

**BADJI MOKHTAR UNIVERSITY-ANNABA -
UNIVERSITE BADJI MOKHTAR-ANNABA-**



**Faculté des sciences de l'ingénieur
Département d'Informatique**

Année : 2009

MEMOIRE

Présentation en vue de l'obtention du diplôme de magister

**Vers un système ouvert d'adaptation multimédia pour téléphones
mobiles**

Option

Intelligence Artificielle

Par : Saighi Asma

Directrice de mémoire: Nacira Ghoualmi

Grade : MC Université d'ANNABA

DEVANT LE JURY

PRESIDENT: Hayet Farida Merouani

Grade MC Etablissement d'Annaba

EXAMINATEURS: Abdallah Boukerame

Grade MC Etablissement de Sétif

Fadila Atil

Grade MC Etablissement d'Annaba

Dédicaces

À la mémoire de mon père et ma grand-mère...

Je dédie ce travail au grand cœur et à ma tendre et géniale mère.

Elle m'a toujours supportée et soutenue dans tout ce que j'ai entrepris. J'espère seulement qu'elle sera fière de moi.

À ma chère grand mère Louisa.

À mes chers frères : Khaled, Tarek et le petit Aimen Wail.

À mes adorables oncles : Abdslem, Yacine, Nasreddine, Djaml, Foued et Farid ainsi qu'à leurs épouses.

À mes tantes : Radia, Houria, Malika, Nadia, Hanifa et Naziha.

À mes cousines et cousins: Samiha, Sirine, Islem, Cheima, Ritej, Abd El Fateh, Marame, Romaissa, Soundous, Ala, Mohammed, Soumia et Sara.

À mes chères amies : Amina, Leila, Soulef, Ahlem et Karima.

« لن تنال قليلا مما تحب حتى
تصبر على كثير مما تكره »

REMERCIEMENTS

Je ne serais pas arrivée jusque là si mon encadreur, Dr. Nacira GHOUALMI, ne m'avait pas prise sous son aile bienveillante pendant cette année. Je tiens à lui exprimer toute ma gratitude pour ses précieux conseils, sa disponibilité ainsi que sa contribution à ce travail. J'aimerais également la remercier pour la pleine confiance qu'elle m'a accordée. Je remercie également, le professeur Yahia SLIMANI de m'avoir accueilli au sein de son laboratoire en Tunisie.

Je remercie également les membres du jury :

Madame Hayet Farida MEROUANI Maître de conférence à l'université Badji Mokhtar de m'avoir fait l'honneur de présider mon jury de Magister,

Monsieur Abdallah BOUKERAME Maître de conférence à l'université de Sétif d'avoir accepté de juger ce travail,

Madame Fadila ATIL Maître de Conférences à l'université Badji Mokhtar d'avoir accepté de faire partie de mon jury.

Résumé

L'information multimédia connaît ces dernières années un essor considérable et s'impose dans tous les secteurs socio-économiques. La vie quotidienne est elle aussi conquises par l'information multimédia et la mobilité ainsi que l'hétérogénéité des mobiles justifient les investigations dans le domaine de l'adaptation multimédia. Le problème de pouvoir envoyer n'importe quel contenu multimédia d'un client possédant un téléphone mobile multimédia à un autre téléphone multimédia sans ce soucier d'avoir un message d'écho. Pour faire face à ce problème, il est évident d'adapter le contenu multimédia avant d'être envoyé au client multimédia final. Dans ce mémoire, nous présentons une architecture d'un système ouvert d'adaptation multimédia. Notre système est basé sur le modèle Client Intermédiaire Serveur où le Proxy constitue le coeur de notre architecture. Le proxy est constitué de quatre composants principaux: module de communication, base de données, module de prise de décision et module d'adaptation. L'objectif de cette architecture est de procurer une adaptation entre clients multimédia en émission et en réception. Nous avons modélisé avec UML la base de données des scénarios illustratifs qui représentent nos études de cas. La base des téléphones mobiles et la base des données multimédia sont modélisées. L'implémentation est faite avec JBUILDER.

Mots clé : Mobile, Adaptation multimédia, Client Intermédiaire Serveur, Scénarios d'adaptation, base de mobiles.

ملخص

لقد عرفت المعلومات المتعددة الوسائط خلال السنوات الأخيرة انطلاقة معتبرة انصبت في كل المجالات الاجتماعية والاقتصادية. الحياة اليومية تأثرت كثيرا بالمعلومات المتعددة الوسائط و حركية و عدم تجانس الهواتف النقالة يبرر تعدد البحوث في مجال تكييف الوسائط المتعددة. و إشكالية التمكن من إرسال أي محتوى متعدد الوسائط من طرف زبون يملك هاتف محمول متعدد الوسائط إلى هاتف محمول آخر متعدد الوسائط دون الاكتراث من أن تكون النتيجة هي تعذر وصول الرسالة. لمواجهة هذا المشكل، من الضروري تكييف المحتوى المتعدد الوسائط قبل إرساله إلى الزبون المتعدد الوسائط النهائي. نطرح في هذه المذكرة ، بنية لنظام مفتوح للتكييف المتعدد الوسائط. نظامنا يقوم على نموذج زبون وسيط خادم أين يمثل الوسيط لب بنيتنا وهذا الأخير(الوسيط) يتكون من أربعة عناصر رئيسية: وحدة اتصال، قاعدة بيانات، وحدة لاتخاذ القرارات و وحدة التكييف. الهدف من نظامنا هو الحصول على تكييف بين زبائن متعددي الوسائط في البعث و الاستقبال. لقد قمنا بتقديم نموذج قاعدة السيناريوهات التوضيحية (التي تمثل حالات دراستنا) و قاعدة الهواتف النقالة و كذلك نموذج المحتويات المتعددة الوسائط باستعمال UML. و قد تم التنفيذ باستعمال JBUILDER .

الكلمات المفتاح : موبايل، تكييف متعددة الوسائط، زبون وسيط خادم ، سيناريو تكييف ، قاعدة موبايلات.

Abstract

The multimedia information in recent years experienced tremendous growth and binding in all socio-economic sectors. Daily life is also conquered by multimedia information; mobility and heterogeneity of mobile justifying investigations in the field of multimedia adaptation. The problem you can send any multimedia content from a client with a multimedia mobile phone to another phone multimedia without worry of having a message echoed. To address this problem, it is easy to adapt the multimedia content before being sent to multimedia final score. In this memory, we present our architecture of an open system for multimedia adaptation. Our system is based on client intermediary server model where the Proxy is the heart of our architecture. The proxy is composed of four main components: communication module, database, making decision module and adaptation module. The purpose of this architecture is to provide accommodation between multimedia clients in emission and in reception. We modelled with UML database of illustrative scenarios that represent our study case. The base of mobile phones and multimedia database are also modelled. The implementation is done with JBUILDER.

Key words: Mobile, Multimedia Adaptation, Client Intermediary Server, Adaptation scenarios, Mobiles base.

Liste des Tableaux

Tab	Titre	Page
Tab. 1 -	Etude comparative entre les approches d'adaptation existantes.	18
Tab. 2 -	Etude comparative entre les architectures présentées.	38
Tab. 3 -	Spécification de quatre téléphones mobile multimédia.	45
Tab. 4 -	Les types de medias et les technologies d'adaptation.	48

Table des figures

FIG	Titre	N°
FIG. 1 -	Stratégie générale d'adaptation.	8
FIG. 2 -	Arbre SMIL (gauche) et structure SMIL (droite) correspondant à la spécification.	9
FIG. 3 -	Spécification d'un document SMIL.	10
FIG. 4 -	Présentation d'un circuit touristique (gauche) et dimension spatiale (droite).	11
FIG. 5 -	Graphe de relations directionnelles adapté avec la contrainte $\{MOO(1)\}$ (gauche) et vue globale de la présentation adaptée (droite).	12
FIG. 6 -	Les relations RCC-8.	12
FIG. 7 -	Graphe de relations topologiques adapté avec la contrainte $\{MOO(1)\}$ (gauche) et vue globale de la présentation ada FIG.	13
FIG. 8 -	Combinaison de l'adaptation de la représentation topologique et directionnelle.	14
FIG. 9 -	Graphe de relations HBGD adapté avec la contrainte $\{MOO(1)\}$ (gauche) et vue globale de la présentation adaptée (droite).	15
FIG. 10 -	Architecture de l'application [8].	19
FIG. 11 -	Environnement d'expérimentation.	20
FIG. 12 -	Architecture logicielle d'adaptation.	21
FIG. 13 -	Hierarchie d'encapsulation multimédia.	21
FIG. 14 -	Organisation générale de l'architecture NAC.	25
FIG. 15 -	L'architecture d'adaptation [30].	28
FIG. 16 -	Génération automatique d'adaptation.	30
FIG. 17 -	Architecture fonctionnelle de fourniture de documents multimédia composés.	31
FIG. 18 -	Graphe d'adaptation.	32
FIG. 19 -	Architecture DCAF.	34
FIG. 20 -	Schéma fonctionnel du Proxy local.	36
FIG. 21 -	L'architecture du système ouvert.	42
FIG. 22 -	Le schéma Fonctionnel du Proxy.	44
FIG. 23 -	Diagramme de classe du scénario 1.	52
FIG. 24 -	Diagramme de class du scénario 2.	53
FIG. 25 -	Diagramme de classe de la base de données multimédia.	54
FIG. 26 -	Diagramme de class de la base de données des téléphones mobile.	55
FIG. 27 -	Base des scénarios d'adaptation.	57
FIG. 28 -	Base des téléphones mobile multimédia.	58
FIG. 29 -	Base des données multimédia.	58
FIG. 30 -	Plan d'adaptation existant dans l'historique de la base du proxy.	59
FIG. 31 -	L'adaptation multimédia n'est nécessaire selon la base des téléphones mobiles multimédia.	60
FIG. 32 -	L'adaptation multimédia est nécessaire (selon le la base des scénarios).	62
FIG. 33 -	Demande de génération du plan d'adaptation adéquat après avoir tester la base des téléphones mobiles.	63

Table des figures

Introduction Générale

1. Contexte du travail

Dans le contexte d'adaptation de contenus dû à la très grande hétérogénéité des moyens de communication, des connexions et des préférences des utilisateurs, beaucoup de travaux ont été effectués pour fournir des outils d'adaptation des contenus multimédia.

Les techniques actuellement disponibles reposent principalement sur la transformation textuelle, le transcodage d'image et le traitement vidéo et audio.

Une des questions majeures en terme d'architecture d'adaptation est de savoir où se réalise la prise de décision et surtout où s'exécutent les opérations d'adaptation. Trois approches élémentaires ont été proposées dans la littérature pour la localisation des processus d'adaptation sur le parcours de transmission de données, entre la source de données (serveur de contenu) et la destination des données (client final).

L'adaptation des contenus multimédias constitue un des mécanismes principaux pour la mise en œuvre des systèmes pervasifs. Les environnements ubiquitaires peuvent inclure des terminaux hétérogènes qui n'ont pas les mêmes caractéristiques en termes de taille mémoire, taille d'affichage, ou les formats supportés. Ceci implique que l'échange de contenu multimédia dans de tel environnement, nécessite un changement parce que le contenu lui-même est hétérogène en termes d'encodage.

Considérons le cas des téléphones mobiles, dont les capacités matérielles et logicielles sont hétérogènes et quelques fois limitées d'une marque à une autre (parlant des formats supportés, de la taille d'affichage de l'écran, etc.). L'avantage de l'adaptation est de permettre aux personnes possédant des téléphones mobiles multimédia (clients multimédia) d'envoyer et de recevoir n'importe quel contenu sans se soucier de l'hétérogénéité.

2. Problématique et Objectifs du travail

L'adaptation présente une solution évidente pour les systèmes contenant des terminaux hétérogènes.

Le problème dans notre cas est que les terminaux sont des téléphones mobiles multimédia qui se différencient en terme de capacité de supporter (recevoir) tout types de données en n'importe quel format et n'importe quelle taille, etc. En plus, l'émetteur (client multimédia émetteur) ne reçoit aucune information lui indiquant pourquoi telle ou telle donnée n'a pas été transmise, il ne reçoit qu'un message d'écho signifiant que l'information n'a pas pu être reçue.

L'adaptation du contenu hétérogène dans un environnement hétérogène en prenant en compte les capacités de l'émetteur et du récepteur devant être délivré au client multimédia cible semble être indispensable pour pallier au problème des messages d'écho.

Généralement, les architectures d'adaptation existantes traitent les données multimédia envoyées par la machine serveur au terminal du client mais dans notre architecture, le traitement d'adaptation est appliqué aux données multimédia envoyées du client multimédia à un autre client multimédia.

Le but essentiel de notre travail est de proposer une architecture ouverte pour l'adaptation multimédia au profil des téléphones mobiles multimédia. Nous allons spécifier tous les composants nécessaires à la mise en place d'une architecture d'adaptation à savoir: son principe de fonctionnement, l'approche d'adaptation considérée, ses différents composants et le rôle de chaque composant.

Les objectifs de notre investigation sont :

- Etude comparative des approches d'adaptation.
- Etude comparative des architectures d'adaptation.
- Proposition d'une architecture d'adaptation.
- Modélisation par UML.
- Réalisation par JBuilder.

3. Organisation du manuscrit

Ce mémoire est organisé en deux parties. La première partie (le chapitre 1 et le chapitre 2) représente un état de l'art et la seconde partie décrit notre contribution théorique ainsi que les implantations qui en découlent. Nous présentons ci-dessous le contenu des chapitres.

Chapitre 1

Le premier chapitre présente l'information multimédia : structure et mécanismes d'adaptation. A travers ce chapitre, nous allons présenter, dans un premier lieu, la structure des documents multimédia. Dans un second lieu, nous présentons le processus d'adaptation de contenu : l'adaptation logique, l'adaptation temporelle et l'adaptation spatiale.

Chapitre 2

Dans ce chapitre, nous présentons les différentes approches et architectures d'adaptation multimédia existantes. Dans la suite, nous dressons dans ce chapitre une étude comparative afin de cerner les approches et les architectures d'adaptation existantes.

Chapitre 3

L'objectif de ce chapitre est de présenter les composants d'architecture proposée pour un système ouvert d'adaptation multimédia. Ce chapitre dans sa première partie, explique les différents composants de notre architecture et leurs rôles ainsi que le fonctionnement général de notre architecture. La deuxième partie de ce chapitre est consacrée à l'étude de cas (à partir de la spécification de quelques téléphones mobiles multimédia). Ainsi, nous présentons cinq scénarios illustratifs.

Chapitre 4

Ce chapitre va être consacré à la modélisation des scénarios d'adaptation qu'en a exposés dans le chapitre 3. Les scénarios choisis sont des instances et des solutions de problèmes réels lors de la transmission des données entre les mobiles multimédia. En a utilisé pour la modélisation des scénarios d'adaptation le langage de modélisation unifié (UML : Unified Modeling Language). La modélisation de la base des téléphones mobiles et des données multimédia a été aussi nécessaire.

Chapitre 5

Dans ce chapitre, nous présentons l'implémentation de la base de données du proxy de notre système. Les études de cas qui sont les scénarios déjà exposés et les téléphones mobiles multimédia spécifiés dans le chapitre 3, servent comme une base pour construire un historique des scénarios. Nous simulons quelques scénarios d'adaptation multimédia. Nous décrivons l'implémentation de la base de données réalisée à l'aide du langage **JBuilder 7**.

Nous terminons ce mémoire par une conclusion générale sur le travail effectué ainsi que nos perspectives.

À la fin de ce document, nous retrouvons une bibliographie contenant les principales références que nous avons utilisées.

CHAPITRE 1

« Données Multimédia : Structure & Mécanismes d'Adaptation »

1.1. Introduction

Dans le contexte d'adaptation de contenus dû à la très grande hétérogénéité des moyens de communication, des connexions et des préférences des utilisateurs ; beaucoup de travaux ont été effectués pour fournir des outils d'adaptation des contenus multimédia.

Les techniques actuellement disponibles reposent principalement sur la transformation textuelle, le transcodage d'image et le traitement vidéo et audio. Assurer la tâche de l'adaptation du contenu dans une architecture globale nécessite des paramètres de l'environnement et pour cela, il faut avoir des mécanismes d'adaptation satisfaisants : l'adaptation logique, l'adaptation temporelle et l'adaptation spatiale (comme que ces présenté dans la section deux de ce chapitre).

Dans ce qui suit, nous allons présenter en premier lieu, la structure (les dimensions) des documents multimédia. En second lieu, nous présentons le processus d'adaptation de contenu de chacune accompagné d'exemples d'adaptation temporelle et spatiale de document SMIL.

1.2. Structure des documents multimédia

Un document multimédia est un document numérique composé des objets de différentes natures : texte, son, image, vidéo, animation, etc. Ces objets et ces compositions sont appelés objets multimédia.

Une bonne modélisation des documents multimédia est nécessaire pour mettre en place des mécanismes d'adaptation efficaces. Les fonctionnalités d'un contenu multimédia peuvent être décrites à l'aide des quatre dimensions suivantes [6]:

1. La dimension logique.
2. La dimension spatiale.
3. La dimension temporelle.
4. La dimension hypermédia.

1.2.1. La dimension logique

La spécification de l'organisation logique correspond à la décomposition d'un système en composants. L'auteur détermine le sujet à présenter et commence à définir les différents composants et la façon selon laquelle ils seront synchronisés. Par exemple, pour créer un contenu multimédia présentant une équipe de recherche, on peut décomposer le sujet en deux composants : une partie générique qui introduit l'équipe et une partie thèmes de recherches qui présente les activités de recherche au sein de l'équipe [3].

1.2.2. La dimension spatiale

La dimension spatiale concerne l'affectation dans le temps des composants multimédia d'un document aux ressources physiques [3].

1.2.3. La dimension temporelle

La spécification de l'organisation temporelle définit l'ordonnancement temporel de la présentation des différents composants afin de construire le scénario souhaité. Quatre formes principales de synchronisation temporelle sont distinguées : la synchronisation intra-objet, la synchronisation inter-objet, la synchronisation avec l'environnement et la synchronisation de groupe [3].

1.2.4. La dimension hypermédia

La dimension hypermédia définit les relations entre les documents et les fragments du document [6].

1.3. Processus d'adaptation de contenu

Assurer la tâche de l'adaptation du contenu dans une architecture globale nécessite la considération de plusieurs paramètres de l'environnement. Dans cet environnement une requête de demande de contenu peut être reçue et doit être satisfaite par le fournisseur du

contenu. Satisfaire une requête revient à satisfaire toutes les contraintes qu'elle peut comporter : le type du contenu, la nature de la connexion, les contraintes du réseau utilisé, etc. D'un autre côté, des mécanismes de base doivent être disponibles pour pouvoir couvrir les différents types d'adaptation [3].

1.3.1. Adaptation logique

Le contenu original peut être vu comme un ensemble d'objets multimédia (texte, images, vidéo, audio, etc.) qui composent le scénario multimédia final. Afin d'adapter la présentation de ces objets sous un contexte particulier, l'organisation logique du contenu peut être réorganisée vers une nouvelle forme qui considère mieux les contraintes du nouveau contexte.

1.3.2. Adaptation temporelle

Le modèle temporel se base sur une unique dimension qui est le temps [10]. Plusieurs langages assurent, de différentes manières, une spécification des règles de présentations temporelles. SMIL [11] exprime le positionnement des objets multimédia en utilisant les opérateurs parallèles et séquentiels. Magic [12] et Madeus [13] utilisent des restrictions de l'algèbre d'Allen.

1.3.2.1. Adapter temporellement un document SMIL

La section suivante présente une application de l'adaptation temporelle aux documents multimédia spécifiés en langage SMIL.

Dans un document multimédia temporel la présentation des objets est organisée dans le temps. Un exemple proposé consiste en une équipe de recherche contenant des régions d'objets graphiques. La première région affiche une image (Titre) et deux vidéos (Auteur et Demo). De plus, un discours est joué durant une partie de la présentation. Chacun de ces objets est représenté par un segment. Une telle description est précise et quantitative car elle définit exactement les instants de début et de fin de chaque objet multimédia. Les spécifications non précises peuvent être exprimées par de relations qualitatives.

Un document SMIL peut être vu comme un arbre. Deux catégories d'éléments sont identifiées : les objets multimédia et les conteneurs de temps (un conteneur séquentiel (*seq*) et un conteneur parallèle (*par*)).

Après la spécification, le système de présentation multimédia (Player) calcule un plan (scénario).

A. Adaptation de document SMIL

Différents contextes de présentation multimédia introduisent différentes contraintes sur la présentation. Donc, les documents multimédia doivent être adaptés. Cette adaptation est réalisée par un programme de transformation du document.

Pour avoir un sous-ensemble d'exécutions compatibles, l'ensemble des interprétations possibles est représenté à l'aide d'un graphe de contraintes résolues. L'algèbre de relations introduite par Allen [4] permet de représenter temporellement les documents. Adapter c'est trouver une distance minimale d entre le graphe initial et le graphe adapté [4].

- Principe d'adaptation de document SMIL

La manière la plus naturelle d'adapter un document SMIL est de rendre l'équivalent en algèbre d'Allen (α), d'adapter et de traduire en SMIL. La figure 1 représente la stratégie générale d'adaptation.

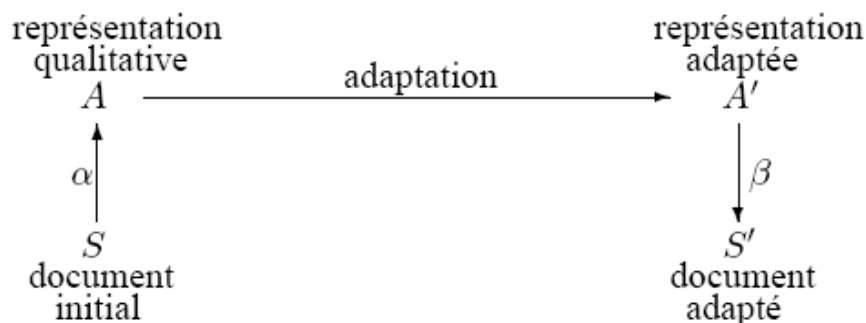


Figure 1: Stratégie générale d'adaptation [4].

