La plate-forme RAMSES pour un triple écran interactif : application à la génération automatique de télévision interactive

Julien Royer

Ecole Doctorale EDITE

Thèse présentée pour l'obtention du diplôme de
DOCTEUR DE L'INSTITUT NATIONAL DES TELECOMMUNICATIONS

Doctorat délivré conjointement par
L'Institut National des Télécommunications et l'Université Pierre et Marie Curie – Paris 6

Spécialité : Informatique, télécommunications et électronique

Par
Julien Royer

La plate-forme RAMSES pour un triple écran interactif : application à la génération automatique de télévision interactive

Soutenue le 16 décembre 2009 devant le jury composé de :

Patrick Gallinari	Président
Claude Timsit	Rapporteur
Azeddine Beghdadi	Rapporteur
Bruno Aïdan	Examinateur
Olivier Martinot	Examinateur
Gérard Mozelle	Examinateur
Françoise Prêteux	Directeur de thèse

Thèse n°2009TELE0020

Remerciements

En premier lieu, je souhaite exprimer au professeur Françoise Prêteux, mon directeur de thèse, l'expression de mes chaleureux remerciements pour sa disponibilité au cours des moments clés de cette thèse ainsi que pour les élans qu'elle a su donner à mes recherches tant lors des phases d'état de l'art que dans l'élaboration des modèles et concepts de l'architecture ou encore dans la valorisation des résultats obtenus. RAMSES lui doit beaucoup et moi également.

J'adresse à Olivier Martinot mon encadrant au sein d'Alcatel-Lucent Bell Labs toute ma reconnaissance. C'est lui qui m'a proposé le premier cette opportunité d'effectuer une thèse dans un domaine aussi riche et dynamique que celui de l'interactivité dans les documents multimédias. Je le remercie vivement pour son soutien permanant durant cette thèse en me laissant à la fois une grande liberté dans la gestion de ma progression et en m'apportant son aide dans les phases décisives.

Bien sûr, je réserve des remerciements tout particuliers au professeur Claude Timsit et au professeur Azeddine Beghdadi qui ont accepté la lourde tâche d'examiner mes travaux. Leurs commentaires stimulants et l'intérêt qu'ils ont porté à ce mémoire ont enrichi la présentation du document. Que le professeur Patrick Gallinari trouve ici l'expression de mes remerciements pour l'honneur qu'il m'a fait en présidant ce jury.

Je remercie toute l'équipe hypermédia application qui a su m'épauler tant dans les moments difficiles que durant les collaborations qui ont été mises en place pour permettre un bon échange des informations. Je remercie notamment Hang Nguyen pour son encadrement au cours de la première année ainsi que Emmanuel Marilly et Abdelkader Outtagarts pour leur participation à la synthétisation des travaux effectués et leur soutien dans la rédaction de ce document. Je remercie plus particulièrement Alexandre Vanbelle qui m'a apporté une aide précieuse notamment pour la concrétisation des modèles à travers l'implantation d'un prototype.

Que l'équipe du département ARTEMIS de Télécom SudParis soit assurée de toute ma sympathie pour son accueil chaleureux et sa disponibilité pour répondre à mes questions sur MPEG. En particulier, un grand merci à Marius Preda et Titus Zaharia.

Une dernière pensée émue à mes parents qui m'ont toujours soutenu dans mes projets ; même s'ils m'ont fait part, au début de cette thèse, de leurs interrogations sur l'intérêt de « continuer les études » plutôt que d'aller « travailler ».

Enfin, je remercie d'une façon plus amicale tous ceux qui m'ont « soutenu » et « supporté » durant toute la durée de cette thèse et en particulier ceux présents durant la phase de rédaction.

Résumé

Avec la révolution du numérique, l'usage de la vidéo a fortement évolué durant les dernières décennies, passant du cinéma à la télévision, puis au web, du récit fictionnel au documentaire et de l'éditorialisation à la création par l'utilisateur. Les médias sont les vecteurs pour échanger des informations, des connaissances, des « reportages » personnels, des émotions... L'enrichissement automatique des documents multimédias est toujours un sujet de recherche depuis l'avènement des médias.

Dans ce contexte, nous proposons dans un premier temps une modélisation des différents concepts et acteurs mis en œuvre pour analyser automatiquement des documents multimédias afin de déployer dynamiquement des services interactifs en relation avec le contenu des médias. Nous définissons ainsi les concepts d'analyseur, de service interactif, de description d'un document multimédia et enfin les fonctions nécessaires pour faire interagir ceux-ci. Le modèle d'analyse obtenu se démarque de la littérature en proposant une architecture modulaire, ouverte et évolutive.

Nous présentons ensuite l'implantation de ces concepts dans le cadre d'un prototype de démonstration. Ce prototype permet ainsi de mettre en avant les contributions avancées dans la description des modèles. Une implantation ainsi que des recommandations sont détaillées pour chacun des modèles. Afin de montrer les résultats d'implantation des solutions proposées sur la plateforme telles que les standards MPEG-7 pour la description, MPEG-4 BIFS pour les scènes interactives ou encore OSGI pour l'architecture générale, nous présentons différents exemples de services interactifs intégrés dans la plateforme. Ceux-ci permettent de vérifier les capacités d'adaptation aux besoins d'un ou plusieurs services interactifs.

L'implantation de la chaîne complète d'automatisation de la génération de médias enrichis depuis l'accès aux médias jusqu'à leur distribution vers l'utilisateur final est validée à travers un service interactif d'informations additionnelles en temps réel sur les personnes présentes (dans la scène) dans le programme TV « Les travaux de l'Assemblée Nationale » sur la chaîne « La Chaîne Parlementaire ».

Abstract

Interactive media technologies which emerge from the convergence of telecommunications and multimedia are now considered as a characteristic of new enriched media. Multimedia analysis is a key element to select, build and adapt interactive scenes relatively to multimedia content.

The concept developed in this thesis is to propose an architecture model allowing automatic multimedia analysis and inserting pertinent interactive contents accordingly to multimedia content.

Until nowadays, studies are mainly trying to provide tools and frameworks to generate a full description of the multimedia. It can be compared as trying to describe the world since the system must have huge description capabilities. Actually, it is not possible to represent the world through a tree of concepts and relationships due to time and computer limitations.

Therefore, according to the amount of multimedia analyzers developed all over the world, this thesis proposes a platform able to host, combine and share existing multimedia analyzers. Furthermore, we only consider user's requirements to select only required elements from multimedia platform to analyze the multimedia.

In order to easily adapt the platform to the service requirements, we propose a modular architecture based on plug-in multimedia analyzers to generate the contextual description of the media. Besides, we provide an interactive scene generator to dynamically create related interactive scenes. We choose the MPEG-7 standard to implement the multimedia's description and MPEG-4 BIFS standard to implement interactive scenes into multimedia.

We also present experimental results on different kind of interactive services using video real time information extraction. The main implemented example of interactive services concerns an interactive mobile TV application related to parliament session. This application aims to provide additional information to users by inserting automatically interactive contents (complementary information, subject of the current session…) into original TV program. In addition, we demonstrate the capacity of the platform to adapt to multiple domain applications through a set of simple interactive services (goodies, games...).

Table des matières

Liste des figures

Liste des Tableaux

Chapitre 0

Contexte : média enrichi interactif

Avec la révolution du numérique, l'usage de la vidéo a fortement évolué durant les dernières décennies, passant du cinéma à la télévision, puis au web, du récit fictionnel au documentaire et de l'éditorialisation à la création par l'utilisateur. Les documents multimédias audiovisuels sont en pleine expansion pour le partage d'informations, d'émotions, de centres d'intérêts...

Ce chapitre, structuré en quatre parties, introduit le contexte de la thèse. La première décrit l'évolution des médias et des modèles économiques. Dans la deuxième partie, nous discutons de l'appropriation des nouveaux médias par les utilisateurs et des nouveaux usages émergeants. Nous montrons ensuite l'évolution des médias vers les documents multimédias composés de différents médias –audio, vidéo, image, texte, graphique…–. Enfin, nous traitons des documents hypermédias proposant une interaction entre les utilisateurs et les documents multimédias ainsi que la nécessité de trouver des solutions pour l'enrichissement automatique des médias au regard du nombre croissant de documents générés chaque jour.

0.1. Rôle des documents multimédias aujourd'hui

Après le cinéma où la vidéo de très haute qualité était reine, la télévision a apporté une première rupture avec une diffusion en direct des vidéos ainsi que la mise en place d'un modèle économique fondé sur la publicité. L'avènement du web marque un nouveau changement dans l'usage de la vidéo en rendant sa diffusion et sa consommation accessible à tous les utilisateurs, à tout moment. Le modèle du web, qui présente des documents multimédias orientés « texte » et qui introduit la notion d'hyperlien et de moteur de recherche, conduit à apparenter la vidéo à du R*ich Media*. La diversification des médias, l'augmentation des débits et la généralisation de l'accès à Internet ont, dès lors, fait évoluer les habitudes des consommateurs. Ceux-ci sont intéressés par l'accès à des documents multimédias améliorés, ou à participer à l'amélioration des contenus multimédias [Rubicon08, Solis08]. La Figure 1 récapitule les diverses possibilités d'enrichissement des documents multimédias. Les médias audiovisuels par exemple sont en pleine expansion pour le partage d'informations ou d'événements notamment grâce des plateformes d'échange de vidéo telles que [YouTube]. Le média est au centre de la Figure 1. Il peut s'agir d'une image, d'un document audiovisuel, d'une page Internet... Tout autour sont représentées les différentes solutions aujourd'hui disponibles pour enrichir un média. Cette étape permet de faire le lien entre le média initial et les activités de reconnaissance, partage… associées. Par exemple, un document audiovisuel auquel on ajoute de l'interactivité permet la création d'un document multimédia interactif (hypermédia). On peut appliquer plusieurs enrichissements successifs et ainsi combiner plusieurs activités complémentaires. L'ajout de métadonnées à cet hypermédia permet de lui ajouter de la connaissance. Enfin, sur la Figure 1, les quatre rectangles renvoient aux domaines dans lesquels les documents multimédias ainsi créés sont aujourd'hui le plus présents.

Ces différentes évolutions ont été rendues possibles grâce à l'engouement des utilisateurs pour le partage des contenus multimédias, l'avènement du web et la démocratisation de l'ordinateur personnel. La possibilité d'annoter des médias sur des sites Internet de partage de vidéo tels que [Flickr, FaceBook] est la solution aujourd'hui utilisée pour interagir tant avec le contenu partagé qu'avec les autres utilisateurs. Ces interactions sont aujourd'hui au cœur des réseaux sociaux ainsi que des communautés virtuelles comme moyen d'expression.

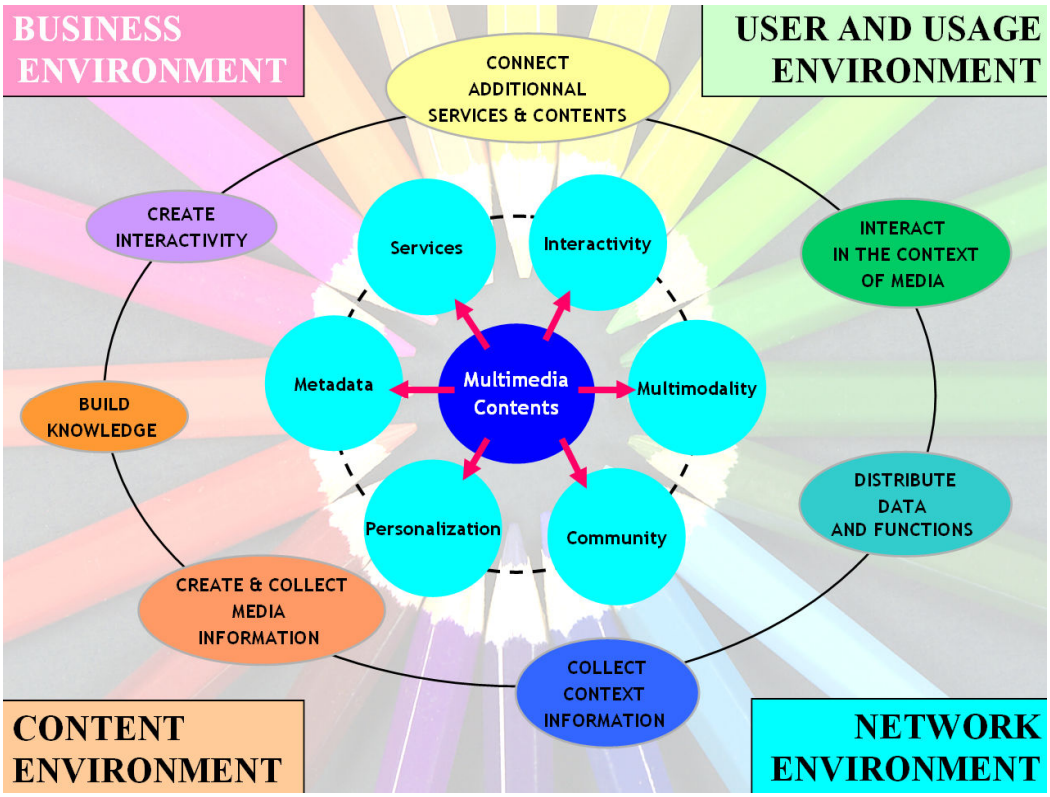


Figure 1 : Illustration des évolutions d'un média

Les utilisateurs utilisent par ailleurs les médias pour échanger [Shao09] des informations, des connaissances, des « reportages » personnels, exprimer leurs réactions à un sujet...

0.2. Evolution vers les documents multimédias

Presse écrite, radio et télévision sont les médias historiques alors que l'Internet et l'IPTV sont les nouveaux médias. L'évolution du marché de la publicité dans les médias est un bon moyen de mesurer l'évolution des modes de consommation (Tableau 1).

Tableau 1 : Evolution du marché de la publicité à travers le monde par média [Arsenault08]

Media	2005	2006	2007	Evolution
Newspapers	30,00%	29,80%	29,40%	- 0,60 %
Television	37,30%	37,40%	37,40%	+ 0,10 %
News magazine	13,50%	13,40%	13,40%	- 0,10 %
Radio	8,50%	8,30%	8,20%	- 0,30 %
Outdoor	5,40%	5,40%	5,50%	+ 0,10 %
Internet	4,10%	4,50%	4,70%	+ 0,60 %
Cinema	0,40%	0,40%	0,40%	no changes

Il est difficile de mesurer l'impact direct des nouveaux médias sur les médias historiques, Cependant, comme cela est détaillé dans [Tessier07], l'accès gratuit à l'information déstabilise les modèles économiques existants. Les consommateurs délaissent les médias historiques pour les nouvelles technologies. Il est nécessaire pour les médias historiques de se moderniser et de fournir de nouvelles fonctionnalités afin de s'adapter aux parts de marché. Le Tableau 2 met en perspective l'évolution globale de l'investissement des publicitaires dans le monde.

Tableau 2 : Evolution globale du marché de la publicité dans le monde [ZenithOptimedia07]

Zone	2005		2007	
	Market (in billion of dollars)	Annual evolution	Market (in billion of dollars)	Annual evolution
North America	174,55	+3,8 %	184,31	+ 4,50%
Europe	107,94	+3,4 %	112,57	+ 4,40%
Asia-Pacific	83,65	+5,1 %	89,12	+7,8 %
South America	17,08	+5,8 %	18,17	+6,0 %
Africa, Oriental countries	20,76	+17,5 %	24,27	+ 14,80%
World	403,98	+4,7 %	428,44	+5,8 %

D'après les Tableaux 1 et 2, il ressort que seules les nouvelles technologies profitent de l'évolution du financement par les publicitaires. Cela est lié à la nouvelle répartition des financements et des consommateurs sur les médias disponibles. Cependant, nous pouvons noter qu'un nouveau média ne remplace pas le précédent ; il apparaît une nouvelle répartition des parts de marché. Par exemple, la télévision n'a pas remplacé le cinéma, au même titre que la cassette vidéo ou le DVD n'ont pas remplacé la télévision.

Néanmoins, la presse écrite est le premier média historique à souffrir de cette répartition des parts de marché. Le nombre de titres dans la presse écrite a chuté de 40% depuis 2005 ; la presse écrite a ainsi dû se concentrer sur ses centres d'intérêts et/ou considérer les nouveaux médias comme Internet par exemple pour consolider l'intérêt des consommateurs et retrouver l'investissement des publicitaires. La télévision est de même directement impactée par cette évolution du marché. La Figure 2 illustre l'évolution de la consommation des médias en Norvège.

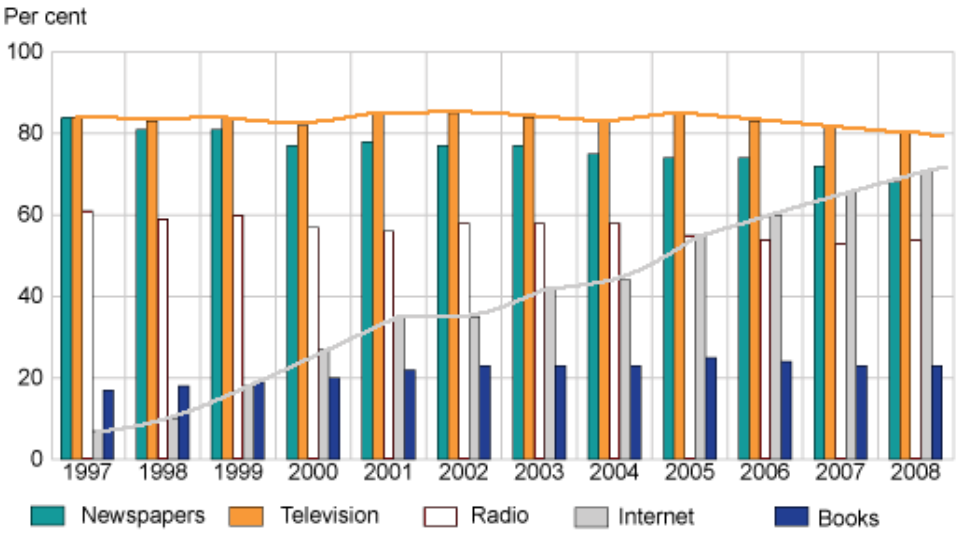


Figure 2 : Pourcentage de la consommation moyenne par jour de la population en Norvège [Norway09]

Cette évolution a aussi été remarquée au Royaume Uni [Ofcom08] ou en France [Mediamat08] avec une réduction en moyenne de cinq minutes par jour de télévision par rapport à 2007 (principalement auprès des 15-34 ans).

0.3. Télévision mobile, sources d'information et interactivité

Face à la multiplicité des sources d'information, les consommateurs veulent aussi bien choisir celle-ci que leur contenu [Saltzman97]. Ainsi, les consommateurs souhaitent recevoir des informations générales sur le monde, mais également pouvoir accéder à des informations très précises sur des sujets spécifiques tels que le sport, l'économie, la culture…

Parmi ces sources d'information, les flux RSS [RSSBOARD] permettent aux utilisateurs de PC et possesseurs d'une connexion Internet d'être informés quels que soient l'heure ou le lieu (PC, mobile…). Le flux RSS est aujourd'hui utilisé pour recevoir les informations relatives à une mise à jour ou une évolution sur un site Internet par exemple. Toutefois, le taux d'information reste élevé et la difficulté est de trier les informations redondantes d'un flux RSS à un autre. De plus, les flux RSS fournissent des informations générales synthétisées. Il est difficile pour l'utilisateur de vérifier la validité de l'information en termes de neutralité et de confiance (dépendant du nombre d'interprétations successives, de la compréhension et de l'analyse de l'information avant sa rediffusion).

Une autre source d'information de plus en plus utilisée pour la recherche d'information spécifique est constituée des encyclopédies ouvertes et libres telles que [Wikipédia]. Dans ce contexte, des efforts sont aujourd'hui mis en œuvre [Auer07] pour extraire de l'information structurée des *wiki* et la rendre disponible sur Internet à travers des applications ou des services web par exemple. Le but est ainsi d'associer les connaissances des bases de données des wiki à des moteurs de raisonnement pour analyser les différents concepts et fournir des résultats. Le moteur

d'inférence pourrait ainsi « naviguer » dans les bases de données des wiki et fournir directement les bonnes informations aux utilisateurs.

0.3.1. Les nouveaux usages des consommateurs de télévision

Traditionnellement, la consommation vis-à-vis des médias historiques est considérée comme passive [Kubey02]. Il n'y a en effet pas d'actions spécifiques requises de l'utilisateur. Pour la télévision, le nombre de chaînes disponibles a explosé ces 10 dernières années avec l'apparition du Câble, de la TNT, des offres satellites… Les préférences des utilisateurs évoluent lentement vers ces nouvelles chaînes, même si en France les utilisateurs conservent une préférence pour les six chaînes historiques (Tableau 3) dont l'audience cumulée est supérieure à 60%. En France, les fournisseurs de contenu TMC(1) et W9(4) ont atteint leurs seuils de rentabilité en 2008, soit trois ans après leur lancement. TMC est détenue par TF1(2) et AB(3) tandis que W9 l'est par M6(5) [Gonzales08].

Tableau 3 : Evolution des audiences en France des chaînes historiques sur la TNT [MediaCabSat]

Audience (in %)	June 2007	June 2006
TF1	25,3	25,5
France 2	13,5	13,9
France 3	9,6	9,5
M6	8,5	8
Canal +	4,7	4
France 5 + Arte	2,6	2,2
Total	**64,2**	**63,1**

Aujourd'hui, comme cela est illustré Figure 3, ce sont les jeunes consommateurs qui évoluent le plus rapidement vers les nouveaux médias. Ils constituent la part d'évolution la plus importante dans la consommation de médias. En conséquence, nous pouvons déduire que ces jeunes consommateurs accumulent et combinent la consommation des nouveaux médias et des médias historiques.

(1) TMC : *Technology Marketing Corporation*, http://www.tmcnet.com
(2) TF1 : *Groupe TF1*, www.tf1.fr
(3) AB : AB *Group*, http://www.abgroupe.fr/
(4) W9 : *M6 Music*, http://www.w9.fr

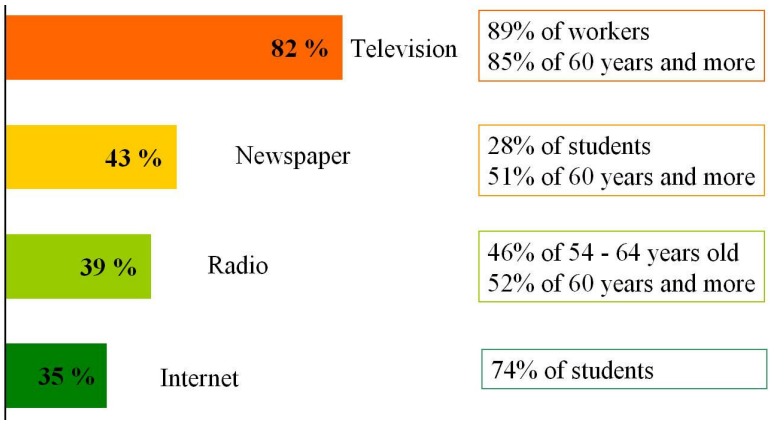


Figure 3 : Préférences des consommateurs en France en 2007 [Teinturier07]

Cette abondance de médias disponibles permet aux consommateurs de passer d'un média à un autre en fonction de leurs besoins, de leurs préférences, de la qualité des programmes proposés, du contexte… Ce phénomène est mis en avant en comparant par exemple les audiences sur Internet et sur la télévision durant le tournoi de football « Euro 2008 ». Une étude [Euro08] a montré que les utilisateurs délaissaient la télévision au profit d'Internet durant les moments du match les moins « intéressants » –juste avant le début du match, durant la mi-temps, ou après le match–. Cette étude souligne également la baisse d'intérêt pour le modèle traditionnel du spot publicitaire de 30 secondes [Arlen81]. Enfin, la vidéo à la demande (V.O.D.) est un autre domaine qui confirme la préférence des utilisateurs à regarder les contenus de leur choix lorsqu'ils le souhaitent. La Figure 4 illustre l'évolution de la consommation « vidéo à la demande » en France en 2007.

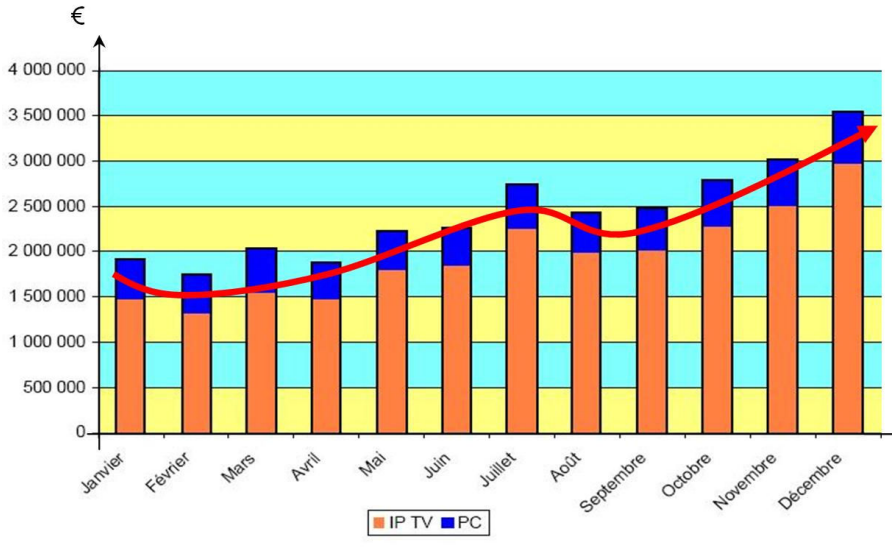


Figure 4 : Evolution de la consommation de la vidéo à la demande en France en 2007 [Franceschini07]

Dès lors, presse écrite, télévision, radio… doivent évoluer pour fournir des services avancés aux consommateurs tels que télévision interactive ou contenus adaptés aux préférences des consommateurs afin de consolider les parts de marché et d'attirer les jeunes consommateurs [Hausenblas07].

