux menés par OASIS sur les environnements d'exécution de Web Services Sémantiques.

Nous sommes convaincus que l'approche proposée par PAAM sera, dans l'avenir, suivie par d'autres travaux de recherche, car elle répond à la nécessité de passer du stockage distribué aux traitements distribués dans les réseaux P2P, en bénéficiant ainsi des ressources logicielles inutilisées et de la bande passante et des capacités de traitement disponibles sur le réseau Internet.

## Références

- [ALMA08] Al-Masri, Eyhab; Mahmoud, Qusay H., Internet Computing, IEEE Publication, 2008, page(s): 74-77 Volume: 12, Issue: 3
- [BERH05] G. Berhe, L. Brunie, J-M. Pierson, Distributed Content Adaptation for Pervasive Systems. Proc. IEEE International Conference on Information Technology, 2005, Las Vegas, Nevada, USA, 234-241.
- [BOSZ03] L. Böszörményi, H. Hellwagner, H. Kosch, M. Libsle, S. Podlipnig, Metadata driven adaptation in the ADMITS project, EURASIP Signal Processing: Image Communication Journal, (Special Issue on Multimedia Adaptation) 18(8), 2003 , 749-766
- [CABR06] Cabral, L., Domingue, J., Galizia, S., Gugliotta, A., Norton, B., Tanasescu, V., Pedrinaci, C. (2006): IRS-III: A Broker for Semantic Web Services based Applications. In proceedings of the 5th International Semantic Web Conference (ISWC 2006), Athens, USA.



- [DELT06] R. Deltour and C. Roisin, The LimSee3 Multimedia Authoring Model. ACM Symposium on Document Engineering, 10-13 October 2006, Amsterdam, The Netherlands, pp. 173-175 ([PDF](#))
- [GIOI04] P. Gioia and al., ISIS: Intelligent Scalability for Interoperable Services, Proc 1st European Conference on Visual Media Production (CVMP), 2004, 295-304.
- [JANN06] D. Jannach, K. Leopold, C. Timmerer, H. Hellwagner, A knowledge-based framework for multimedia adaptation, Applied Intelligence, 24(2), 2006, 109-125.
- [JAXWS] Home page of the project JAX-WS: <https://jaxws.dev.java.net/>
- [LAPA05] J.-C. Lapayre, F. Renard, Appat: a New Platform to Perform Global Adaptation, Proc. The First IEEE International Conference on Distributed Frameworks for Multimedia Applications, 2005, 351-358.
- [LEML04] T. Lemlouma and N. Layaïda, Context-Aware Adaptation for Mobile Devices, Proc. IEEE International Conference on Mobile Data Management , 2004.
- [KAZI08] Z. Kazi-Aoul and I. Demeure and J. C. Moissinac. PAAM: A WEB SERVICES ORIENTED ARCHITECTURE FOR THE ADAPTATION OF COMPOSED MULTIMEDIA DOCUMENTS, PDCN 08
- [KIMI05] M. Kimiaei-Asadi, J-C. Dufourd, Context-aware Semantic Adaptation of Multimedia Presentations, Proc. IEEE International Conference on Multimedia & Expo (ICME 2005), Amsterdam, 2005.
- [SMIL] D. Bulterman and al., Synchronized Multimedia Integration Language , W3C 12/2005
- [WSDL01] W3C: WSDL: Web services Description Language (WSDL) 1.1, (2001). (<http://www.w3.org/TR/2001/NOTE-wsdl20010315>)
- [WSMO] WSMO Working Group (2004), D2v1.0: Web service Modeling Ontology (WSMO). Working Draft, (2004). (<http://www.wsmo.org/2004/d2/v1.0/>).
- [WSMX07] WSMX Working Group (2007), The Web Service Modelling eXecution environment, <http://www.wsmx.org/>.
- [ZENG03] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, Q.Z Sheng, Quality Driven Web Services Composition, Proc. International WWW Conference, 2003, Budapest.