Il est nécessaire d'injecter les informations de fonction minimale par une fonction γ . Dans le cas d'une fonction qui ne nécessite pas l'adaptation, la traduction ne change rien. C'est la propriété de *neutralité*.

- Fonctions de traduction

Une structure SMIL doit associer à chaque nœud un index temporel. Ces indexes sont des triplets $[b; d; e]$ tel que b , d et e correspondent respectivement aux valeurs des attributs temporels *begin*, *dur* et *end*. Si l'attribut n'a pas de valeur, cela est noté par le symbole "-".

- L'exemple de la figure ci dessous représente Arbre SMIL (gauche) et structure SMIL (droite) correspondant à la spécification.

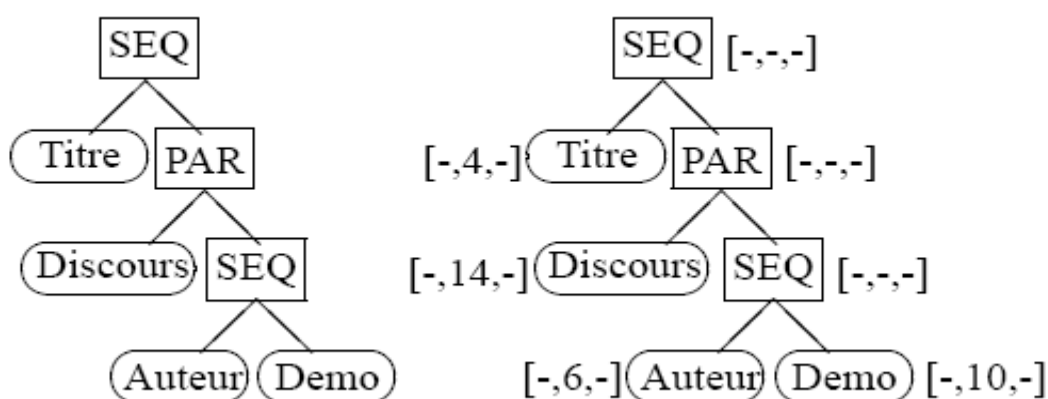


Figure 2 : Arbre SMIL (gauche) et structure SMIL (droite) correspondant à la spécification [4].

- Maintenir la cohérence entre relations et structure

À partir du document SMIL S , $\alpha(S) = \langle A, T \rangle$, il est possible d'appliquer l'adaptation sur $A \langle \text{Adapt}(A), T \rangle$. Cependant, il est nécessaire d'avoir une structure SMIL cohérente avec le graphe de relation pour que n'en aura pas des documents SMIL illégaux. Cette cohérence peut venir des conteneurs de temps non adaptés aux relations Allen. L'incohérence peut être détectée à l'intérieur d'un graphe SMIL étendu. Des expérimentations sur les travaux précédents ont été réalisées mais qui ce limite sur la dimension temporelle et une minimalité entière de la distance n'est obtenue. Des travaux antérieurs se concentrent sur la génération de documents SMIL adapté que d'adapter des documents SMIL existants.

Pour conclure, [4] applique l'adaptation sémantique d'un document SMIL selon sa dimension temporelle. Cette adaptation doit être essentiellement minimale.

Enfin, une implémentation a été réalisée.

1.3.3. Adaptation spatiale

Ce type d'adaptation vise à optimiser les ressources spatiales du système de présentation cible [14], [15] en décrivant les différentes relations qui peuvent exister entre les objets multimédia. Ces relations peuvent représenter les types de placement possibles.

1.3.3.1. Adaptation spatiale effective de documents SMIL

L'objectif de [5] est de trouver une représentation spatiale qualitative permettant de calculer un ensemble de solutions d'adaptation proche du document initial en mesurant les deux niveaux de la qualité d'une adaptation (expressivité des solutions d'adaptation et rapidité de calcul).

A. Spécification spatiale

Dans un document multimédia, la présentation des objets multimédia visibles est organisée dans l'espace. L'exemple proposé dans [5] consiste en une présentation d'un circuit touristique dont chacun des objets est représenté par un rectangle défini à l'aide d'un point d'origine (dans l'exemple, c'est le point gauche d'origine), d'une hauteur ainsi que d'une largeur. Par exemple l'objet Logo a une largeur de 240 pixels, une hauteur de 80 pixels et un point d'origine (0,0). Dans [5], les documents multimédia sont exprimés avec le langage SMIL. La figure 3, représente la spécification d'un document SMIL.

```

<root-layout height="450" width="700"/>
<region id="Logo" height="80" width="240"
top="0" left="0"/>
<region id="Texte" height="360" width="450"
top="0" left="240"/>
<region id="Photo" height="235" width="350"
top="80" left="0"/>
<region id="Carte" height="235" width="350"
top="205" left="200"/>

```

Figure 3 : Spécification d'un document SMIL [5].

Un exemple de la présentation d'un circuit touristique (gauche) et dimension spatiale (droite) est présenté dans la figure 4.

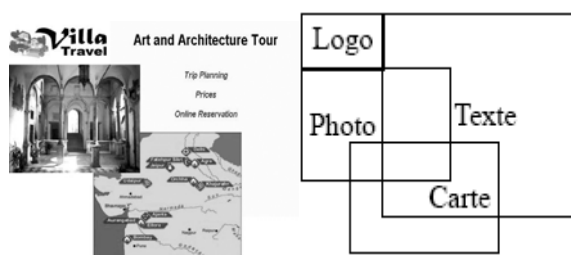


Figure 4 : Présentation d'un circuit touristique (gauche) et dimension spatiale (droite) [5].

Il existe plusieurs types d'adaptation. [5] a été consacré à l'adaptation globale c'est à dire liée à l'organisation du contenu de la présentation. L'adaptation peut être implicite et cela est permis par SMIL en offrant une balise *switch*. L'adaptation peut être aussi explicite en utilisant la sémantique du document (cette approche permet une adaptation efficace en fournissant plus de flexibilité).

B. Adaptation sémantique

Pour réaliser l'adaptation sémantique, l'ensemble des interprétations possibles sont représentés à l'aide d'un graphe de relation. Le graphe de relation $g_s = \langle N, E, \lambda \rangle$ de la

spécification $s = \langle O, C \rangle$ associe à chaque arc un ensemble r de relation tel que si $x r y \in C$, $\lambda(\langle x, y \rangle) \subseteq r$. Suite aux contraintes d'adaptation certaines relations du graphe seront interdites, il sera nécessaire de les adapter par des relations voisines.

1.3.3.2. Représentation spatiale directionnelle

La représentation spatiale directionnelle permet de définir des orientations dans l'espace entre les objets. Il est possible d'étendre les relations spatiales en ajoutant des relations de chevauchement.

A. Adaptation sémantique de la représentation directionnelle

Une relation spatiale directionnelle comprenant deux relations d'Allen, un pour l'axe horizontal et un autre pour l'axe vertical.

La distance entre graphes de relations Allen2D est définie comme suit : $d(\lambda, \lambda') = d_x(\lambda, \lambda') + d_y(\lambda, \lambda')$

La distance d par rapport au graphe de relations initial est de 5 ($dx = 4$ et $dy = 1$). Un exemple de graphe de relations directionnelles adapté avec la contrainte $\{MOO(1)\}$ (gauche) et vue globale de la présentation adaptée (droite).

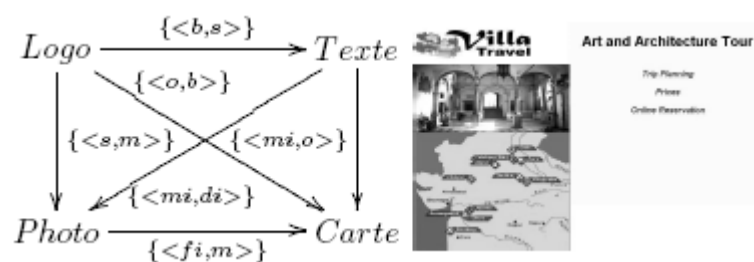


Figure 5 : Graphe de relations directionnelles adapté avec la contrainte $\{MOO(1)\}$ (gauche) et vue globale de la présentation adaptée (droite) [5].

1.3.3.3. Adaptation de la représentation spatiale topologique

A. Représentation spatiale topologique

Ce formalisme définit de nombreuses propriétés et relations spatiales. En particulier une de ces spécifications est l'ensemble des huit relations (RCC-8). La figure 6, représente les relations RCC-8 précédemment citées.

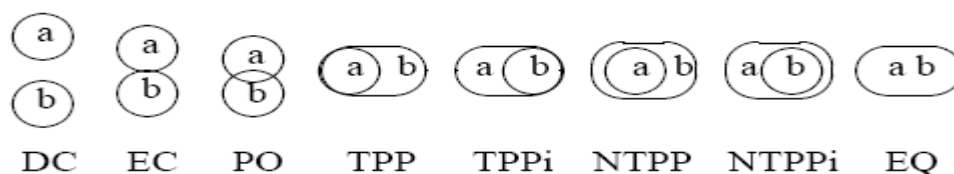


Figure 6 : Les relations RCC-8 [5].

Ces huit relations sont :

- DC (*is disconnected from*) : les deux régions sont disjointes.
- EC (*is externally connected with*) : les deux régions ne sont pas incluses l'une dans l'autre mais sont en contact.
- PO (*partially overlaps*) : les deux régions se chevauchent.
- TPP (*is a tangential proper part of*) : les deux régions sont incluses l'une dans l'autre et sont en contact.
- TPPi (inverse de TPP)
- NTPP (*is a nontangential proper part of*) : les deux régions sont incluses l'une dans l'autre mais ne sont pas en contact.
- NTPPi (inverse de NTPP)
- EQ (*equals*) : les deux régions sont identiques [5].

B. Adaptation sémantique de la représentation topologique

Supposant que l'on ne peut chevaucher des régions d'affichage et que le graphe de relation initiale ne respecte pas les contraintes d'adaptation. Pour cela, des relations de proximité entre relations spatiales topologique sont introduites. Un exemple de relation adapté avec

la contrainte $\{MOO(1)\}$ (gauche) et la vue globale de la présentation adaptée (droite) est représenté dans la figure ci-dessous.

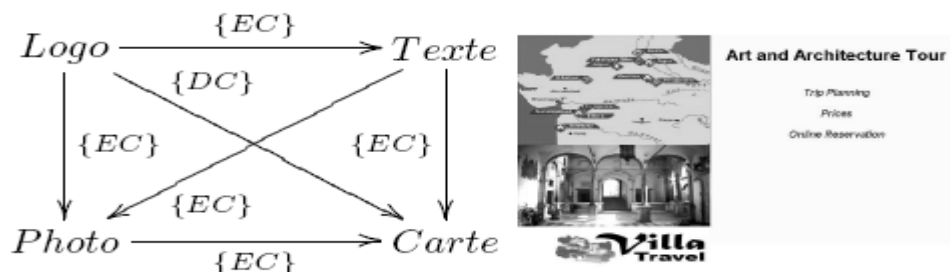


Figure 7 : Graphe de relations topologiques adapté avec la contrainte $\{MOO(1)\}$ (gauche) et vue globale de la présentation adaptée (droite) [5].

L'exécution présentée dans la figure 7, respecte les contraintes imposées par le graphe de relation adapté mais, propose un document adapté assez éloigné du document initial. Pour améliorer le résultat de l'adaptation, dans [5] il ya une proposition de raisonner sur la combinaison de la représentation directionnelle et topologique.

C. Association avec la représentation directionnelle

Il est important de noter que l'ensemble des relations directionnelles couvrent l'ensemble des relations topologiques.

Une stratégie d'adaptation dans [5] a été envisagée :

1. Appliquer la généralisation topologique sur le document SMIL initial

$$S : \alpha_{RCC}(S) = R_{RCC}.$$

2. Adapter la représentation topologique R_{RCC} : $adapt_{RCC}(R_{RCC}) = R'_{RCC}$.

3. Transformer la représentation topologique R'_{RCC} en représentation directionnelle RA_{2D} : $T(R'_{RCC}) = RA_{2D}$.

4. Adapter la représentation directionnelle RA_{2D} : $adapt_{A2D}(RA_{2D}) = R'_{A2D}$.

5. Appliquer l'instanciation sur la représentation directionnelle adaptée

$$R'_{A2D} : \beta_{A2D}(R'_{A2D}) = S'$$

La stratégie d'adaptation par combinaison de l'adaptation de la représentation topologique et directionnelle est représentée dans la figure 8.

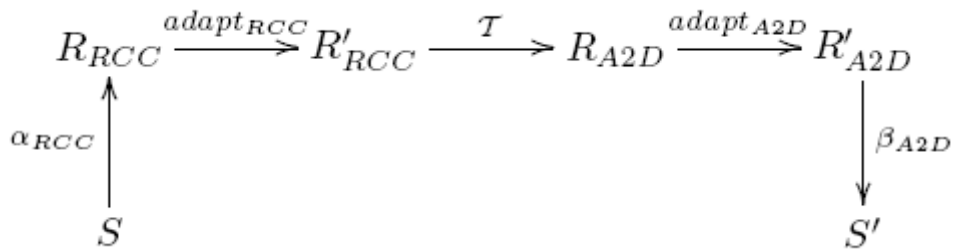


Figure 8 : Combinaison de l'adaptation de la représentation topologique et directionnelle [5].

1.3.3.4. Adaptation de la représentation HBGD (Haut, Bas, Gauche, Droite)

Définition de la présentation spatial HBGD: Soient A le point supérieur gauche et B le point inférieur droit d'une région α (respectivement C et D pour une région β). Considérant des régions rectangulaires où il est admis que. $A_x \leq B_x, A_y \leq B_y, C_x \leq D_x$ et $C_y \leq D_y$ 36 relations spatiales mutuellement exclusives sont identifiées.

La figure 9 présente un graphe de relations adapté. Avec ce type de représentation, on calcule un ensemble de solutions minimal d'adaptation.

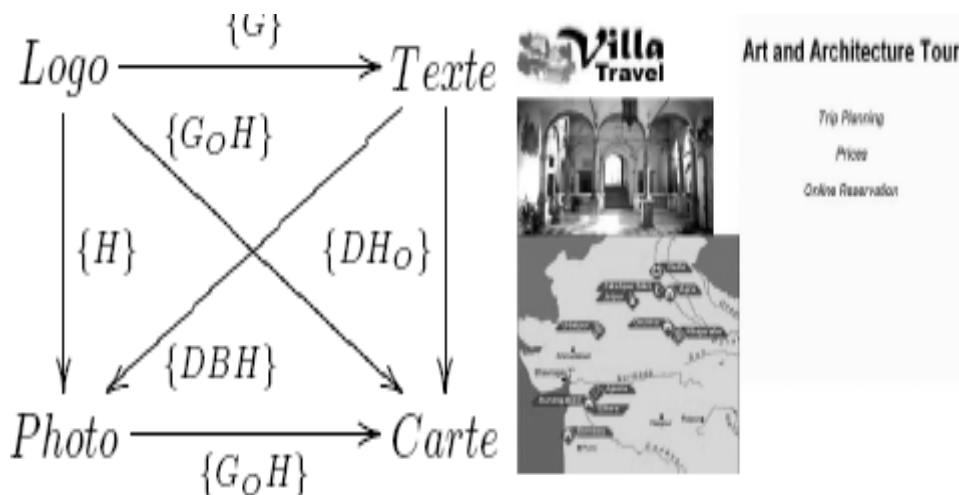


Figure 9 : Graphe de relations HBGD adapté avec la contrainte $\{MOO(1)\}$ (gauche) et vue globale de la présentation adaptée (droite) [5].

1.4. Conclusion

Dans ce chapitre, nous avons commencé par la présentation de la structure des documents multimédia : dimension logique, dimension spatiale, dimension temporelle et dimension hypermédia. Puis, nous avons présenté le processus d'adaptation de contenu. Nous avons expliqué les différents types d'adaptation : l'adaptation logique, l'adaptation temporelle avec des exemples de documents multimédia spécifiés en langage SMIL et l'adaptation spatiale (aussi avec des exemples de documents multimédia spécifiés en SMIL).

Pour l'exemple d'adaptation temporelle de document SMIL, nous avons présenté : le principe d'adaptation du document, les fonctions de traduction, la structure SMIL et la maintenance de la cohérence entre les relations et la structure.

Dans les exemples de l'adaptation spatiale, nous avons présenté : l'adaptation spatiale effective de document SMIL, la représentation spatiale directionnelle et l'adaptation de la représentation topologique.

Des approches et des architectures qui utilisent les différents types d'adaptation et qui considèrent les entités de l'environnement, font l'objet du chapitre suivant.

CHAPITRE 2

« *Approches & Architectures d'Adaptation Multimédia* »

2.1. Introduction

L'échange de données dans les environnements ubiquitaires requière une adaptation. Une des questions majeures en terme d'architecture d'adaptation est de savoir où se réalise la prise de décision et surtout où s'exécutent les opérations d'adaptation. Trois approches ont été proposées dans la littérature pour la localisation des processus d'adaptation sur le parcours de transmission de données, entre la source de données (serveur de contenu) et la destination des données (client final). Ces approches sont : *l'approche centrée serveur*, *l'approche centrée client* et *l'approche centrée intermédiaire*. En s'appuyant sur les approches (modèles) d'adaptation, beaucoup d'architectures ont été proposées.

Dans ce qui suit, nous allons présenter les approches d'adaptation existantes suivies d'une étude comparative entre ces approches. Enfin, nous présentons quelques architectures d'adaptation existantes (leurs composants en insistant sur les méthodes et les stratégies d'adaptation de chacune de ces architectures), la présentation des architectures est suivie également d'une étude comparative.

2.2. Les approches d'adaptation existantes

2.2.1. L'approche centrée serveur – [31], [32], [33], [34], [35]

Le serveur se charge de découvrir les capacités du client et la bande passante disponible. Il décide de la meilleure stratégie d'adaptation. Cette approche permet à l'auteur (créateur) du contenu de formuler des conseils ou des contraintes sur l'adaptation. De plus, les solutions centrées serveur permettent la mise en œuvre de mécanisme d'adaptation de contenu dynamique (*à la volée*) ou statique (*hors ligne*). Ces approches présentent deux inconvénients majeurs : les ressources de transformation induisent une charge de calcul et une consommation de ressources importantes sur les serveurs que celui-ci n'a souvent pas la capacité de fournir.

2.2.2. L'approche centrée client – [36], [37], [38], [39], [40]

Le processus d'adaptation est réalisé au niveau du terminal client en fonction des capacités de celui-ci. Le serveur demeure donc inchangé et transmet ses contenus dans leur format d'origine. Le terminal du client utilisateur à la réception du contenu transforme celui-ci de manière à pouvoir l'afficher correctement. Avec ces caractéristiques ou transformation ad hoc effectuée par le terminal lui-même. La sélection des contenus compatibles peut être réalisée dans le cadre de négociation avec le serveur de contenus : en réponse à une requête client, le serveur transmet la liste des variantes disponibles pour le contenu disponible répondant à la requête. Le terminal client sélectionne les (variantes des) contenus. A titre d'exemple, certains terminaux Windows CE (comme le SmartPhone) modifient la résolution chromatique des images (passage d'une profondeur de couleur de 24 bits au niveau de gris sur 4 bits).

L'adaptation centrée client requière des modifications minimales au niveau du terminal de connexion et peut être extrêmement efficace dans la personnalisation du contenu et l'adaptation aux caractéristiques du terminal. Toutefois, cette solution est très mal adaptée aux situations où les contraintes de transmission réseaux sont prégnantes, ce qui est typiquement le cas de nombreuses applications pervasives intégrant des dispositifs nomades. En outre, la complexité élevée des mécanismes d'adaptation interdit l'utilisation générique de cette solution dans la mesure où une proportion considérable des dispositifs pervasif reste limitée en termes de puissance de calcul, de mémoire, d'énergie et de stockage. Ainsi, les approches centrées client ne sont en pratique pertinentes que vis-à-vis de problématiques simples d'affichage lorsque les contraintes réseaux sont traitées ailleurs.

Le dispositif du client est responsable d'adapter le contenu selon les caractéristiques du dispositif, les préférences utilisateur et le contexte opérationnel. En raison de la capacité limitée des dispositifs client, cette approche est difficile à mettre en place.

2.2.3. Approche centrée Proxy -[41], [42], [43], [44], [45], [46], [47]

Place un proxy entre le client et le serveur de contenus. Ce proxy est alors responsable de l'adaptation des contenus. Un exemple de cette approche: le proxy intercepte la réponse du serveur, décide et effectue l'adaptation en fonction des préférences de l'utilisateur, de

capacité des terminaux et de l'état du réseau, puis envoie le contenu adapté à l'utilisateur. Même si cette approche résout à un certain degré les problèmes rencontrés avec les approches orientées client et serveur, elle présente des limitations en termes d'extensibilité en raison du coût élevé des opérations d'adaptation.

2.2.4. Etude comparative entre les approches d'adaptation existantes

Le tableau ci-dessous (tableau 1) présente une étude comparative entre les trois approches d'adaptation existantes [52], [53].

Approche	Prise de décision et adaptation	Avantages	Inconvénients
Centrée serveur	Au niveau serveur.	+L'auteur formule des conseils ou des contraintes sur l'adaptation. +Mise en œuvre de mécanismes d'adaptation dynamique et statique.	-Le fournisseur intègre des mécanismes d'adaptation. -Charge de calcul sur le serveur.
Centrée client	Au niveau client par deux méthodes : sélection des contenus ou transformation ad hoc.	+Pour les problématiques simples.	-Mal adaptée aux situations où les contraintes réseau sont prégnantes. -N'est pratique.
Centrée proxy	Sur un nœud intermédiaire : proxy	+Mise en cache des résultats. +La charge de calcul est sur le proxy. +Dispose d'une vue globale sur l'environnement.	-Passe mal à l'échelle. -Problème de sécurité -Outils d'adaptation amenés à évoluer.