0.3.2. L'interactivité

La définition communément admise de l'interactivité est que celle-ci repose sur une communication bidirectionnelle. L'interactivité peut être une interaction entre deux personnes ou plus, mettant en œuvre un ou plusieurs systèmes… L'interactivité n'est pas quelque chose de nouveau ou de mystérieux comme le rappelle [Dufresne96]. La Figure 5 illustre de gauche à droite le degré d'interactivité en fonction du taux de présence des critères d'interaction tels que le retour des utilisateurs (*Feedback*), le *contrôle*, la *communication*…

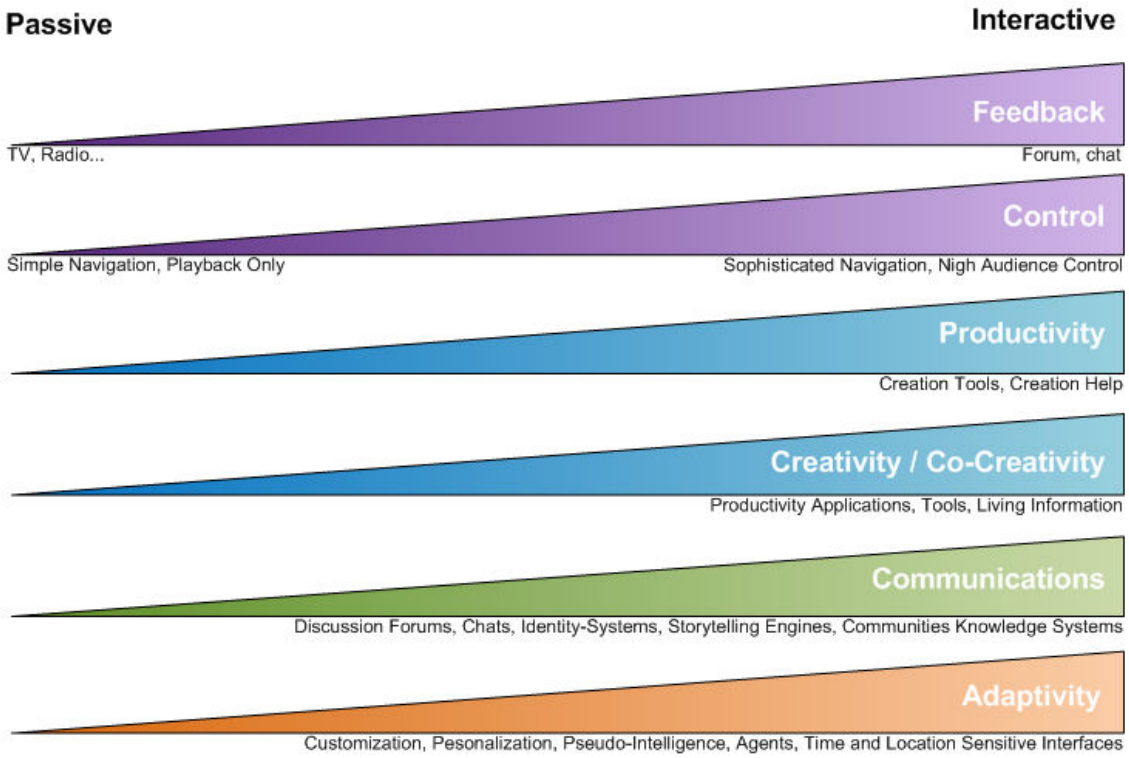


Figure 5 : Evaluation des critères liés à l'interactivité [Shedroff94]

Un professeur par exemple vit constamment dans l'interactivité avec ses élèves lorsqu'il pose des questions à sa classe, lorsqu'il donne des exercices à faire, lorsqu'il anime des groupes de discussion… Dans ce contexte, « l'enseignement interactif » est fondé sur des échanges plus ou moins dirigés, contrôlés dans un souci d'optimisation d'une acquisition de compétences ou de connaissances. En reportant cet exemple à la Figure 5, nous vérifions bien qu'il nécessite un haut niveau de communication, de contrôle, d'adaptation…

0.3.3. L'interactivité dans les médias

L'interactivité dans la télévision mobile est un moyen d'enrichir la télévision vers un mode de consommation proche du modèle du web [Bara05, Shin06]. L'interactivité permet à l'utilisateur d'interagir à différents niveaux avec le contenu des programmes diffusés (informations additionnelles), de naviguer à travers les contenus (VOD), ou de participer directement à des quizz télévisés….

Une implantation élémentaire de l'interactivité dans la télévision est le déploiement de services interactifs « directs » tels que T-commerce, l'accès immédiat à des bulletins d'information, de la vidéo à la demande, des quizz, des votes… Selon les critères de la Figure 5, ce type d'interactivité correspond aux composantes de la *communication*, des *retours utilisateurs*, et du *contrôle*. Un second niveau d'interactivité, fondé sur l'*adaptation* et le *contrôle* est l'introduction de la personnalisation à l'aide d'un profil utilisateur ou de préférences par exemple. Ces préférences visent à permettre d'adapter la navigation des utilisateurs dans les contenus disponibles à travers la création de grilles de programmes personnalisées [Rowe00].

0.3.4. Evolution de la télévision vers le modèle du web

Même si la citation de Steve Jobs [co-fondateur d'Apple Computer and Pixar, dans Macworld Magazine, Février 2004] est caricaturale :

"You watch television to turn your brain off and you work on your computer when you want to turn your brain on."

Il est reconnu que l'utilisateur est passif devant la télévision qui n'était technologiquement pas plus « évoluée » qu'un appareil ménager [Manjoo03].

Des technologies et standards ont été développés durant ces 10 dernières années dans le but de faire évoluer les médias historiques notamment à travers l'ajout d'interfaces interactives. Les systèmes de télévision interactive tels que [MHP] ont ainsi été distribués pour lancer l'accès à l'interactivité dans les foyers.

Des systèmes propriétaires tels que *MediaHighway* [NDS] et *OpenTV* [Bouilhaguet] ont été installés dans des *Set-Top-Box (STB)* par les distributeurs de contenus diffusés par câble ou par satellite. Ces appareils n'ont cependant pas connu de succès pour plusieurs raisons. La première est l'obligation pour l'utilisateur d'installer un boîtier supplémentaire en plus de ceux déjà présents –enregistreur vidéo, lecteur DVD, décodeurs…–. La seconde est que les programmes interactifs n'ont pas été déployés en raison des coûts des licences et de l'absence d'infrastructure… Enfin, la télévision permettait l'affichage de quelques informations texte [Ceefax] mais n'était pas conçue pour l'affichage et la navigation dans des contenus textuels riches.

Aujourd'hui la situation a évolué, les *STB* fournies par les opérateurs Internet sont présentes dans 63% des foyers européens en 2007 [Loof08]. Les STB permettent d'embarquer des fonctionnalités pouvant modifier les téléviseurs en appareils multimédias multi-usages. En effet, les STB tels que [Freebox] ont déjà commencé à proposer des fonctionnalités telles que l'enregistrement de contenus, l'accès à la vidéo à la demande, l'accès au répondeur téléphonique, le *time shifting* (mode *pause*) pour le contenu diffusé…

Par ailleurs, le marché de la diffusion de contenu évolue et s'ouvre aux nouvelles technologies, notamment avec l'apparition de la « téléréalité » encourageant les utilisateurs à interagir avec l'émission (vote, quizz, informations…) par le biais du téléphone, des SMS, des sites Internet … Les séries télévisées de la même façon tirent profit des nouveaux moyens de communication comme Internet par exemple pour développer la vente de produits dérivés, de contenus additionnels, pour créer un lien avec les consommateurs. Dès lors, les utilisateurs utilisent aujourd'hui un appareil supplémentaire (téléphone mobile, ordinateur…) pour accéder à ces contenus ou services.

Dans ce contexte, l'insertion de ces services ou contenus additionnels directement dans les documents multimédias faciliterait l'accès aux utilisateurs et permettrait une interaction directe avec les contenus.

0.3.5. Evolution des documents multimédias

Comme introduit précédemment, les utilisateurs veulent consommer des contenus multimédias enrichis. De plus, une partie des usagers souhaite enrichir et partager des médias. La Figure 6 présente la règle généralement admise 90-9-1 qui décrit les contributions des utilisateurs au sein des communautés. L'audience représente quatre-vingt dix pour cent des utilisateurs qui ont donc tendance à lire et à observer, mais qui ne contribuent pas activement. Neuf pourcents des utilisateurs sont des « éditeurs » qui modifient du contenu existant ou ajoutent des parties à un média existant, mais ceux-ci ne créent pas des articles à partir de zéro. Le pourcentage des utilisateurs restants correspond aux « créateurs » qui dirigent en grande partie l'activité du groupe social.

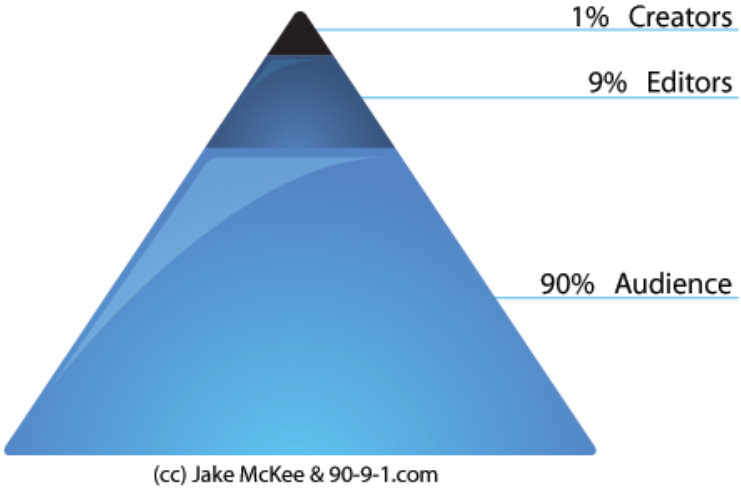


Figure 6 : Représentation de la règle du 90-9-1 [McKee]

L'évolution technologique dans les documents multimédias permet aux utilisateurs de modifier et d'enrichir les médias selon leurs opinions ; ainsi ils peuvent partager leurs points de vue avec leurs amis, les membres de leurs familles ou avec toute personne en général. La Figure 7 montre une liste non exhaustive des technologies

qui permettent cela. Celle-ci évolue constamment puisque les utilisateurs sont en permanence à la recherche de nouvelles approches pour publier leurs opinions. Nous pouvons également remarquer que presque toutes les solutions techniques proposent des possibilités de distribution avec envoi de données –image, vidéo, audio…– ou publication de liens vers les médias pour maximiser le potentiel de partage.

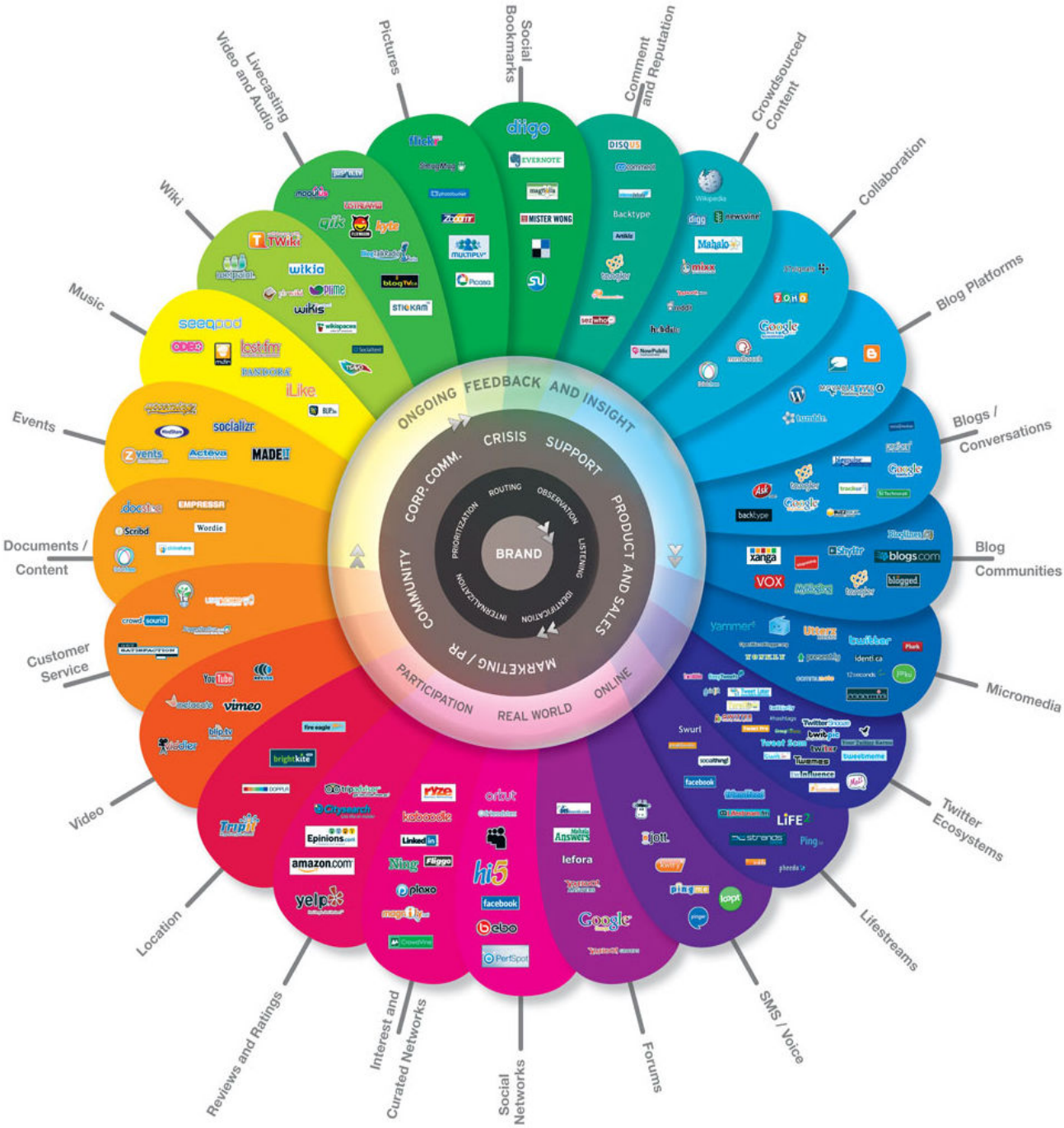


Figure 7 : Prisme de la conversation [Solis08]

Au regard de la Figure 7, il apparait que l'évolution technologique est plus que jamais motivée par l'activité des utilisateurs sur les plateformes de discussion, de partage et d'interaction avec les contenus multimédias. De plus, l'ajout de logiciels intégrés d'édition multimédia permet d'encourager les personnes à enrichir les médias. Par exemple, il y a plus de deux millions d'articles en anglais sur Wikipédia en 2009 [WikipediaToday].

0.4. Télévision interactive

Les personnes utilisent de plus en plus les nouvelles technologies pour s'informer [Tessier07]. Dans ce contexte, la télévision interactive a pour but l'amélioration de la télévision historique vers un mode de consommation sur le modèle d'Internet.

Figure 8 : Scepticisme à propos de la télévision interactive [Burke]

L'interactivité permet aux consommateurs d'avoir différents niveaux d'interaction avec les contenus multimédias [Bara05, Rowe00]. Avec celle-ci, de nombreuses applications et services sont envisageables. La liste ci-dessous ne présente qu'un petit ensemble des applications possibles :

- rechercher des contenus multimédias associés ;
- définir une alerte sur des événements spéciaux (retransmission d'un match de football, d'un concert…) ;
- donner un accès direct aux publicitaires aux préférences des clients en utilisant des annonces interactives ;
- partager des émotions avec une communauté avec du texte, des images, des sons, des vidéos… ;
- jouer en temps réel (questionnaire en ligne, quête dans un jeu multi-joueurs…) ;

- etc.

Certains de ces services existent déjà sur la télévision, mais ils sont disponibles uniquement en utilisant un autre périphérique tel que le téléphone mobile (SMS, numéro de téléphone…) ou un ordinateur personnel (Internet, messagerie). De nos jours, des scènes interactives peuvent être ajoutées dans des contenus multimédias. Par conséquent, l'utilisation des services peut être améliorée en permettant aux utilisateurs de se connecter directement aux services avec des contenus vidéos interactifs.

Cependant, changer l'infrastructure pour déployer la télévision interactive reste un enjeu économique ouvert.

0.4.1. Télévision interactive mobile

Puisque la télévision mobile est une technologie qui est actuellement déployée, il est possible de définir de nouvelles interfaces et de nouveaux services et donc d'introduire les possibilités de la télévision mobile interactive. Dans ce contexte, l'infrastructure déployée en France repose sur un mode de communication point-à-point [Boni04]. Cependant, au regard des limites techniques en bande passante, ce type d'infrastructure réduit le nombre d'accès simultanés. En effet, le flux du média est envoyé indépendamment et individuellement à chaque utilisateur. La prochaine génération de télévision mobile s'appuiera sur un mode de communication en diffusion afin de réduire les besoins en bande passante.

0.4.2. Interactivité diffusée

Le schéma de l'architecture décrite Figure 9 est sur le point d'être déployé en France pour la télévision mobile [DVB-H]. Cette solution décrit le système complet de communication entre les diffuseurs et les utilisateurs finaux : chaîne de production multimédia, transmetteurs, etc.

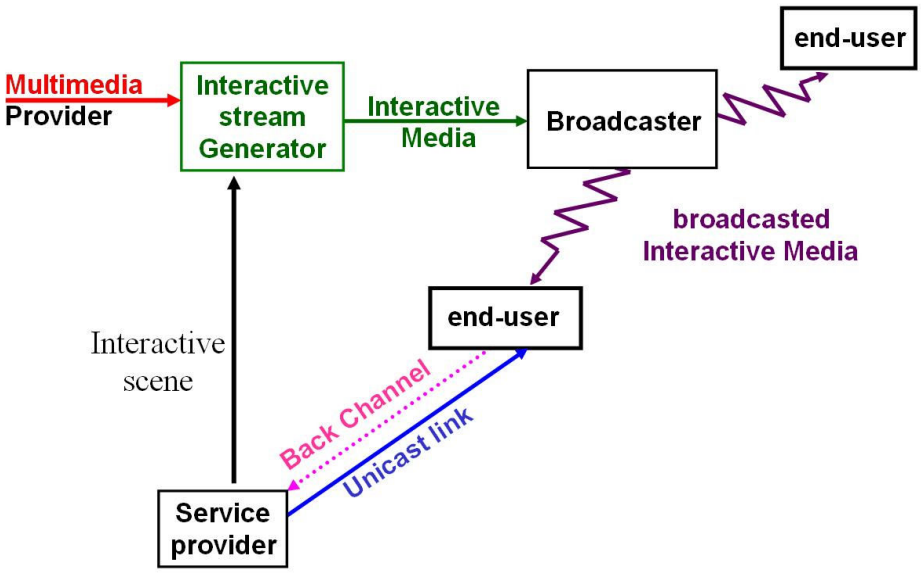


Figure 9 : Architecture pour la diffusion et l'interactivité

De plus, les communications point-à-point (GRPS, EDGE, 3G) sont possibles entre les utilisateurs et les fournisseurs de services. Ainsi, l'utilisateur reçoit le flux de manière diffuse et il peut interagir en s'appuyant sur les technologies point-à-point. Pour la télévision traditionnelle, il est nécessaire de modifier les récepteurs et les télécommandes pour que les utilisateurs interagissent avec le contenu télévisuel. En effet, un moteur de rendu est requis pour décoder le flux et lui ajouter les scènes interactives MPEG-4 BIFS [Concolato02], LASeR [Dufourd05].

0.4.3. « Télévision IP » interactive

La télévision interactive est déjà possible en utilisant les boîtiers ADSL. La Figure 10 détaille la répartition des souscripteurs à la télévision IP au niveau mondial fin 2007.

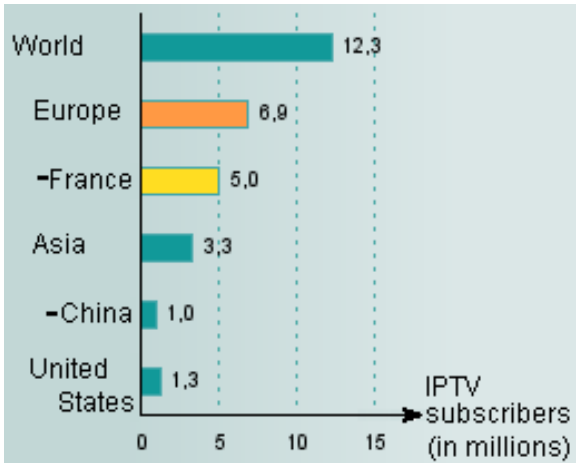


Figure 10 : Quantité et répartition des souscripteurs IPTV en 2007 [Informa]

L'expérimentation de services interactifs simples a déjà été testée en France pendant les élections locales 2008 [Dang08]. Les boîtiers ADSL semblent être le moyen privilégié pour déployer la télévision interactive puisqu'ils ont accès au moteur de rendu multimédia et qu'ils ont une télécommande adaptée (Freebox par exemple). Les problèmes techniques sont toujours présents car il y a peu d'utilisateurs disposant d'une bande passante suffisamment large pour accéder aux services IP. En France, même si le nombre de souscripteurs augmente continuellement, l'accès à la télévision IP reste le privilège des utilisateurs résidant dans les grandes villes (Figure 10). L'implantation de la télévision interactive mobile semble plus accessible grâce à l'évolution rapide des « Smartphones » en termes de connectivité et de puissance.

La prochaine étape pour l'évolution des documents multimédias est l'interactivité [Lemuet08], le chiffre d'affaire est estimé à 240 millions d'euro en 2005. Par ailleurs, sous l'impulsion de la télévision numérique et le développement de la télévision par ADSL, ce chiffre devrait être multiplié par quatre pour atteindre les 8.3 milliards d'euro en 2010.

Enfin, les fournisseurs de contenu historiques ont un modèle de distribution stable et sont capables d'ajouter les évolutions nécessaires pour la création de nouveaux usages [Hausenblas07].

0.5. Documents multimédias

Dans les années 1980, le multimédia est essentiellement déployé dans les domaines professionnels de l'impression et des médias (télévision, radio, journaux…). C'est pourquoi, de nombreux standards ont été créés pour construire la chaîne de diffusion entre les créateurs de médias et les utilisateurs finaux.

Cependant, les contenus générés par les utilisateurs ont explosé en 2005 avec le développement de la publication sur Internet et de nouveaux *cercles de production* de documents multimédias [Wiki_UGC]. Les utilisateurs profitent des avantages des évolutions des documents multimédias en terme de gestion multipistes (sous-titres, interactivité, métadonnées…) et de compression numérique pour augmenter le nombre de canaux télévisuels disponibles [Feldman97].

Même s'il y a de nombreux standards multimédias, les concepts utilisés pour représenter les données sont similaires. La Figure 11 représente le format général des fichiers multimédia (« Avi », « Windows Media », « Real Audio », « MP3 », …). Le conteneur encapsule les différentes pistes qui sont définies dans son entête. Les pistes peuvent être audio, vidéo, sous-titres, etc. Le type de piste et l'algorithme de compression des données (codec), le taux pour afficher les données sur le *Player* (bitrates), les dimensions et la résolution sont également déclarés dans l'entête.

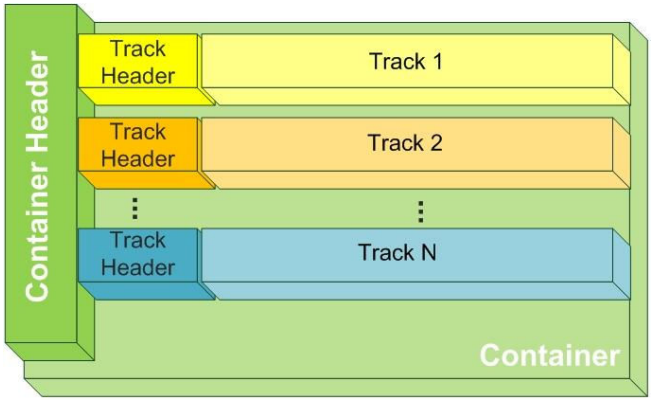


Figure 11 : Format général des fichiers multimédia

Les conteneurs ont évolué pour aboutir à des formats de plus en plus avancés. Par exemple, le consortium MPEG (Moving Picture Experts Group) a modifié et amélioré ses standards multimédia ainsi :

[MPEG-1] est le standard initial pour la compression vidéo et audio. Il a été choisi pour le VCD et il inclut le format populaire MP3 pour la partie audio [LeGall91].

[MPEG-2] est l'évolution de MPEG-1 pour considérer le transport. Il contient des standards vidéo et audio pour la transmission diffusée [Tranchard94]. Il est utilisé pour les solutions « dans les airs » (ATSC, DVB et ISDB, service de TV satellite numérique, télévision numérique) et, avec quelques légères modifications, pour le disque DVD.

[MPEG-4] est la dernière version de standard de format de fichier multimédia. Il étend MPEG-1 afin de supporter les objets vidéo et audio, les contenus 3D, la scalabilité des flux, la gestion des DRMs… Par ailleurs, il offre de nombreux standards

d'encodage avancé de la vidéo tels que MPEG-4 partie 10 (Advanced Video Coding) [MPEG4-part10].

Il existe également de nombreux formats disponibles pour générer, diffuser et jouer les scènes interactives. Il est nécessaire de considérer les caractéristiques des différents formats riches [Marilly06, LeBonhomme08] pour sélectionner celui qui correspond le mieux à un problème donné.

D'une façon générale, sept composants sont nécessaires pour gérer les documents multimédias :

1 – une interface utilisateur (Agent conversationnel, interface graphique…),

2 – un format enrichi (contrôle avancé, interactivité…),

3 – des outils d'annotations manuelles,

4 – des outils d'annotation automatique élémentaires (détection de couleurs, bruits…),

5 – des outils d'analyses multimédias spécifiques (reconnaissance de visages, textes…),

6 & 7 – des outils d'analyse sémantiques (ontologies, moteurs d'inférence…).

Nous en avons créé une représentation graphique (Figure 12), afin d'en simplifier la compréhension.