Tableau 1 : Etude comparative entre les approches d'adaptation existantes [52], [53].

2.3. Les architectures d'adaptation existantes

2.3.1. Adaptation d'une application multimédia par un code mobile

Dans [8], une application multimédia répartie a été présentée en utilisant un agent mobile programmé en java. Un agent est envoyé à la machine distante qui fournit les données multimédia (le serveur ou le proxy) afin d'adapter les données en fonction des contraintes du client et/ou du réseau.

2.3.1.1. Architecture de l'application

L'architecture de l'approche considérée dans [8] est basée sur le modèle client/serveur. Les sites proxy permettent de distinguer les machines sur lesquelles sont gérées des logiciels standards (le client et le serveur). La figure 10 représente l'architecture de l'application [8].

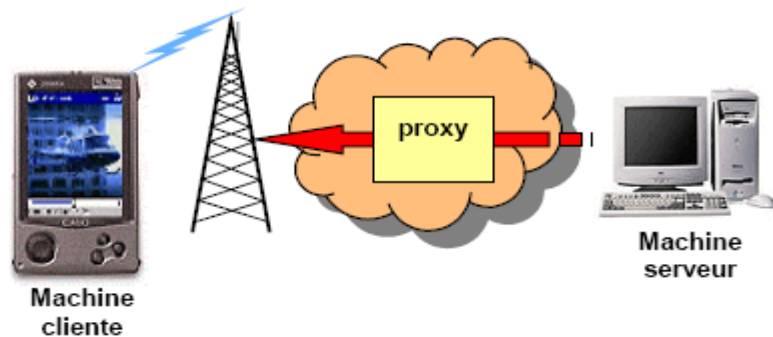


Figure 10 : Architecture de l'application [8].

2.3.1.2. Les adaptations à base d'agent mobile

Dans le système [8], une adaptation est instanciée sous la forme d'un agent mobile déployé sur le site proxy afin d'effectuer un traitement sur la vidéo envoyée au client. Cette modification peut diminuer la charge de calcul sur le client qui présente la vidéo et a un impact aussi sur la consommation de la bande passante sur le réseau.

Deux exemples illustrent ces deux motivations :

Un PDA disposant d'un écran petit doit redimensionner une vidéo encodée pour être restituée sur un écran de station de travail alors qu'il ne dispose pas de la vidéo. Une solution simple illustre l'adaptation à la variation de bande passante du réseau. Dans ce cas, l'adaptation consiste à envoyer sur le proxy un agent qui adapte la vidéo en fonction de la bande passante disponible, ce qui réduit significativement le volume d'information émis.

Certaines adaptations peuvent cumuler ces deux avantages. Le fait de modifier l'encodage de la vidéo par un agent permet d'économiser des cycles de calcul sur le terminal client et de réduire la bande passante consommée.

2.3.1.3. Environnement d'expérimentation

La figure 11 présente l'architecture de l'application. Un client joue une présentation vidéo chargée depuis un serveur Web. Le client se connecte à un site proxy en lui passant en paramètre l'URL du document multimédia à présenter. Le proxy accède au document multimédia avec le protocole HTTP. La communication entre le client et le proxy utilise le protocole RTP (Real Time Protocol) [28].

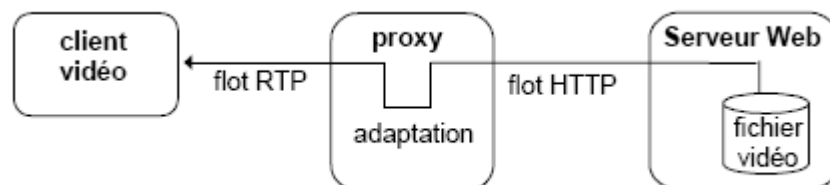


Figure 11 : Environnement d'expérimentation [8].

La figure 12 représente l'architecture logicielle d'adaptation dont laquelle, il ya :

Un serveur web Apache et un client de visualisation multimédia (JMStudio de l'environnement JMF).

Un agent est envoyé vers le site proxy. C'est l'agent d'adaptation. Ces adaptations modifient le contenu du flot vidéo. L'installation dynamique de l'agent d'adaptation sur le site proxy s'appelle : configuration du proxy.

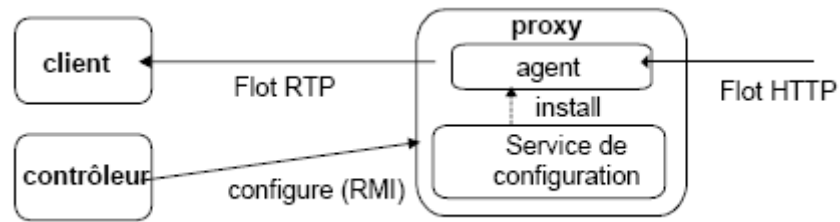


Figure 12 : Architecture logicielle d'adaptation [8].

Le code du proxy (le service de configuration) et le code des agents d'adaptation sont programmés en Java. Ils reposent sur Java Media Framework (JMF), un environnement de gestion de données multimédia.

2.3.1.4. Les adaptations du flot multimédia

La figure.13 représente l'hierarchie d'encapsulation multimédia. Un **flot multimédia** est composé de l'assemblage d'un ensemble de **flots média**. La transmission dans [8], se focalise sur l'échange de flot vidéo. Ce dernier est une séquence d'images encodée. Une image non encodée (**image brute**) est représentée dans un format linéaire.

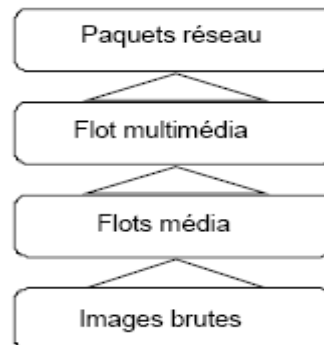


Figure 13 : Hiérarchie d'encapsulation multimédia [8].

Dans l'architecture de la figure 10, une vidéo est transmise depuis un site Web vers une machine cliente qui présente la vidéo. Le flot vidéo passe à travers un site intermédiaire appelé *proxy*. Le proxy peut être configuré à distance avec des agents d'adaptation qui appliquent une transformation sur le flot vidéo transmis à la machine cliente (l'agent modifie l'encodage de la vidéo). La mise en œuvre de l'application [8] repose sur l'infrastructure Java Media Framework.

2.3.2. Une architecture générique pour la fourniture de services multimédia adaptables - illustration par un scénario

De façon informelle, un service d'adaptation est défini comme un service qui évolue pour prendre en compte les changements intervenant de son environnement (par exemple baisse du niveau de batterie du terminal, requête de l'utilisateur ou changement de terminal).

Plusieurs approches ont été proposées en vue de l'adaptation des services offerts à l'utilisateur. L'architecture [2] vise à regrouper deux techniques d'adaptation (celles qui se concentrent sur l'adaptation de la logique de service [17], [18], [19] et les architectures de description de format de contenu [20], [21], [22]). L'objet est de fournir une architecture d'adaptation de la logique de service en utilisant des composants et à l'adaptation des flux multimédia en s'appuyant sur les travaux MPEG-21 et MPEG-7 [23], [24].

2.3.2.1. Une architecture pour la fourniture de services adaptable

Trois acteurs y sont représentés : l'utilisateur, le fournisseur de service et le gestionnaire de services.

L'utilisateur envoie une requête au fournisseur de service par le biais du gestionnaire de service. Ce dernier se charge aussi de trouver la version du service correspondant à la requête, aux références, au profil de l'utilisateur ainsi qu'au contexte d'exécution du service.

La requête donne lieu à l'envoi d'une liste des offres. Lorsque l'utilisateur choisit son service dans cette liste, le gestionnaire déclenche une 'boucle de configuration d'adaptation'. Cette boucle commence par une collecte des paramètres de l'environnement. Le gestionnaire d'adaptation détermine la configuration du service la plus adaptée pour cet utilisateur. Cette même boucle d'adaptation se répète tant que le service s'exécute afin d'adapter ce dernier aux changements survenant dans l'environnement d'exécution.

2.3.2.2. Illustration de l'architecture par le scénario

Phase 1 :

Pierre souhaite voir un match de football sur son ordinateur de bureau. Par le biais de son interface d'accès aux services, Pierre demande la liste des vidéos des matchs disponibles. Pierre sélectionne dans cette liste le match qu'il souhaite voir et commence à le regarder.

Pierre demande le match souhaité, le gestionnaire de service contacte le prestataire de service et retourne les profils locaux. Le gestionnaire de service fait appel à son module de supervision qui collecte le profil de Pierre et ces préférences ainsi que les paramètres de son environnement (Internet, réseau). Ce module envoie un rapport au gestionnaire d'adaptation qui se base sur la politique d'adaptation et le profil service qui choisit la bonne configuration du flux multimédia du match de foot pour avoir le bon mode d'exécution de service.

Il y a donc un 'dialogue' entre l'utilisateur et son prestataire de service.

Phase 2 :

Pierre doit alors quitter son bureau. En se connectant depuis son PDA, une boucle d'adaptation est déclenchée par le module de supervision qui détecte dans l'environnement le changement d'équipement. Il récupère alors le profil du nouveau terminal depuis la base de profils. Le gestionnaire d'adaptation détermine les actions à entreprendre, le gestionnaire de service envoie une requête au fournisseur de service pour télécharger la version de service adaptée au PDA. Pierre regarde le match mais avec un flux adapté à son PDA. Le fournisseur de service notifie Pierre de la possibilité de ne regarder que quelque séquence du match. Si Pierre fait ce choix, une version alternative du service peut être obtenue dynamiquement par décomposition temporelle du flux multimédia correspondant au match et forme des segments de flux. Le gestionnaire de services est chargé de réaliser cette décomposition et permet la visualisation du flux multimédia. En fin, le visionnage de la vidéo en noir et blanc suppose une adaptation au niveau du prestataire. En effet, la vidéo étant stockée avec la couleur dans la base de données du fournisseur de service, des outils tels que les DIA de MPEG-21 sont utilisés pour choisir par exemple de ne pas transmettre une couche d'adaptation contenant l'information en couleur.

Phase 3 :

Pierre se déplace et renonce à regarder la vidéo pour n'avoir que le son en meilleure qualité.

Une nouvelle boucle d'adaptation est déclenchée. Le gestionnaire de flux télécharge un flux audio scalaire et réalise le traitement sur ce flux qui correspond à une qualité de base et un ou plusieurs ensembles de données d'amélioration. Le gestionnaire de service choisit facilement les données à prendre en compte. Ceci suppose aussi un changement de code. Ce changement est conduit par le gestionnaire de service et résulte d'une action d'adaptation déclenchée par le gestionnaire d'adaptation.

Le travail présenté dans [2] a été entrepris pour faire cohabiter au sein de la même architecture, différents moyens d'adaptation [2].

2.3.3. Architecture de Négociation et d'Adaptation de Services Multimédia dans des Environnements Hétérogènes

L'architecture d'adaptation et de négociation NAC (Negociation and Adaptation Core) [3], a pour objectif de contribuer à l'adaptation et à la négociation des contenus en considérant les limitations des terminaux et les contraintes de leurs environnements.

NAC inclut deux grandes parties : la description du contexte et l'adaptation pour satisfaire le contexte. L'approche adoptée dans l'architecture NAC essaye de réaliser l'objectif de l'utilisation automatique d'un contenu distant par une variété de clients qui peuvent être différents en terme de capacités de traitement, format supportés, protocoles de communication, etc.

L'accès universel du web nécessite d'effectuer un traitement d'adaptation sur le contenu avant de procéder à la transmission. Pour analyser le contexte de l'environnement, le proxy présente l'entité idéale. En confiant au proxy les traitements relatifs à l'adaptation et la négociation du contenu, on allège le reste des tâches aux autres composants du réseau.

Il est fréquent que l'utilisateur change ces préférences (langue, style de contenu, filtrage..) vis-à-vis du contenu demandé. Le système de cache des profils peut donc être augmenté par des mécanismes qui assurent plus d'interaction orientés négociation entre le client et le serveur. Qui dit interaction, dit protocole de communication qui assure une sorte de dialogue sous forme de question réponse entre les deux parties de la négociation. L'utilisateur de protocoles de négociation doit être optionnel pour les terminaux qui ne

peuvent pas supporter l'exécution de protocole de communication autre que leurs propres protocoles de transfert. Les services Web s'avèrent intéressants pour assurer l'interaction entre les entités consommatrices des descriptions et la gestion de ces descriptions.

Du côté de l'adaptation, le format et la structure d'origine du contenu joue un rôle important dans le processus d'adaptation. Afin de faciliter ce processus, le contenu ne doit pas être lié, au moment de sa création, à une plate forme particulière.

L'architecture NAC, part de ses constats pour proposer une solution au problème de négociation et d'adaptation. NAC tire avantage des standards existants, elle définit un ensemble d'entités et associe un ensemble de fonctionnalités à chacune de ces composants. Cette architecture est organisée sous forme de cinq entités qui coopèrent. L'organisation générale de l'architecture NAC est présentée dans la figure 14.

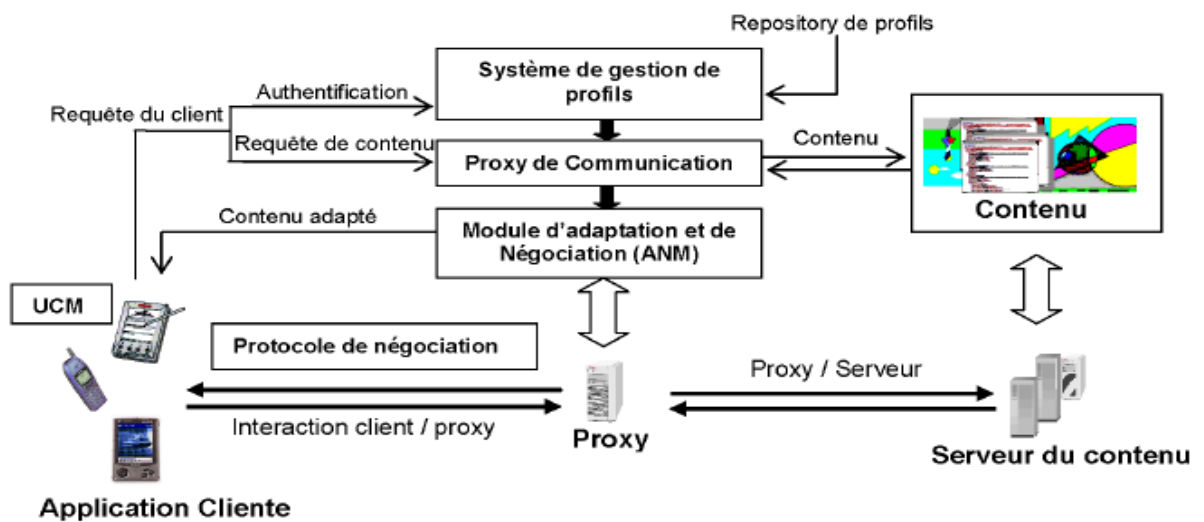


Figure 14 : Organisation générale de l'architecture NAC [30].

- **Le proxy de communication** : il assure l'adaptation et la négociation du contenu de l'architecture. L'adaptation est dans la majorité du temps dynamique. Ce module coopère avec les autres entités de l'architecture afin de rendre la meilleure décision de négociation.
- **Module du contexte de l'utilisateur (UCM)** : assure une négociation avancée du contenu.
- **Protocole de négociation** : assure une stratégie de négociation avancée, définit l'interaction entre le module UCM et le processus de négociation du module ANM (Adaptation and Negotiation Module).

- **Système de gestion de profil** : assure l'analyse et la gestion des descriptions du contexte (la description du client, du contenu, des méthodes d'adaptation..) au profil de l'adaptation du contenu.

Le profil peut être simple, par exemple sous forme d'un contenu textuel ou d'une page HTML, ou complexe, par exemple présentation multimédia.

Le proxy de communication reçoit la requête de l'application cliente. Le proxy essaie d'identifier le profil courant du terminal et son contexte grâce au protocole de négociation qui peut interroger directement l'UCM. La connaissance du contexte client eut faire appel au système de gestion de profils afin d'exploiter les services des repository des profils existants. L'ANM doit prendre en compte la requête courante avant la transmission du contenu final. A n'importe quel moment, et afin de faire la correspondance entre le contenu source, et la capacité des systèmes d'adaptation et les exigences du terminal. Le module ANM peut enrichir son ensemble de contraintes à l'aide du protocole de négociation et des repository des profils. L'ANM essaie d'établir un graphe d'adaptation qui liste toute les étapes et techniques d'adaptation possibles. Le module essaie de trouver le chemin, composé d'un ensemble de tâches d'adaptation qui respecte le mieux le contexte de l'environnement et transmet le contenu adapté à l'application cliente grâce au proxy de contenu.

2.3.3.1. Présentation des principaux acteurs de l'architecture NAC

A. Les applications clientes

C'est cet acteur qui permet au terminal de communiquer avec le reste du système, de recevoir le contenu et de le présenter à l'utilisateur. Afin de supporter la diversité des applications clientes qui peuvent exister dans l'environnement hétérogène, NAC sépare les deux aspects de négociation et d'accès au contenu. Après avoir effectué la configuration du réseau, toutes les requêtes passeront par le module ANM qui assurera la transmission d'un contenu compatible avec les exigences de l'application cliente.

B. L'application PocketSMIL

SMIL permet de créer des présentations riches qui combinent plusieurs ressources de médias (Audio, vidéo, etc.) synchronisées. L'architecture NAC inclut l'application PocketSMIL qui a été développée pour les systèmes embarqués et les terminaux mobiles tels que les ordinateurs de poche (PDA). L'adaptation est effectuée selon les caractéristiques du terminal cible et de son environnement (bande passante disponible, charge mémoire du terminal, etc.).

C. Module d'adaptation et de négociation (ANM)

L'interface de l'ANM comporte un menu et deux zones de texte : une utilisée pour l'affichage des requêtes des applications clientes et une pour l'affichage des réponses des serveurs d'origines. L'ANM peut être exécuté dans le serveur de contenu ou dans un proxy intermédiaire [29].

D. Module du contexte de client (UCM)

UCM (User Context Module) est un module qui fait partie de l'architecture NAC et qui fonctionne du côté du client. Lorsqu'un module UCM est installé du côté d'un terminal, le système d'adaptation et de négociation de NAC peut assurer une stratégie de négociation avancée grâce au protocole de négociation et l'échange de caractéristiques en temps réel entre les modules UCM et ANM. Le module ANM utilise une gestion de cache afin de garder la trace des caractéristiques des terminaux et des changements qu'ils subissent. L'identification de ces caractéristiques (i.e. les profils) est basée sur l'adresse IP des terminaux. Le module permet de configurer les paramètres de négociation avec l'ANM, et de visualiser les événements liés au protocole de négociation.

2.3.3.2. L'utilisation des services Web au profit de la négociation

N'importe quelle entité du système : client, serveur, proxy peut exécuter la tâche de la gestion des profils. En effet, l'exécution de ces traitements peut ralentir le fonctionnement des différentes entités.

Un « repository » basé sur les services SOAP (Simple Object Access Protocol) [25], [26] répond à des requêtes spécifiques en envoyant des profils. Le module ANM est considéré comme un client SOAP, les requêtes RPC (Remote Procedure Calls) sont utilisées pour interroger les profils du « repository » pour extraire le contenu contextuel du terminal concerné. Le « repository » utilise le langage XQuery [27].

2.3.3.3. Adaptation du contenu dans NAC

La figure.15 présente l'architecture d'adaptation.

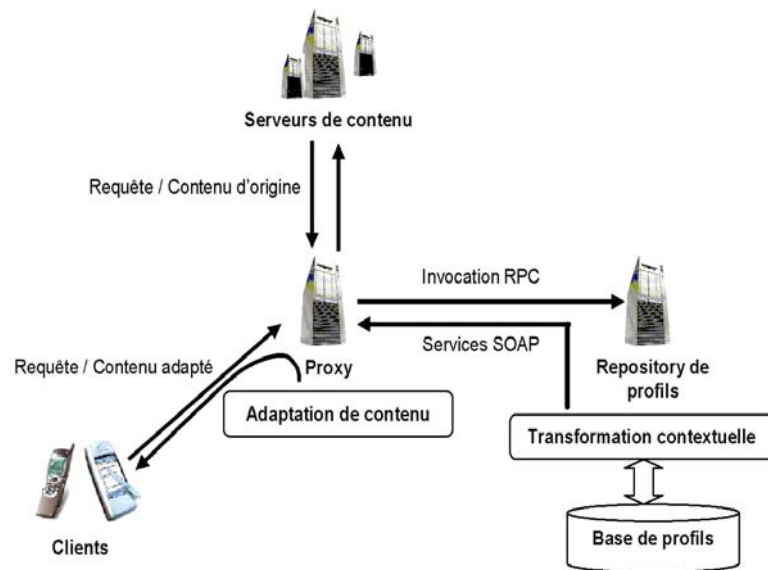


Figure 15 : L'architecture d'adaptation [30].

A. Modèle de description pour l'adaptation

L'architecture NAC est extensible à tout moment pour enrichir les capacités d'adaptation. NAC utilise le modèle de description pour l'adaptation UPS (Universal Profiling Schema) pour la description des profils de l'environnement selon un format descriptif. La description d'une entité est sous forme de *composants*. Par exemple, la description d'un écran est décomposée en la description du nombre de couleurs supportées, la longueur du dispositif d'affichage, la largeur, etc. Les caractéristiques du contenu et du contexte de transmission sont extraites afin de voir si la transmission du contenu source est conforme aux exigences de l'application cliente cible ou non. Le système d'adaptation applique les

mécanismes nécessaires pour que toutes les caractéristiques non conformes soient cohérentes avec les contraintes posées.