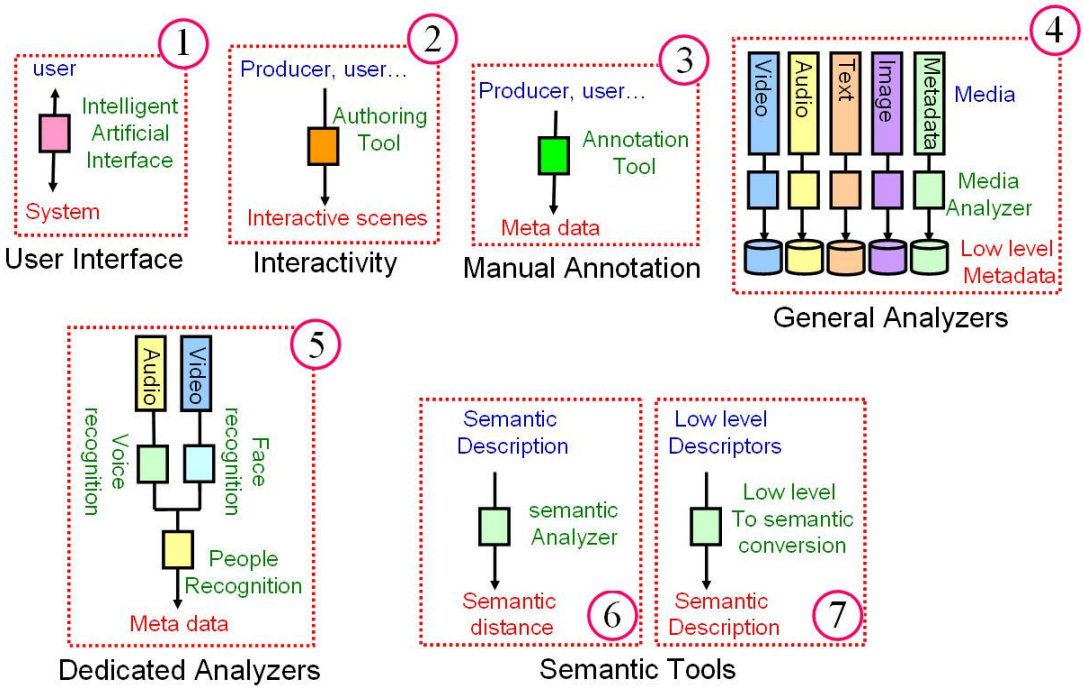


Figure 12 : Schéma introduisant les principaux domaines technologiques

Proposer une solution pour faciliter l'insertion de services interactifs contextuels dans un flux multimédia revient à disposer des modules 1 et 2 (Figure 12). Il est dès lors nécessaire d'extraire la description contextuelle des médias (modules 3, 4 et 5,

Figure 12). Il est ainsi possible de comparer le contexte du document multimédia avec les différentes spécifications d'un service interactif (modules 6 et 7, Figure 12).

0.5.1. La notion de *Rich Media*

Le principe du *Rich Media* est d'intégrer différents médias (audio, vidéo, image, texte, graphique…), dont l'interactivité est simplifiée par une ergonomie s'appuyant sur l'utilisation d'animations et des médias eux-mêmes. La capacité à synchroniser ces divers médias est l'une des caractéristiques du *Rich Media* qui renvoie souvent à la notion d'interfaces riches.

On peut par le biais de cette technique obtenir une vidéo en streaming chapitrée et agrémentée de documents, ce qui augmente l'interactivité avec l'utilisateur. Dans le modèle du web, Silverlight (Microsoft), JavaFx (Sun MicroSystems), FLEX (Adobe System), AJAX (Asynchronous JavaScript And XML) et HTML5 sont des exemples de technologies permettant la création d'interfaces interactives avancées entre les utilisateurs et les sites Internet. Cependant, même s'il est possible de créer des temporisations pour suivre une planification d'événements (animation), ces technologies s'appuient sur des communications asynchrones (pas de synchronisation directe sur un média). Ces technologies ne sont pas développées pour l'insertion d'interactivité dans les médias [Viljoen03, Tran03].

Des technologies dédiées aux documents multimédias audiovisuels sont préférées dans le cadre de l'édition de services interactifs pour les chaînes de diffusion de programmes interactifs.

La quantité de technologies disponibles pour générer des scènes interactives croît quotidiennement. Presque tous les formats existants de *Rich Media* permettent de diffuser le contenu d'une animation et alors, de la jouer localement sur le mobile de l'utilisateur final (format flash par exemple). Pour fournir une vue générale des fonctionnalités principales, nous sélectionnons et détaillons uniquement un ensemble de principes des standards. Pour répondre à nos besoins d'application de télévision interactive, nous nous limitons à considérer deux objectifs multimédia principaux : l'audiovisuel (TV, vidéo à la demande, …) et le multimédia sur Internet (interfaces utilisateur avancées).

0.5.2. Introduction au contenu audiovisuel interactif

Le consortium MPEG et des entreprises telles que QuickTime, RealPlayer fournissent un ensemble de normes ou standards de fait qui permettent de créer et diffuser du contenu audiovisuel interactif :

- MPEG-4 LASeR (Lightweight Application Scene Representation [MPEG4-part20]), MPEG-4 BIFS (Binary Format for Scene description [MPEG4-part11]) ;
- W3C SVG (Scalable Vector Graphics [SVG]), SMIL (Synchronized Multimedia Integration Language [SMIL]) ;
- Format Real Player ;
- Format QuickTime ;
- …

Le *Rich Media* concerne donc entre autres les principaux formats vidéo dont Windows Media, Flash Video, Real Media, MPEG-4, QuickTime. Les technologies telles

que MPEG-4 BIFS, SVG du W3C… permettent de créer et diffuser des documents multimédias enrichis [Concolato07].

Le tableau 4 représente les principaux standards existants qui disposent des caractéristiques requises telles que « interactivité », « compression », « diffusion en flux », « composition dynamique », etc. Cependant, les formats doivent également permettre de diffuser les mises à jour de scène pour rendre possible l'envoi en temps réel du contenu d'animation. Par exemple, les standards [VRML97] et [SMIL] n'ont pas été retenus en raison du manque de compression et de mise à jour en temps réel.

Tableau 4 : Comparaison des formats existants de vidéo interactive

	BIFS	**LASER**	**SVG**	**FLASH**
Standard name	MPEG-4 (part 11)	MPEG-4 (part 20)	W3C	Adobe System
Authoring tools	Not available in industry	Proprietary authoring tool	Available	Proprietary authoring tool
Player	Need implemented BIFS decoder	Proprietary player	Available	Compatible with every platform and browser
Scripting facilities	Supported	Supported	Not yet supported	Supported on server
Scalability	Supported (MPEG-4 properties)	Supported (MPEG-4 properties)	Not yet supported	Not yet supported
Dynamic composition	Using update mechanism	Using update mechanism	Not supported	Using update mechanism from Server
Compression	Binary format	Binary format	Zip compression	Flash Format

Deux principaux standards MPEG sont indiqués pour fournir des interactions avec la scène audiovisuelle, LASeR et BIFS.

0.5.2.1. MPEG-4 partie 11 – BIFS

Fondé sur le conteneur orienté objet média MPEG-4, MPEG-4 BIFS est la première tentative du consortium dans le domaine du codage de composition. Il est accompagné d'outils innovants pour la création de contenu multimédia mixant graphiques 2D et 3D, pour la mise à jour incrémentale de scène, pour la diffusion de scènes longues et pour une synchronisation entre les différents éléments audiovisuels

d'une scène. MPEG-4 BIFS fournit des facilités pour intégrer et synchroniser, spatialement et temporellement, de nombreux objets interactifs.

La Figure 13 illustre un exemple de *Rich Media* créé à l'aide de la norme MPEG-4 BIFS. La norme intégrant un langage de description de scène multimédia, il est possible de combiner et synchroniser des médias de différents types et de différentes sources. Il est également possible de décrire les interactions entre les utilisateurs et les objets ainsi qu'entre les objets eux-mêmes. Cette description peut être de haut-niveau comme illustré Figure 13, mais aussi de bas-niveau pour la description des actions («bouger », « tourner », « cacher ») et des primitives (« champs texte », « rectangles », « cercles », etc.).

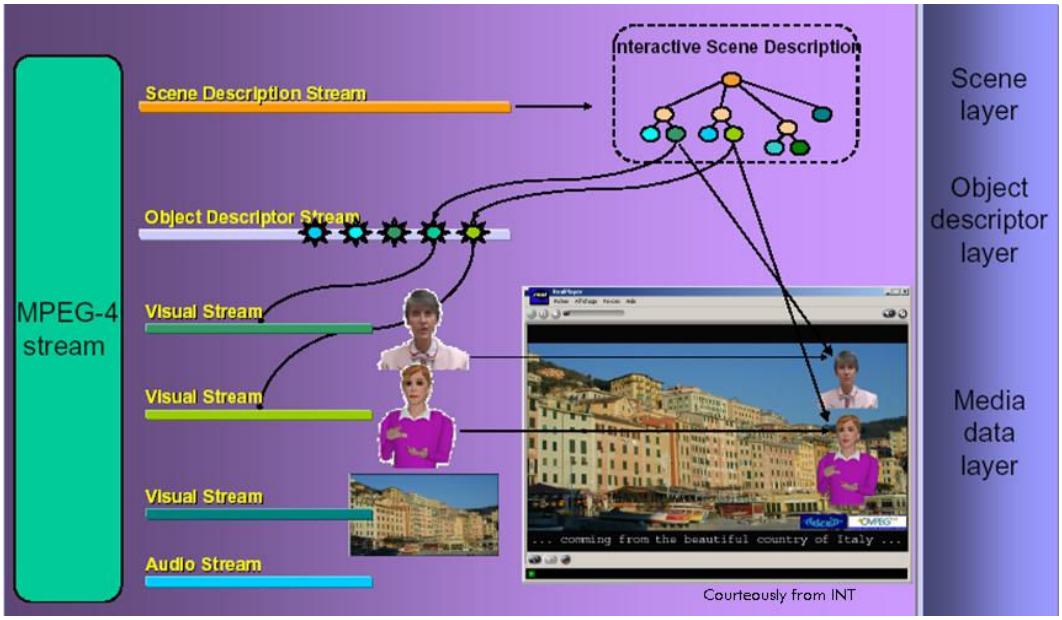


Figure 13 : Exemple de Rich Media créé avec la norme MPEG-4 BIFS. (courtoisie Département ARTEMIS-Télécom SudParis)

Cependant, les principales limitations restent la complexité du codec et le trop bas niveau de la programmation de scène interactive. Le manque d'outils pour l'édition de scènes BIFS dans les documents multimédias limite actuellement le recours à BIFS [Shao06].

0.5.2.2. MPEG-4 partie 20 – LASeR

Le standard LASeR offre les solutions, de bout en bout, à la chaîne de publication de *Rich Media* : facilité de création de contenu, optimisation de la livraison des données du *Rich Media* et rendu amélioré sur tous les périphériques [Dufour05]. Inspiré par les meilleurs concepts des solutions à l'état de l'art (W3C/SVG, Macromedia Flash, ISO/IEC MPEG/BIFS). La norme MPEG-4 LASeR modifie et optimise chaque fonctionnalité requise par les services *Rich Media* afin de répondre efficacement aux besoins d'un standard ouvert.

Le standard LASeR spécifie la représentation codée de la présentation multimédia. Dans sa spécification, une présentation est une collection comprenant une

description de scène et un média (zéro, un ou plus). Un média est un contenu audiovisuel individuel d'un des types suivants : image (statique), vidéo (succession d'images), texte (police de caractère), audio... Une description de scène est composée de texte, de graphiques, d'animations, d'interactivité et de description spatiale et temporelle.

Une description de scène LASeR spécifie quatre aspects d'une présentation :

- Comment les éléments de scène sont organisés spatialement (description spatiale des éléments visuels),
- Comment les éléments de scène sont organisés temporellement (s'ils sont synchronisés, quand l'animation commence et quand elle se termine),
- Comment interagir avec les éléments dans la scène. Par exemple, définir l'action à « lancer » lorsqu'un élément est cliqué,
- Comment ces changements prennent effet (si la scène change).

La spécification définit un moteur LASeR (Figure 14) pour visualiser les présentations LASeR. LASeR est basé sur le standard [SVG]. Parmi toutes les primitives de composition, LASeR spécifie des capacités d'hyperliens, d'embarquement d'audio et de vidéo, de représentations de graphiques vectoriels, de fonctionnalités d'animation et d'interactivité. Le module « Updates and Extensions » augmente globalement le nombre de fonctions. Le module « Binary Encoding » permet de convertir une scène en format binaire afin d'en faciliter la diffusion.

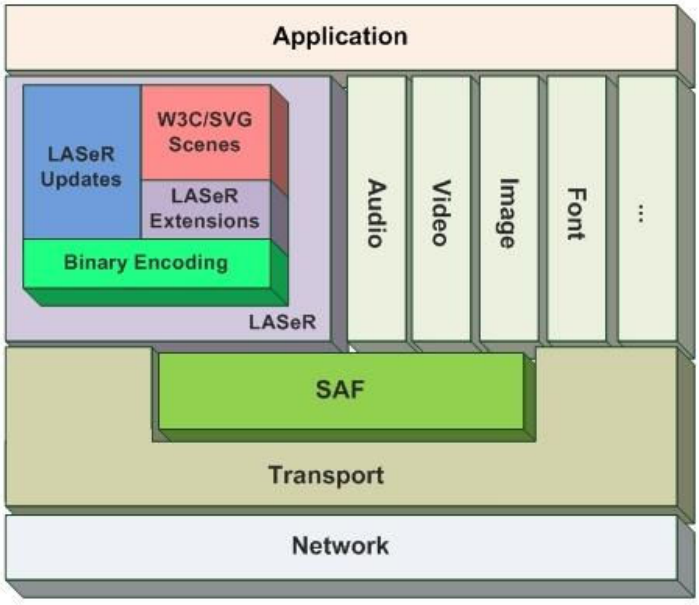


Figure 14 : L'architecture du moteur LASeR

La principale limitation de MPEG-4 LASeR est de ne pas considérer actuellement les contenus 3D. LASeR, BIFS, Flash sont des standards avancés pour transporter et représenter l'interactivité aux utilisateurs. Néanmoins, il est toujours nécessaire de générer au préalable les scènes interactives et de définir quand les scènes doivent être insérées dans le média.

Dans ce contexte, deux types d'usage et de diffusion de la vidéo coexistent : en temps différé, comme dans YouTube, ou en temps réel, avec enrichissement dynamique à la volée.

0.6. Mise en œuvre du Rich Media

La solution mise en œuvre aujourd'hui pour ajouter de l'interactivité consiste à créer et insérer manuellement des scènes interactives dans les documents multimédias. Cette opération est réalisée, soit de manière différée pour les médias stockés, soit à « la volée » avec un temps de latence pour les médias diffusés en « live ». Ces limitations apparaissent par exemple lors de la diffusion des *Travaux de l'Assemblée nationale* où l'on peut observer un retard d'environ cinq minutes entre ce qui est filmé dans l'hémicycle et ce qui est rediffusé sur les chaînes de la vidéo d'origine et des informations contextuelles ajoutées (noms des personnes, sujets traités…).

Au regard du nombre de documents multimédias créés chaque jour (on prévoit 1000 heures par heure de contenu diffusé pour les télévisions dans le monde en 2010), il est impossible de générer et d'insérer manuellement les services interactifs dans les médias. C'est pourquoi, des solutions fondées sur l'analyse du contenu des médias sont aujourd'hui en cours de développement pour automatiser ces étapes [Bara05]. Celles-ci reposent notamment sur des analyseurs de contenu multimédia, développés à travers le monde [Zhou03, Zhang06, Verilook…].

0.6.1. Les analyseurs de documents multimédias

Un analyseur est un algorithme d'analyse d'un ou plusieurs flux médias pour en extraire des informations. Par exemple dans le cadre d'un flux vidéo, un analyseur fournit un histogramme des couleurs, la localisation d'un mouvement dans la scène, la détection d'un son…

Il existe deux types d'analyseurs : ceux qualifiés de « bas niveau » qui permettent l'extraction d'informations élémentaires (couleurs, fréquences…) et ceux dits de plus haut niveau (détection de personnes, reconnaissance d'objets, reconnaissance de la parole) qui sont généralement des combinaisons d'analyseurs de bas niveau.

Les analyseurs de bas niveau ont individuellement de bons résultats [Pfeiffer96, Rabiner78]. Cependant, les informations fournies ne sont réellement exploitables que si elles sont combinées avec celles d'autres analyseurs [Mezaris04].

0.6.2. Exposé des problèmes

Si combiner de tels analyseurs ne présente aucune difficulté technique, la limitation est liée à ce que les analyseurs de plus haut niveau ainsi obtenus sont généralement *statiques*, c'est-à-dire figés et dédiés à une application spécifique. On ne dispose pas à l'heure actuelle d'une méthodologie et d'outils pour capitaliser sur la construction d'analyseurs complexes. En outre, les analyseurs restent généralement limités dans le nombre de flux médias qu'ils exploitent simultanément.

Quant aux analyseurs de plus haut niveau, ils sont classés en deux groupes : les analyseurs de haut niveau et ceux de très haut niveau. Les premiers manipulent des concepts objectifs en combinant des analyseurs de bas niveau. Leurs performances sont liées au contexte de fonctionnement spécifié. Ainsi, pour la détection et la

reconnaissance de visages dans les vidéos par exemple, on définit un taux de « réussite » de l'analyseur en fonction des conditions de luminosité, d'orientation du visage à détecter, de contraste de l'image…

Les analyseurs de très haut niveau constituent aujourd'hui le plus haut niveau d'analyse avec la manipulation de concepts subjectifs. Ils doivent par exemple être capables d'identifier des concepts subjectifs comme l'humeur d'une personne, la nature de la relation entre deux personnes dans une scène... La détection d'un sourire par exemple repose non seulement sur l'extraction de caractéristiques visuelles, mais aussi sur une modélisation conceptuelle, un sourire recouvrant des acceptions différentes d'un continent à un autre, et se distinguant d'un rictus par exemple. Ce type d'analyseur fort complexe dans toute sa généralité qui n'existe pas de façon opérationnelle aujourd'hui.

Le recours à la combinaison d'un très grand nombre d'analyseurs pour pallier cette limitation nécessite dès lors de disposer non seulement d'une description de chacun d'eux pour les caractériser, les identifier et les classer, en vue d'en proposer une composition dynamique en fonction du service ciblé. Cette modélisation à la fois locale et globale d'une gestion adaptative des analyseurs n'existe pas.

Enfin, les systèmes d'analyse développés qui sont aujourd'hui intrinsèquement fermés ne permettent pas de tirer le meilleur parti de l'évolution rapide des technologies d'analyse, des langages de description sémantique et des progrès apportés en termes de performances. A titre d'exemple, une détection de mouvement peut être réalisée selon le contexte de l'application par différence d'images, par analyse à base d'ondelettes, ou encore par exploitation opportuniste d'informations fournies par la compression. Le système doit alors être capable de supporter et de gérer de façon optimale ces diverses solutions. Ajoutons que les analyseurs de haut niveau tels ceux de détection ou de reconnaissance de visage sont souvent fermés afin de protéger la logique métier qu'ils exploitent.

Modularité, extensibilité et ouverture des systèmes restent donc des enjeux actuels dans un contexte où les modèles d'affaire sont à repenser pour prendre en compte la notion de *prosumer,* ce nouveau genre d'utilisateur à la fois consommateur et producteur de contenus et de médias.

0.7. Contenu de la Thèse

Dans ce contexte, le chapitre 1 de cette thèse s'attaque aux verrous ci-dessus et propose des solutions de modélisation des analyseurs, des mécanismes de reconfigurabilité et de dynamique temporelle, d'ouverture, de modularité et d'extensibilité mises en œuvre au sein d'une architecture intitulée ReconfigurAble Multimedia Service EnablerS (RAMSES). Cette description est mise en parallèle avec une description de l'état de l'art pour chacun des modèles proposés.

Le chapitre 2 propose une implantation des modèles proposés au sein de cette plateforme RAMSES. Celle-ci, opérationnelle sous forme d'un prototype, permet l'analyse d'un document multimédia afin de sélectionner et déployer de façon automatique des scènes interactives en relation avec le contenu, créant ainsi du *Rich Media* personnalisable et adaptatif.

Le chapitre 3 détaille les résultats obtenus dans le cadre du déploiement automatique de services interactifs sur une plateforme de diffusion de télévision mobile interactive. Enfin, la discussion sur les différents modèles implantés met en

évidence les contributions apportées vis-à-vis des verrous identifiés et propose des recommandations pour les améliorations à apporter.

La conclusion générale sur les enjeux de l'analyse des documents multimédias pour l'enrichissement automatique de ceux-ci et les perspectives en termes d'industrialisation au sein d'Alcatel-Lucent sont l'objet du dernier chapitre de ce mémoire.

Sept annexes complètent cet ensemble en présentant principalement les publications et brevets effectués durant cette thèse et les exemples de codes implantés pour la mise en place de la plateforme de démonstration.

Chapitre 1

RAMSES (ReconfigurAble Multimedia Service Enablers) : Modélisation

Les limites des systèmes actuels seront tout d'abord analysées et discutées, puis une modélisation des évolutions à mettre en œuvre pour lever les verrous technologiques identifiés sera proposée.

1.1. Spécification et modélisation de l'architecture

1.1.1. Analyse d'une architecture de référence

La Figure 15 rappelle le schéma traditionnel d'un analyseur de média. Cet exemple nous permet d'illustrer et de mettre en évidence les principaux verrous énoncés précédemment. Bien que cet exemple soit extrait d'une publication de 2004, ce type d'architecture est toujours utilisé [Mei07, Radhakrishnan08] et *de facto*, fait référence.

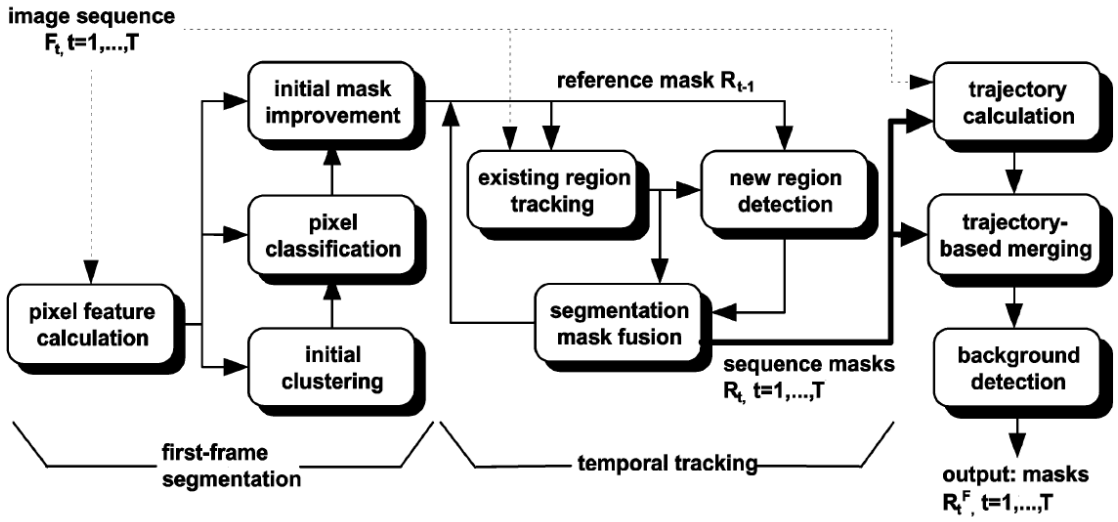


Figure 15 : Exemple d'architecture d'analyse de média [Mezaris04]

Même s'il présente un niveau de complexité élevé, ce schéma n'intègre que trois algorithmes d'analyse et ne tient compte que *d'un seul type de flux multimédia* (vidéo).

Le couplage fort d'une part entre le format d'entrée (*séquence d'images*) et les modules d'analyse et d'autre part entre les modules eux-mêmes lui confère son caractère *non évolutif*. En effet, une modification d'un des analyseurs ou du type du format d'entrée (HSV, YUV…) remet en cause toute l'architecture du système.

Les analyseurs n'incluent *pas de description* de leur fonctionnement (entrées sorties ou pré-conditions). Il est dès lors nécessaire d'analyser les fonctions et les algorithmes mis en œuvre dans les différents modules pour comprendre les interactions et la logique métier implantée. Les analyseurs sont par ailleurs souvent *fermés* afin de protéger ce savoir / savoir faire métier et l'expertise mise en œuvre.

Dans une telle configuration statique, il devient dès lors plus difficile de tirer parti des avancées en termes de performances ou de mise au point de nouveaux analyseurs. En pratique, *l'évolution rapide* des technologies fait que si l'on doit adapter le système à de nouvelles conditions, on repart de zéro pour développer plus rapidement un nouveau système. Capitalisation et réutilisation sont non valorisées.

Enfin, observons que dans le schéma Figure 15, le *bas niveau d'information* fourni par cet analyseur ne permet pas l'utilisation immédiate des résultats pour décrire un média. Le système ne fournit en effet en sortie que les zones de « mouvements » détectées dans l'image (une personne, une voiture, la pluie…) et la déduction de l'arrière plan de la scène est considéré ici comme immobile.

En conséquence, pour une telle architecture, soulignons que dans le cas d'un mouvement de caméra (induisant par définition un changement de valeurs de tous les pixels de l'image), l'analyseur n'est plus efficace puisque l'image entière est détectée comme étant en mouvement. Ce système ne permet donc pas de suivre le déplacement d'un objet lors d'un effet de caméra (poursuite, zoom, fondu…). La solution serait d'inclure dans le schéma un analyseur permettant de détecter, puis de compenser les mouvements de la caméra. On se retrouve alors face au dilemme suivant : modifier le schéma pour insérer un nouvel analyseur ce qui aura pour conséquence directe de le rendre encore plus complexe ; ou repartir de zéro pour tenir compte d'emblée dans l'architecture de cette nouvelle fonctionnalité. Et ce jusqu'à la prochaine fonctionnalité à ajouter. Cet exemple met pleinement en évidence les verrous énumérés dans l'introduction.

La Figure 16 représente une schématisation fonctionnelle de l'exemple donné Figure 15 à laquelle est ajoutée l'insertion d'interactivité dans le flux multimédia. Cela montre que des architectures existantes peuvent ainsi répondre aux enjeux d'enrichissement de flux multimédia à la volée.

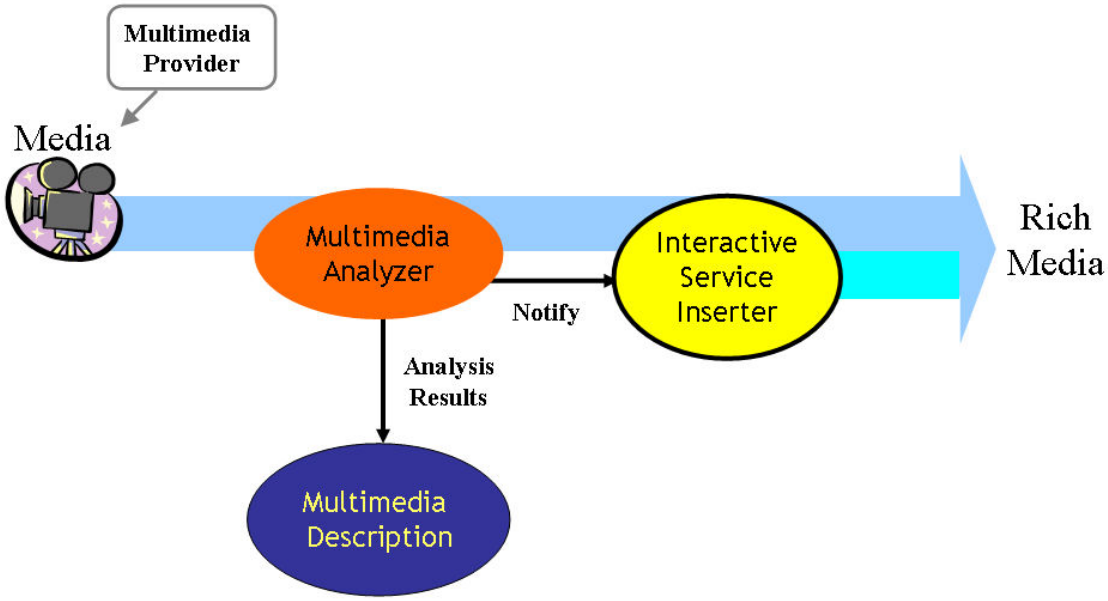


Figure 16 : Schéma fonctionnel construit à partir de l'architecture de la Figure 15

1.1.2. Verrous identifiés et modélisation proposée

Dans cette thèse, nous proposons la nouvelle architecture, RAMSES, (Figure 17) permettant un enrichissement des flux multimédia à la volée.

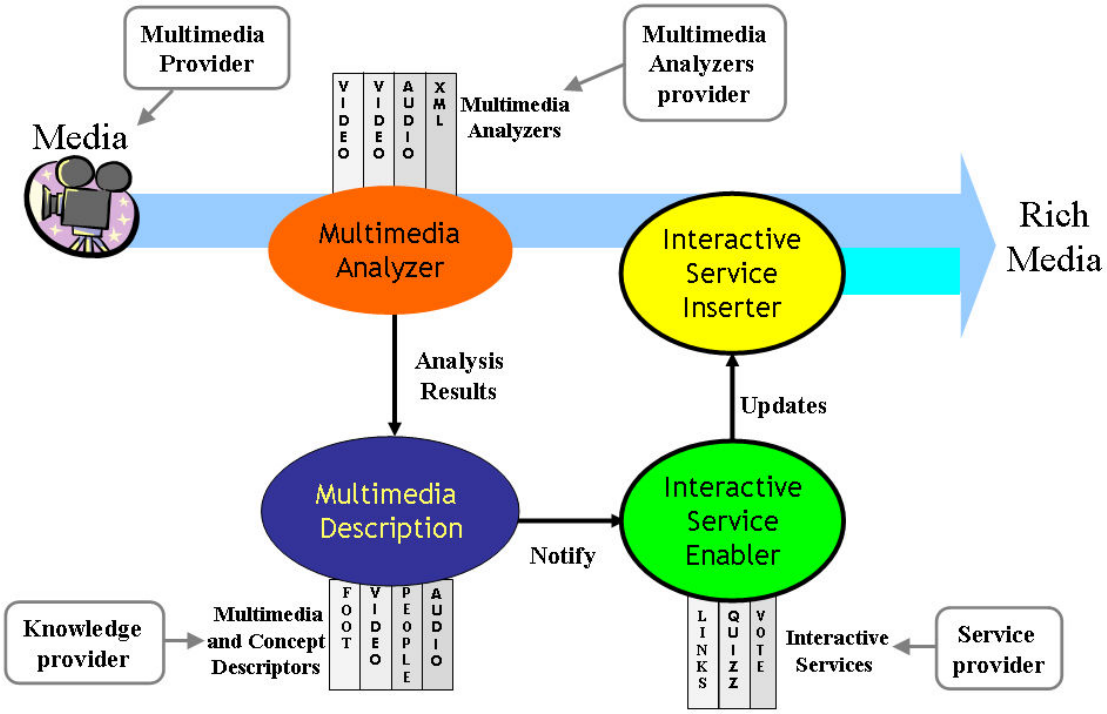


Figure 17 : Schéma de l'architecture RAMSES proposée

Cette architecture lève les verrous technologiques précédemment identifiés, puisqu'elle est :

- *dynamique* : en assurant la flexibilité dans la combinaison des analyseurs, elle rend le système évolutif ;
- capable *d'adapter* le niveau d'analyse en gérant les résultats d'analyse combinant informations de bas et haut niveau sur les médias ;
- enrichie *d'outils de description sémantique* capables de fournir des informations sur les analyseurs eux-mêmes : elle lève donc le verrou sur la description des analyseurs multimédias ;
- *générique* : en assurant la *synchronisation* des différents flux à analyser, elle s'affranchit de la dépendance au type de média et au nombre de flux considérés ;
- *ouverte* : elle permet l'intégration de nouveaux analyseurs dans un contexte de réutilisation/capitalisation ;
- *extensible* : elle *s'adapte* aux nouveaux langages de description et outils.

Détaillons à présent l'architecture RAMSES (Figure 17) au niveau conceptuel en termes de :

- modélisation, notamment des services interactifs, de l'analyse multimédia, de la description du document multimédia et du contextuel virtuel ;
- fonctionnalité et validation.

1.1.3. Etat de l'art et objectifs

L'objectif est de fournir des modèles et fonctionnalités afin d'enrichir automatiquement des flux multimédia à la volée.

L'analyse critique de l'état de l'art scientifique et technologique montre qu'il existe un grand nombre de plateformes d'analyse des documents multimédias avec des objectifs et des clients très différents : depuis les systèmes vidéo de détection d'intrusion jusqu'aux plateformes ouvertes pour l'annotation manuelle des images [Flickr] en passant par les systèmes d'analyse semi-automatique des médias [Schallauer06]. Toutefois, on observe qu'il est toujours nécessaire, pour un besoin particulier, de recréer un système spécifique. Cela traduit à l'évidence un manque de systèmes modulaires ou ouverts qui s'adaptent aux besoins.

RAMSES propose une architecture nouvelle combinant les avantages des domaines, standards, ou architectures qui existent. La Figure 18 détaille ces différents domaines et schématise l'architecture, les communications logiques, les processus et les fonctionnalités mis en œuvre pour lever les verrous précédemment identifiés.

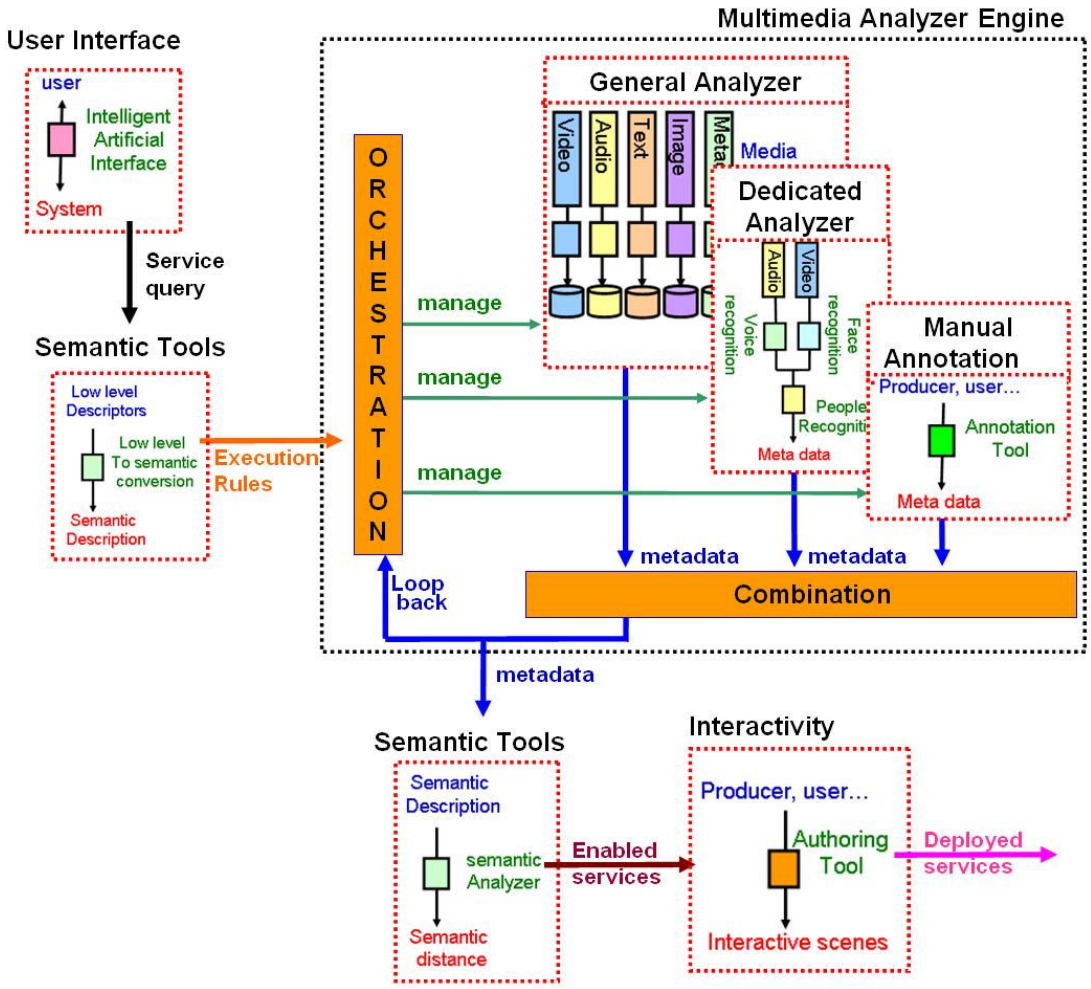


Figure 18 : Schéma des différents domaines et technologies mis en œuvre pour l'architecture RAMSES

L'architecture proposée est prévue pour sélectionner à partir d'un contexte identifié d'un document multimédia une liste de services interactifs à déployer dans le média. L'*interface utilisateur* et les *outils sémantiques* sur la gauche de la Figure 18 représentent l'interface au niveau des fournisseurs de services pour l'enregistrement de nouveaux services interactifs et l'analyse des nouveaux contextes associés à détecter pour les déployer. L'idée principale consiste à segmenter les analyseurs multimédias *complexes* en un ensemble d'analyseurs *simples* hébergés dans l'architecture (*Multimedia Analyzer Engine*). Le système est dès lors conçu comme une architecture de plugins modulaires fondés sur des analyseurs de médias pour extraire des concepts et des relations entre ceux–ci. Enfin, les outils sémantiques et les composants liés à l'interactivité contrôlent le déploiement et le maintien dans le média des scènes interactives.

1.1.4. Description et contributions

L'architecture que nous proposons est ici illustrée dans le cadre de l'insertion automatique d'interactivité dans les médias [Royer07b, Royer08]. Dès lors, les fournisseurs de services sont vus comme les clients de l'architecture, puisqu'ils lui

procurent d'une part les services interactifs à insérer automatiquement dans les médias et d'autre part les conditions de leur déploiement.

La Figure 17 modélise la chaîne fonctionnelle complète de diffusion et d'insertion automatique de services interactifs. Dans un premier temps, les fournisseurs de services mettent à disposition les services interactifs qu'ils souhaitent mettre en œuvre (module *Interactive Service Enabler*) et spécifient les informations nécessaires à leur déploiement. Lors de la diffusion du flux multimédia, le module *Multimedia Analyser* analyse le flux. Les résultats sont utilisés pour vérifier la présence des informations / conditions associées aux services interactifs (module *Multimedia Description*). Cette vérification faite, le cas échéant, les services interactifs sont déployés dans le flux multimédia (module *Interactive Service Inserter*).

La capacité d'analyse de l'architecture est fondée sur les analyseurs de médias. Dans nos développements, nous avons exploité et implanté les analyseurs existants. Notre contribution porte donc sur la capacité de l'architecture à combiner dynamiquement les analyseurs pour extraire les informations nécessaires au déploiement des services interactifs. L'implantation opérationnelle de cette contribution sera détaillée section 2.

1.2. Modélisation des services interactifs

Précisons tout d'abord les définitions et concepts qui sont au cœur de nos recherches.

Une *scène interactive* est un composant logiciel mis en œuvre en étant associé, inséré ou synchronisé avec un contenu multimédia afin de proposer des interactions homme/machine à l'utilisateur final.

Un *service interactif* est un ensemble constitué d'une scène interactive et des paramètres permettant son utilisation. Par abus de langage, on associe souvent la scène interactive au service interactif : par exemple un service de vote lié à une émission de télévision, un service proposant des informations additionnelles sur les personnes présentes dans une scène, ou encore un service de vente en ligne.

Dans le domaine des services interactifs, un *template* est un modèle décrivant la logique fonctionnelle du service. Ces *templates* contiennent le squelette de la scène interactive, les conventions de couleur, l'emplacement des logos, les logos…

Notre objectif est de définir un modèle de description des services interactifs qui permette au fournisseur de services d'ajouter de nouveaux services interactifs de manière modulaire, de spécifier les conditions de déploiement de la scène interactive dans le média, ainsi que de caractériser les besoins en information à extraire du document multimédia.

1.2.1. Description

La Figure 19 illustre la modélisation d'un service interactif.

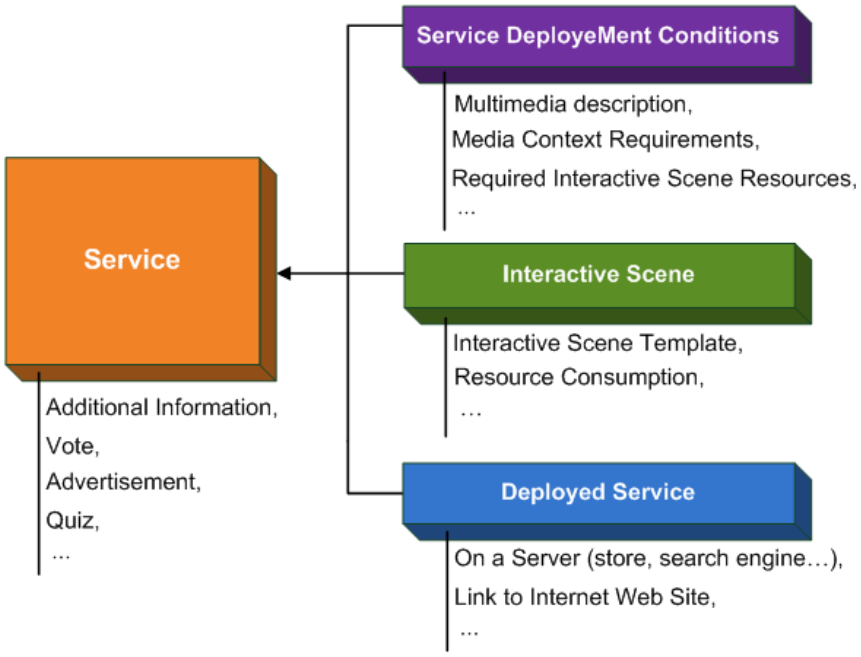


Figure 19 : Modélisation des services interactifs

Un service interactif est complètement décrit par les trois composantes suivantes : conditions d'insertion dans la scène, description du service interactif et enfin paramètres de déploiement.

1.2.2. Conditions d'insertion de la scène interactive

Il s'agit d'un ensemble de descriptions : la *requête du service interactif ainsi que* la *description des ressources nécessaires* pour le déploiement d'une scène interactive.

La *requête du service interactif* permet dans un premier temps au fournisseur du service de définir le contexte multimédia à détecter pour déclencher l'insertion de la scène interactive dans le média. Cette première partie peut être exprimée en langage naturel. « Détecter des personnes dans la scène », « le fond de la scène est une plage » ou encore « détecter un but dans un match de football » sont des exemples de telles requêtes. Cette requête doit en outre définir le ou les documents multimédias à analyser ainsi que la description du multimédia (localisation, URI…) si celui-ci diffère du document multimédia à analyser.

La *description des ressources nécessaires* pour le déploiement d'une scène interactive permet d'estimer les besoins conditionnant le déploiement et le maintien d'une scène interactive dans le média. Cela concerne par exemple la bande passante, la taille ou la position de la scène interactive dans le média. Ces informations permettent à l'architecture de ne pas autoriser l'insertion dans le média de deux scènes interactives qui pourraient se superposer ou saturer la bande passante disponible par exemple.

Ces deux descriptions doivent définir complètement les caractéristiques des scènes interactives ainsi que les conditions à détecter pour déployer celles-ci dans le média.

1.2.3. Scènes et services interactifs

La spécification des scènes interactives passe par l'utilisation soit d'un langage de description de scènes, soit de *templates* de scènes interactives. Ces derniers facilitent l'automatisation de l'insertion et de la mise à jour des scènes interactives. Ils peuvent être dans un format de description de scène MPEG-4 BIFS par exemple.

La spécification du déploiement du service interactif correspond à la description de l'implantation de ce service. Il s'agit ici par exemple d'un serveur de vote, un site de vidéo à la demande, une plateforme d'achat en ligne…

Comme il n'existe actuellement pas de modèle complet d'un service interactif et des moyens de le mettre en œuvre, nous proposons dans cette architecture un modèle de description de ces spécifications s'appuyant sur la structuration illustrée Figure 19.

1.3. Modélisation des analyseurs

Aujourd'hui, les services interactifs sont créés et insérés manuellement dans les documents multimédias. Cependant, les contraintes de temps font que ceux-ci ne sont pas forcément directement liés au contexte du multimédia. Il n'existe pas en effet d'analyseurs capables d'analyser un document multimédia en temps réel pour générer une description sémantique complète de celui-ci. Il est dès lors impossible de produire de façon automatique des documents multimédias enrichis. Cette section décrit les différentes technologies existantes qui nous permettrons d'introduire des solutions.

Nous décrirons dans un premier temps les ressources associées aux utilisateurs en termes de calcul (« human computing »), puis un état de l'art sur les analyseurs, enfin nous présenterons une solution permettant l'analyse automatique des documents multimédias.

1.3.1. Facteurs humains (« *Human Computing* »)

Il s'agit dans ce concept de confier à une personne des tâches à accomplir qu'un ordinateur ne peut pas effectuer. Ainsi, aujourd'hui les jeux vidéo reposent de plus en plus sur la créativité des joueurs. Par exemple, [Spore] est considéré comme un jeu individuel en ligne fondé sur le partage de données de manière asynchrone. Ce jeu vidéo permet à un joueur de contrôler l'évolution d'une espèce depuis son origine en tant qu'organisme microscopique jusqu'à l'exploration interstellaire. A chaque étape, l'utilisateur a la possibilité d'utiliser les outils d'édition du jeu disponibles pour créer du contenu (objets, créatures…) et agrémenter l'expérience de jeu. Le contenu ainsi créé est automatiquement transféré vers une base de données centralisée, référencé, noté pour sa qualité (liée au nombre de téléchargements par les autres utilisateurs) et enfin redistribué pour enrichir l'environnement de jeu des autres joueurs [Wright06].

Le jeu [Little Big Planet] est un jeu de plateforme solo ou multi-joueurs dont l'objectif est la résolution d'obstacles ou d'énigmes. Ce jeu reposant sur un enrichissement produit par les joueurs eux-mêmes approfondit le concept de création par l'utilisateur en fournissant un éditeur complet pour la création de niveaux de jeu et en motivant la créativité des joueurs à travers un concours du « meilleur niveau ». Le

nombre de 900.000 niveaux créés et publiés par les utilisateurs a été atteint en six mois environ et ces différentes versions ont été jouées deux cent millions de fois [LBPWiki].

Après ce succès, d'autres jeux exploitant de plus en plus la participation des joueurs pour l'enrichissement du jeu sont en préparation. Par exemple, le jeu [ModNation] dont la sortie est prévue en 2010 est un jeu de karting fondé sur la création de contenu par les joueurs. Ceux-ci disposeront d'une suite complète d'outils d'édition pour la création et le partage en ligne d'avatars, de voitures, de circuits... comme illustré Figure 20 (image de gauche).

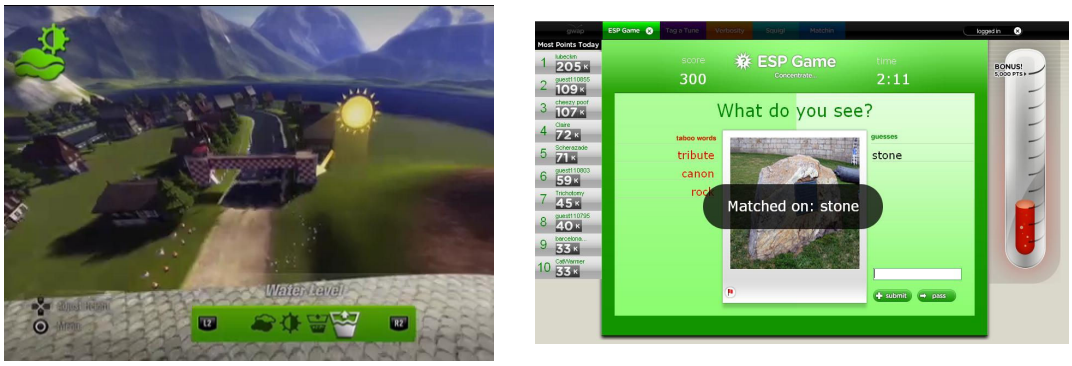


Figure 20 : Exemples d'environnement de jeux exploitant la création de contenus par les joueurs eux-mêmes

[AhnVideo] introduit la capacité de calcul par les utilisateurs à travers les neuf milliards d'heures-hommes passées en 2003 à jouer au jeu *solitaire*. En transférant ces *ressources* au domaine de l'indexation des images, [AhnVideo] réutilise l'environnement de jeu pour permettre aux joueurs de « travailler » (décrire le contenu des images) tout en ressentant l'expérience du jeu. Par conséquent, les *ressources humaines* lorsqu'elles sont mises à profit représentent une puissance de calcul très importante. Le site Internet [Gwap] par exemple regroupe une suite d'outils et de jeux exploitant des personnes comme ressources pour effectuer différentes tâches que les ordinateurs ne peuvent encore effectuer telles que la détection et la reconnaissance de concepts par exemple [Ahn08]. Dans ce contexte, l'entreprise Google a déjà acheté une licence du jeu *ESP game* [WikiESP] en 2006 pour améliorer la base de données de métadonnées sur les images et fournir ainsi de meilleurs résultats via son moteur de recherche [GoogleLabel]. Le principe de ce jeu est la localisation et l'extraction des concepts contenus dans l'image (cf. l'image de droite, Figure 20).

Dans le cas d'un document multimédia évoluant dans le temps (audio, vidéo…), il n'existe pas encore d'équivalent pour l'indexation par des personnes de façon ludique. Toutefois, au regard des statistiques [Junee09] qui recensent 20 heures de contenu audiovisuel déposé chaque minute en 2008, il semble aujourd'hui que les ressources humaines seraient insuffisantes pour analyser un média audiovisuel de

façon complète en temps soit réel, soit différé. Le passage à l'échelle joue ici dans toute sa complexité.

1.3.2. Evolution des analyseurs de médias

Les analyseurs de médias sont développés depuis l'avènement des médias et de nombreuses améliorations ont été apportées même si les résultats et performances d'analyse de chaque type de contenu média (audio, vidéo) sont aujourd'hui hétérogènes.

1.3.2.1. Analyseurs vidéo

Aujourd'hui les analyseurs de vidéos disponibles offrent aussi bien une analyse spatiale [Manerba08, Liu05] que temporelle [Ferman02, Joyce06] du contenu, et sont, à des degrés divers, opérationnels pour détecter et identifier des visages [Kienzle05, Chellappa95].

Détection, reconnaissance et suivi d'objets en général restent en revanche des exemples de domaines encore difficilement opérationnels et sujets à de nombreuses recherches par des approches fort différentes : extraction de points principaux d'une scène pour en construire une représentation vidéo en trois dimensions [Zhou03, Ahmed97, Lee99…], focus sur l'information de mouvement pour détecter des objets [Sifakis02, Amintoosi07, Mech98…] ou sur les techniques d'apprentissage pour la reconnaissance d'objets spécifiques dans une vidéo [Zhang06, Lee04…]… La difficulté reste entière dès lors que l'on s'attaque à des objets quelconques représentés sous diverses formes et sous des conditions variables d'acquisition et de capture, sans compter qu'il s'agit souvent de différentier plusieurs objets simultanément. C'est pourquoi, l'état de l'art opérationnel aujourd'hui consiste en la détection et l'identification de quelques objets simples (non déformables et de texture déterminée) dont toutes les représentations sont connues à l'avance et répertoriées dans une base de données. Toutefois, même dans ce cadre simplifié, les algorithmes qui restent complexes sont appliqués en temps différé sur le média : la contrainte de temps réel n'est donc jamais satisfaite.

Enfin, dans le but de construire dynamiquement la description sémantique d'une scène (actions, principaux objets, personnages…), il est nécessaire d'identifier les relations (« les personnages discutent ») entre les objets et leur environnement (la scène se déroule en extérieur, à paris, à la piscine…). Cet aspect reste une voie de recherche ouverte.

1.3.2.2. Analyseur audio

Les analyseurs de documents multimédias ont orienté leurs recherches durant les 10 dernières années vers l'analyse du média audio. En effet, suite aux difficultés rencontrées dans la description des médias vidéos, l'analyse de la composante audio pour la segmentation en musique/parole/bruit [Pfeiffer96, Lu02] ou encore la conversion audio vers texte [Rabiner78]… permet l'extraction d'informations souvent directement exploitables. Citons, à titre d'exemple, les bonnes performances obtenues pour la transcription texte/parole pour les journaux d'information, mais leurs faibles performances dans le cadre de films d'action riches en fonds sonores, bruits, nombre de personnes qui parlent simultanément…

Toutefois, là encore et dans un contexte très général, l'information contenue dans le seul média audio ne suffit pas pour décrire du contexte du média : la voix en fond n'est pas forcément liée à la personne présente dans la scène, et le discours peut ne pas être relié au contenu…

1.3.2.3. Analyseurs multimédias complexes

Afin d'améliorer la capacité d'analyse d'un document multimédia, l'approche la plus utilisée consiste à combiner les résultats d'analyse de différentes sources de média. Dans [Nefian02, Albiol05], analyses des pistes audio et vidéos sont exploitées conjointement. Ces combinaisons de différents types d'analyseurs permettent dès lors une évolution des descriptions du document multimédia vers un niveau d'abstraction plus élevé. En effet, la combinaison d'un détecteur de visages, combiné à une reconnaissance des voix et une conversion des paroles vers du texte permet l'association du contenu audio aux différentes personnes présentes.

Les premières analyses complètes des documents multimédias sont effectuées via des analyseurs *semi-automatiques* fournissant une description fondée sur l'analyse spatiale (objets) et temporelle (événements) d'un document multimédia [Schallauer06]. Le document multimédia est par exemple découpé en scènes, avec une segmentation des formes en mouvement au sein de chaque scène.