B. Adaptation contextuelle

A cause de la variété des caractéristiques du contenu et des autres dimensions du contexte, le nombre des mécanismes de transformation nécessaires peut être très important. NAC définit un mécanisme d'adaptation contextuel qui permet la réutilisation des transformations pour plusieurs contextes. NAC s'appuie sur deux principales approches : la définition des méthodes et des règles d'adaptation paramètres et d'adaptation dynamique.

C. Stratégie d'adaptation

Après la réception d'une requête cliente et dans le cas où il n'y a pas de correspondance entre le contenu source et le contexte du client, le système d'adaptation cherche une variante (une version) du contenu source qui pourra être supportée par le client cible. Si une telle variante n'existe pas, le système cherche une méthode d'adaptation. Le système choisit la version qui a la plus petite taille.

Après avoir présenté les principales entités de l'architecture de négociation et d'adaptation NAC. Il ressort que la conception d'une architecture d'adaptation et de négociation efficace doit fournir :

- Un modèle de description de l'environnement et une stratégie d'analyse et de gestion du contexte,
- La définition des règles d'interaction entre les différentes entités du système et l'exploitation des descriptions pour l'adaptation du contenu,
- Des mécanismes d'adaptation qui assurent la transformation du contenu de son état initial vers un état final qui répond au contexte des clients cibles [3].

2.3.3.4. Techniques et méthodes d'adaptation du contenu dans NAC

Avant d'appliquer un processus d'adaptation sur le contenu, le système de négociation doit déterminer le couple (contexte initiale, contexte final). L'architecture NAC distingue deux types d'adaptation selon les variations des dimensions de contexte. Ces deux types sont l'adaptation statique et l'adaptation dynamique.

A. Adaptation statique

Consiste à préparer une collection riche de méthodes pour différents contextes. Chaque méthode considère un unique couple (contexte initiale, contexte final) et ne peut pas être appliquée si le contexte change de dimension. NAC inclut des types de transformation pour l'adaptation telle que la compression des images et des vidéos, la couverture de types de médias, le taillage des ressources, etc.

B. Adaptation dynamique

B.1. Adaptation dynamique paramétrée

L'approche adoptée dans NAC est de définir une feuille de transformation générique sur le profil pour générer une nouvelle feuille de transformation. La feuille de transformation sélectionne et instancie les règles de transformation selon les contenus du profil. La génération automatique d'adaptation est présentée dans la figure 16.

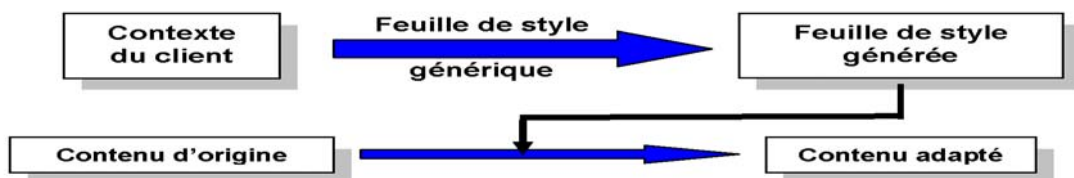


Figure 16 : Génération automatique d'adaptation [3].

B.2. Adaptation dynamique pendant l'exécution

On considère les terminaux qui ont des systèmes de présentation non évolués. Suite à cette limitation des terminaux, dans NAC la tâche de sélection de contenus est attribuée au module de négociation et d'adaptation. NAC inclut l'adaptation des présentations SMIL incluant de la vidéo vers un flux vidéo unique. Ce type d'adaptation inclut l'analyse du scénario temporel SMIL en entrée et l'encodage de la vidéo en sortie.

2.3.4. Une architecture de fourniture de documents multimédia adaptables dans une logique paire à paire

2.3.4.1. Vers une architecture paire à paire...

• Les architectures d'adaptation d'aujourd'hui suivent le modèle client/intermédiaires/serveur (C/I/S).

• L'objectif de [7] est d'aller plus loin :

Tout participant peut être potentiellement consommateur, fournisseur ou adaptateur de documents multimédia. La solution proposée dans [7] est une logique pair à pair. La figure 17 représente l'architecture fonctionnelle de fourniture de documents multimédia composés.

• Une telle architecture doit pouvoir découvrir, composer et instancier des ressources d'adaptation.

• L'objectif est la conception et la réalisation d'un système :

– Permettant à une **large** communauté d'échanger des documents **multimédia composés**

– **Adaptés** aux **contextes changeants** des utilisateurs

– En fournissant les outils pour mettre en commun et **composer** les différentes **ressources d'adaptation** que les usagers souhaitent **partager**.

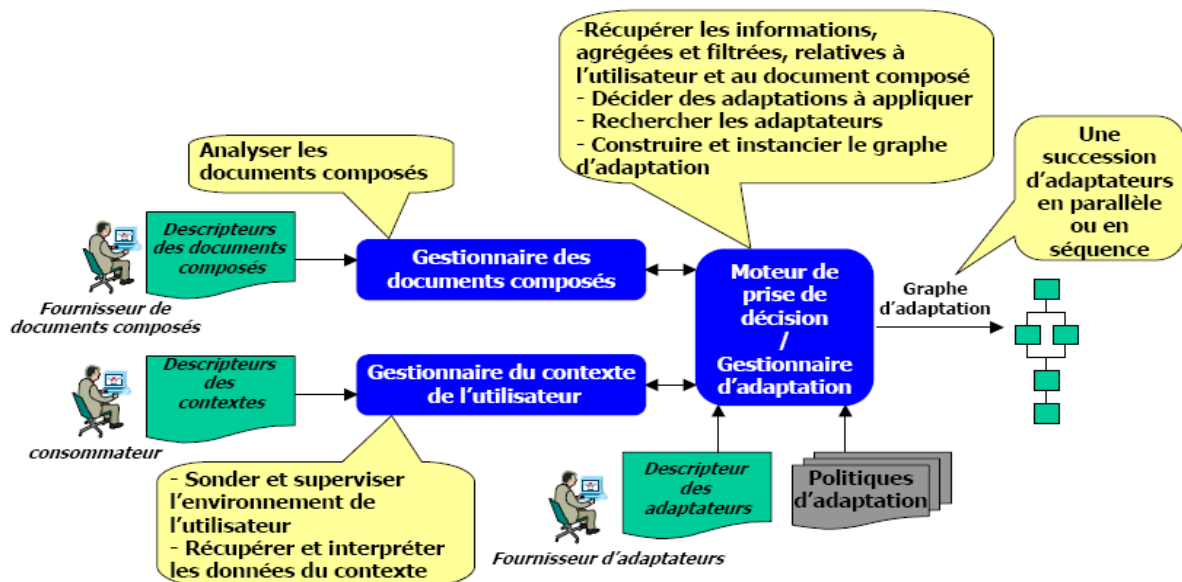


Figure 17 : Architecture fonctionnelle de fourniture de documents multimédia composés [7].

2.3.4.2. Document composé

- Un document composé (Pas de streaming) est :
 - un document dans lequel sont **synchronisés** temporellement et/ou spatialement plusieurs médias élémentaires (texte, vidéo, image, audio).
- Un document n'est fourni à l'utilisateur qu'après la synchronisation finale
 - Pas de considération des contraintes de temps réel (streaming).
 - La notion de temps de réponse est prise en compte.

2.3.4.3. Graphe d'adaptation

- Le graphe d'adaptation tel que illustré dans la figure 18, est une succession d'adaptateurs en parallèle ou en séquence
- Il faut un outil pour décrire et découvrir les adaptateurs
 - WSDL et UDDI dans le cas d'adaptateurs sous forme de services Web.

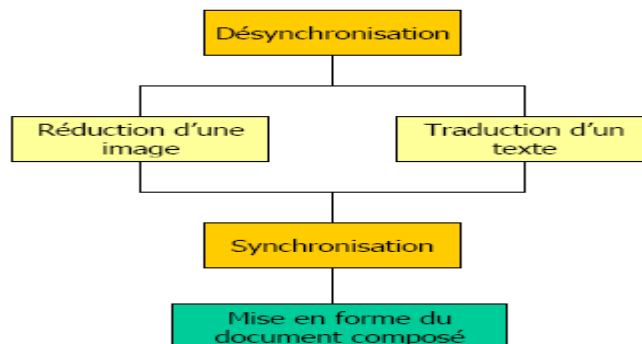


Figure 18 : Graphe d'adaptation [7].

2.3.4.4. Architecture fonctionnelle : Gestion dynamique du contexte utilisateur

Un **contexte** se caractérise par: les préférences de l'utilisateur, les capacités de son terminal et les caractéristiques de son réseau d'accès.

- Plusieurs outils pour décrire le contexte, le choix dans [7] se porte sur MPEG-21.
- Des **sondes** permettent de détecter **dynamiquement** des changements dans l'environnement d'exécution :
 - Sonde pour la bande passante au niveau du réseau.
 - Sonde CPU ou mémoire au niveau du terminal.

- L'**interaction** avec l'utilisateur permet au module de gestion du contexte de tenir compte des changements **dynamiques** (relatifs aux préférences de l'utilisateur ou à certaines capacités de son terminal).

2.3.4.5. Architecture fonctionnelle : Gestion dynamique du contexte de l'adaptation

- Un pair qui offre un adaptateur peut à tout moment se **déconnecter**..., le gestionnaire d'adaptation et l'adaptateur s'échangent des messages à intervalles réguliers, en cas de non réponse, une déconnexion est détectée.

- La gestion dynamique implique la **reconstruction** du graphe d'adaptation

- Transition (basculement) d'un adaptateur vers un autre sans perte d'état de l'adaptation en cours.

- La recherche d'autres adaptateurs se fait sur la base des mêmes critères que l'adaptateur disparu.

- Cela implique la gestion d'un cache contenant les critères de recherche des adaptateurs tant que les adaptations appliquées à un document donné sont en cours.

2.3.4.6. Gestionnaire de prise de décision

- Dialogue avec le gestionnaire de contexte de l'utilisateur

- Récupère les informations du contexte de l'utilisateur telles que la langue parlée, les formats préférés, les moyens de paiement, etc.

- Dialogue avec le gestionnaire des documents composés

- Récupère les informations caractérisant un document composé ainsi que les médias élémentaires le composant telles que les formats et dimensions des médias, la taille du document, etc.

- Les prises de décision se basent sur les politiques d'adaptation.

On peut conclure que l'originalité de cette approche [7] est de généraliser les architectures client / intermédiaire(s) / serveur aux architectures basées sur un modèle pair à pair [7].

2.3.5. L'architecture DCAF

Une architecture d'adaptation de contenu générique, extensible et interopérable est constituée de six composants principaux, l'architecture DCAF est représentée dans la figure ci-dessous [16]:

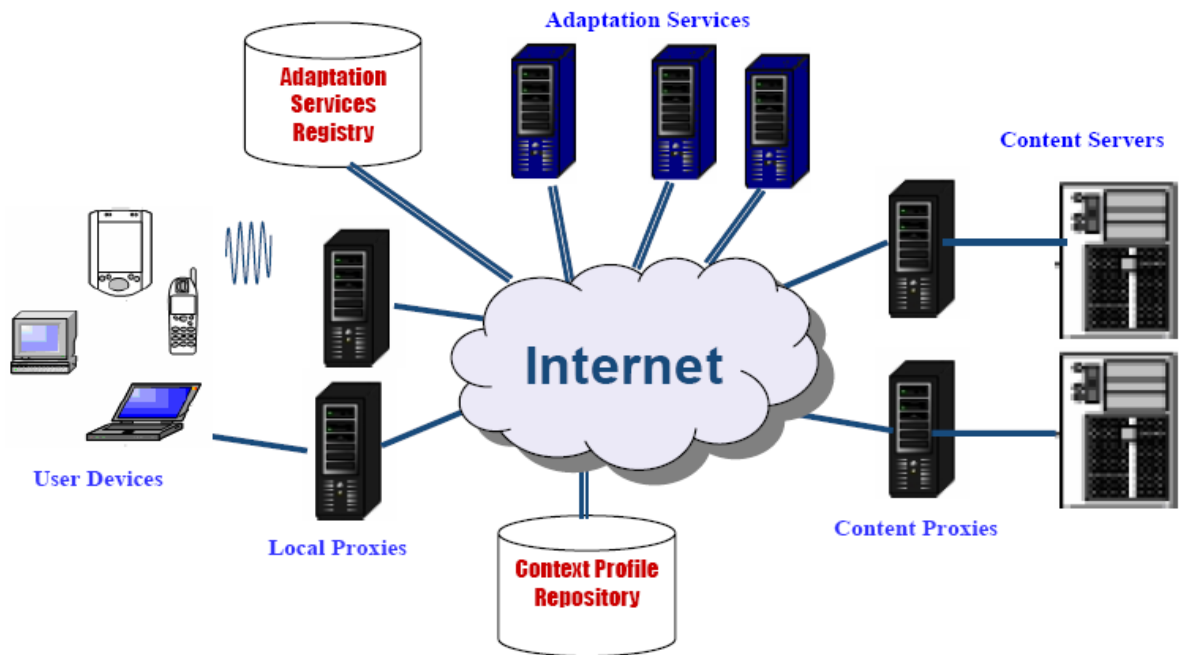


Figure 19 : Architecture DCAF [16].

1. **Serveur de contenus** : Ce sont des entrepôts de données.
2. **Proxy de contenus** : Fournissent un accès aux serveurs de contenu.
3. **Gestionnaire de profils utilisateur** : Ce serveur stocke les informations du contexte utilisateur.
4. **Répertoire des services d'adaptation** : Pour faire face au problème de services web d'adaptation disponibles, un annuaire d'adaptation est introduit.
5. **Services d'adaptation** : Serveur hébergeant un ou plusieurs services d'adaptation.
6. **Proxy locaux** : Ces proxys prennent en charge la récupération et le traitement des profils de contexte. Un élément fondamental du proxy local est le module de négociation et d'adaptation de contenu (CNAM).

2.3.5.1. Module de négociation et d'adaptation de contenu

CNAM est la partie fondamentale du proxy local. Le CNAM est composé de moteur de décision et de planificateur d'adaptation.

A. Générateur de graphe d'adaptation multimédia (MAGG)

Dans un scénario, un client demande un contenu multimédia à un serveur au moyen d'un dispositif connecté au serveur par un réseau donné. L'objectif de l'adaptation est de transformer le contenu multimédia stocké sur le serveur en fonction des capacités du terminal, de la connexion réseau et des préférences de l'utilisateur avant de le transmettre au client. Par exemple, si le contenu est rédigé en anglais et que l'utilisateur préfère le français, deux transformations sont nécessaires: une conversion du format textuel en séquence audio et une traduction de l'anglais au français. Comme sur chaque transformation peut être effectuée un ou plusieurs services d'adaptation, cela amène à rechercher un scénario d'adaptation optimal. Un scénario d'adaptation correspond à un chemin de la racine à l'extrémité du graphe construit par le module 'Générateur de graphe d'adaptation multimédia' et exécuté par le proxy local. Un graphe d'adaptation est un graphe orienté acyclique DAG. Dans le but de construire un DAG, un analyseur XML est utilisé. Cet analyseur prend en entrée le fichier de description d'un service et crée l'opérateur d'adaptation correspondant. Le développement de l'algorithme MAGG permet de générer le graphe d'adaptation.

2.3.5.2. Implémentation du prototype

Une architecture logicielle DCAF a été développée et le prototype est composé de :

A. Moteur de décision d'adaptation

Le moteur de décision est chargé de la construction du graphe d'adaptation et du calcul d'un chemin optimal.

B. Proxys

Trois types de proxys ont été développés. Des proxys de contenu chargés de la gestion des serveurs de contenus et des métas donnés. Des proxys locaux qui constituent le cœur de l'architecture ont été mis on place. Enfin, des proxys services d'adaptation ont été implémentés. L'élément au cœur de l'adaptation est le proxy local. Un Proxy local est composé de cinq éléments principaux comme illustré dans la figure 20.

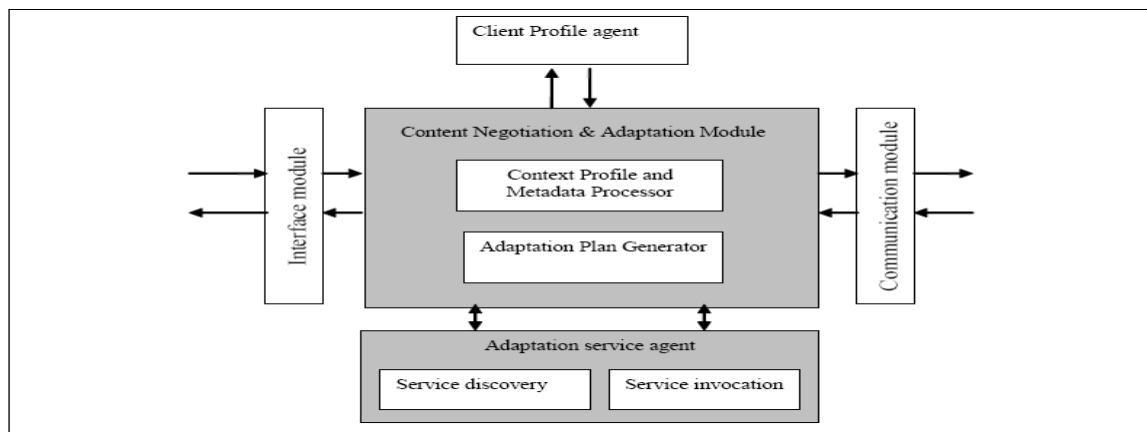


Figure 20 : Schéma fonctionnel du Proxy local [1].

C. Bases de données utilisées dans le profil

Quatre bases de données ont été développées dans le prototype. Le répertoire des profils qui stocke les informations relatives aux utilisateurs telles que le nom, l'identifiant, l'annuaire des services d'adaptation et les liens vers les fichiers de description (OWL-S), une base de contenus héberge les données d'application et les contenus téléchargeables par l'utilisateur. Enfin, la base de méta donné accueille les informations de recherche sur le contenu et un lien vers le fichier MPEG-7, les BDD sont développées sous MySQL-4.0.12.

Pour concrétiser ces concepts, une implémentation typée de DCAF a été développée pour une application médicale.

D. Interface visuelle

Plusieurs interfaces visuelles ont été développées pour la communication entre le système et l'utilisateur : interface de connexion, interface de saisie de requêtes, interface de saisie des profils et interface de présentation des résultats. Les interfaces les plus significatives sont : l'interface de saisie des profils et l'interface d'affichage de données adaptées.

D.1. Interface de saisie des profils

L'interface de saisie des profils est composée de trois panneaux : panneau de spécification du réseau, panneau de saisie des préférences de l'utilisateur, panneau de sélection du type du terminal.

D.2. Interface d'affichage de données adaptées

Dans un système pervasif, l'adaptation a deux aspects principaux. Adaptation des données/contenu et adaptation des interfaces utilisateur. [1] se focalise sur l'adaptation de contenus. Comme l'adaptation du contenu nécessite une adaptation de l'interface, l'équipe qui a travaillé sur [1] et l'équipe SEFAGI (Simple Environnement For Adaptable Graphical Interfaces) ont collaboré. Cette collaboration a permis de disposer d'un système complet d'adaptation qui délivre les informations médicales adaptées au contexte de l'utilisateur en termes de contenu et d'interface.

Les principales contributions de [1] sont les suivantes :

- L'introduction d'une nouvelle architecture répartie d'adaptation de contenu multimédia fondée sur l'utilisation de l'architecture DCAF.
- L'application de la technologie de services web sur l'informatique pervasif (pour l'adaptation de contenu).
- La spécification d'une ontologie pour la description de service d'adaptation multimédia.
- Une méthodologie d'optimisation et de composition de service a été proposée.
- Enfin, et pour évaluer la proposition, un prototype a été réalisé [1].

2.3.6. Etude comparative entre les architectures d'adaptation étudiées

Le tableau suivant représente une étude comparative entre les architectures d'adaptation présentées ci-dessus [52], [53].

Architecture	Objectif	Proxy	Adaptation	Gestion de profils
Adaptation d'une application multimédia par un code mobile [8]	Adaptation d'une application multimédia répartie par un code mobile	Sur le site proxy est déployé un agent mobil d'adaptations	-Une vidéo est transmise d'un site web au client. -La vidéo passe par le proxy. -Des agents d'adaptation déployés sur le proxy modifient le flot vidéo.	N'est spécifiée
Architecture génique pour la fourniture de service adaptable. [2]	Architecture qui s'attaque Simultanément à l'adaptation de la logique de services en utilisant des composants et à l'adaptation des flux multimédia.	Pas de proxy l'intermédiaire est un gestionnaire de services	-Le module de supervision détecte le changement. -Le gestionnaire détermine les actions d'adaptation. - Le gestionnaire de service envoie la requête de téléchargement de la version adaptée.	Base de profils
NAC [30]	Assurer dans un environnement hétérogène une	Proxy de communication orientée	-ANM établit un graphe d'adaptation.	Annuaire des profils

	transmission de contenu adapté en mettant l'accent sur la négociation.	négociation.	-Adaptation statique. -Adaptation dynamique paramétrée. -Adaptation dynamique pendant l'exécution.	
Architecture d'adaptation dans une logique pair à pair [7]	Tout participant doit être consommateur, fournisseur ou adaptateur.	Pas de proxy	-Récupérer les informations relatives à l'utilisateur et au document composé. -Décider les adaptations a appliqué et rechercher les adaptateurs. -Instancier le graphe d'adaptation.	Gestionnaire de contexte de l'utilisateur
DCAF [16]	Architecture orientée services d'adaptation multimédia dans un environnement pervasif en résolvant le problème de L'interopérabilité, la flexibilité et le passage à l'échelle.	-Proxy de contenu. -Proxy locaux. -Proxy de services d'adaptation.	-Basée sur des services d'adaptation tiers. -Introduit un annuaire de services d'adaptation (ASR). -Assurer des services web	Par le CPR (Context Profil Repository)

			d'adaptation disponibles implémentés indépendamment de DCAF.	
--	--	--	---	--

Tableau 2 : Etude comparative entre les architectures présentées [52], [53].