En étendant et généralisant ce principe de combinaison des analyseurs, il est possible d'améliorer les capacités de description d'un système : cela dépendra du nombre d'analyseurs de médias interconnectés en termes tant d'analyse globale qu'en profondeur d'analyse.

Ces analyseurs de média fournissent un niveau de description plus élevé (détection et reconnaissance de visages par exemple), mais créent corrélativement des problèmes pour l'évolution et la maintenance Figure 21.

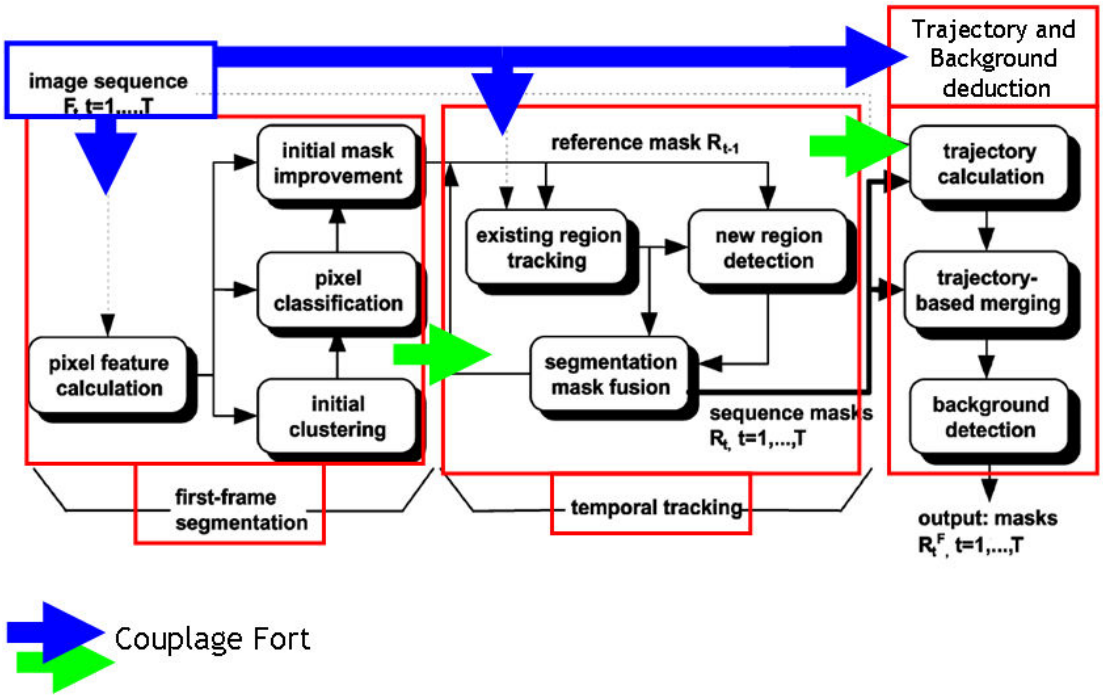


Figure 21 : Schéma d'architecture complexe exhibant les limitations qui en résultent [Mezaris04]

La Figure 21 présente un exemple d'analyseur complexe combinant un ensemble d'analyseurs *simples* pour améliorer le fonctionnement global de l'analyse en termes de détection et de suivi d'objets dans une scène. Sur cet exemple, la difficulté à maintenir le système en cas d'évolution des besoins d'analyse (nouvel analyseur à insérer ou retirer) ou d'un des analyseurs (fort couplage entre les modules) apparaît à l'évidence. Il est dès lors nécessaire de mettre en place une architecture flexible capable de s'adapter aux évolutions des analyseurs en fonction des avancées technologiques.

L'objectif est donc de modéliser les analyseurs de médias afin de pouvoir les réutiliser de manière modulaire et dynamique. Cela nécessite de décrire les entrées, sorties et pré-conditions ainsi que le but de chacun des analyseurs. Par exemple, un analyseur simple ayant comme finalité de « détecter les visages » à partir d'une image en entrée fournira en sortie la position de chacun des visages détectés.

1.3.3. Description

La Figure 22 schématise la modélisation générale de la partie analyse du document multimédia. Elle comporte la représentation des différents analyseurs présents dans le système. Les résultats d'analyse sont utilisés pour maintenir à jour un contexte virtuel de description du multimédia et déployer les services interactifs dont les conditions de déploiement sont validées.

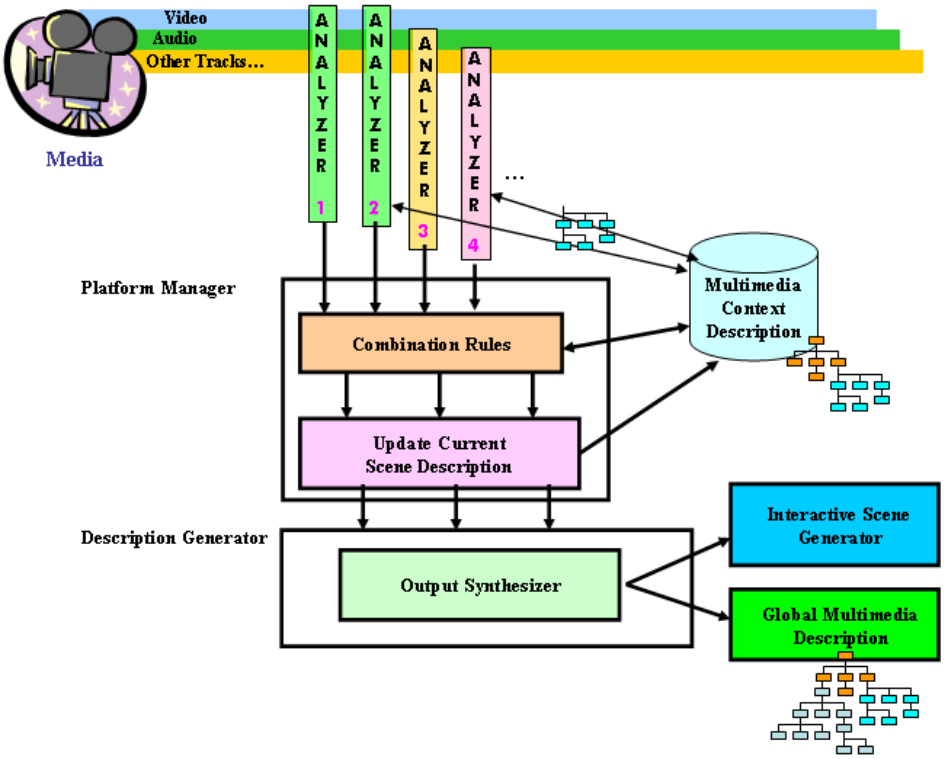


Figure 22 : Modélisation de l'analyse multimédia

La Figure 22 reprend et illustre les concepts de « *contexte virtuel* » et « d'analyseurs de médias ».

1.3.4. Etat de l'art et contributions

Comme mentionné précédemment, les analyseurs de documents multimédias reposent sur une architecture fixe. Il n'est pas possible de modifier l'ordonnancement ou le nombre d'analyseurs à utiliser pour scruter un ou plusieurs médias. Il n'existe par ailleurs pas de description des analyseurs directement réutilisables.

La Figure 23 met en avant les possibilités de capitalisation du grand nombre d'analyseurs multimédias développés à travers le monde.

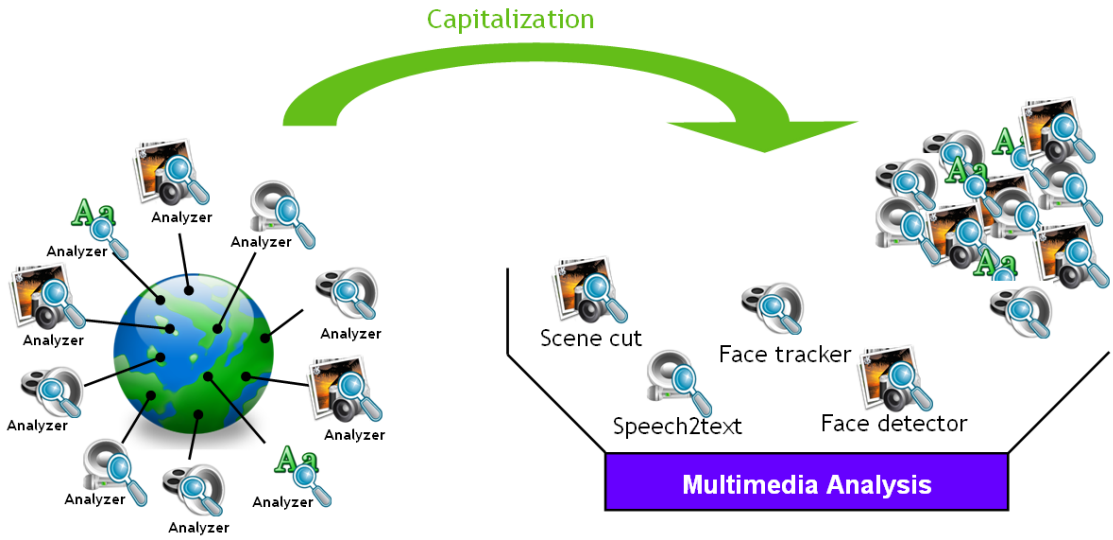


Figure 23 : Diversité des analyseurs de médias disponibles dans le monde

Le principal objectif du système est de permettre la combinaison d'analyseurs et de l'optimiser (en termes d'efficacité de résultat et de complexité minimale). Le principe consiste à décomposer les analyseurs multimédias complexes en analyseurs élémentaires de façon modulaire. Pour y parvenir, nous proposons une architecture (Figure 24) en plugin des analyseurs.

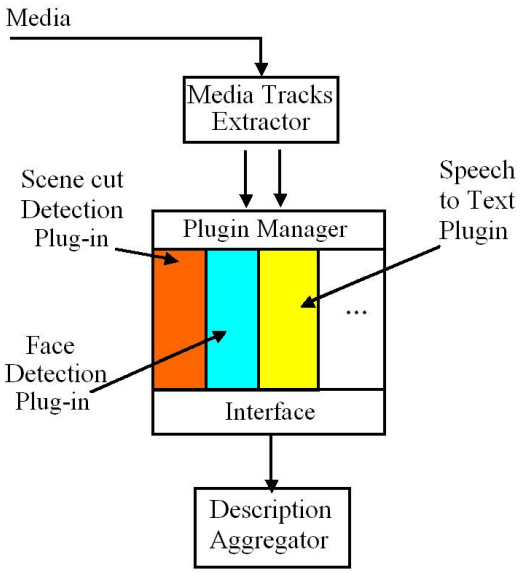


Figure 24 : Architecture en plugins pour la gestion des analyseurs

Comme illustré Figure 24, l'encapsulation des différents modules est nécessaire : cette modularité permet un fonctionnement dynamique et adaptatif de l'architecture.

Pour mettre en œuvre cette architecture en plugins, nous proposons un modèle de description des analyseurs (Figure 25) qui puisse être évolutif.

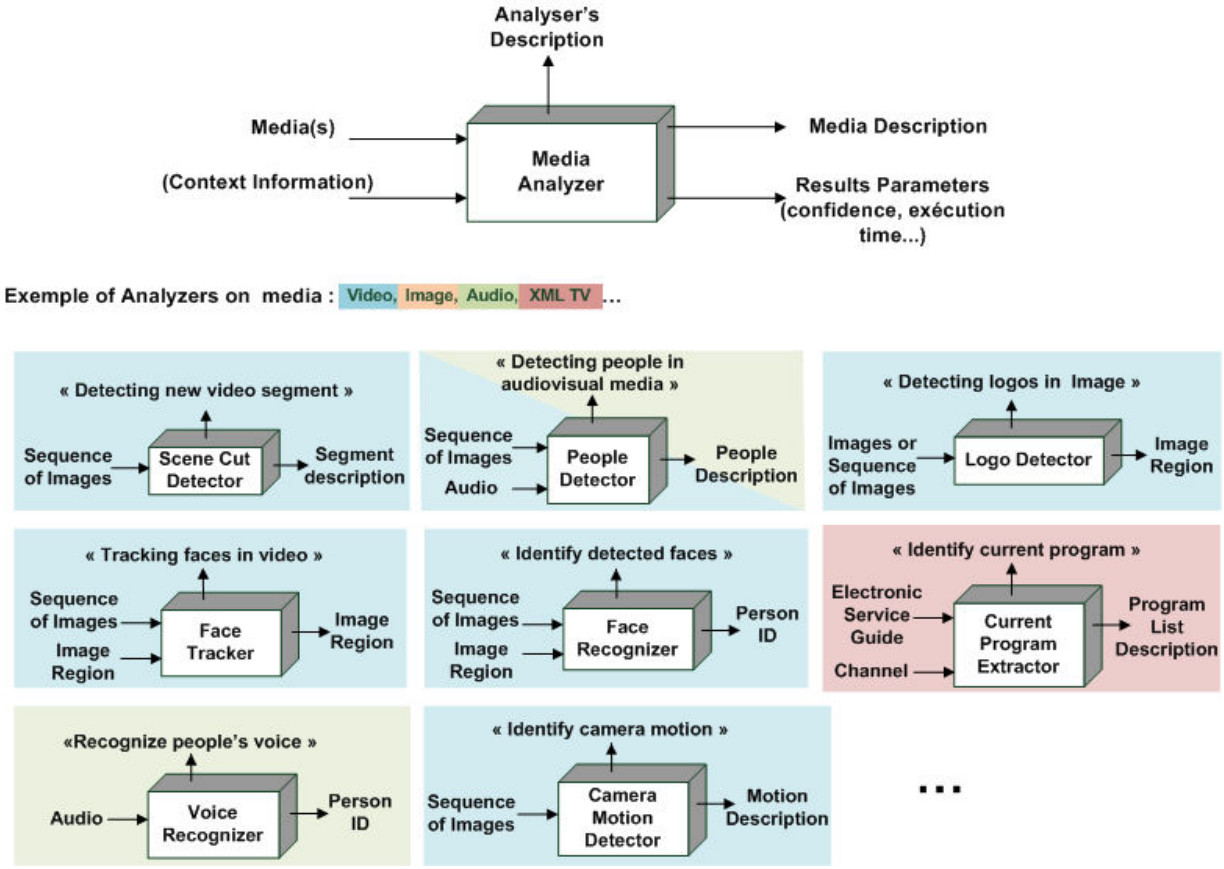


Figure 25 : Modélisation et exemples d'analyseurs de médias

Les entrées (*Inputs*) définissent les types de médias pour lesquels l'analyseur est capable de fournir des descripteurs en sortie (*Outputs*). Le but de l'analyseur (*Goal*) permet de décrire la fonction d'analyse.

1.4. Modélisation de la description d'un document multimédia

Ce système est modulaire puisque l'utilisateur peut choisir les modules d'analyseurs média à insérer dans le système afin de générer la description multimédia contextuelle souhaitée. En outre, il est facilement extensible en modulant le nombre et les versions des différents modules d'analyseurs.

1.4.1. Modularité et agrégation de sources multiples d'information

Cette architecture modulaire permet également de combiner des descriptions générées avec des descriptions multimédia existantes (manuel ou descriptions précédemment générées). Finalement, comme chaque fournisseur de services peut avoir des besoins redondants de descripteurs (détection d'objets en mouvement, identification de fond...), le système combinera des analyseurs de faible complexité pour obtenir une description multimédia de haut niveau.

Cette modularité permet le choix contextuel des analyseurs de média à utiliser aussi bien en termes de description à fournir pour le choix du type d'analyseur que pour les limitations de ces analyseurs pour le choix parmi plusieurs analyseurs de même type mais de technologies différentes.

L'avantage de ce système est la possibilité de combiner des analyseurs complémentaires avec différents niveaux d'abstraction. Il est possible de combiner les analyseurs multimédias avec différents types de ressources extérieures. Dans le cas de la télévision interactive, des éléments de l'architecture permettent d'obtenir beaucoup d'information sur les médias. Les guides de programmes ou guide des services électronique (*EPG* et *ESG*) permettent l'accès à des informations sur les programmes diffusés comme le temps, le titre, la chaîne, le genre… Enfin, les formats de description propriétaires déjà générés, les forums, les pages d'accueil web… fournissent des informations sur les documents multimédias.

1.4.2. Répartition des responsabilités

Nous proposons de séparer de l'architecture générale tous les algorithmes qui génèrent des résultats de description. La Figure 26 représente le rôle des analyseurs multimédias dans l'architecture proposée.

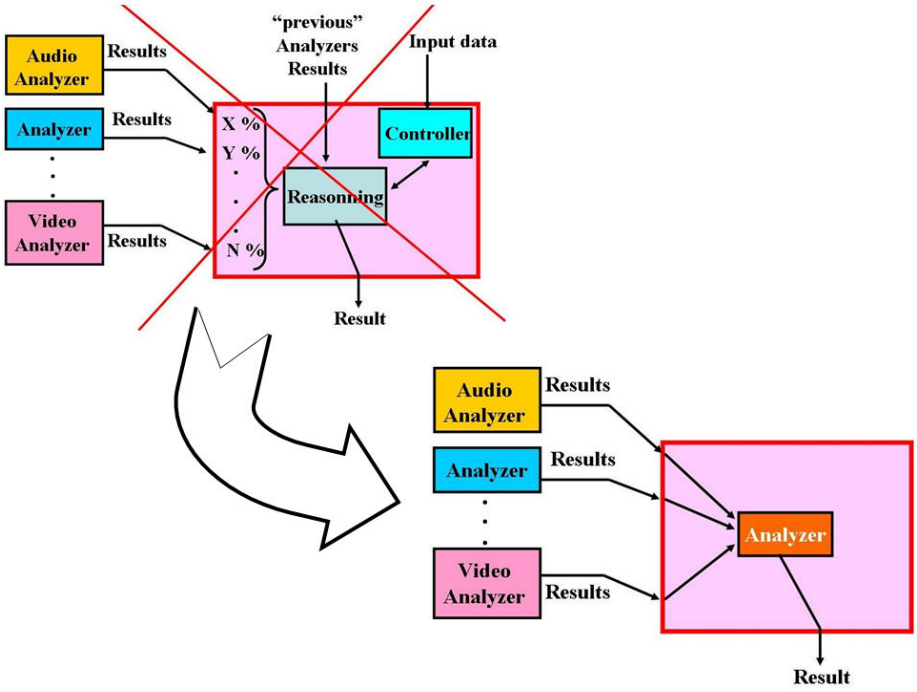


Figure 26 : Solution permettant de gérer les lacunes sémantiques

Comme *l'adaptabilité* de l'architecture proposée repose sur un couplage faible entre les modules, nous encapsulons chaque algorithme dans un module indépendant défini par ses entrées, ses sorties et ses capacités de description.

1.5. Métadonnées pour décrire un document multimédia

Comme détaillé dans [Nack04] et [Nack05], les métadonnées sont des "données sur d'autres données" de toute sorte dans tous les médias : données d'archivage, descriptive, de production, d'affaires, techniques, de postproduction, de programmation, de marketing, d'identification, etc.… Les métadonnées constituent un document de données sur les éléments de données ou attributs (nom, taille, type de données…), des données sur des dossiers ou des structures de données (longueur, champs, colonnes, etc.) et les données sur les données (localisation, propriété…). Les métadonnées peuvent inclure des informations descriptives sur le contexte, la qualité ou les caractéristiques des données. Elles peuvent être enregistrées selon différentes granularités.

Selon [Wendler99], les métadonnées pour les objets d'information digitaux, incluant les vidéos, peuvent être associées à l'une des trois catégories suivantes :

- Descriptive : facilitant l'identification et l'exploration de la ressource.
- Administrative : aidant à la gestion des ressources au sein d'une collection.
- Structurelle : associant entre eux les composants d'objets d'information.

Les métadonnées sont principalement organisées en fonction des exigences de domaine (tags *ID3* pour des fichiers audio « mp3 », *EXIF* pour des images « jpeg »...). Les métadonnées sont hétérogènes, mais ne sont pas interopérables. Ci-dessous sont listés de façon non exhaustive différents formats de métadonnées disponibles:

- consortium W3C : *RDFS*, *OWL*, *DAML*.
- normalisation ISO/MPEG : *MPEG-7*.
- divers : *XMP* (Adobe extensible Metadata Platform base sur RDF).
  FOAF (description d'une personne)
  IPTC (description sémantique d'une image).
  EXIF (description technique d'une image).
  XHTML.
  EUDICO Annotation Format.
  …

Des outils pour convertir un format en un autre sont également disponibles :

- Convertisseur de *EXIF* / *IPTC* / *ESW* / *XMP* / *Flickr* à *RDF*
- Convertisseur de *Flickr* à *FOAF*
- …

Dans les domaines de l'audiovisuel comme introduit dans [Wactlar02], les fournisseurs de contenu sont très intéressés par les standards de description des documents multimédias depuis l'apparition de la numérisation des contenus et les équipements permettant l'enregistrement simultané de descripteurs tels que les dates, les durées, les lieux...

1.5.1. Description textuelle des modules MPEG-4

OCI (Object Content Information) est une "première version" d'un format de description textuelle défini dans la norme MPEG-4 pour transporter dans le flux multimédia les informations sur les médias objet en général : un ensemble de descripteurs OCI (*Rating Descriptor, Language Descriptor, Media Time Descriptor, OCI Creator, Name Descriptor*…) et un format OCI [MPEG-4]. OCI a été créé à une époque où il n'y avait pas de format disponible pour l'insertion de métadonnées dans le format de fichier vidéo. Aujourd'hui, ce format est remplacé par le standard [MPEG-7Ov] offrant plus de possibilités.

1.5.2. Standard MPEG-7

La norme [MPEG7] est un format de description pour décrire les contenus multimédias. Avec l'augmentation exponentielle du nombre de documents multimédias générés, il est nécessaire d'être capable de retrouver un média facilement comme cela est possible aujourd'hui pour les documents textuels avec les moteurs de recherche (Copernick, Google Desktop...). La norme MPEG-7 permet la description des contenus multimédias dans le cadre d'un large éventail d'applications [MPEG7Ov] telle que l'archivage, l'accès, la navigation, la recherche et la gestion de l'information audiovisuelle. MPEG-7 propose un ensemble de descripteurs audio-visuels au format XML compréhensible par l'homme. Le standard MPEG-7 est par ailleurs subdivisé en « profils », ces sous-ensembles du standard de métadonnées permettent de réduire le domaine d'application. Plusieurs profils ont

été examinés pour la normalisation et trois profils ont été normalisés (ils constituent la « *partie 9* » de la norme, leurs schémas XML sont définis dans la « *partie 11* »). Le modèle de description MPEG-7 peut également être mis à jour et consulté de manière dynamique. Les informations échangées entre les différents modules (et terminaux) nécessitent de maintenir à jour le *profil*. Cela peut être fait directement à l'aide de fichiers XML au format MPEG-7. MPEG-7 permet une description complète de contenus multimédias en offrant un ensemble de descripteurs normalisés très riche comme l'illustre la Figure 27. En fonction des besoins de l'application, il est possible d'instancier seulement des sous-parties de ces descripteurs. Les descriptions MPEG-7 peuvent de plus contenir des informations décrivant la création et le processus de production du contenu (réalisateur, titre, etc.), les informations liées à l'utilisation du contenu (les pointeurs de droit d'auteur, l'historique d'utilisation...), des informations sur la classification (Evaluation des parents), etc.

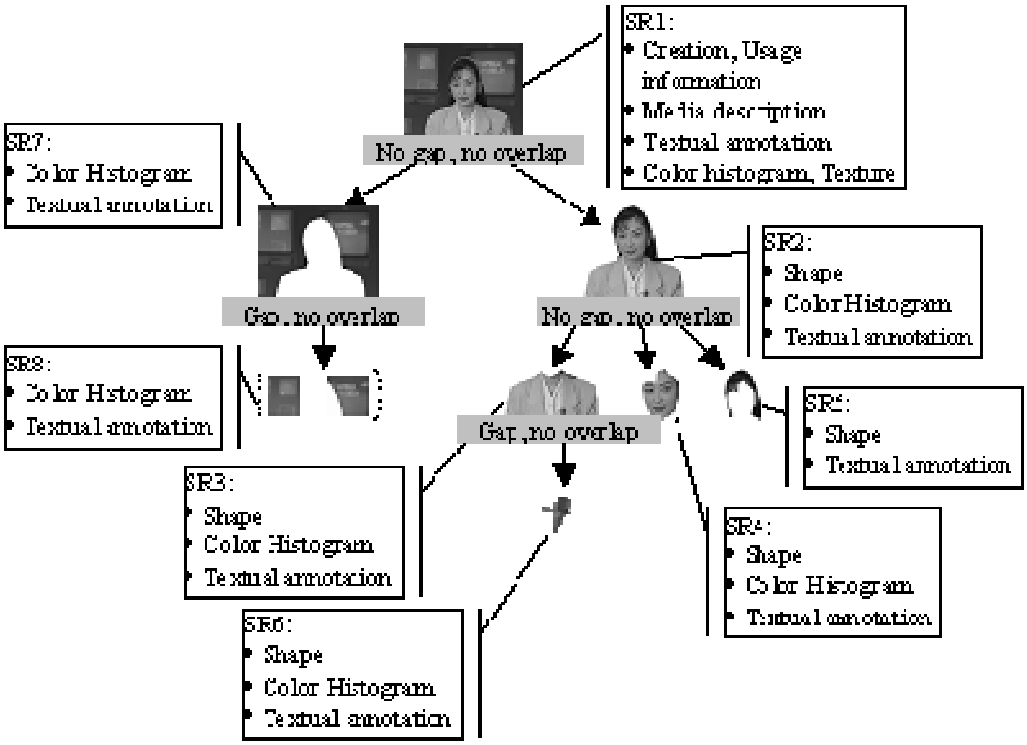


Figure 27 : Exemple d'un schéma de description d'une scène à l'aide de MPEG-7 [MPEG7Ov]

Pour la description temporelle d'un média, celui-ci est divisé en plusieurs niveaux de détails. On retrouve cette notion dans le schéma de la Figure 28. Un ensemble d'images provenant d'une même caméra sans interruption forment un plan (*shot*), un ensemble de plans d'un même contexte forment une scène et enfin l'ensemble des scènes forment le média.

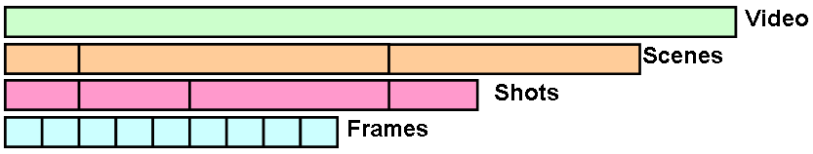


Figure 28 : Schéma de découpage temporel d'un média vidéo

Pour la description spatiale, un média peut être décrit par le lieu où la scène est enregistrée (intérieur, extérieur, dans une ville, dans le désert, etc.), par la position des objets dans la scène (voiture, arbre, montagne), etc.

Toutefois, il n'est pas possible de déduire des descriptions de haut niveau sémantique directement à partir de descripteurs de bas niveau. La Figure 29 par exemple peut être décrite avec des descripteurs de bas niveau de la façon suivante : « La couleur dominante de la scène est le vert ; une forme circulaire en bas de l'image constituée de parallélogrammes blancs et noirs ; un rectangle blanc non rempli constitue le fond de la scène ». Une autre description valable de cette image est : « Le ballon qui se trouve à 20 mètres du but est sur le point d'être joué" ».

Figure 29 : Exemple pour illustrer une description de bas et haut niveau

Comme détaillé ci-dessus, la description de bas niveau, tel que le standard MPEG-7, permet de décrire précisément une partie spécifique des médias (les mouvements de la caméra, les couleurs principales, les objets en mouvement...).

1.5.3. Limitations de la description bas niveau d'un document multimédia

La Figure 30 montre un exemple de description de bas niveau, incluant les caractéristiques du contenu tels que : les couleurs, les textures, les timbres sonores, la description de la mélodie, etc.

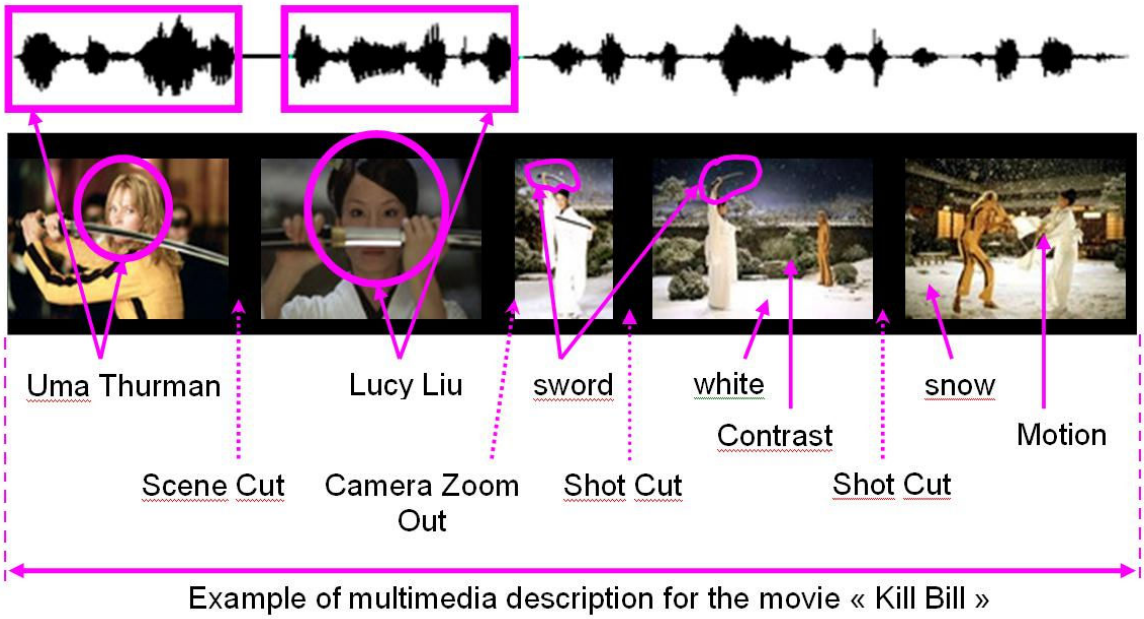


Figure 30 : Exemple de description d'une scène d'un multimédia

On remarque qu'il n'est pas possible de décrire l'action en cours dans la scène avec précision: «Uma Thurman et Lucy Liu se battent en utilisant des épées ». Par conséquent, le groupe de travail sur MPEG-7 a ajouté la gestion d'une description sémantique afin de permettre un plus haut niveau de description du média. Cependant, le principal inconvénient de MPEG-7 réside dans la possibilité de décrire les mêmes concepts utilisant différents descripteurs comme détaillé dans [Arndt07]. A titre d'exemple, il existe différentes possibilités pour décrire une image (<Structured Annotation>, <text>, <TextAnnotation>...). Par conséquent, une requête sur un élément spécifique peut ne pas retourner les résultats escomptés. Il est dès lors impossible de surcharger directement le schéma de description MPEG-7 avec des descriptions sémantiques afin d'utiliser des moteurs d'inférence.

1.6. Sémantique et ontologies

De nombreux efforts ont été effectués jusqu'ici dans le domaine de la sémantique sur les ontologies ou les moteurs d'inférence [Lacot05, Antoniou04, Passin04...]. Nous présentons dans cette section les principaux formats et leurs liens avec l'hypermédia. Comme détaillé dans [Huang07], une ontologie est une représentation formelle d'un ensemble de concepts dans un domaine ainsi que des relations entre ces concepts. L'ontologie est utilisée pour raisonner à propos des objets du domaine concerné.

La spécification [TV-Anytime] par exemple est un format d'abord dédié aux services télévisuels. Cette spécification offre très peu d'informations de bas niveau, mais offre un large panel d'informations d'un niveau sémantique supérieur (par exemple le titre, le synopsis, le genre, les crédits, etc.).

1.6.1. RDF

A l'inverse, le *Resource Description Framework* (RDF), comme son nom l'indique, est un *framework général* pour la description et l'échange de métadonnées. Les déclarations RDF définissent les relations entre les concepts (nœuds du graphe) qui sont des sous-parties du domaine sémantique. Comme détaillé dans [Bray98], le *framework* RDF est construit sur les règles suivantes:

- Une *Ressource* est tout ce qui peut avoir une URI. Cela inclut les pages Web du monde entier, ainsi que des éléments individuels d'un document XML ;
- Un *PropertyType* est une *Ressource* qui a un nom et peut être utilisé comme une propriété, par exemple Auteur ou Titre ;
- Une *Property* est la combinaison d'une *Resource*, d'un *PropertyType*, et d'une *value ;*
- Il existe une méthode simple pour exprimer ces propriétés abstraites en XML.

L'environnement a évolué et une introduction plus récente en RDF peut être trouvée dans [Tauberer06]. À titre d'exemple, dans la description suivante: *« A man is driving the taxi »*, le triplet est constitué d'un sujet : *man*, d'un objet : taxi, et d'un attribut : drive. La description correspondante en RDF / XML pourrait ressembler à la description illustrée Figure 31.

```
<rdf:Description rdf:about=« http://person.org#man »>
<drive>
      <rdf:Description rdf:about=« http://car.org#taxi »>
</drive>
</rdf:description>
```

Figure 31 : Exemple de description RDF

Cependant, il ya encore des limitations qui empêchent d'utiliser cette norme pour développer un *framework* de description. Comme RDF ne précise pas de mots clés à utiliser pour définir les triplets, le raisonnement est nécessaire pour résoudre la synonymie ou l'équivalence entre les concepts. De plus, une attention particulière est requise lors de la définition du domaine car il n'y a pas de distinction entre les schémas et les données dans les triplets. Enfin, la simplicité des expressions exige un nombre plus élevé de triplets pour éviter toute ambiguïté, mais il introduit des inconvénients en termes de volume pour la lecture et les calculs.

1.6.2. RDF-S

RDF for Schema (RDF-S) tente de résoudre ce problème en fournissant un modèle de données. Par exemple, FOAF [Brickley07] est un exemple de RDF-S qui fournit un modèle pour décrire une personne. Comme illustré dans [RDFS_Wiki], une instance de la classe *foaf: Person* est une ressource liée à la classe en utilisant un *rdf: type* comme prédicat, comme dans l'expression formelle de la phrase suivante en langage naturel : «Jean est une personne». Néanmoins, RDFS reste principalement un format « catalogue » créé dans le contexte XML.

Depuis que nous avons besoin de créer, d'enrichir, et de représenter des informations sur les documents multimédias, le langage *Ontology Web Language* (OWL) est le plus expressif qui ait été créé. Celui-ci pourrait être la norme à utiliser car il permet le traitement du contenu de l'information. Toutefois, de nombreux composants RDFS sont inclus dans ce langage.

1.6.3. OWL

Comme détaillé dans [OWL04], OWL permet une meilleure interprétation par les systèmes du contenu Web que ceux proposés par XML, RDF et RDF Schema (RDF-S) en fournissant un vocabulaire supplémentaire avec une sémantique formelle. OWL propose plus de vocabulaire pour décrire les propriétés et les classes: les relations entre les classes (disjointes, par exemple), la cardinalité (« un seul caractère » par exemple), l'égalité, un typage plus riche des propriétés, les caractéristiques des propriétés (la symétrie par exemple)…

OWL a trois niveaux croissants d'expression. Le plus simple est *OWL Lite*, celui-ci convient aux utilisateurs qui ont besoin avant tout d'une hiérarchie de classification et de contraintes simples. *OWL-DL* propose une solution pour les utilisateurs qui souhaitent une expressivité maximale, tout en conservant l'intégrité du calcul (toutes les *conclusions* sont garanties d'être calculable) et de décidabilité (tous les calculs se terminent en un temps fini). Enfin, *OWL Full* est destiné aux utilisateurs qui souhaitent un maximum d'expressivité et la liberté syntaxique de RDF mais sans garanties de calcul. Le développent d'ontologies se compose de 6 volets [Isaac05] :

- *Spécification :* Spécifier l'utilisation de l'ontologie;
- *Conceptualisation :* Construire un modèle de données en fonction du domaine de connaissances ;
- *Formalisation* : Conversion du modèle conceptuel au modèle formel ;
- *Intégration* : Réutilisation autant que possible des ontologies existantes ;
- *Implantation* : Construire des modèles compréhensibles ;
- *Maintenance* : Maintenir l'ontologie à jour.

Des outils similaires tels que *Protégé* [Dong07] permettent la construction d'ontologies comme détaillé dans [Horridge04]. L'étape suivante consiste à vérifier la syntaxe et la cohérence de l'ontologie en utilisant *WonderWeb OWL Ontology Validator* [Bechhofer04] pour faciliter l'interopérabilité et le raisonnement.

1.6.4. Modèle de description sémantique multimédia

En théorie, les ontologies et la sémantique apportent à la description multimédia d'énormes capacités d'enrichissement de la description. Nous devrions être capables de décrire le contenu multimédia et de créer une représentation « virtuelle » du contenu multimédia au moyen des descripteurs. Cependant, nous ne sommes pas encore en mesure de regarder un film en « rejouant » cette description sémantique. Ainsi, il n'est pas possible de retrouver une scène dans une liste du document multimédia en utilisant un langage de description naturel comme « Tous les médias où Bill se bat avec son ami ». Cela est dû au fossé sémantique (Figure 32).

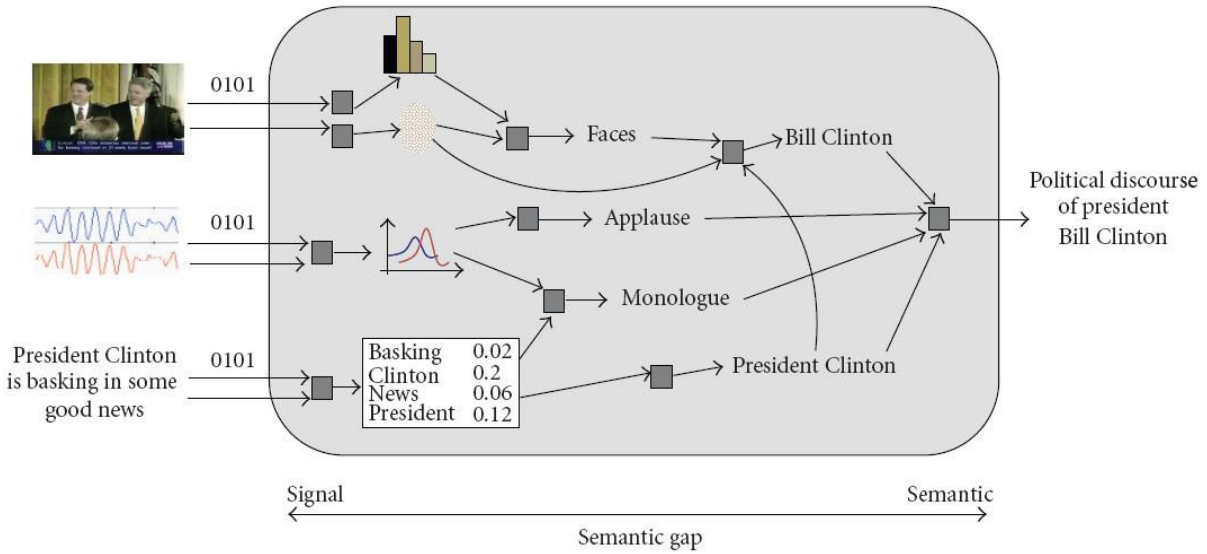


Figure 32 : Illustration du fossé sémantique [Ayache07]

Le fossé sémantique est un problème bien connu dans le multimédia [Smith07]. Par exemple, Google a amélioré son moteur de recherche d'images en déployant une solution de traitement fondée sur les utilisateurs pour décrire les images web [GoogleLabel].

La sémantique fournit des solutions et un grand nombre de personnes utilisent déjà des outils incluant de la sémantique sans le savoir. À titre d'exemple, Microsoft a déployé *Embodied Conversational Agent* (ECA) dans Microsoft Office Assistant [Cassell01] et plus récemment dans l'aide aux utilisateurs *FAQ*. La solution ECA s'appuie sur un raisonneur sémantique pour classifier les demandes des utilisateurs, analyser automatiquement les mots-clés et établir les relations permettant de trouver des sujets ou des liens à renvoyer aux utilisateurs.

1.6.4.1. Descripteurs de bas niveau vers des concepts de haut niveau

Le principal problème est la combinaison de descripteurs à différents niveaux de granularité pour construire une description virtuelle du contexte multimédia [Delezoide06]. Sur la base des travaux de [Martin98] et [Bennett02], la fusion de concepts de haut niveau et des concepts de classification n'est pas possible en

raison de la quantité et de la complexité des informations et des relations à considérer et à maintenir. En outre, la difficulté d'utiliser des concepts pour décrire des descripteurs de bas niveau comme les couleurs est également impossible à maintenir. La fusion des concepts et des descripteurs de bas niveau permet d'obtenir des descripteurs de bas niveau décrivant un objet (Figure 33).

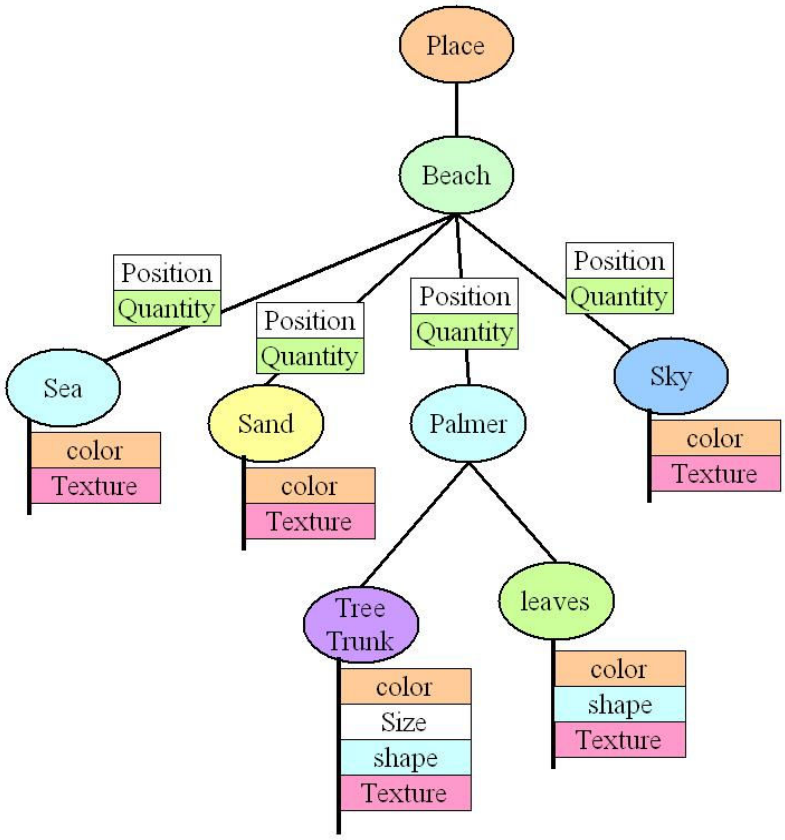


Figure 33 : Exemple d'ontologie décrivant une plage

[Delezoide06] introduit la solution « *Late Fusion* » (fusion tardive) proposée par [Luo01], qui repose sur un réseau bayésien où les descripteurs sont tenus d'être indépendants les uns des autres (les résultats d'un descripteur ne fournissent pas d'information sur un autre descripteur). En outre, des différents concepts peuvent donc être déduits de la même combinaison de descripteurs. Afin de compenser ces manques, la solution *Multinet* proposée par [Naphade00] présente des relations positives ou négatives entre chaque concept. Par conséquent, comme représenté Figure 33, le concept de plage aura un *lien positif* avec d'autres concepts comme la mer, le sable, le ciel... mais un *lien négatif* avec le concept d'intérieur. La détection du concept plage réduit la probabilité de concepts d'intérieur. Une fois encore, le principal problème introduit avec cette dernière solution est l'obligation de décrire les relations entre tous les concepts. Cette solution est difficile à maintenir et nécessite une énorme quantité de calcul pour réaliser le raisonnement... La combinaison de la solution *late fusion* et de la solution *Multinet* est ainsi présentée comme une solution temporaire.

1.6.4.2. Gap sémantique

[Rui07] présente un exemple de recherche dans ce domaine. L'idée principale, nommée *paradigm annotation*, est d'étiqueter manuellement un ensemble de concepts présents dans un contenu multimédia (image, vidéo...) pour établir des relations entre les descriptions de bas niveau et les concepts de haut niveau. Il est donc possible de créer un concept de détecteur permettant d'extraire des concepts connus dans d'autres contenus multimédias.

La modélisation de la description d'un document multimédia permet de fournir une description sémantique de celui-ci à différents niveaux : bas, haut ou très haut niveau. Il faut donc gérer les enjeux du gap sémantique. Ce dernier définit la difficulté de mesurer la similarité de deux concepts exprimés dans des langages différents. Ce problème se retrouve ainsi dans la conversion d'une description de bas niveau (numérique) vers une description au haut niveau (symbolique).

La description d'un média correspond à des concepts élémentaires et/ou composés. Les concepts élémentaires ou « bas niveau » correspondent à des champs descriptifs indépendants les uns des autres. Par exemple, les zones de couleurs dominantes, l'histogramme des couleurs d'une image, le nom de l'auteur du média...

Les concepts composés correspondent à la mise en relation de champs descriptifs. La valeur de ces champs peut être spécifiée ou non. La nature des relations (« est composé de », « appartient à », « est le fils de »…) peut également être spécifiée ou non.

1.6.5. Etat de l'art et limitations

La Figure ***34*** présente un schéma synoptique des différents formats de description disponibles et de leurs *relations*.

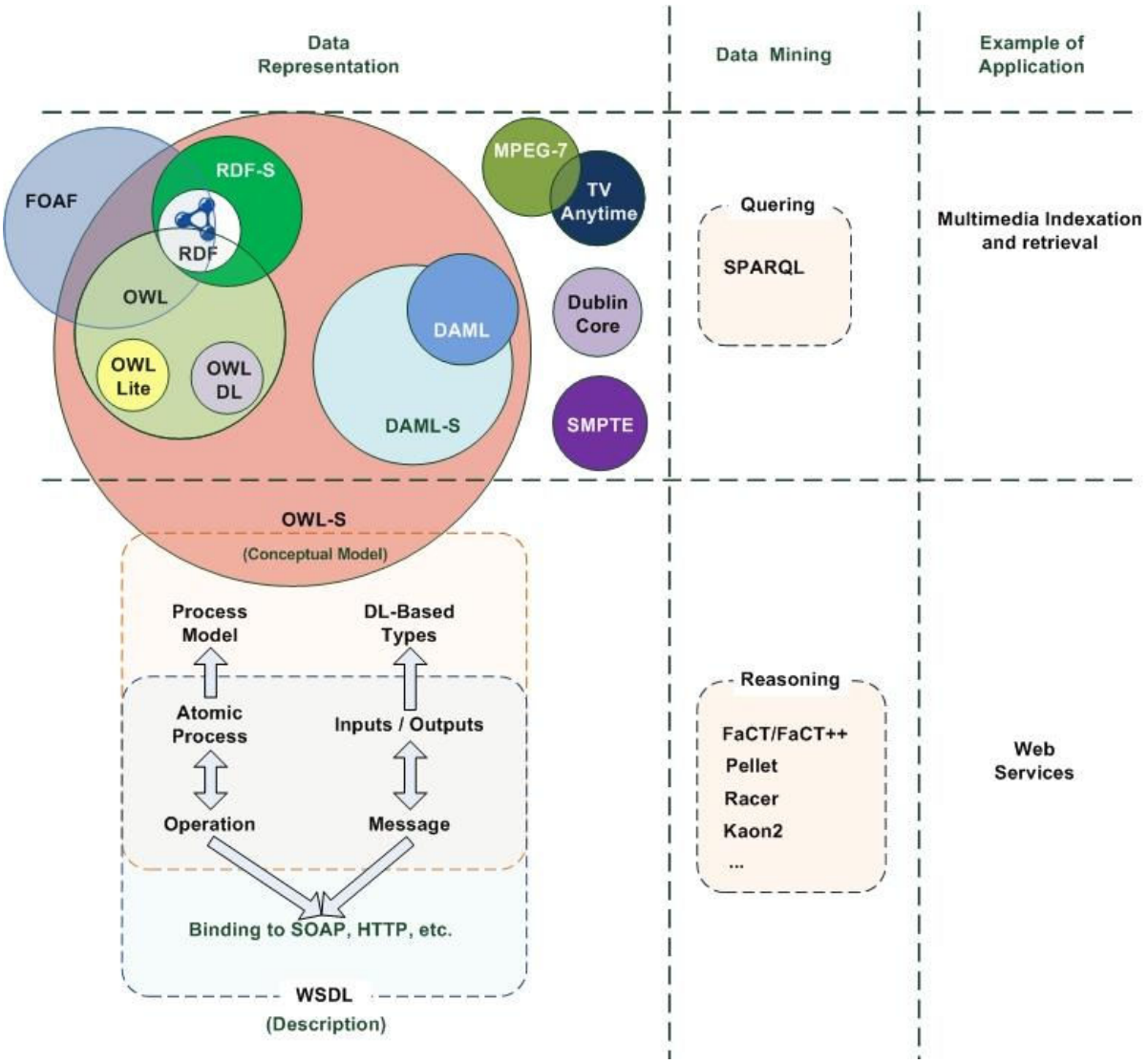


Figure 34 : Synopsis des différents formats de description et leurs interrelations

La Figure *35* est une synthèse de l'état de l'art concernant la vision sur la description d'un document multimédia. L'idée générale qui en ressort consiste tout d'abord à exploiter les descripteurs de bas niveau pour relier la description au contenu du média.

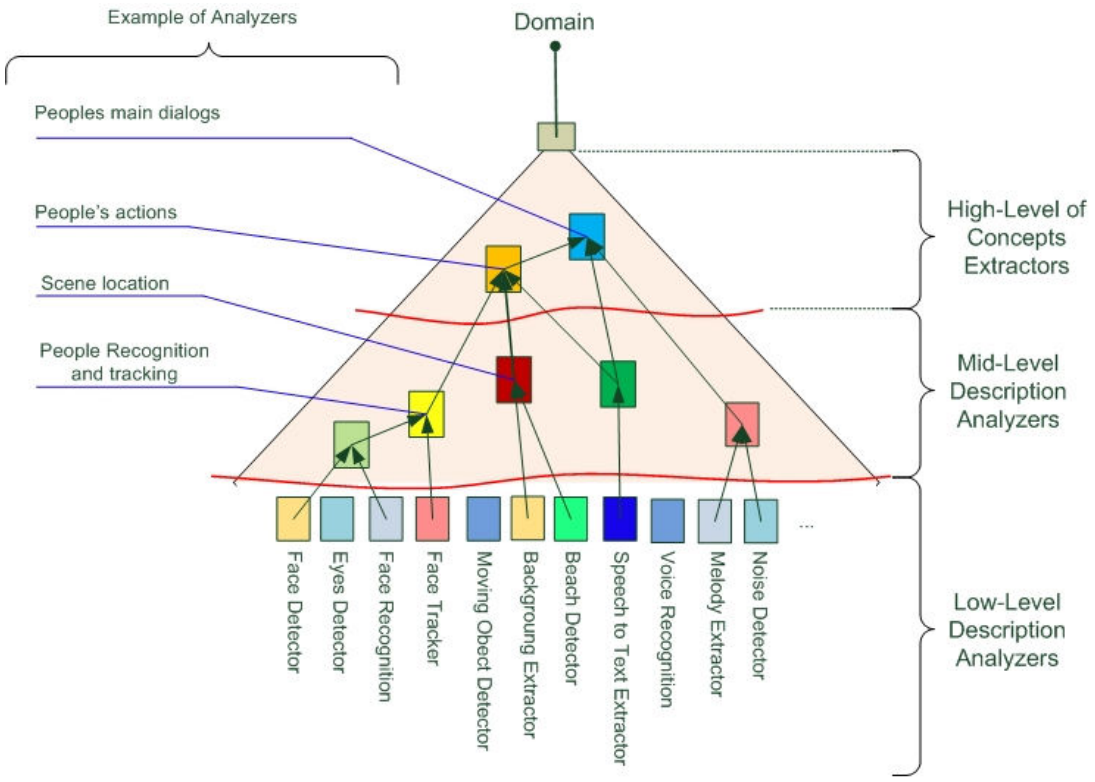


Figure 35 : Modélisation d'un exemple de l'analyse d'un domaine en fonction des différents niveaux d'abstraction

Ensuite, on réalise des combinaisons de ces descripteurs pour monter en niveau d'abstraction. De cette façon, en combinant un analyseur de détection de visages à celui de la reconnaissance des visages, le système peut fournir les noms des personnes détectées. Toujours dans le même esprit, il est possible de combiner ces résultats avec ceux d'un analyseur de voix pour associer les discours aux différentes personnes détectées et reconnues…

D'une part, nous avons donc des descripteurs de bas niveau possédant des relations hiérarchiques explicites (« arbre »). D'autre part, les descriptions de concepts de haut niveau ont des relations implicites, contextuelles et subjectives, non-hiérarchiques dont la signification se déduit par des inférences au travers d'un moteur de raisonnement.