2.4. Conclusion

Dans ce chapitre, nous avons présenté un état de l'art dans le but d'introduire les concepts qui nous sont nécessaires pour proposer notre architecture d'adaptation. Nous avons présenté les modèles d'adaptation existants suivie d'une étude comparative entre ces modèles. Puis, dans ce chapitre nous avons présenté les architectures proposées dans la littérature pour l'adaptation dans les environnements hétérogènes avec une étude comparative entre ces architectures.

Après cette étude, nous proposons dans le chapitre suivant une architecture d'adaptation multimédia pour un système ouvert pour téléphone mobile.

CHAPITRE 3

«Proposition d'un Système Ouvert pour l'Echange d'Information Multimédia»

3.1. Introduction

Nous présentons dans ce chapitre notre architecture basée sur le modèle Client/Intermédiaire/Serveur. Nous distinguons quatre parties principales : le client multimédia émetteur, le client multimédia récepteur, le serveur avec les descripteurs de l'environnement et le proxy comme des services web. Notre investigation a pour objectif de concevoir un système qui intègre des téléphones mobiles hétérogènes. Cette architecture ouverte vise à améliorer la qualité du service (Qos) entre le client multimédia émetteur et le client multimédia receveur.

Les limitations des capacités des téléphones mobiles à transmettre tout type de données sans problèmes, requièrent une adaptation de contenu ou un autre type d'adaptation.

Notre architecture permet aux clients multimédia de délivrer le contenu multimédia par rapport à la spécification du téléphone mobile receveur. Comme une étude de cas, nous présentons la spécification de quelques téléphones mobiles : Nokia 2610, Samsung X640, Sony Ericsson K320, Siemens CX65 et Nokia N93i avec quatre scénarios d'adaptation illustratifs.

3.2. L'architecture proposée

L'architecture proposée dans ce mémoire est illustrée dans la figure 21. Cette architecture est basée sur le modèle Client/Intermédiaire/Serveur. Cette architecture ouverte vise à procurer la flexibilité et l'adaptabilité du service (Qos) entre l'émetteur multimédia et le récepteur multimédia. Notre architecture permet aux clients multimédia de délivrer le contenu par rapport à la spécification du téléphone mobile receveur. Elle intègre des téléphones mobiles hétérogènes et fournit un service d'adaptation pour ces téléphones mobiles d'une manière transparente [52], [53] aux utilisateurs finaux.

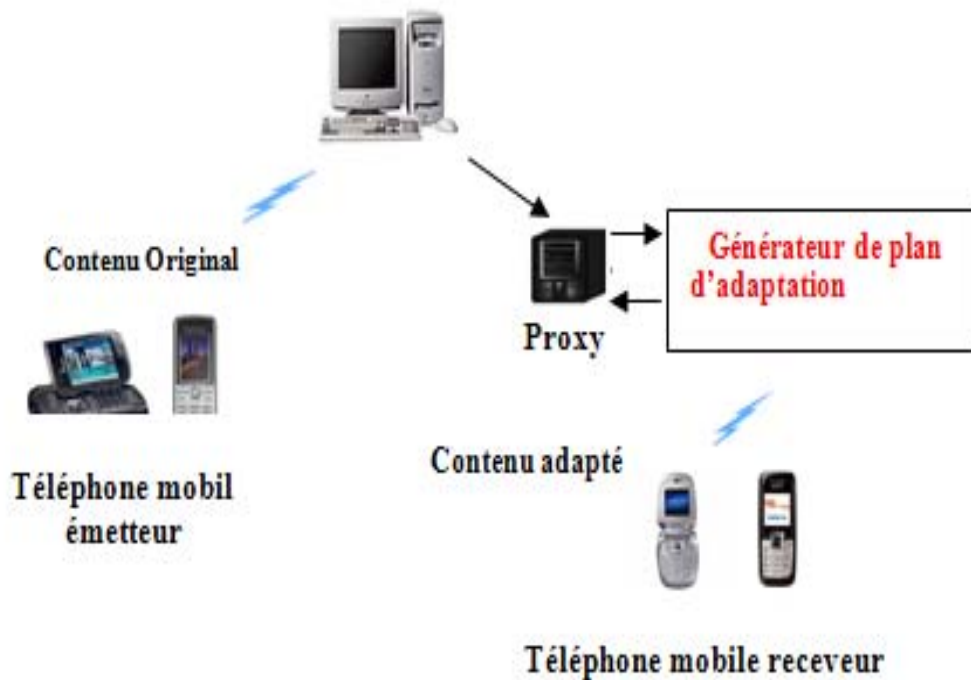


Figure 21 : L'architecture du système ouvert [52], [53].

Il y a deux types de clients multimédia: le client multimédia émetteur et le client multimédia receveur.

3.2.1. Le Serveur

Le serveur a deux descripteurs structurés comme une base de données. Chaque téléphone multimédia a différentes caractéristiques (identifiant, etc.). Le descripteur du document multimédia contient les données originales reçues du client multimédia émetteur. Il est important de souligner que le serveur supporte tous les types de données multimédia. Par conséquent, nous supposons que chaque message émis par le client multimédia émetteur passera directement et d'une manière transparente au serveur. Puis, le serveur sélectionne de ce message tous les paramètres de l'environnement. A savoir: les paramètres du téléphone du client multimédia receveur (comme l'affichage de l'écran, les contenus supportés) et les paramètres du contenu multimédia (comme le format, la taille, la dimension de l'image, etc). Après avoir collecté les paramètres de l'environnement, le serveur les vérifie dans les descripteurs. Si ces descripteurs n'existent pas, il les enregistre.

3.2.2. Le proxy

Le proxy constitue le noyau de notre architecture, c'est un web service qui assiste le serveur avec ces deux modules : le module de prise de décision et le module d'adaptation. La Figure 22, montre le schéma fonctionnel du proxy. Du moment que le succès de l'adaptation dépend de la qualité et la quantité de connaissances acquises sur l'environnement, le module de communication dans le Proxy reçoit les paramètres de l'environnement représentant une requête d'adaptation (1) de la part du serveur. Puis, le module de communication envoie à la base de données les nouveaux paramètres de l'environnement (2).

Si les nouveaux paramètres de l'environnement existent dans la base de données ; cette dernière envoie le type d'adaptation enregistrés par rapport à ces nouveaux paramètres de l'environnement au module de prise de décision (3). Sinon, le module de prise de décision dans le proxy sélectionne le type d'adaptation correspondant aux nouveaux paramètres de l'environnement dans la base de données d'adaptation. Sinon si le type d'adaptation correspondant n'existe pas, la base de données envoie seulement les nouveaux paramètres d'environnement (4). Dans ce cas, il (le module de prise de décision) crée un (des) nouveau (x) type (s) d'adaptation, l'envoie à la base de données dans le but de la mettre à jour (5). Après cela, le module de prise de décision envoie les nouveaux paramètres de l'environnement et le type d'adaptation généré au générateur de plan d'adaptation (6) pour avoir le plan d'adaptation optimal déjà enregistré (7).

Si le plan d'adaptation optimal n'existe pas, le générateur de plan d'adaptation génère cet ensemble d'actions correspondantes aux paramètres donnés. Avant d'envoyer le message au client multimédia receveur (8), le module d'adaptation exécute le plan d'adaptation optimal [52], [53].

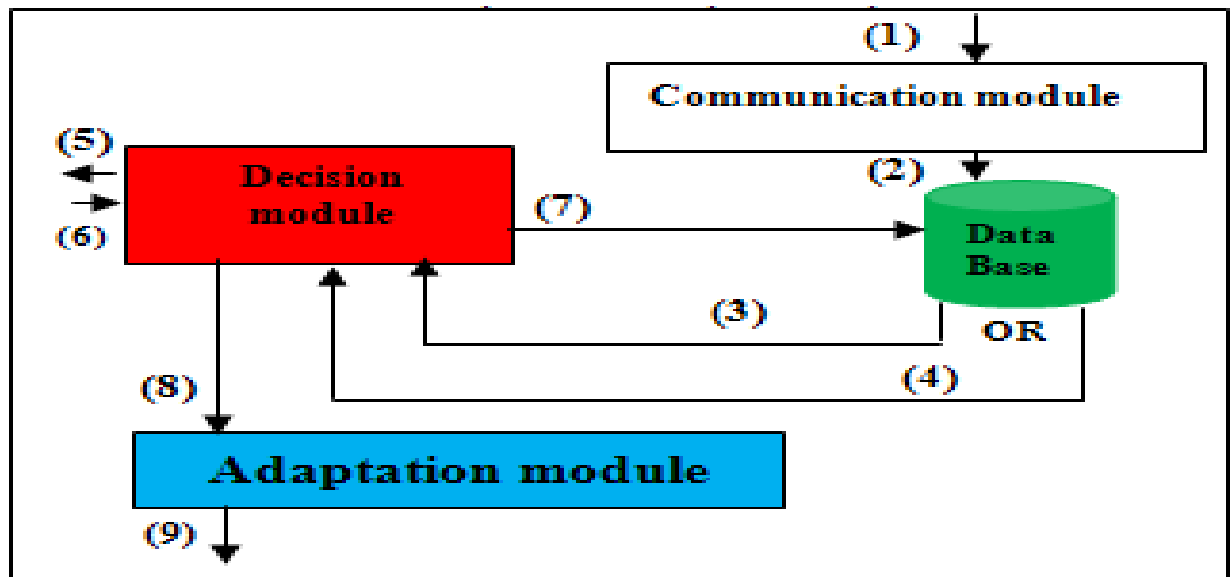


Figure 22 : Le schéma Fonctionnel du Proxy [52], [53].

Après la réception du message, le serveur envoie au téléphone mobile multimédia du client émetteur un message (SMS) de confirmation quand le message a été bien reçu. Dans le cas contraire, un message d'erreur est envoyé au client multimédia (téléphone mobile) émetteur.

3.2.2.1. Le générateur de plan d'adaptation

Le rôle du générateur du plan d'adaptation est de générer le plan d'adaptation optimal des paramètres d'environnement donnés et aussi d'enregistrer tout les types d'adaptation et l'ensemble des actions d'adaptation correspondants.

3.3. Domaine d'application

3.3.1. Téléphonie mobile

Chaque téléphone mobile multimédia a une spécification ou un contexte du terminal. Pour cette raison, nous ne sommes pas capables de spécifier tous les téléphones mobiles multimédia existants dans le marché. Comme une étude de cas, nous choisissons de spécifier les dimensions, le type, la taille d'affichage, les types de sonneries, les slot de la carte mémoire, GPRS, HSCSD, EDGE, WLAN, Bluetooth, le port Infrarouge, le port

USB, les formats d'image supportés, les formats de vidéo supportés, les formats d'audio supportés, la messagerie, la batterie etc. de quelques téléphones mobiles [49].

3.3.1.1. Caractéristiques des téléphones mobiles

La spécification de quatre téléphones mobiles multimédia est présentée dans le tableau 3 [52], [53].

Les caractéristiques techniques	Nokia 2610	Samsung SGH-X640	Sony Ericsson K320	Nokia N93i	Siemens CX65
Dimensions	104 x 43 x 18 mm	87.4 x 47 x 23 mm	101 x 44 x 18 mm	108 x 58 x 25 mm, 115 cc	132x176
Type	CSTN, 65K couleurs	UFB, 65K couleurs	UBC, 65K couleurs	TFT, 16M couleurs	TFT, 65K couleurs
Taille d'affichage	128 x 128 pixels	128 x 160 pixels	128 x 160 pixels, 1.8 inches	240 x 320 pixels	162x176 pixels
Types de sonneries	Polyphonique (24 canaux), MP3	Polyphonique (40 canaux)	Polyphonique (40 canaux), MP3, AAC	Polyphonique (64 canaux), MP3	Polyphonique (40 canaux)
Slots de carte mémoire	Non	Non	Non	miniSD, hot swap	Non
GPRS	Oui	Oui	Oui	Classe 32, 107.2/64.2 kbps	Classe 10 (4+1/3+2 slots), 32 - 48 kbps
HSCSD	Non	Non	Oui	Oui (via PC dial-up)	Non
EDGE	Non	Non	Non	Classe 32, 296 kbps; DTM Classe 11, 236.8 kbps	Non
WLAN	Non	Non	Non	Wi-Fi 802.11b/g	Non
Bluetooth	Non	Non	Oui	Oui	Non

Port infrarouge	Non	Non	Oui	Oui	Oui
Appareil photo	Non	Disponible	Disponible	Disponible	Disponible
Format d'images supportées	GIF, PEG, PNG, BMP	BMP, GIF, JPEG, PNG, X-NP-WPNG	GIF, JPEG, WBMP, BMP, PNG, VND.WAP, WBMP, CVG	GIF, JPEG, JP2, JPG, PNG, SVG+WMP, TIFF.	BMP, GIF, PNG, JPEG, SVG, +xml, VND.wap.W B
Caméra vidéo	Non	Non	Disponible	Disponible	Disponible
Formats de vidéo supportés	Non	Non	Mpeg, mp4, 3gpp, mpeg4, mp4v-es	3gpp, mp4, vnd.rn-real	3gpp
Formats d'audio supportés	Midi, mid, mp3, x-mid, amr, amr-wb, mpeg, x-amr	Mélodie, midi	Amr, rhz, midi, x-midi, sp-midi, midi melody, mpeg, mpeg3, mp3, wav, 3gpp, mp4, x-wav, xmf	3pgg, aac, amr, amr-wb, au, basic, mid, midi, mobile-xinf, mp3, mp4, mpeg, rmf, sp-midi, vn d.rm-real audio, wav, x-amr, x-au, x-beatnik-rmf, x-mid, x-midi, x-pn-real audio, x-pn-real audio plugin, x-rmf, x-wav	Midi, wav, amr
Messagerie	SMS, MMS, Email, Message Instantané	SMS, EMS, MMS	SMS, MMS, Email, Message Instantané	SMS, MMS, Email, Message Instantané	SMS, MMS, Email
Navigateur	WAP 2.0/xHTML	WAP 2.0/xHTML	WAP2.0/xHTML, HTML (NetFront)	WAP 2.0/xHTML, HTML	WAP 2.0/xHTML
Batterie	Standard Li-Ion, 970	Standard, Li-Ion 800 mAh	Standard, Li-Ion 750 mAh	Standard, Li-Ion 950 mAh (BL-	Standard, Li-Ion 750 mAh

	mAh (BL-5C)		(BST-36)	5F)	(EBA-660)
Jeux	Coin Jeux + téléchargeable	2 -Snowball fighter, Bubble smile téléchargeable	Oui	Oui	Oui

Tableau 3 : Spécification de quatre téléphones mobiles multimédia [52], [53].

3.3.2. Les adaptations projetées

Beaucoup de techniques d'adaptation ont été développées pour délivrer les données multimédia au client multimédia receveur dans un environnement hétérogène (de téléphones mobiles hétérogènes). D'habitude, les techniques disponibles appliquent les transformations textuelles, le transcodage de l'image et la procession de l'audio. Une liste de technologies d'adaptation de contenu qui peut être appliquée aux types de média depth de base : texte, image, audio et vidéo qui sont présentés dans le tableau 4 [50], [51].

Catégorie	Texte	Image	Vidéo	Audio
Transcodage	-Conversion de format. - réduction de la taille du fond.	- réduction de la taille des données. - réduction de la dimension. - réduction profondeur-couleur. - réduction color-to-grayscale. -conversion du format.	- réduction du frame rate. - réduction de la résolution spatiale. - réduction de la résolution temporelle. - réduction du font de la couleur. - conversion du format.	- réduction audio vers stéréo- mono. - conversion du format.

Transmodage	-texte-vers-audio.	-Image vers texte.	-vidéo-vers-image. -vidéo-vers-texte. -vidéo-vers-audio.	-audio-vers-texte.
Résumé	-Résumé du texte.		- Extraction du key frame.	-audio highlight.
translation	- translation du langage.		- translation du langage.	- translation du langage.

Tableau 4 : Les types de medias et les technologies d'adaptation [50], [51].

En général, les techniques d'adaptation de contenu peuvent être classifiées comme adaptation sémantique et adaptation physique. Dans notre étude, nous sommes intéressés aux techniques d'adaptation physique (l'adaptation au niveau du contenu).

3.3.3. Les scénarios d'adaptation proposés

Scénario 1: Le client multimédia émetteur est le téléphone mobile **Nokia N93i**. Il transmet une image à un autre téléphone mobile receveur **Nokia 2610**. L'image est enregistrée sous format TIFF coloré. Le tableau 3, spécifie que **Nokia 2610** ne supporte pas le format d'image TIFF et la dimension de l'image est plus grande que la taille d'affichage de l'écran du **Nokia 2610**. Donc, en a besoin de deux transformations : l'adaptation de la dimension et l'adaptation du format.

Scénario 2: Le client multimédia émetteur est le **Sony Ericsson K320**. Il transmet une vidéo à un autre client multimédia receveur **Samsung X640**. Le client receveur dans ce cas ne peut pas recevoir la vidéo. Donc, il est nécessaire d'avoir une image de la séquence vidéo, convertir l'audio en texte et le changement de la dimension.

Scénario3: Le client multimédia receveur est le **Siemens CX65** qui ne supporte pas la vidéo enregistrée dans le format MPEG envoyé du téléphone mobile **Sony Ericsson K320**. Par conséquent, en a besoin d'une conversion du format de la vidéo.

Scénario4: L'audio est enregistré dans le format .wav est envoyé par le client multimédia émetteur **Nokia N93i** au client receveur **Samsung SGHX640**. Dans ce cas, il est nécessaire de faire une conversion du format de l'audio pour être reçu par le téléphone mobile du client multimédia receveur [52], [53].

3.4. Conclusion

Dans ce chapitre, nous avons présenté l'architecture de notre système ouvert pour l'échange de données multimédia pour les téléphones mobiles multimédia. L'architecture est basée sur le modèle Client/ Intermédiaire/ serveur. Le principal objectif de notre système ouvert est d'améliorer la qualité du service (Qos) dans l'échange de données multimédia en utilisant des téléphones mobiles hétérogènes et d'intégrer les divers types de téléphones mobiles. La conception de notre architecture va être présentée dans le chapitre suivant en utilisant la modélisation orientée objet par UML.

CHAPITRE 4

« *Conception & Modélisation* »

4.1. Introduction

Ce chapitre a pour objectif de modéliser la base des scénarios dans le proxy qui est un point clé pour avoir un historique des scénarios d'adaptation. Pour cela, nous utilisons le langage de modélisation unifié UML. Les classes principales dans chaque modèle sont: téléphone mobile multimédia, le serveur et le proxy. Nous avons instancié la classe téléphone mobile et la classe multimédia parce que c'est intéressant dans notre cas. Nous utilisons aussi les relations d'association, de généralisation et de composition. Nous présentons aussi la base de données multimédia et la base de données des téléphones mobiles.

4.2. La modélisation de la base de données d'adaptation

4.2.1. Les scénarios d'adaptation

Nous avons choisi des scénarios pour modéliser la base de données d'adaptation (les scénarios déjà cités dans le chapitre précédent).

A. Scénario 1

Le client multimédia émetteur est le téléphone mobile **Nokia N93i**. Il transmet une image à un autre téléphone mobile receveur **Nokia 2610**. L'image est enregistrée sous format TIFF coloré. Le tableau 3, spécifie que **Nokia 2610** ne supporte pas le format d'image TIFF et la dimension de l'image est plus grande que la taille d'affichage de l'écran du **Nokia 2610**. Donc, en a besoin de deux transformations : l'adaptation de la dimension et l'adaptation du format.

La figure 23 représente le diagramme de classe qui modélise le premier scénario d'adaptation. Nous commençons la modélisation par l'instanciation de la classe téléphone mobile (deux objets sont instanciés : Nokia 2610 et Nokia N93i). Après l'envoi de l'image originale par le Nokia N93i, le serveur reçoit d'une manière transparente le message, extrait les paramètres de l'environnement et envoie ses paramètres d'environnement au proxy [48], [54].