1.6.6. Contributions

Nous proposons une architecture modulaire qui permet l'insertion des deux principales approches descriptives : par descripteur de bas niveau et par ontologies afin d'offrir une capacité de description d'un plus haut niveau sur les concepts à détecter.

Nous proposons par ailleurs d'introduire la logique métier (autour des analyseurs) et de faire intervenir un utilisateur (*Media Analyser Provider*) dans la préparation des

choix associés à cette logique métier, afin de diminuer fortement la complexité comparée à la complexité du passage des descriptions bas niveau à haut niveau.

Après analyse de l'état de l'art, et pour garantir l'interopérabilité des services et des applications, en même temps que le caractère ouvert de notre architecture, nous avons choisi d'exploiter le modèle de description proposé par la norme MPEG-7.

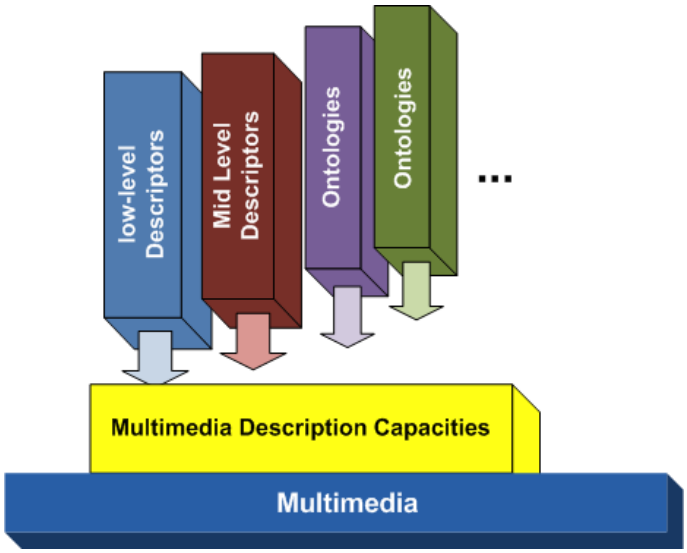


Figure 36 : Modèle de description extensible

Toutefois, nous sommes bien conscients que la modélisation proposée ne résout pas le problème du gap sémantique dans l'analyse du média. Notre objectif est de déporter le problème du gap sémantique au niveau des analyseurs tout en leur fournissant le maximum d'information pour le résoudre. En effet, seuls les analyseurs ont le pouvoir de fournir des résultats d'analyse du média. Ils disposent ainsi des ressources nécessaires pour analyser les résultats provenant d'autres analyseurs et pour déduire des résultats sur des concepts subjectifs.

L'architecture fournit donc, par l'intermédiaire des moteurs d'inférence, un moyen de mettre en relation les analyseurs à travers les outils de description disponibles (haut ou bas niveau). Dans ce contexte, si un analyseur de haut niveau prend en entrée un joueur de football, le moteur d'inférence pourra lui proposer un détecteur de personne dans la scène si ces deux concepts sont définis comme étant suffisamment proches dans une ontologie.

1.7. Modélisation du contexte virtuel

[Dimitrova03] introduit le concept de génération et de maintenabilité de la description du contexte multimédia. L'exploitation du «cycle de vie » de descripteur du média en termes de mémoire à court et à long terme est nécessaire pour fournir un niveau supérieur et amélioré de la description multimédia temporelle. En fait, la sortie formelle du contenu multimédia final (analyse hors ligne) est l'ensemble des contextes média instantanés (analyse en temps réel) ; ce qui nécessite d'offrir

différents niveaux de granularité temporelle pour décrire des scènes, plans, événements, actions…

Comme nous sommes en mesure d'extraire des informations du contenu des médias, les métadonnées fournissent des solutions pour les représenter, stocker et retrouver. La modélisation du contexte virtuel d'un flux multimédia permet aux analyseurs de partager à tout moment l'état d'avancement de l'analyse de ce flux et de la compléter à l'issue de leur analyse. L'objectif majeur est de lever le verrou associé au traitement en temps réel en partageant un contexte unique d'analyse.

Le contexte virtuel est défini par l'état instantané des connaissances sur la scène en cours. Il se différencie de la description globale du média qui contient l'ensemble de toutes les descriptions qui ont déjà été effectuées sur ce média. Le contexte virtuel permet ainsi de décrire un média sur plusieurs « niveaux temporels ».

Dans les systèmes existants, ce problème de description globale est résolu en choisissant une définition unique pour chaque concept à travers des diagrammes UML par exemple. Or, combiner des diagrammes UML ayant des définitions différentes pour un même concept conduit à des incohérences de modèle.

Nous proposons un modèle où la représentation du contexte virtuel (Figure 37) est au cœur des interactions entre les services interactifs, l'état de la description en cours du média et les analyseurs.

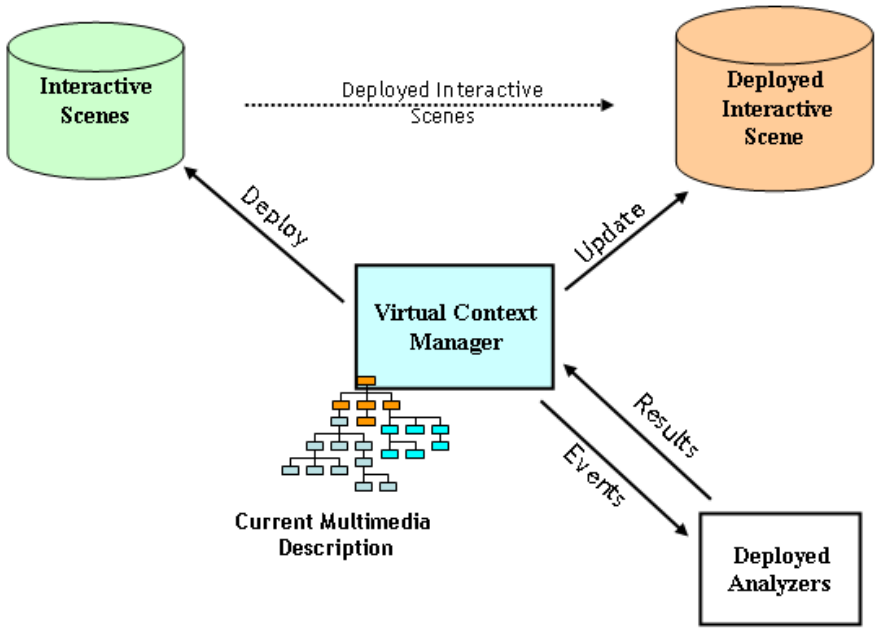


Figure 37 : Modélisation des interactions du contexte virtuel

Il n'y a pas d'échange d'informations directement entre les analyseurs. Le contexte virtuel a pour rôle de maintenir à jour la description « instantanée » du média en fonction de la durée de vie des descripteurs (les informations relatives à une personne durant un plan ou une scène par exemple, les paroles d'une personne…). Le contexte virtuel est mis à jour en temps réel par les analyseurs.

Ce modèle permet par ailleurs un accès facilité pour l'insertion d'outils d'analyse sur les performances de l'analyse, ainsi que des outils de raisonnement entre la description du média en cours et les analyseurs. Il est par exemple possible d'améliorer la sélection de l'analyseur de façon dynamique en fonction du contexte du média, les performances d'un détecteur de visage dans une séquence d'images pouvant varier en fonction de la luminosité, le contraste, le zoom, les mouvements de la caméra.

1.7.1. Agrégation des résultats des analyseurs de médias

L'agrégation des résultats d'analyse des documents multimédias est un élément déterminant dans le système proposé. Ce module inclut la représentation de la description du contexte virtuel du document multimédia et gère les évolutions de la description. Deux exemples de gestion de l'évolution de la description du document multimédia sont détaillés ci-dessous.

Le système combine les descripteurs générés à partir des différents analyseurs. Deux analyseurs différents peuvent alors fournir des informations pour les mêmes descripteurs. Afin d'éviter les incohérences dans les résultats d'analyse, un module aura pour objectif la gestion de la fiabilité, la cohérence et la pertinence de l'évolution de la description des documents multimédias.

[Dung95] propose une méthode pour déterminer les résultats d'analyses (descripteurs) à conserver ou encore pour identifier ceux qui ont échoués. Par ailleurs, certains analyseurs de médias fournissent des résultats d'analyse pondérés [Verilook] permettant d'estimer la fiabilité de la description multimédia et faciliter la prise de décision.

Dans notre implantation, nous avons dans un premier temps privilégié la synchronisation de l'accès en écriture aux descripteurs (*section critique*). Cette synchronisation maintient la cohérence de la description du média mais réduit les performances du système en interdisant la parallélisation des analyses des médias.

1.8. Web Sémantique

Dans [Passin04], la sémantique est présentée comme la signification des données sur le Web pouvant être découverte non seulement par les personnes, mais aussi par des ordinateurs. De façon contrastée, actuellement la signification des données sur le Web provient des personnes qui lisent des pages web. Le Web sémantique représente une vision dans laquelle les ordinateurs ainsi que les personnes peuvent trouver, lire, comprendre et utiliser les données du « World Wide Web » pour répondre aux attentes des utilisateurs. [Auer07] présente des travaux permettant d'extraire la signification des pages web pour créer une base de données de connaissance, puis un moteur de recherche permettant d'analyser cette base de données générée et de fournir une réponse à la requête des utilisateurs. D'autres services web simplifiés sont par exemple disponibles dans [XMethods].

Nous avons introduit le Web sémantique du fait des similarités existantes avec la représentation du contexte multimédia et des analyseurs de médias. Nous sommes à nouveau confrontés à la connaissance du « World Wide Web » qui peut être considéré comme une agglomération de bases de données, d'ontologies... et les services web ont à traiter des données à différents niveaux de granularité.

1.8.1. Web Services

Un Service Web est un composant logiciel identifié par une URI accessible via des protocoles réseau standard tels que SOAP (Simple Object Access Protocol) sur HTTP

La Figure 38 présente les principaux outils et spécifications des Web services :

- *Découverte (Discovering)* utilisant [UDDI] pour trouver la localisation des services web et leurs activités.
- *Description* utilisant [WSDL] pour décrire un service web et la manière d'interagir avec lui.
- *Encapsulation (Packaging)* utilisant SOAP pour embarquer les interactions avec le service web.
- *Transport* utilisant HTTP ou TCP/IP pour véhiculer l'enveloppe de données à travers le réseau.

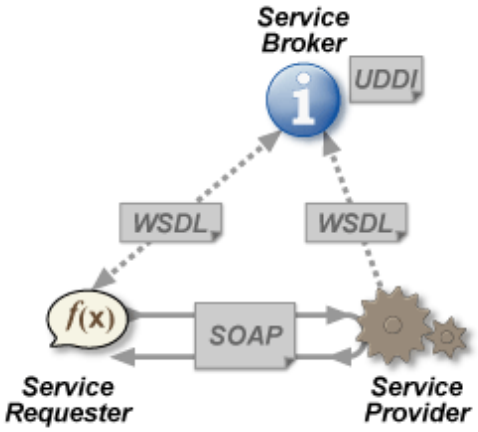


Figure 38 : Synoptique de l'architecture Service Web [Juszczyk05]

Le Web sémantique fournit des solutions permettant de déterminer les services pertinents et de les lier en fonction de leurs entrées et sorties à partir des spécifications suivantes OWL-S [Martin04], [WSDL-S], [WS-BPEL]...

Une étude récente sur le service Web sémantique (SWS) est présentée dans [Wu08]. L'objectif principal de OWL-S et WSDL-S est d'établir un cadre dans lequel les descriptions de service sont construites et partagées.

1.8.2. WSDL-S

Le *Web Service Description Language* (WSDL) est un format de document XML qui décrit le service Web. WSDL-S signifie *Web Services Description Language - semantic*. Il étend WSDL afin d'utiliser les capacités sémantiques du langage OWL et de fournir des significations sémantiquement enrichies de descriptions de service [Herrmann07].

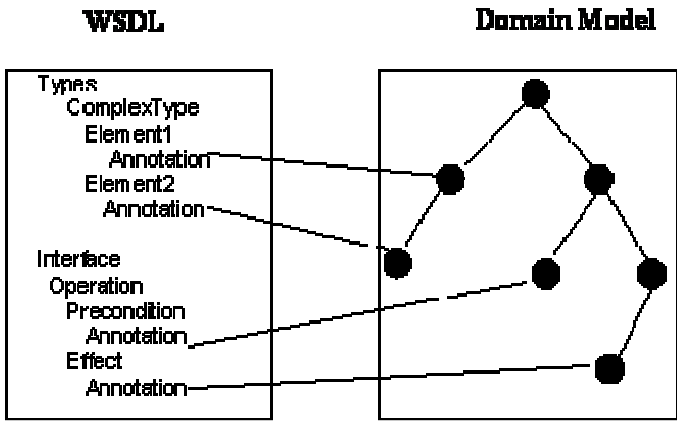


Figure 39 : Association de sémantiques à des éléments WSDL [WSDL-S]

Comme détaillé dans [WSDL-S], la Figure 39 montre comment les annotations sémantiques sont associées à divers éléments d'un document WSDL (y compris les entrées, les sorties et les aspects fonctionnels comme les opérations, les conditions préalables et effets) en faisant référence à des concepts sémantiques dans un modèle sémantique externe. Le modèle de domaine peut être constitué d'une ou plusieurs ontologies.

1.8.3. OWL-S

OWL-S signifie *Ontology Web Language* pour les *Services*. Suivant l'approche à plusieurs niveaux pour baliser le développement de langage, OWL-S est construit sur la recommandation *Ontology Web Language* (OWL).

La Figure 40 présente une ontologie permettant la découverte automatique de service, l'invocation, la composition et le suivi d'exécution. Cette composition est fondée sur des pré-et post-conditions.

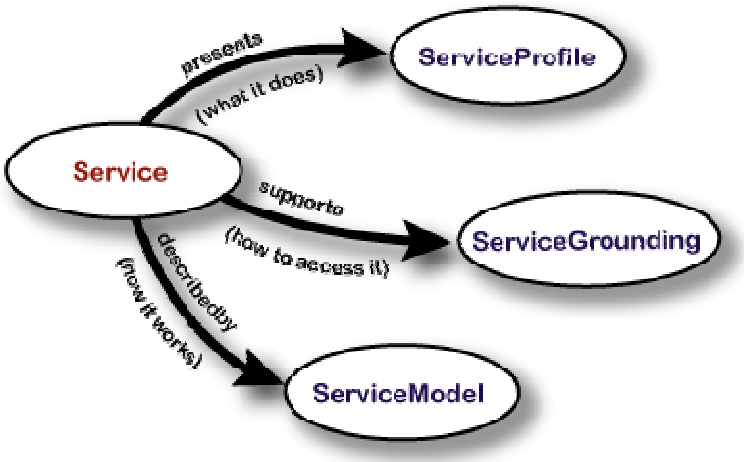


Figure 40 : Description haut niveau de l'ontologie de service [Martin04]

La composition des services Web est la tâche de combinaison et le regroupement des services Web existants pour créer de nouveaux processus Web afin d'ajouter de la valeur à la collection de services.

1.8.4. WS-BPEL

WS-BPEL signifie Web Services Business Process Execution Language. WS-BPEL est la grammaire XML définissant et standardisant les structures nécessaires pour l'orchestration de services web. La composition repose sur une gestion de processus pré-modélisés. WS-BPEL [WS-BPEL] définit un langage permettant de spécifier le comportement des processus d'affaires basés sur des services Web. Traités dans les fonctionnalités d'exportation et d'importation de WS-BPEL en utilisant des interfaces de services Web exclusivement, les processus d'affaires peuvent être décrits de deux manières : les modèles de processus opérationnels exécutables qui décrivent le comportement réel d'un participant à une interaction opérationnelle ; les processus opérationnels abstraits qui sont des processus partiellement spécifiés et qui ne sont pas destinés à être exécutés. Un processus abstrait peut cacher des détails opérationnels. Les processus abstraits ont un rôle descriptif, avec plus d'un cas d'utilisation possible, incluant des modèles du comportement observable et des processus. WS-BPEL est destiné à être utilisé pour modéliser le comportement des processus exécutables et des processus abstraits.

WS-BPEL étend le modèle d'interaction de services Web et lui permet de supporter les transactions opérationnelles. Il définit un modèle d'intégration interopérable qui devrait faciliter l'intégration des processus automatisés dans les échanges intra-entreprise et business-to-business. Le langage BPEL spécifie le comportement des processus opérationnels tant que les activités du processus sont les services Web. Les interactions humaines ne sont pas dans son domaine. Malgré une large acceptation des services Web dans des applications opérationnelles distribuées, l'absence d'interaction humaine est une lacune importante pour de nombreux processus opérationnels dans le monde réel. Pour combler cette lacune, [BPEL4People07] étend WS-BPEL de l'orchestration de services Web seule à l'orchestration du rôle des activités humaines.

1.8.5. Limitations de la composition des services Web

Comme décrit ci-dessus, le web sémantique fournit des spécifications et des outils pour identifier, localiser et composer des services Web Services.

1.8.5.1. Différentes façons de décrire le monde

[Beek06] présente un panorama des solutions permettant la composition des services et effectue une comparaison entre elles à partir de différents critères et exigences. Toutefois, la composition de services Web présente la même problématique que la description des contenus multimédias car la connaissance est toujours distribuée. La subjectivité est également un problème à prendre en compte (Figure 41). [Smeulders00] s'appuie sur l'exemple du concept de « chaise » pour développer ses recherches. Dans cet exemple, l'auteur affirme que nous pouvons nous contenter de n'importe quel objet connu sous ce nom. Lorsque nous cherchons une chaise, nous ajoutons des contraintes supplémentaires à la catégorie générale de façon à restreindre la classe. La même chose se produit quand nous recherchons

un fauteuil rouge en ajoutant une condition indépendante de la contrainte géométrique. Lorsque l'on cherche une chaise équivalente à une chaise donnée, une correspondance physique et géométrique doit être effective. Enfin, lorsque l'on recherche une chaise à l'identique, il doit y avoir correspondance littérale en ignorant encore toute variation.

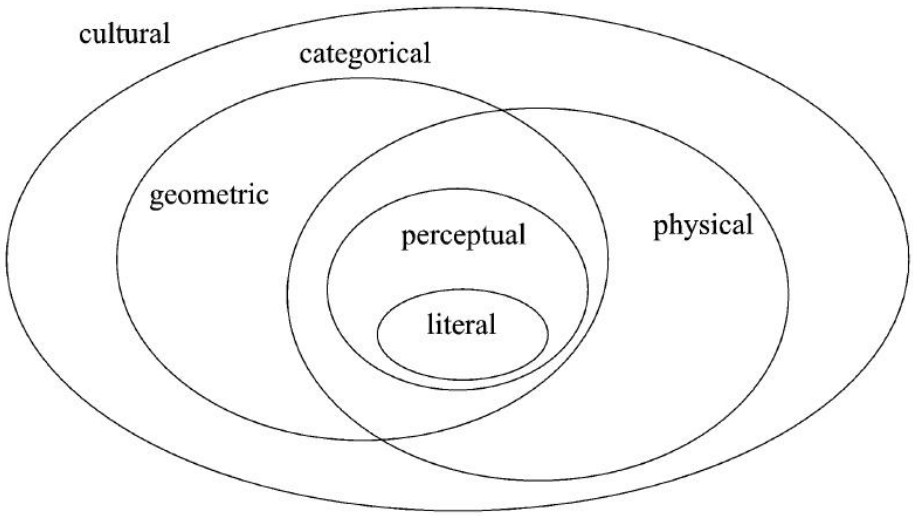


Figure 41 : Connaissances générales classées comme classes d'équivalences [Smeulders00]

1.8.5.2. Information à différents niveaux de granularité

Puisque des concepts peuvent être définis de différentes façons, la réutilisation d'ontologies existantes est complexe à réaliser [Isaac05]. Dans ce contexte, les travaux pour fournir des outils et méthodologies sont toujours d'actualité. L'état de l'art effectué par [Antoniou05] fournit une perspective des efforts consacrés au développement de langages plus expressifs incluant l'union expressive de règles et l'ontologie. [Cuenca-Grau06] a également fait des propositions pour étendre la couche sémantique d'ontologie Web avec des règles destinées à faciliter la fusion d'ontologies. Dans la gestion des problèmes de cohérence, [Ghilardi06] et [Lutz07] étudient de nouveaux problèmes de raisonnement fondés sur la notion d'extension d'ontologies conservatrice.

1.8.5.3. Composition linéaire non certifiée

Nous avons détaillé des services Web sémantiques pour illustrer les analogies avec l'analyse de contenu multimédia. Toutefois, la composition des services existants générés est principalement utilisée de façon linéaire et directe. Le processus de composition de services a été conçu pour le calcul de trajectoire dynamique utilisant les services nécessaires pour atteindre les résultats recherchés à partir des informations fournies. Le chemin correspondant à la composition de services étant fondé, il est utilisé et supprimé. La réutilisation multiple de l'analyse itérative des informations n'est donc pas directement liée aux préoccupations de composition des services web.

Il est d'ailleurs à noter que le temps de calcul et la composition des services en termes d'enchaînement des services Web est un réel problème à prendre en compte notamment dans la transposition qui peut être faite par analyse en temps réel des contenus multimédias [Cheng07]. Cheng propose une solution de composition dynamique des services pour éviter la création de chemins non co-accessibles. Cela conduit à une réduction d'environ 60% des besoins en mémoire, avec une augmentation d'environ 60% du temps de décodage en raison des coûts supplémentaires liés à la composition dynamique.

Finalement, les résultats de composition de services web ne permettent pas d'assurer de bons résultats car la définition de nouveaux processus interagissant avec des processus existants doit être réalisée manuellement, ce qui est complexe, chronophage et sujet à erreur [Beek06].

1.8.5.4. Composition générée dynamiquement

La génération de la composition d'analyse dynamique d'un document multimédia est fondée sur les capacités des modèles de description précédemment décrits. Le modèle de description global définit toutes les « voies » possibles pour accéder à toutes les définitions des descripteurs et des concepts. Il est possible grâce au mécanisme de raisonnement ontologique d'associer les descriptions syntaxiquement hétérogènes et donc de construire la liste d'analyseurs permettant d'accéder à l'information requise.

La Figure 42 représente le synoptique du processus de composition proposée. La première étape du processus de composition est d'extraire les concepts de la requête de service et d'identifier tous les analyseurs en mesure de fournir des descripteurs sur ces concepts. Dans cet exemple, les descripteurs « people », « position », « name » et « place » sont accessibles via des analyseurs sélectionnés. Dans un deuxième temps, le système identifie les analyseurs nécessaires pour lier les descripteurs précédemment sélectionnés à la description du document multimédia racine. Le détecteur de segmentation en scènes (*Scene cut detector*) est nécessaire pour remplir l'arbre de description du document multimédia.

Finalement, le système identifie les analyseurs nécessaires. Il vérifie que chaque analyseur sélectionné a accès à l'information nécessaire pour fonctionner correctement. A titre d'exemple, l'analyseur de reconnaissance du visage a nécessairement besoin d'un visage à reconnaître en entrée laquelle devant être validée par l'analyseur de détecteur de visage.

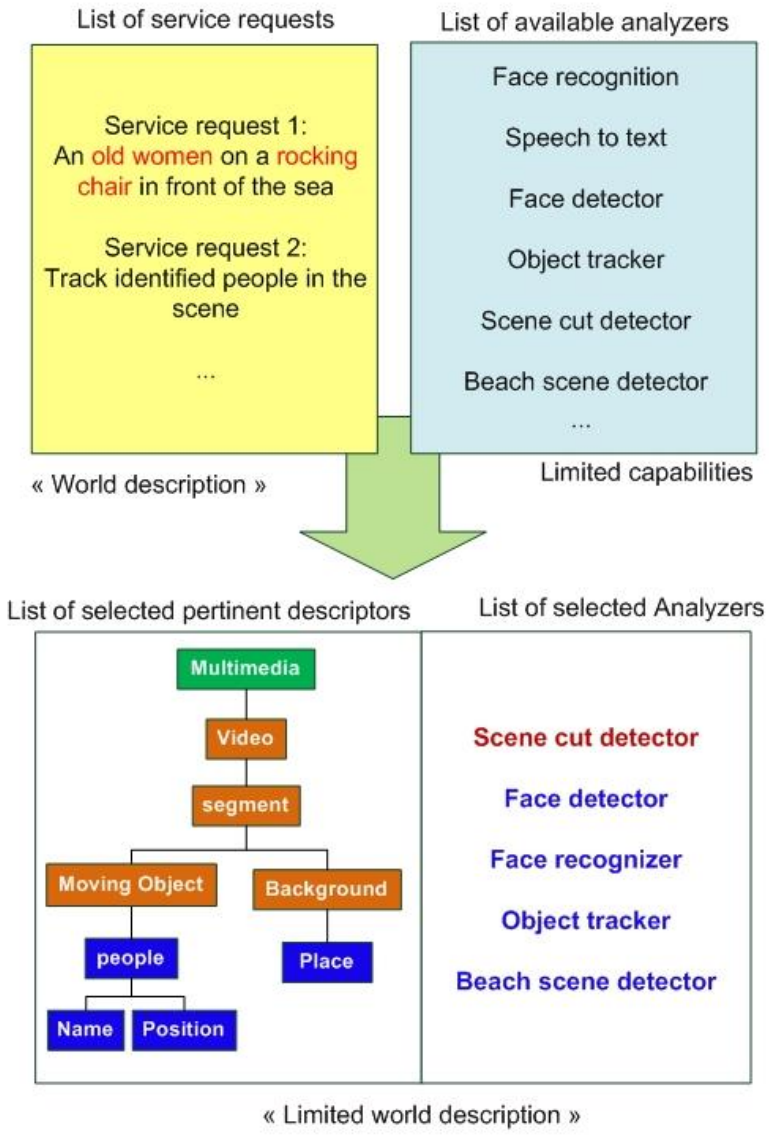


Figure 42 : Synoptique du processus de composition proposée

On peut remarquer que, dans cet exemple, les concepts d'«âge» et de «genre» ne peuvent pas être identifiés par les analyseurs de la liste des analyseurs disponibles. La première requête de service n'est donc pas accessible. La composition finale proposée est fondé sur la liste des analyseurs sélectionnés disponibles. En outre, le choix de l'analyseur « Détecteur de Scène Plage » dépendra de services permettant de maintenir ou non la requête de service demandée à partir des concepts non résolus car aucune information supplémentaire ne peut être fournie.