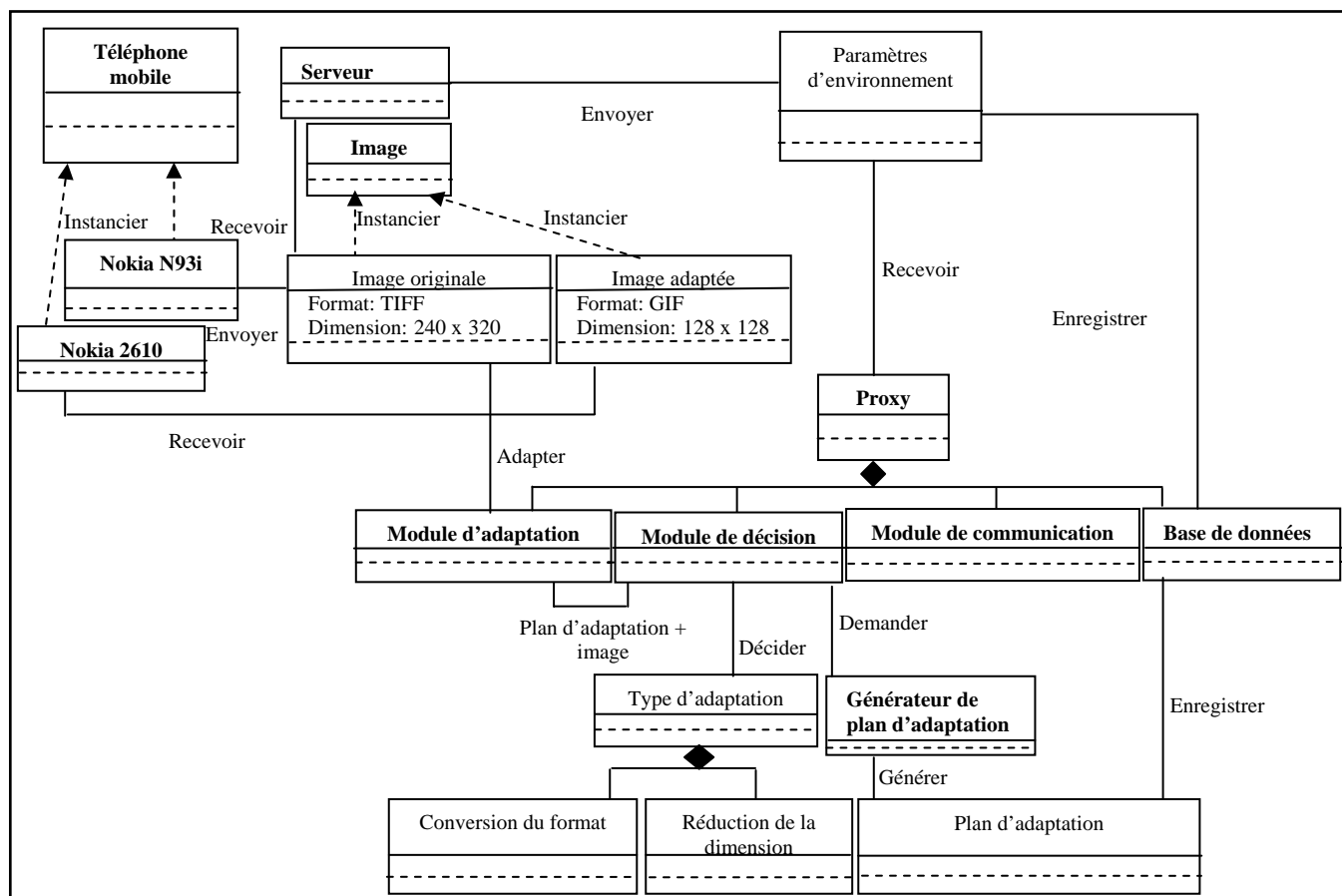


Figure 23 : Diagramme de classe du scénario 1 [48], [54].

Le proxy est composé d'un module d'adaptation, d'un module de prise de décision, du module de communication et la base de données.

Le module de prise de décision décide du type d'adaptation à exécuter dans un cas précis. Dans le premier scénario par exemple, les types d'adaptation sont : la conversion du format et la réduction de la dimension. Après l'étape de la précision du type d'adaptation, le module de prise de décision demande la génération du plan d'adaptation. Comme une troisième étape, ce module (de prise de décision) envoie le plan d'adaptation et l'image au module d'adaptation.

Le module d'adaptation adapte l'image par rapport au plan d'adaptation généré.

Finalement, Nokia 2610 reçoit l'image adaptée.

La figure 24 représente la base de données d'adaptation multimédia par rapport au scénario 2.

B. Scénario 2

Le client multimédia émetteur est le **Sony Ericsson K320**. Il transmet une vidéo à un autre client multimédia receveur **Samsung X640**. Le client receveur dans ce cas ne peut pas recevoir la vidéo. Donc, il est nécessaire d’avoir une image de la séquence vidéo, convertir l’audio en texte et le changement de la dimension.

Dans ce scénario, seulement les objets et les types d’adaptation changent. Les objets instanciés qui vont être changés sont : le téléphone mobile et les objets multimédia [48], [54].

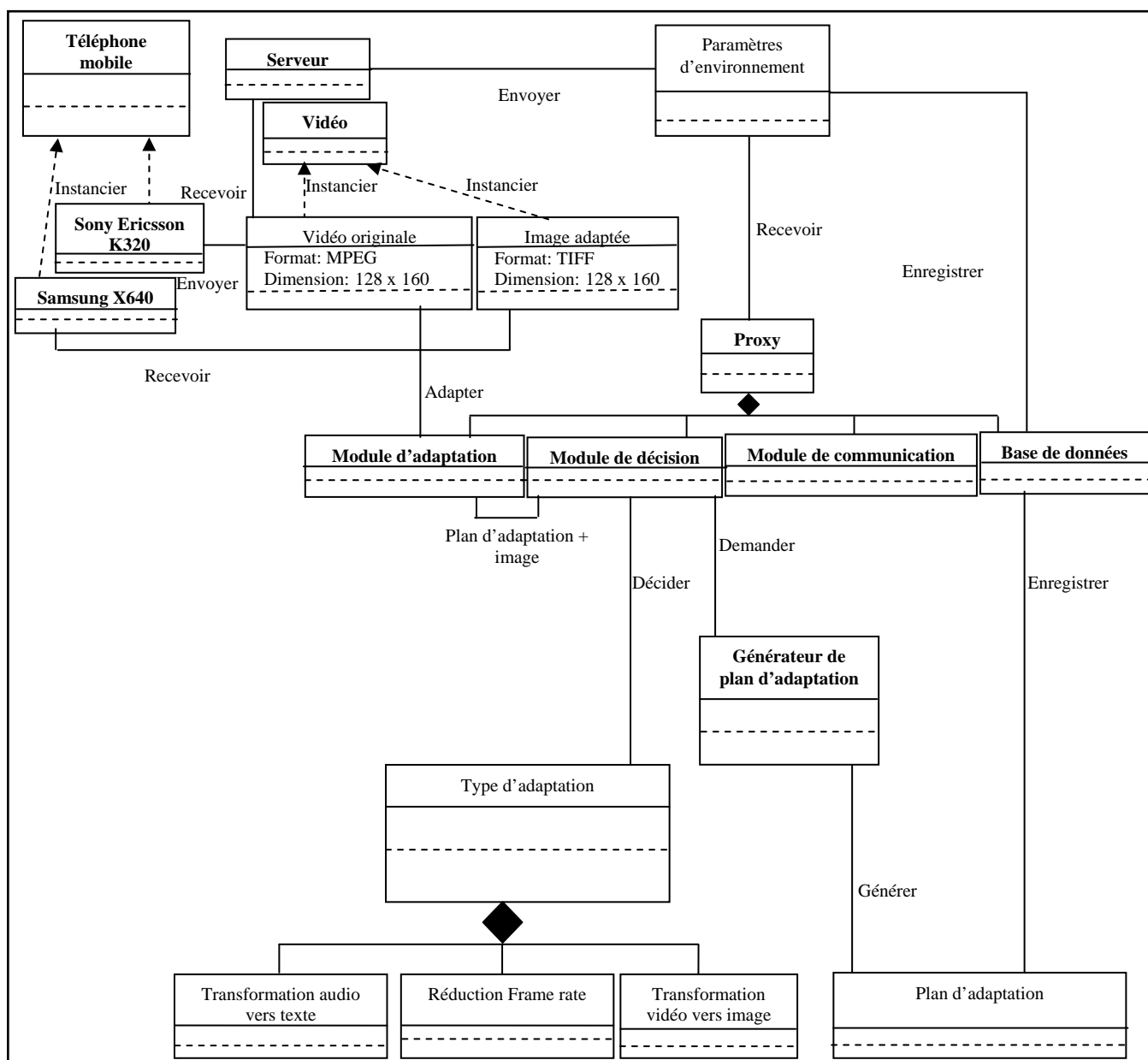


Figure 24 : Diagramme de classe du scénario 2 [48], [54].

C. Scénario 3

Le client multimédia receveur est le **Siemens CX65** qui ne supporte pas la vidéo enregistrée dans le format MPEG envoyée du téléphone mobile **Sony Ericsson K320**. Par conséquent, en a besoin d'une conversion du format de la vidéo.

D. Scénario 4

L'audio est enregistré dans le format .wav est envoyé par le client multimédia émetteur **Nokia N93i** au client receveur **Samsung SGHX640**. Dans ce cas, il est nécessaire de faire une conversion du format de l'audio pour être reçu par le téléphone mobile du client multimédia receveur [52], [53].

Par mesure de clarté, nous avons omis les attributs et les méthodes des classes.

Dans la base de données des téléphones mobiles, nous avons choisi cinq marques: **Nokia N93i**, **Siemens CX65**, **Sony Ericsson K320**, **Samsung SGH-X640** et **Nokia 2610**.

La figure 25 et la figure 26 représentent respectivement le diagramme de classe de la base de données multimédia et de la base de données des téléphones mobiles [48], [54].

Dans [52], nous avons cité la spécification complète de chaque marque de ces téléphones mobiles multimédia [48], [54].

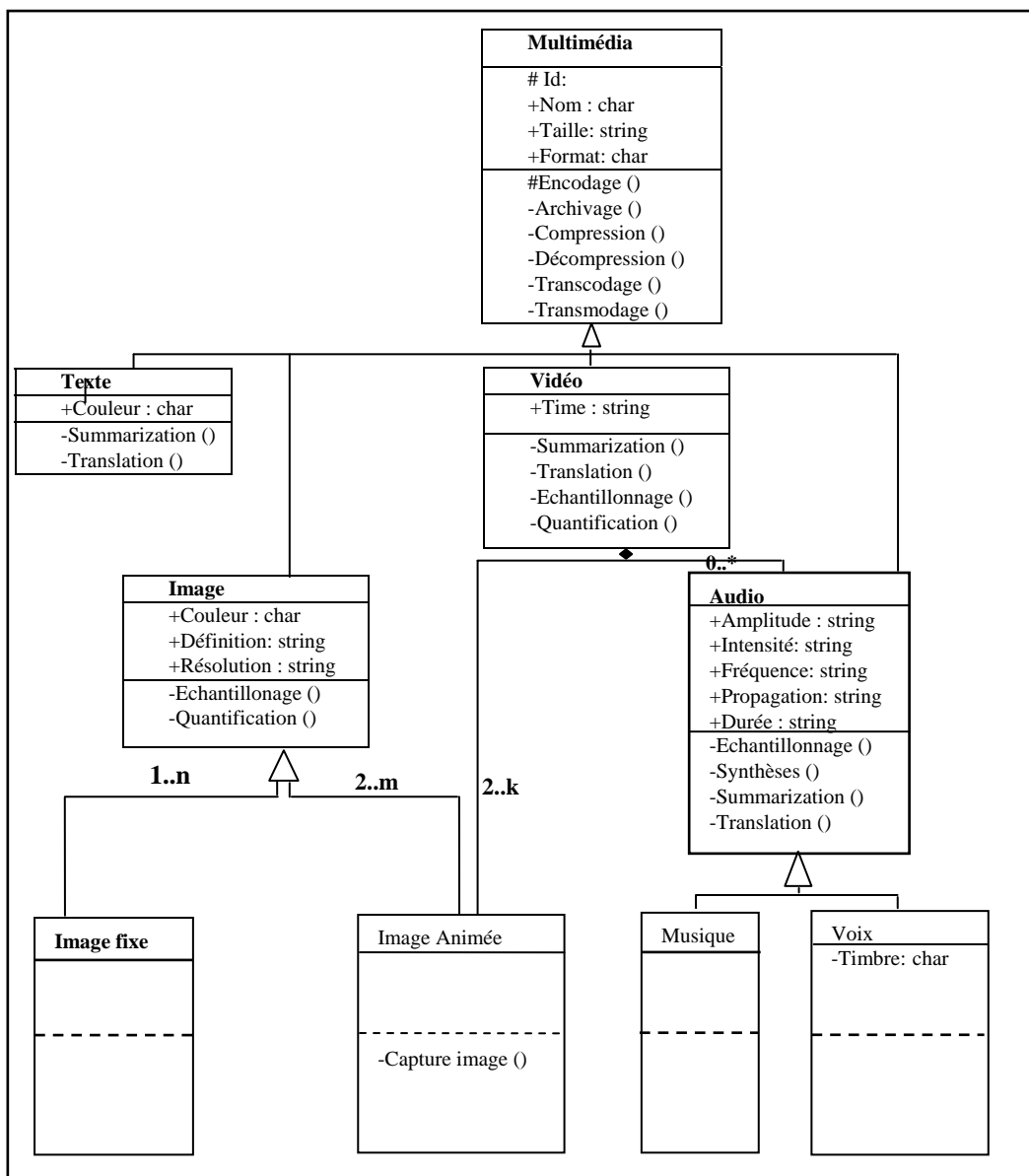


Figure 25 : Diagramme de classe de la base de données multimédia [48], [54].

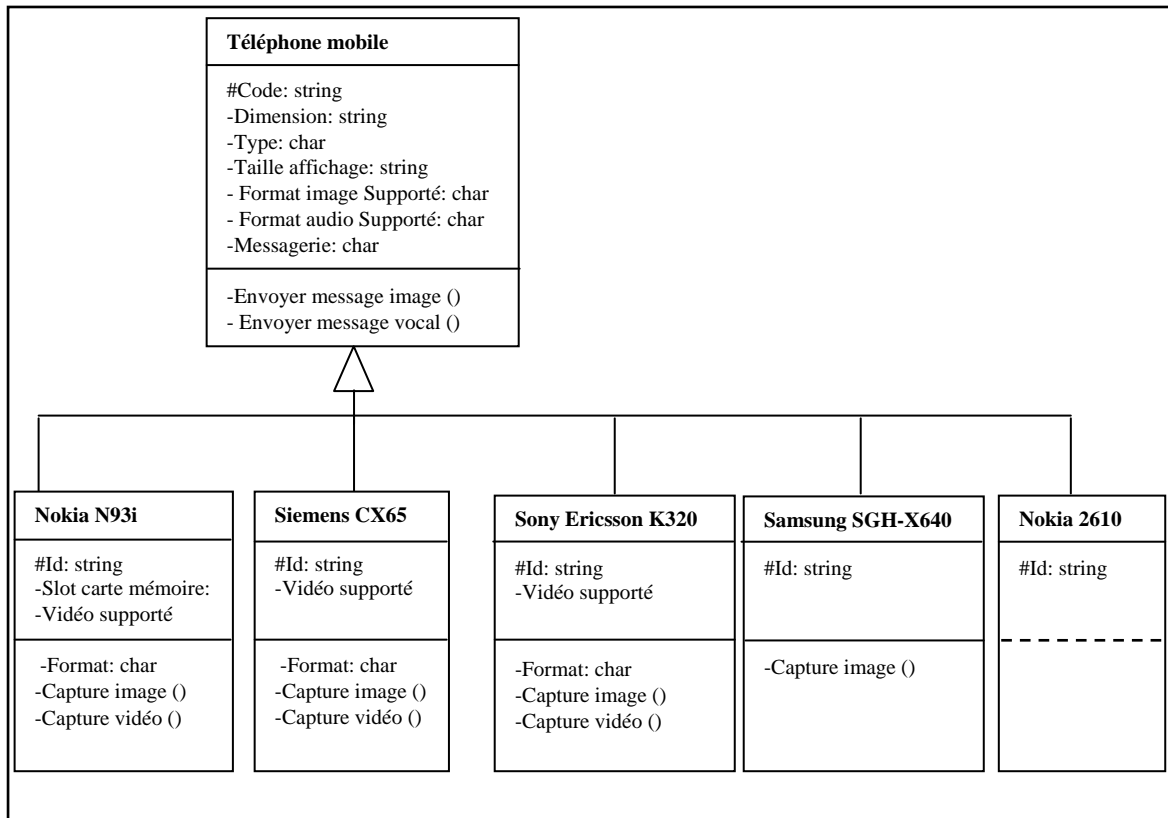


Figure 26 : Diagramme de classe de la base de données des téléphones mobiles [48], [54].

4.3. Conclusion

Dans ce chapitre, nous avons présenté la modélisation avec UML de la base de données multimédia en utilisant des scénarios d'adaptation.

Pour mieux expliquer la classe multimédia et la classe des téléphones mobiles, nous avons modélisé le diagramme de la classe multimédia et le diagramme de la classe téléphone mobile avec tout les attributs et les opérations et en particulier les opérations d'adaptation.

Dans la suite de ce travail, nous présentons l'implémentation des composants de notre architecture en donnant la priorité aux éléments du proxy.

CHAPITRE 5

«Implémentation de Prototype de la Base de Données du Proxy du Système Ouvert»

5.1. Introduction

Après avoir modélisé la base des scénarios, nous allons entamer dans le présent chapitre la réalisation de la base de l'historique des scénarios et de mise à jour de la base en cas de nouveaux événements : la réception de nouveaux paramètres de l'environnement qui vont représenter un nouveau scénario. Le proxy dans notre système doit demander au générateur de plan d'adaptation de générer la(es) opération(s) d'adaptation adéquate(s). Si les nouveaux paramètres d'environnement reçus par la base des scénarios existent déjà, la base fournit la(s) opération(s) nécessaire(s) à exécuter.

5.2. Environnement d'implémentation

Nous optons pour JBuilder Entreprise. Il est un environnement multi-plate-forme pour la construction d'applications Java pour l'entreprise. JBuilder 7 Enterprise simplifie le développement EJB, Web, XML et d'applications **de base de données** grâce à ses concepteurs visuels bidirectionnels et au déploiement rapide vers les principaux serveurs d'applications de la plate-forme J2EE notamment BEA WebLogic, IBM WebSphere, iPlanet, Oracle9i et Borland Enterprise Server intégré. Il permet aussi d'améliorer la productivité des développeurs et de tirer partie de la programmation extrême avec la visualisation du code UML, de développer et de déployer des applications sur les plates-formes Windows, Linux, Solaris et Mac OS [9].

5.2.1. Implémentation de la base de données du proxy

5.2.1.1. Création des bases de données

Afin de créer la base de données du proxy, nous avons utilisé le **JDataStore Explorer** qui permet d'examiner le contenu du JDataStore. Le répertoire du JDataStore est vu dans un arbre de contrôle avec chaque table et ces indexes regroupés ensemble. Quand on sélectionne un flux de données dans l'arbre, son contenu apparaît. Beaucoup d'opérations du JDataStore peuvent être accomplies sans écrire une ligne de code. Nous pouvons créer un nouveau fichier JDataStore, créer les tables et les indexes, importer des fichiers texte

délimités aux datasets, importer des fichiers comme étant des flots de fichier, supprimer des tables et des indexes, supprimer des datasets ou des autres flots de données et de vérifier l'intégrité du JDataStore. Il permet aussi de gérer les requêtes qui fournissent les données dans les datasets du JDataStore, d'éditer les datasets, etc. Nous pouvons administrer les propriétés de sécurité du JDataStore, à titre d'exemple les : utilisateurs, mots de passe et la cryptographie.

A. Création de la base des scénarios

La création des tables dans le **JdataStore Explorer** est très facile, dans le menu **File** du **JdataStore Explorer**, il ya **New table**, en donne un nom à la table et en localise l'emplacement de l'enregistrement de la table (**Table local**). La figure 27 représente une vue de la table des scénarios. Cette dernière est structurée sous forme de colonnes et de lignes. La première colonne contient le nom de la colonne (Column Name), la deuxième colonne contient le type de la colonne, etc.

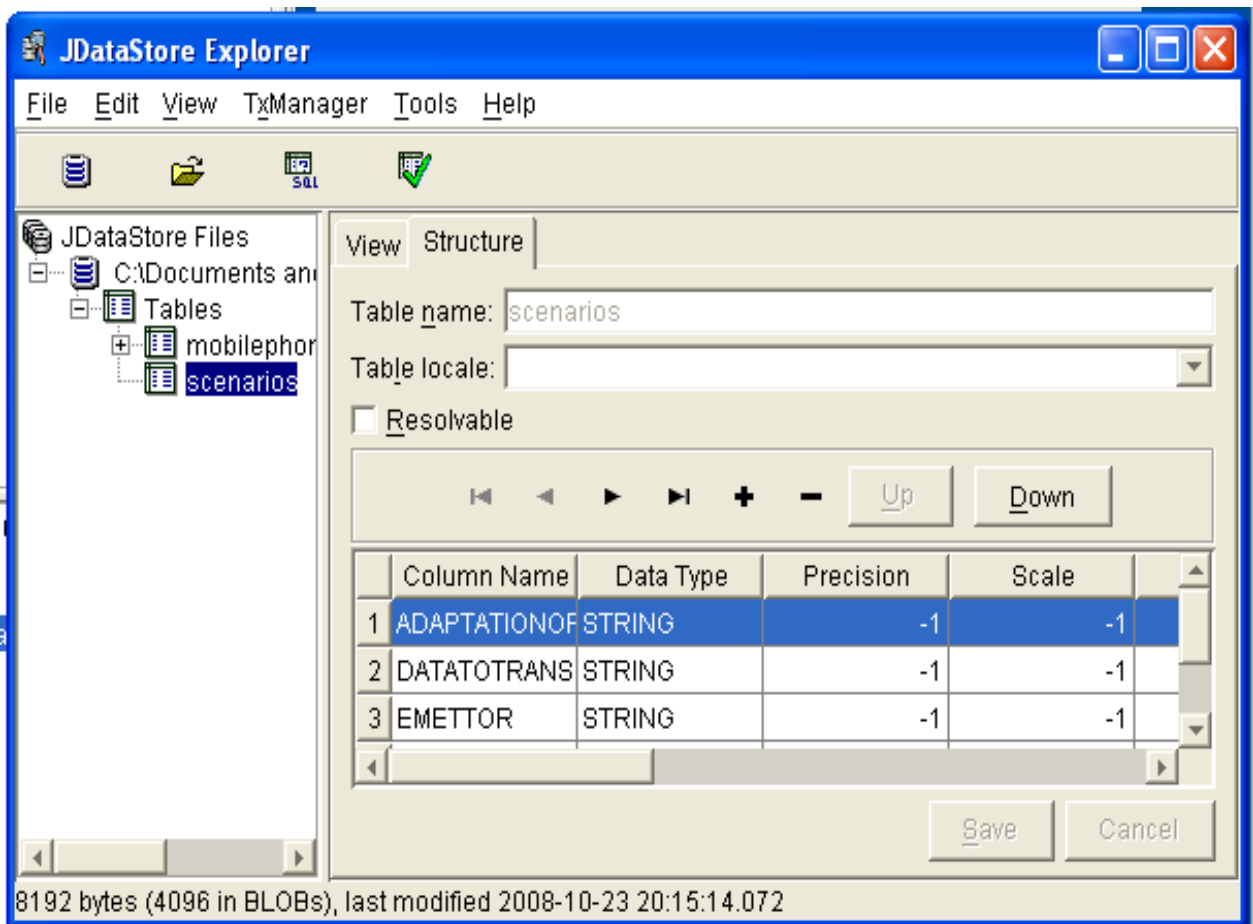


Figure 27 : Base des scénarios d'adaptation.

B. Création de la base des téléphones mobiles multimédia

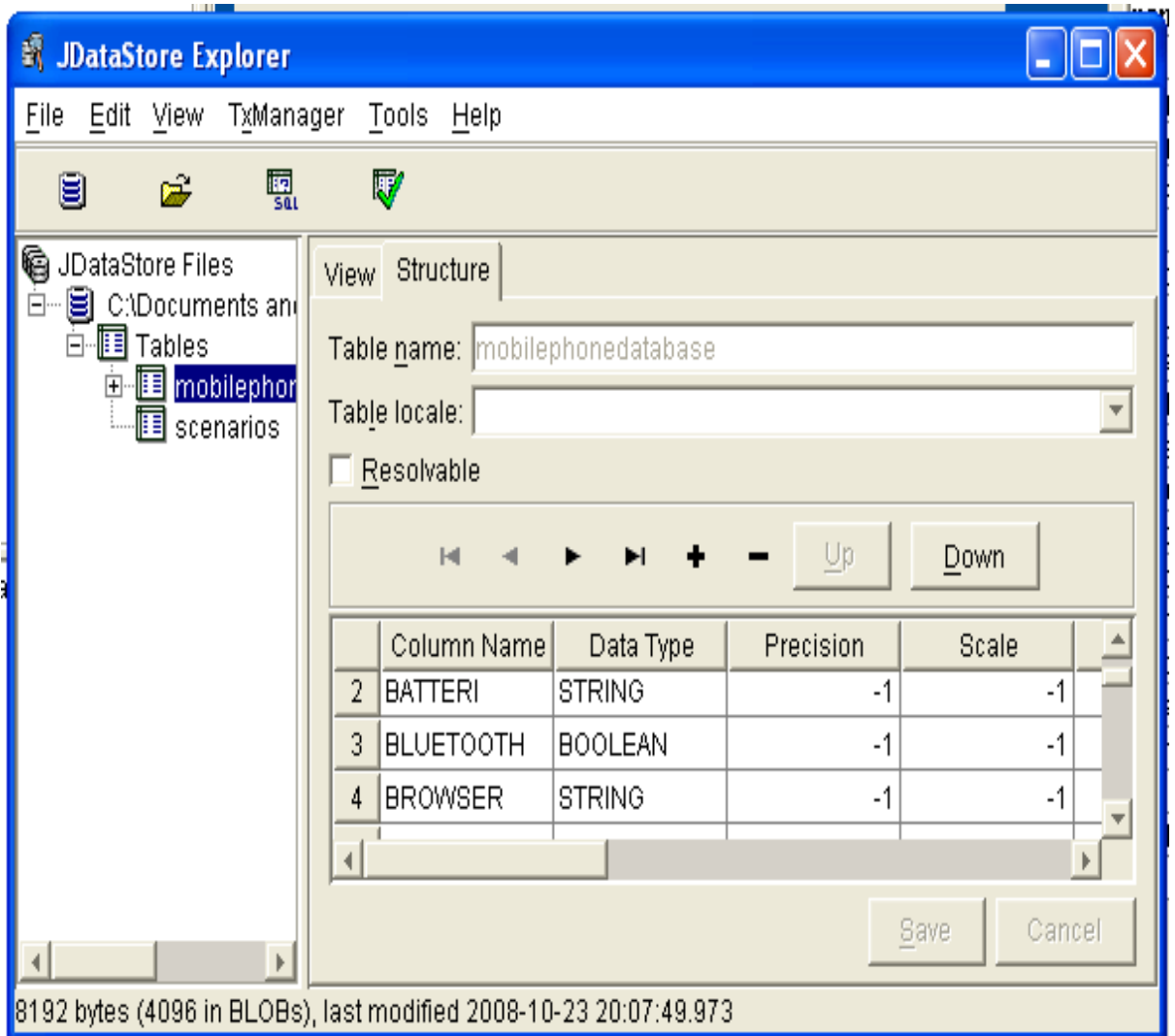


Figure 28 : Base des téléphones mobiles multimédia.

L'interface représentée dans la figure 28, montre la table des téléphones mobiles, le nom de la table est : mobilephonedatabase. Le nom de la première ligne de cette table (par exemple), est BATTERI, son type est STRING et ainsi de suite pour les autres lignes.

C. Création de la base des données multimédia

De la même manière en procède à la création de la table des informations multimédia. La figure 29 montre la base des données multimédia (image animée, vidéo, audio, etc.).

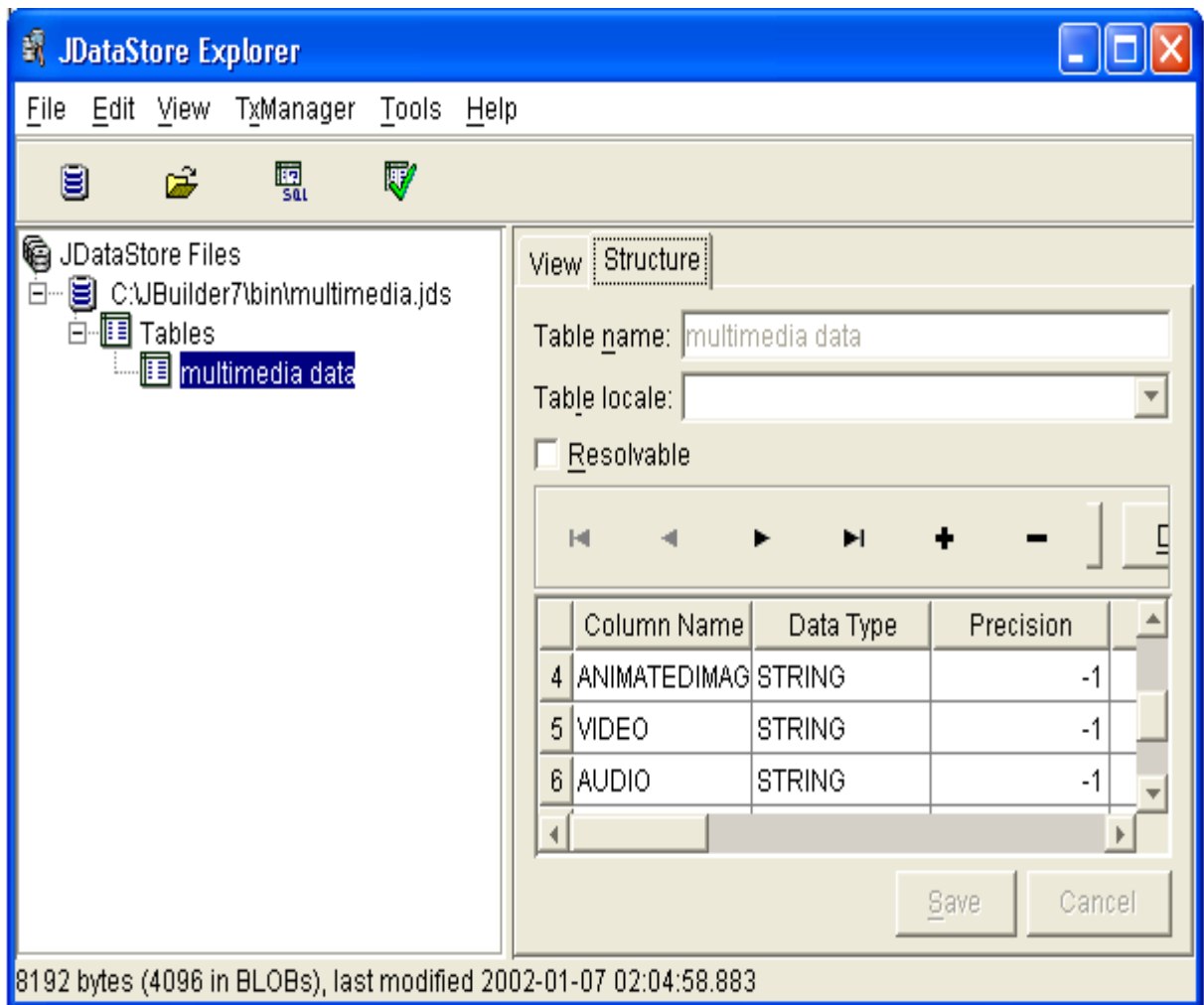


Figure 29 - Base des données multimédia.

5.3. Interrogation de la base du proxy

5.3.1. Affichage du plan d'adaptation existant

Avec l'arrivée des nouveaux paramètres de l'environnement (Emettor, Receiver et Data transmitted), la base du proxy vérifie dans la base des scénarios. Si les paramètres d'environnement ont été déjà reçus et traités c'est-à-dire que les paramètres existent déjà dans l'historique de la base des scénarios. La (s) opération(s) d'adaptation adéquate(s) est (sont) alors affichées (la(s) opération(s) qui a (ont) été(s) précédemment généré(s) par le système).

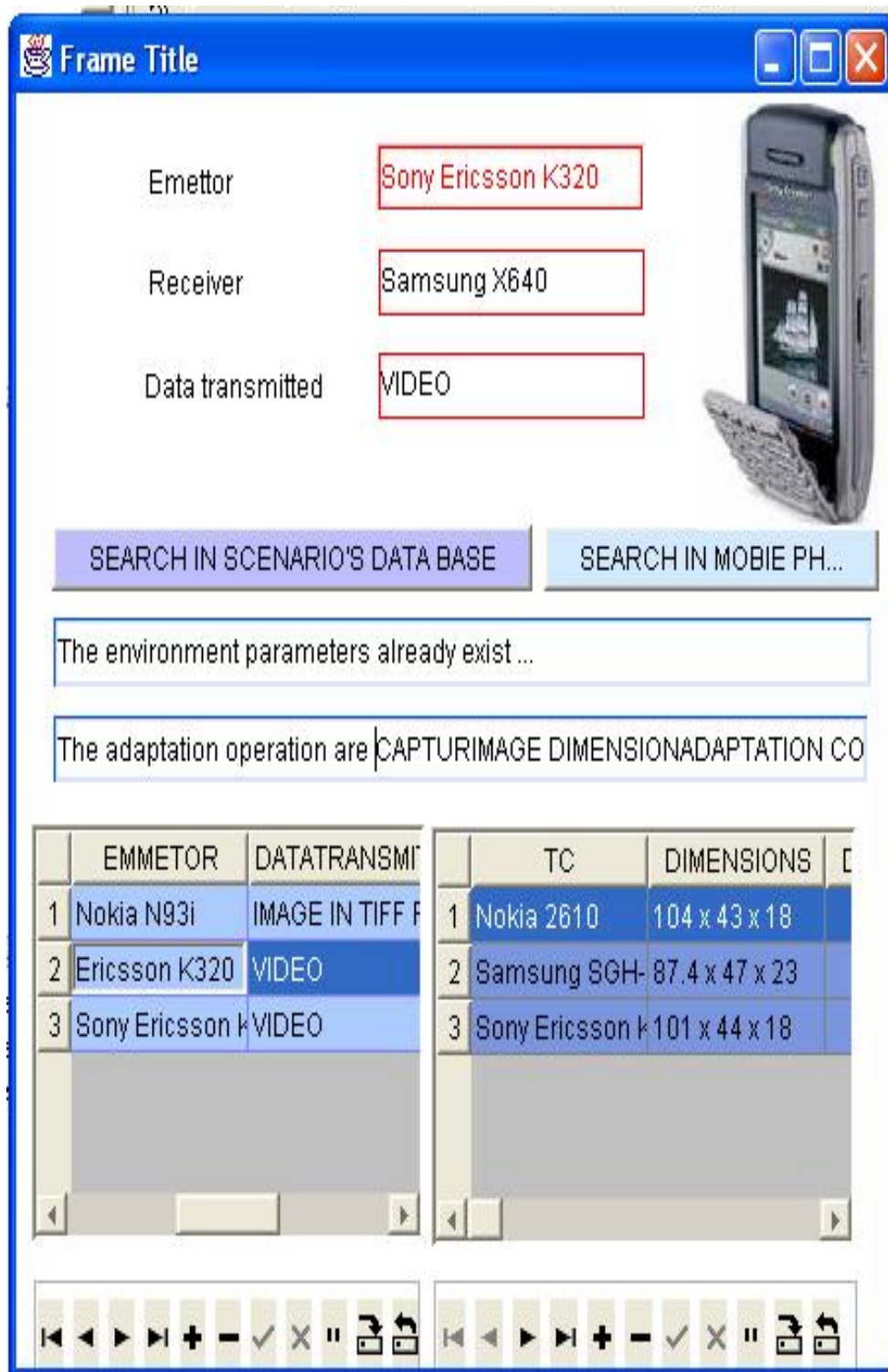


Figure 30 : Plan d'adaptation existant dans l'historique de la base du proxy.

5.3.1.2. Echange avec compatibilité des mobiles

Si les paramètres d'environnement ne sont conformes à aucun des paramètres existants dans la base des scénarios, seulement que le téléphone multimédia receveur existe déjà

dans la base de données des téléphones mobiles multimédia, une vérification des propriétés du téléphone multimédia receveur doit être faite pour savoir si les données peuvent être transmises sans nécessité d'adaptation c'est-à-dire que le receveur supporte les données qu'en souhaite transmettre. Un exemple de résultat de ce cas est représenté dans la figure 31.

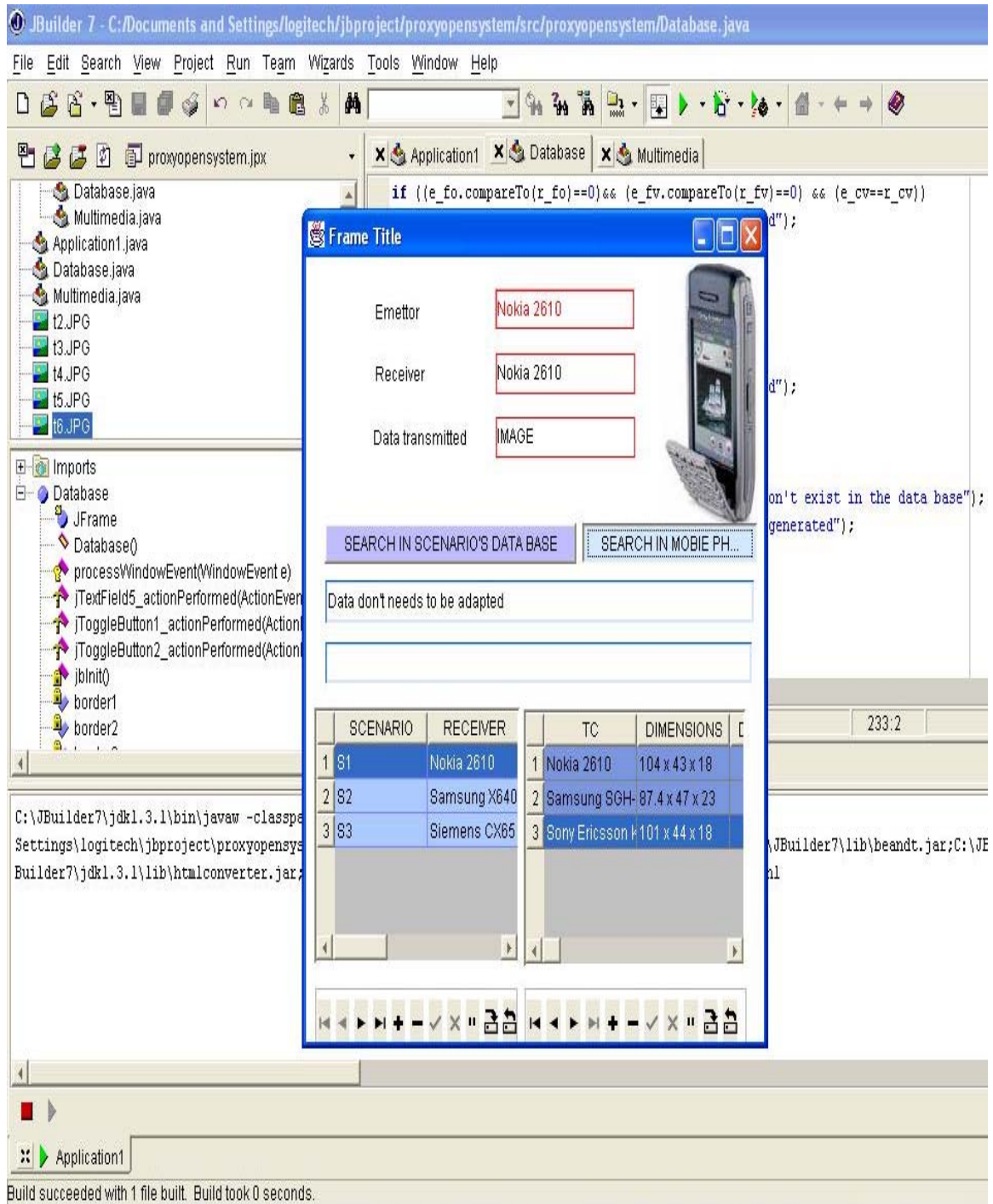


Figure 31 : L'adaptation multimédia n'est pas nécessaire selon la base des téléphones mobiles multimédia.

La figure 32, représente un exemple de cas où les paramètres de l'environnement saisis n'existent ni dans la base des scénarios ni dans la base des téléphones mobiles après avoir fait le test dans notre application.

5.3.2. Décision de la nécessité de la génération de plan d'adaptation

5.3.2.1. Test de la base des scénarios

Un autre cas où les paramètres d'environnement ne sont semblables à aucune des propriétés de la base de données des scénarios, un message apparaît informant que les paramètres d'environnement reçus n'existent pas, en attente que le générateur de plan d'adaptation génère la (es) opération(s) d'adaptation conforme au cas pour les ajouter avec leurs paramètres d'environnement dans la base des scénarios afin d'avoir un nouveau scénario et c'est de cette manière que la base du proxy est enrichie.

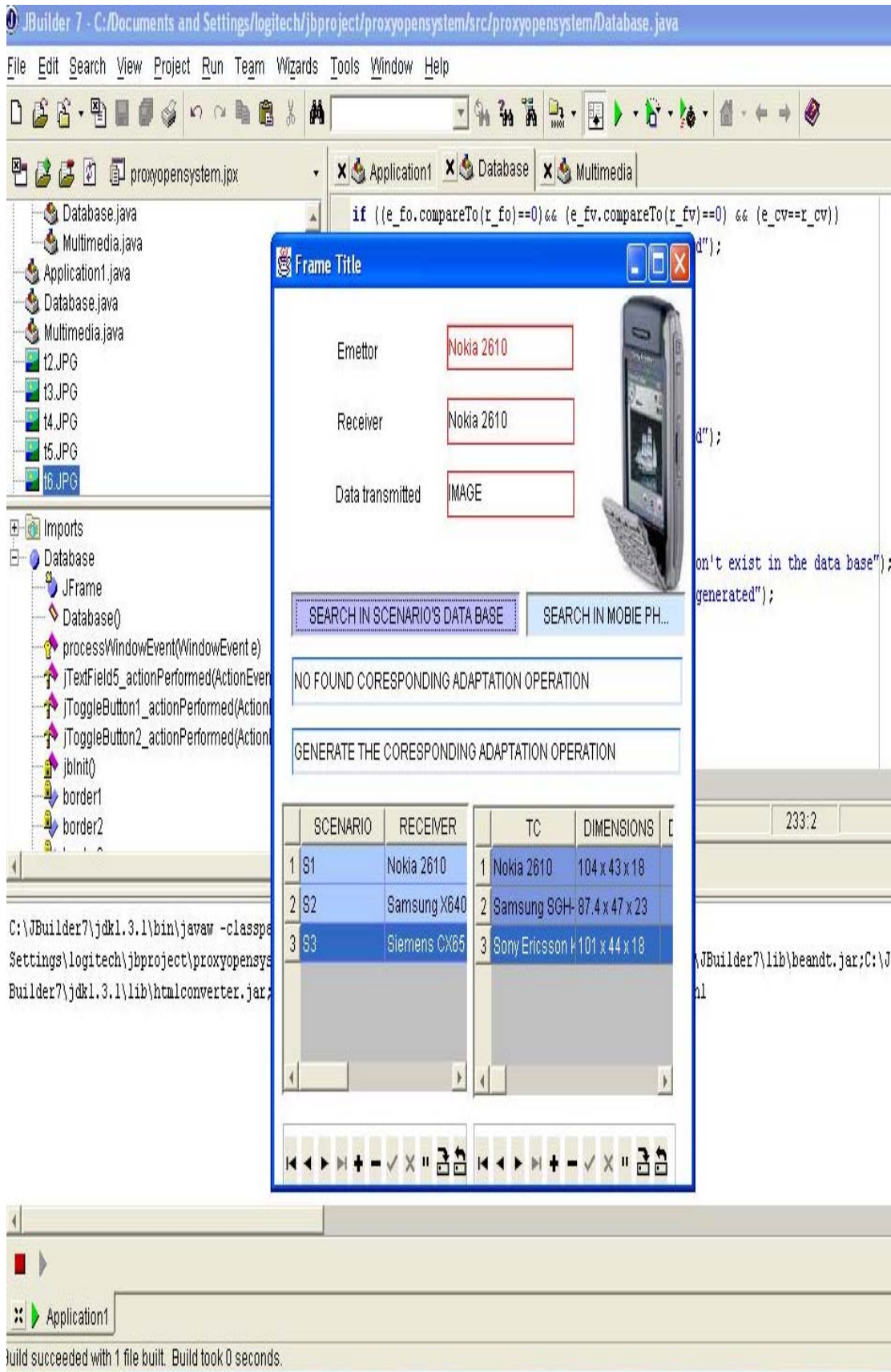


Figure 32 : L’adaptation multimédia est nécessaire (selon la base des scénarios).

5.3.2.2. Test de la base des téléphones mobiles

Si après avoir testé la base des scénarios on a trouvé que les nouveaux paramètres de l'environnement ne se trouvent pas dans notre base des scénarios, on est amené à tester leur apparence ou non dans la base des téléphones mobiles multimédia. La figure 33 représente un cas où les paramètres d'environnement saisis n'existent pas dans la base des téléphones mobiles. Par conséquent, une demande de génération du plan d'adaptation multimédia est forcément nécessaire.

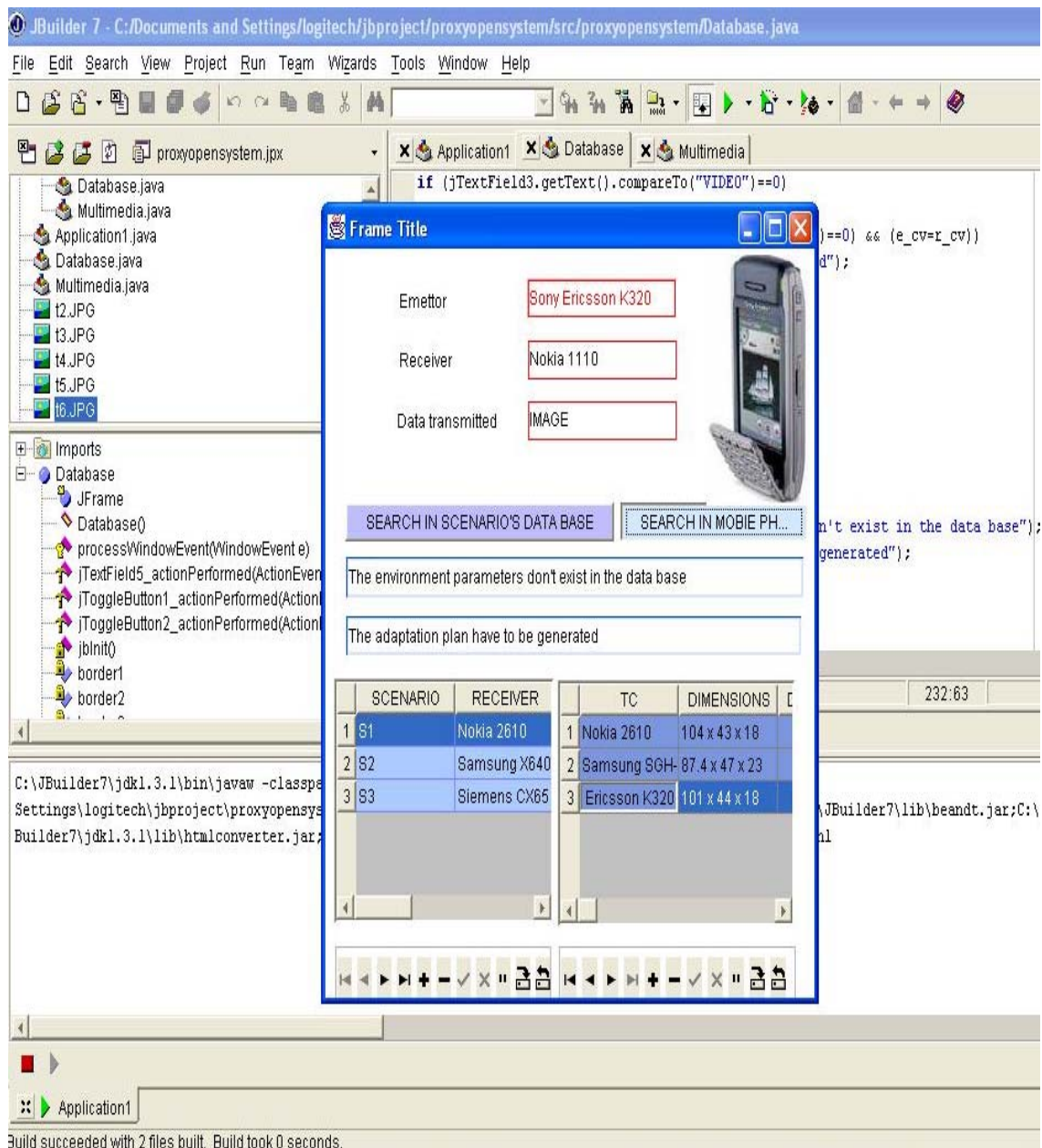


Figure 33 : Demande de génération du plan d'adaptation adéquat après avoir testé la base des téléphones mobiles.

5.4. Conclusion

Dans ce chapitre, nous avons présenté l'implémentation de la base de données du proxy de notre système ouvert pour l'échange de données multimédia entre les téléphones mobiles multimédia. Nous avons expliqué et présenté les interfaces du système qui gère la base du proxy qui contient : la base des scénarios et la base des téléphones mobiles.

Conclusion Générale

1. Bilan du travail

Le travail présenté dans ce mémoire s'inscrit dans le cadre de l'adaptation multimédia au profil des téléphones mobiles multimédia. Nous avons proposé un système ouvert d'adaptation multimédia pour l'échange de données multimédia entre les téléphones mobiles. L'architecture est basée sur le modèle Client\ Intermédiaire\ Serveur, où l'intermédiaire est un proxy. Le serveur et après avoir reçu un message du client multimédia émetteur, extrait les informations nécessaires du message reçu et les envoie au proxy comme étant des paramètres de l'environnement. Le proxy et le générateur de plan d'adaptation vont prendre en considération les capacités du téléphone mobile multimédia émetteur et celle du receveur ainsi que les étapes nécessaires pour faire le choix et l'exécution des opérations d'adaptation adéquates. En fin, les données adaptées seront transmises au client multimédia de la destination.

En modélisant cette architecture, nous avons essayé de mettre l'accent sur le comportement rationnel du proxy. Afin de valider les propriétés du système proposé et vérifier sa faisabilité, nous avons effectué une étude de cas dans le domaine des téléphones mobiles multimédia. Nous avons modélisé notre architecture à l'aide du langage UML, cette modélisation comprend trois principaux diagrammes (modèles) : *des scénarios*, *des téléphones mobiles multimédia* et *des données multimédia*. Notre modélisation (diagramme de classe) se base essentiellement sur les opérations de composition d'instanciation et de généralisation. Notre premier et primordial modèle qui est la base des scénarios prend en compte le comportement général du système suite à l'envoi de données multimédia d'un client multimédia émetteur. Les deux autres diagrammes (modèles) des téléphones mobiles multimédia et des données multimédia, considèrent la généralisation de la classe téléphone mobile en plusieurs autres classes et la généralisation de la classe multimédia aussi.

Afin de valider les fonctions assurées par le système proposé et vérifier la faisabilité de ce dernier, nous avons implémenté comme premier pas, la base des scénarios du proxy en s'appuyant sur l'étude des cas présentés. L'implémentation de la base des scénarios est faite par JDataStore Explorer qui est un composant du langage JBuilder Entreprise. La connexion entre la base du proxy et JBuilder est réalisée à l'aide des composants DataSet et DataQuery du DataExpress.

Conclusion générale

Ainsi, nous avons pu montrer que notre architecture d'adaptation multimédia pour l'échange des données multimédia entre les téléphones mobiles multimédia présente des avantages aux clients multimédia.

Généralement, les architectures d'adaptation existantes traitent les données multimédia envoyées par la machine serveur au terminal du client mais dans notre architecture, le traitement d'adaptation est appliqué aux données multimédia dès leur émission du client émetteur multimédia vers le client multimédia récepteur grâce au proxy.

Perspectives

Dans ce travail, nous avons proposé un système ouvert pour l'adaptation des données multimédia. Comme déjà vu, notre architecture comprend trois composants principaux: les clients (émetteur et receveur), le serveur et le proxy. Nous avons réussi après avoir modélisé quelques cas (scénarios), d'implémenter notre base du proxy.

Nous comptons compléter notre système ouvert d'adaptation multimédia au profit des téléphones mobiles.

Afin de réaliser l'architecture d'adaptation proposée dans sa globalité (après avoir réalisé chaque composant), nous avons la perspective de tester s'il est intéressant ou non d'utiliser les threads, pour cela il faudra que nous déterminions:

- ❖ le nombre de threads nécessaires pour réaliser notre système ;
- ❖ La fonction de chaque thread ;
- ❖ La relation entre les threads.

En étudiant la présence de potentiel de parallélisme entre les threads, il est intéressant d'utiliser les threads pour des raisons de performances.

Dans le cas contraire (la relation dominante entre les threads est la précédence), il n'est pas intéressant d'utiliser les threads. Dans ce cas, la programmation classique convient.

Références bibliographiques

- [1] Hagos B. “Accès et adaptation de document multimédia pour les systèmes pervasifs”, thèse de doctorat, L’institut National des Sciences Appliquées, Lyon, France, 25 septembre 2006.
- <http://docinsa.insa-lyon.fr/these/2006/girma/these.pdf>
- [2] Kazi-Aoul Z., Demeure I. et Moissinac J C. “Une architecture générique pour la fourniture de services multimédia adaptables - illustration par un scénario”, Ubimob, Nice, France, juin 2004.
- <http://www.infres.enst.fr/~demeure/PUBLIS/Ubimob'04.pdf>
- [3] Lemlouma T., Cécile R. et Layaida N. “Architecture de Négociation et d’Adaptation de Services Multimédia dans des Environnements Hétérogènes”, thèse de doctorat, L’institut National Polytechnique de Grenoble, France, juin 2004.
- [4] Laborie S., Jérôme E. et Layaida N. “Adapter temporellement un document SMIL”, AFIA, Nice (FR), France, pp. 47-58, 2005.
- [5] Laborie S., Jérôme E. et Layaida N. “Adaptation spatiale efficace de documents SMIL Effective spatial adaptation of SMIL documents”, AFIA-AFRIF, Tours (FR), pp. 127, 2006.
- [6] Jérôme E., Layaida N. et Victor D. “A semantic framework for multimedia document adaptation”, Proceeding of 18th International Joint Conference on Artificial Intelligence (IJCAI), Acapulco (MX), San-Mateo (CA US), pp. 31-36, 2003.
- [7] Kazi-Aoul Z., Demeure I. et Moissinac J C. “Une architecture de fourniture de documents multimédia adaptables dans une logique pair à pair”, France, 06 Décembre 2005.
- <http://adapt.asr.cnrs.fr/reunions/20051206/slide>
- [8] Hagimont D. et Layaida N. “Adaptation d’une application multimédia par un code mobile”, Technique et Science Informatique (TSI), numéro spécial “Agents et code mobile”, Vol 21, France, pp. 6, 2002.
- <http://wam.inrialpes.fr/people/layaida/research/tsi2002/tsi2002.pdf>

- [9] [http://www.ciao.fr/JBuilder Enterprise Edition 7 Ensemble de mise a niveau de ve
rsion_produit_425950](http://www.ciao.fr/JBuilder_Enterprise_Edition_7_Ensemble_de_mise_a_niveau_de_version_produit_425950)
- [10] Kate W. T., Deunhouwer P., Clout R. et Timesheets. “Integrating Timing in XML”, Proceedings WWW9 Workshop: Multimedia on the Web, Amsterdam, Hollande, 15 Mai 2000.
- [11] Synchronized Multimedia Working Group du W3C, Synchronized Multimedia Integration language (SMIL 2.0), W3C Recommendation,
<http://www.w3.org/TR/smil20/>, 7 Aout 2001
- [12] Dalal M., Feiner S., McKeown K., Pan S., Zhou M. X., Hollerer T., Shaw J., Feng Y. et Fromer J. “Negotiation for automated generation of temporal multimedia Presentations”, ACM Multimédia conférence, Boston, MA, US, pp. 55-64, 1996.
- [13] Layaida N., Madeus. “Système d’édition et de présentation de documents structurés multimédia”, Thèse de doctorat en Informatique, Université Joseph Fourier, France, Juin1997.
- [14] Burning A., Lin R. et Marriot K. “Constraint-based Document Layout for the Web”, Multimedia Systems, USA, ISSN: 0942-4962, Vol. 8, No.3, pp. 177- 189, 2000.
<http://www.csse.monash.edu.au/~marriott/BorLinMar00.pdf>
- [15] Borning A., Lin R. et Marriott K. “Constraints for the Web”, Proceedings of the fifth ACM international conference on Multimedia, USA, Seattle, pp. 173-18, Novembre 1997.
- [16] Hagos B., Lionel B. et Pierson J.M. “Planning-Based multimedia Adaptation Services Composition for Pervasive Computing”, SITIS, 2006.
<http://www.u-bourgogne.fr/SITIS/06/Proceedings/IMRT/f57.pdf>
- [17] Mandato BRAIN D. “Concepts for Service adaptation, scalability and QoS handling on mobility enabled networks”, IST-1999-10050 project, deliverable 1.2., 31 Mars 2001.
<http://jungla.dit.upm.es/~ist-mind/publications/MAN02s.pdf>

- [18] Marquet B., Gustave C., Lefebvre A., Nemchenko S. et Chassande-Barrioz S. “Secured services in a multi-tier architecture”, World Telecommunications Congress, WTC 2002, Paris, Septembre 2002.
- [19] Panagiotakis S., Houssos N. et Alonistioti N. “Integrated generic architecture for flexible service provision to mobile users”, PIMRC 2001, San Diego, USA, 30 Septembre- 2 Octobre 2001.
- [20] Böszörményi, Hellwagner H., Kosch H., Libsie M. et Podlipnig S. “Metadata driven adaptation in the ADMITS project”, EURASIP Signal Processing: Image Communication Journal, Vol. 18, No. 8, pp. 749-766 , Septembre 2003 (Special Issue on Multimedia Adaptation).
- [21] <http://isis.rd.francetelecom.com/>
- [22] Kim J.-G. et Chang S.-F. “Content-Adaptive Utility Based Video Adaptation”, icme, International Conference on Multimedia and Expo - Volume 3 (ICME '03), pp. 281-284, 2003.
- [23] Fouial O. et Demeure I. “Fourniture de Services Adaptatifs dans les Environnements Mobiles”, Journées sur les systèmes à composants adaptables et extensibles, Poster, Grenoble, France, 17-18 Octobre 2002.
- [24] Kazi-Aoul Z., Ferraz S., Fouial O. et Demeure I. CAAS. “An architecture for component based adaptable service provision”, 2nd ANWIRE Workshop on Wireless, Mobile & Always Best Connected, Mykonos, Greece, 25-26 Septembre 2003.
- [25] SOAP Version 1.2 Part 1: Messaging Framework, W3C Recommendation,
[http : //www.w3.org/TR/2003/REC-soap12-part1-20030624/](http://www.w3.org/TR/2003/REC-soap12-part1-20030624/), 24 Juin 2003.
- [26] Lemlouma T. et Layaïda N. “Context-Aware Adaptation for Mobile Devices”, IEEE International Conference on Mobile Data Management, Berkeley, California, USA, pp. 106-111, 19-22 Janvier 2004.
- [27] XQuery 1.0: An XML Query Language. W3C Working Draft,
[http ://www.W3.org/TR/xquery/](http://www.W3.org/TR/xquery/), 22 Aout 2003.
- [28] IETF. RTP: A Transport Protocol for Real-Time Applications. Internet Draft, version 8, revision of RFC 1889, Juillet 2000.

- [29] Lemlouma T. et Layaïda N. “NAC: A Basic Core for the Adaptation and Negotiation of Multimedia Services”, OPERA Project, INRIA Rhône Alpes, Septembre 2001.
- [30] Lemlouma T. “Adaptation et négociation de contenus”, Équipe WAM, INRIA Rhône-Alpes journée GT Documents multimédia - Groupe de travail 3.3, GDR I3, Grenoble, France, 10 décembre 2004.
- [31] Margaridis M. et Poloyzos G.C. “Adapatation Technique for Ubiquitus Internet Multimedia”. Wireless Communication and Mobile Computing, John Wiley & Sons, Vol. 1, No.2, pp. 141-164, 2001.
- [32] Mogul J.C. “Server-Directed Transcoding”. Computer Communications, Vol.24, No.2, pp. 155-162, Février 2001.
- [33] Mhan R., Smith J.R. et Chung-Sheng Li. “Adapting Multimedia Internet Content for Universal Access”, IEEE Trans On Multimedia, ISSN: 1520-9210, Vol.1, No.1, pp. 104-114, 1999.
- [34] Bharghavan V. et Gupta V. “Framework for Application in Mobile Computing Environment”, In processing of the IEEE composec’97, Washington, D.C, USA, 11-15 Août 1997.
- [35] Bolot J.C. et Turlitti T. “Experience with rate control mechanisms for packet video in the Interet”, Computer Communication Review, ACM, Vol. 28, No.1, pp. 4-15 Janvier 1998.
- [36] Marriott K., Meyer B. et Tardif L. “Fast and Efficient client-Side Adaptivity for SVG”, ACM press: In proc. Of WWW 20002, Honolulu, Hawaii, USA, pp. 496-507, Mai 2002.
- [37] Lei Z. et gerganas N.D. “Context-besed Media Adaptation in Pervasive Computing”, In Proc. Of Canadiau Conference on Electronic and Computer Engineering (CCECE), Toronto, Mai 2001.
- [38] Bjork S., Hoimquist L.E., Redstrom J. et WEST Al. “a Web browser for small terminals”, In Proceedings of the 12th annual ACM symposium on user interface software and technology, Asheville, North Carolina, USA, pp. 187-196, 17-10 Novembre 1999.

- [39] Yoshikawa C., Chun B., Eastman P. et al. "Using smart clients to build scalable services", In Proc. Winter 1997 Usenix Technical Conf, Anaheim, California, USA, pp. 105-117, 6-10 Janvier 1997.
- [40] Fisher B., Agelidis G., Dill J. et al. "Fis-Eye Views for Visualizing the World-Wide Web", Proc. Of the 7th Int. Conf. on Human-Computer Interaction (HCI International '97), Elsevier, Amsterdam, Pays-Bas, pp. 719-722, 1997.
- [41] Singh A., Trivedi A., Ramamritham K. et al. "Proxies that Transcode and Cache in Heterogeneous Web Client Environments", World Wide Web Journal, Vol.7, No.1, pp. 7-28, 2004.
- [42] Wee S. et Apostolopoulos J. "Secure scalable streaming and secure transcoding with JPEG-2000", IEEE International Conference on Image Processing (ICIP), Barcelona, Spain, Vol.1, ISBN: 0-7803-7750-8, pp. 205-212, 14-17 Septembre 2003.
- [43] Kim J.G., Wang Y. et Chang S.F. "Content-adaptive utility-based video adaptation", In Proc, of IEEE Int'l Conference on Multimedia & Expo(ICME), Baltimore, MD, pp. 281-284, 6-9 Juillet 2003.
- [44] Chandra S., Ellis C.S. et Vahdat A. "Differentiated Multimedia Web Services Using Quality Aware Transcoding", 19th Annual Joint of the IEEE Computer and Communication Societies (Infocom 2000), Israel, pp. 961-969, Mars 2000.
- [45] Fox A., Gribble S., Brewer E. et Amir E. "Adapting to Network and Client Variation via On-Demand, Dynamic Distillation", In Processing ASPLOS-VII, Cambridge, MA, USA, pp. 160-170, Octobre 1996.
- [46] Bickmore T., Girgenson A. et Sullivan J.W. "Web Page Filtering and Re-Authoring for Mobile Users", In The Computer Journal, Vol. 42, No.6, pp. 534-546, 1999.
- [47] Noble B.D., Satyanarayanan M., Narayanan D. et al. "Agile application-aware adaptation for mobility", In Proceedings of the 16th ACM Symposium on Operating Systems Principles, Vol. 31, No.5, pp. 276-287, Décembre 1997.
- [48] Saighi A., Goualmi-Zine N. et Roose P. "Modeling Scenarios for the Open System for Multimedia Mobile phone Exchange", ICEEDT'08, Hammamet, Tunisie, 8-10 Novembre 2008.
- [49] <http://www.mobilemultimedia.be/fr/index.php>

- [50] Shanableh T. et Ghanbari M. "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats", IEEE Trans. Multimedia, Vol. 2, pp. 101-110, Juin 2000.
- [51] Lei Z. "Media transcoding for pervasive computing", In: Proceedings of the ninth ACM international conference on Multimedia, ACM Press, Vol. 9, Ottawa, Canada, ISBN 1-58113-394-4, pp. 459-460, 2001.
- [52] Saighi A., Goualmi-Zine N. et Roose P. "Towards an Open system for Multimedia Mobile Phone Exchange: adaptation architecture", Caise-DC'08, CEUR-WS, Montpellier , France, ISSN 16130073, Vol. 343, pp. 53-61 , 16-17 Juin 2008.
- [53] Saighi A., Goualmi-Zine N. et Roose P. "Towards an open exchange system for multimedia mobile phone ", SSNE'08, Alger, Algérie, pp. 50, 5-6 Mai 2008.
- [54] Saighi A., Goualmi-Zine N. et Roose P. "Modeling the proxy's data base for a multimedia adaptation open system", JSIO'08, Oran, Algérie, pp. 53-60, 16-17 Décembre 2008.

Références Bibliographiques