Les utilisateurs auront la possibilité d'accéder à et de valider la composition d'analyseurs générée à travers une interface utilisateur. Il ne sera pas possible de garantir qu'un analyseur sera meilleur qu'un autre, chaque analyseur multimédia ayant ses propres caractéristiques et capacités pour un contexte défini et connu.

1.9. Conclusion

Nous avons tout d'abord analysé et identifié les limites des systèmes actuels – fermés, pas adaptatifs, non évolutifs…– pour l'analyse des documents multimédias. Nous avons dès lors proposé les modélisations des évolutions à mettre en œuvre pour lever les verrous technologiques identifiés.

Ces modélisations concernent les services interactifs et les analyseurs en termes de définition des entrées, des sorties, des déclarations des fonctions... La modélisation de la description d'un document multimédia a ensuite permis dans un premier temps de mettre en avant le fossé sémantique entre les descriptions de bas niveau et de haut niveau des médias ; puis d'introduire la modélisation d'une représentation "virtuelle" et "instantanée" de la description d'un document multimédia. Le contexte virtuel est ainsi défini par l'état instantané des connaissances sur la scène en cours. Il permet de décrire un média sur plusieurs « niveaux temporels ». Cette représentation virtuelle permet la liaison entre les services interactifs, l'état de la description en cours du média et les analyseurs. L'agrégation des résultats d'analyse des médias rend possible le déploiement des services interactifs pertinents vis-à-vis des documents multimédias. Nous avons enfin introduit les outils de composition issus du domaine de la sémantique pour effectuer la sélection et la combinaison des analyseurs en fonction des besoins des services interactifs.

Après avoir décrit dans cette deuxième section la modélisation des différents modules de l'architecture proposée, nous en détaillons, dans le chapitre 2, les choix d'implantation mis en œuvre.

Chapitre 2

RAMSES : Intégration & validation

L'intégration des modèles et fonctions précédemment présentés a abouti à ce que nous désignerons par la plateforme RAMSES (Figure 43) qui permet le déploiement des différents composants constituant l'architecture RAMSES.

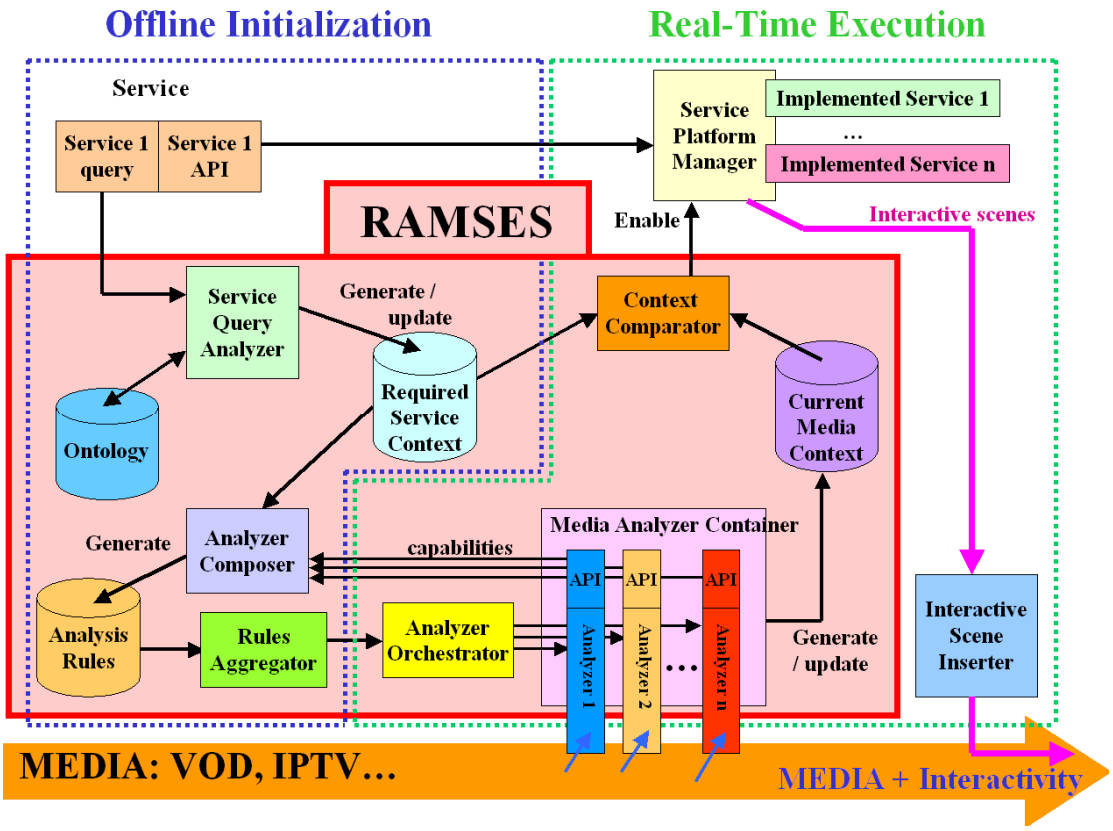


Figure 43 : Schéma de la plateforme RAMSES développée

La plateforme RAMSES (Figure 43) est constituée de deux parties logiques. La partie « Offline Initialization » gère la configuration de tous les éléments de l'architecture en fonction des analyseurs disponibles et des requêtes des services interactifs. La seconde partie intitulée « Real Time Execution » spécifie le fonctionnement logique de l'orchestration des analyseurs ainsi que le déploiement et le maintien des services interactifs.

2.1. Depuis l'analyse du monde à la recherche d'informations ciblées

La plupart des recherches se fondent sur une mise en place d'architectures capables de générer automatiquement une description complète des documents multimédias. La Figure 44 représente un schéma global du cycle de vie actuel des informations sur les médias. La description générale du document multimédia (*General Multimedia Description*) est générée par un système d'analyse (*Multimedia Analyzer*) de la façon la plus complète possible sur le document. Cette description est rendue accessible aux services (interactifs, vidéo à la demande, guide des

programmes…). Les fournisseurs de services peuvent ainsi extraire des informations qui leur sont utiles pour le déploiement de leurs services. Dans ce contexte, on se rend compte que les descriptions de documents multimédias doivent être les plus exhaustives que possible afin de pouvoir proposer les informations nécessaires au fonctionnement des différents types de services. Ce type d'information peut ainsi servir à des moteurs de recherche pour retrouver des médias, obtenir un résumé du contenu du média pour les services de vidéo à la demande, obtenir des informations de présence de personnes pour proposer des services d'information additionnelles, etc.

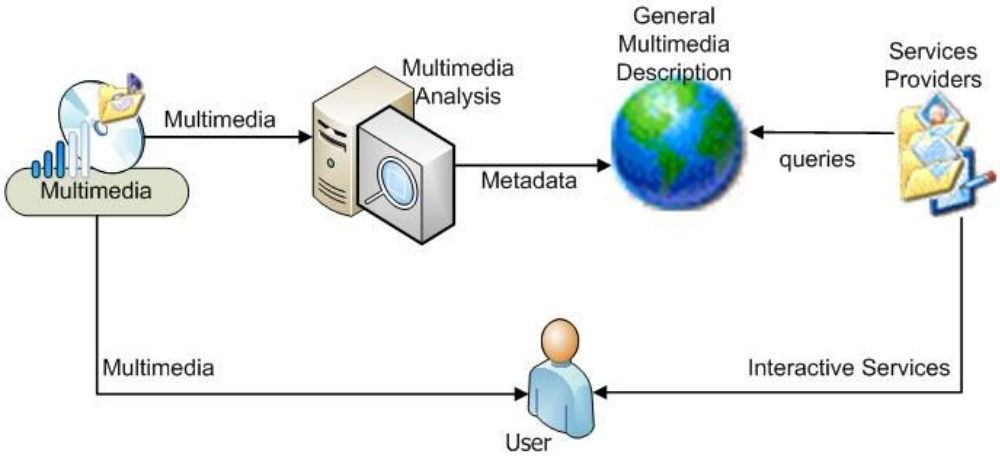


Figure 44 : La description du monde aujourd'hui

« Décrire le monde » n'est pas un objectif accessible aujourd'hui de par les limitations de nos ordinateurs en termes de puissance de calcul. Dans ce contexte, nous proposons une approche inverse, nous proposons de limiter l'analyse des documents multimédias aux informations requises par les services qui se sont « enregistrés » pour l'obtention d'informations sur des documents multimédias. Dès lors, le but de l'architecture d'analyse des documents multimédias consiste à mesurer la distance entre le contexte du document multimédia et le contexte recherché par les fournisseurs de services. Le contexte représente ici l'ensemble des informations requises par un service pour son déploiement.

La Figure 45 illustre le schéma du cycle de vie d'une description d'un document multimédia en fonction de l'architecture proposée.

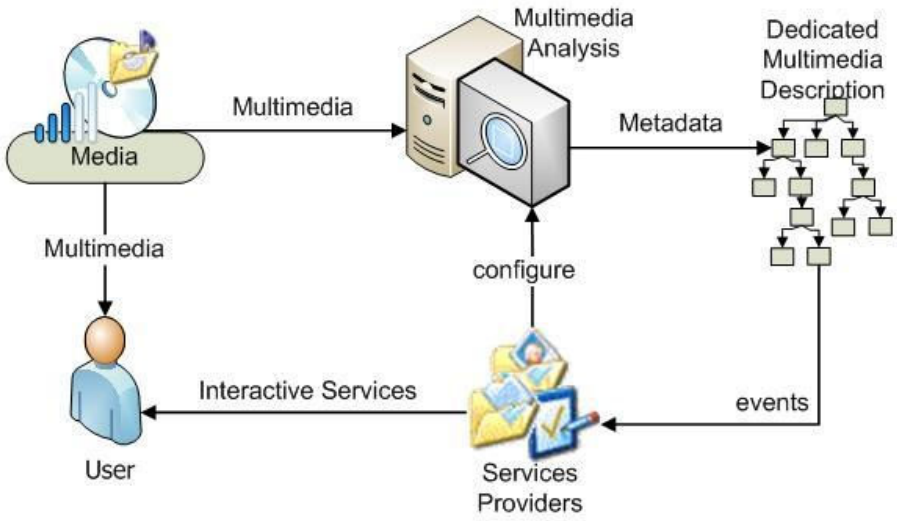


Figure 45 : Schéma proposé pour l'analyse restreinte de documents multimédias

Le système d'analyse (*Multimedia Analyzer*) est maintenant configuré à l'aide des différentes requêtes des services interactifs pour cibler les informations à analyser dans les documents multimédias. Il n'est dès lors plus nécessaire par exemple de déterminer si une scène se déroule en intérieur ou en extérieur si les informations à extraire concernent la détection, le suivi et la reconnaissance des personnes présentes dans la scène. De plus, le fait de ne se focaliser que sur les informations pertinentes pour le déploiement des services interactifs permet de restreindre le domaine d'analyse et ainsi le nombre de concepts à manipuler dans la génération de la description. Il est dès lors possible de prévoir si les analyseurs média présents dans les systèmes seront capables de fournir les descripteurs requis par les services ainsi que les ressources nécessaires pour y parvenir.

2.2. Choix d'implantation

Afin de répondre aux critères de modularité, d'évolutivité, de *dynamicité*, d'adaptabilité et de fonctionnement temps réel, nous avons dû effectuer un ensemble de choix.

2.2.1. Modularité et extensibilité

La plateforme doit permettre aux développeurs de créer et d'insérer le plus facilement possible des analyseurs média, des combinaisons d'analyseurs, des services interactifs… à l'aide des API fournies. La courbe d'apprentissage pour le développement de modules d'analyse ou de services pour la plateforme doit être accessible à tous les développeurs. Enfin, la plateforme doit être facile à prendre en main pour un simple utilisateur pour l'insertion, la mise à jour ou la suppression pour chacun des types de modules.

La modularité est le point clé de la plateforme. Nous avons dans un premier temps étudié les modèles d'implantation existants. Par ailleurs, dans l'objectif d'une implantation rapide de tous les modèles et fonctionnalités, nous nous sommes

tournés vers une implantation en Java. Nous avons retenu une architecture fondée sur le standard OSGI. La notion « Tout est service » dans le standard OSGI fournit *de facto* à la plateforme RAMSES une architecture modulaire et adaptative.

2.2.1.1. Framework OSGI

L'Alliance OSGi (initialement Open Services Gateway Initiative) [OSGi] est un organisme ouvert de standardisation fondé en mars 1999. L'Alliance et ses membres ont spécifié une plateforme de services s'appuyant sur Java pouvant être gérée à distance. Le cœur de cette spécification (appelé *framework*) définit un modèle de gestion du cycle de vie d'une application, un service d'enregistrement, un environnement d'exécution, et des modules. De nombreuses spécifications enrichissant ce *framework* tels que les couches OSGI, APIs, services sont décrites dans [WikiOSGi]. L'Alliance fournit des spécifications, des implantations de référence, des suites de tests et des certifications afin de permettre la mise en place d'un écosystème pertinent regroupant les industriels concernés.

La technologie OSGi fournit une plateforme de développement d'applications à la fois modulaire et orientée service. L'adoption de cette plateforme modulaire permet une intégration facilitée de composants ou modules en phase amont, c'est-à-dire avant qu'ils soient totalement développés ou testés.

Nous avons choisi le *framework* [Equinox] pour l'implantation de la plateforme RAMSES. Equinox est le nom du projet Eclipse s'appuyant sur le langage de programmation Java qui fournit une implantation certifiée de la spécification « R4 » du cœur de la spécification du *framework* OSGI. L'architecture de cette plateforme décrite Figure 46 montre les principes de modularités, de dépendance entre modules, et le lien avec l'environnement logiciel et matériel existants.

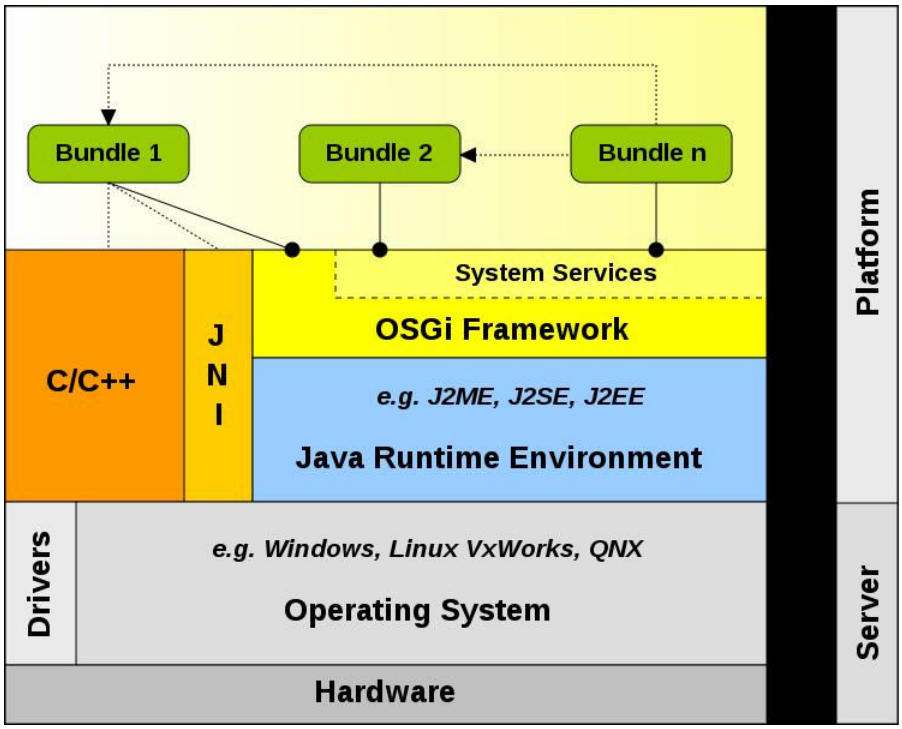


Figure 46 : détails des couches OSGi [Grammling06]

Le *framework* Equinox apporte les aspects modulaires au travers de la notion de "bundles" qui permet aux développeurs une exploitation facilitée de l'infrastructure et des services communs [Equinox]. Depuis la version 3.0, Eclipse a également choisi OSGi afin de remplacer les technologies de modularisation des versions précédentes.

2.2.1.2. Définition des « bundles » OSGI

Dans le contexte OSGi, la notion de module ou « Bundles » est standardisée, et correspond à la représentation de la Figure 47. Les « bundles » ont la possibilité d'importer ou d'exporter des packages logiciels (au sens Java) afin de pouvoir instancier un service au sens OSGi. Les entrées et sorties de ces « bundles» sont des services déclaratifs. Ces services déclaratifs sont exploités pour pouvoir sélectionner et implémenter dynamiquement les services OSGi, nécessaires : cette étape est effectuée par un moteur sémantique de raisonnement.

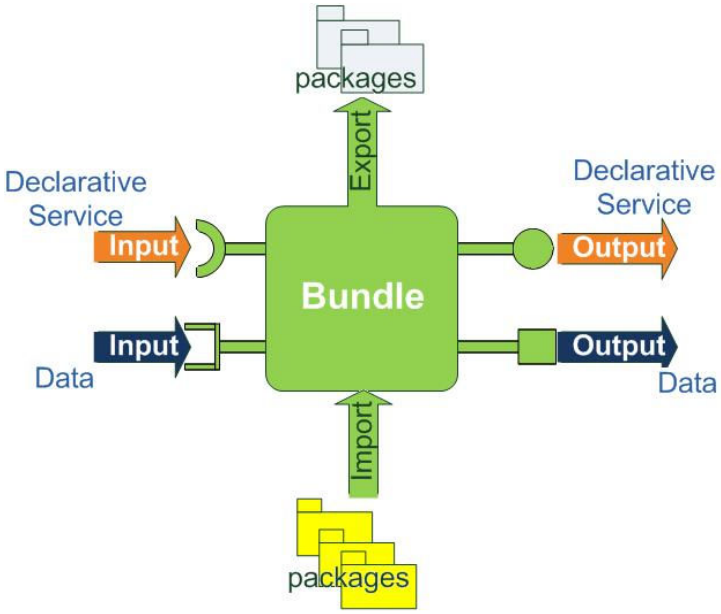


Figure 47 : Représentation du modèle « Bundle » décrit par OSGi

Tous les modules décrits dans la suite de ce document doivent respecter la formalisation des "bundles" OSGi.

2.2.1.3. Les notions majeures : modules et API

La plateforme RAMSES s'appuyant sur OSGi. La Figure 48 représente les différents types de modules mis en œuvre dans cette plateforme et les API correspondantes.

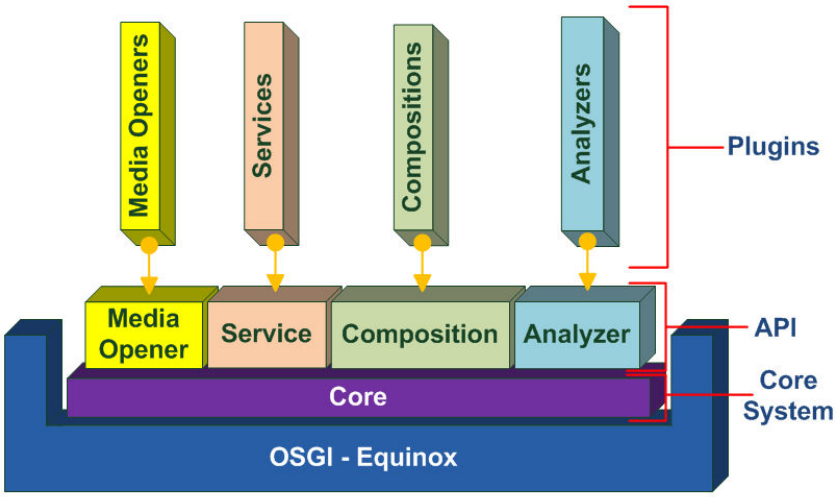


Figure 48 : Schéma de l'architecture en Plugins

De manière opérationnelle, le principe de fonctionnement est de fournir un ensemble de composants par défaut qui peuvent être enrichis et remplacés par des modules spécifiques appropriés aux applications à mettre en œuvre.

Service API : cette API est conçue de manière à permettre au gestionnaire de la plateforme la mise en place modulaire de services interactifs, rendant ainsi possible l'évolutivité rapide de ces services en fonction des besoins.

Analyzer API : cette API est centrale dans la plateforme RAMSES. En effet, grâce à cette API, le gestionnaire de la plateforme pourra ajouter et améliorer les analyseurs de média en fonction des besoins des services interactifs et des contraintes de performance de manière simple, puisque s'appuyant sur l'architecture modulaire décrite ci-dessus.

Composition API : cette API joue également un rôle important dans la plateforme RAMSES puisqu'elle permet d'apporter les modèles de description sémantiques permettant de faire le lien entre services interactifs et analyseurs de média, ainsi que de fournir un modèle sur lequel s'appuyer pour décrire les résultats des analyses mise en œuvre.

Media Opener API : cette API est conçue de manière à permettre au gestionnaire de la plateforme de mettre en œuvre et d'utiliser le module le plus approprié pour accéder au média en fonction du format, du niveau de description nécessaire ou des performances. Pour l'accès au média, le module implémenté par défaut s'appuie sur le projet open-source [Xuggle] et la libraire logicielle associée.

Le *framework Equinox* reposant sur le langage Java, les analyseurs devront de préférence être développés en Java afin d'être implantés directement dans la plateforme. Il est cependant possible d'implanter des analyseurs, des accesseurs média, des services interactifs… développés en C ou C++ grâce à la librairie *Java Native Interface* (JNI).

2.3. *Services interactifs et scènes interactives*

Les services interactifs proposés par les fournisseurs de services contiennent les *templates* des scènes interactives. Ces *templates* contiennent le squelette de la scène interactive, les conventions de couleur, l'emplacement des logos, les logos… ainsi que la définition l'ensemble du fonctionnement du service. La Figure 49 illustre un exemple de *template* mis en œuvre pour le service interactif « puzzle ». Le *template* (au centre de la figure) permet la mise en place rapide d'un service complet via la modification des champs adaptatifs.

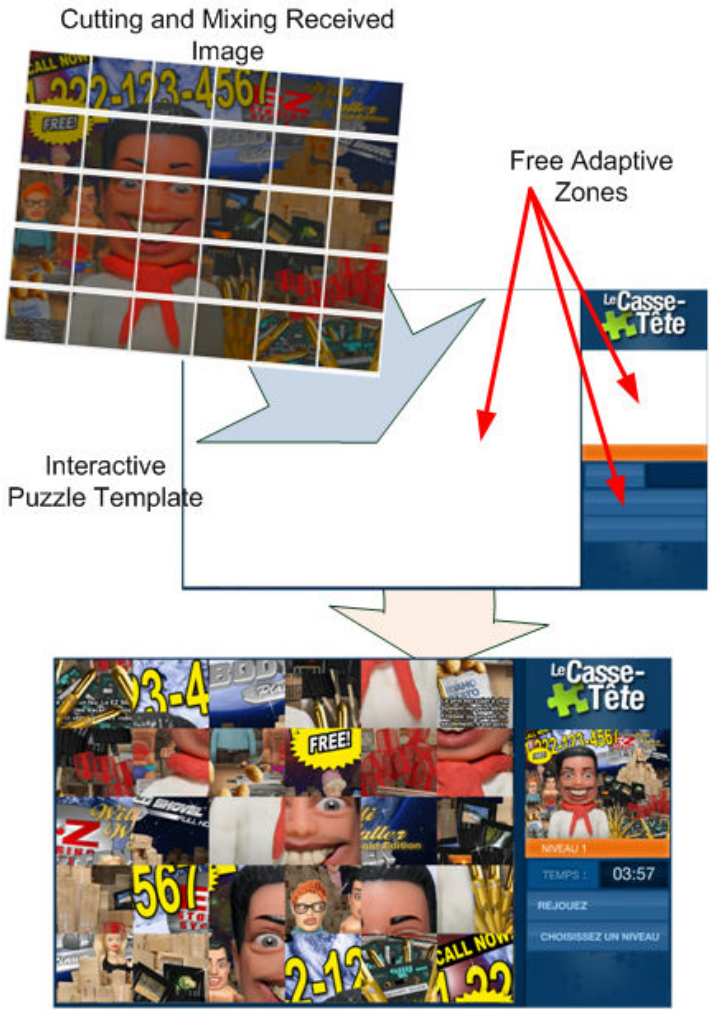


Figure 49 : Exemple d'un template et de son implantation utilisé pour un service interactif de jeu (puzzle) [TAC]

Les Services interactifs sont différenciés des autres « plugins » installés sur la plateforme RAMSES grâce à la présence du mot « Media Interactive Service » dans l'entête du champ de description du service interactif. Le champ « *Interactive_Service_Request* » contient la requête du service interactif sous forme de triplets comme détaillé précédemment. La vérification des services interactifs est effectuée lors de la génération de la composition de l'analyse. La plateforme vérifie

que l'ensemble des analyseurs présents permettent de fournir les informations nécessaires pour le déploiement du service interactif.

2.3.1. Scènes Interactives

Il existe différentes solutions pour enrichir un document multimédia. Une modification « définitive » du média est la modification du contenu vidéo par exemple (Titres, Copyrights, condition d'utilisation…). Cependant, nous ne souhaitons pas ici modifier le contenu du média, les enrichissements du média proposés doivent pouvoir évoluer ou être remplacés par des services plus en rapport avec les préférences des utilisateurs. Nous avons ainsi choisi un format de gestion de scène interactive qui se « superpose » au média diffusé. Il est dès lors nécessaire de disposer d'un système de diffusion ainsi que des récepteurs compatibles pour diffuser et accéder aux documents multimédias enrichis. Il existe différents standards pour générer des flux interactifs dans les documents multimédias. La difficulté est de trouver des *Players multimédias* capables de recevoir et décoder ces flux. Notre choix s'est porté sur le standard de description de scène MPEG-4 BIFS. [Concolato05] propose un ensemble d'outils comprenant notamment un *encodeur live*, un *broadcaster* et un *Player* compatible avec ce standard.

2.3.2. Architecture type de diffusion de services interactifs

Cette section présente l'implantation d'un modèle de diffusion de services interactifs général. L'implantation de cette architecture combinée avec les capacités d'adaptation, de sélection et d'insertion de services interactifs illustre les possibilités de déploiement pour la télévision interactive. La Figure 50 présente les possibilités pour un récepteur multimédia, un téléphone mobile par exemple, de réception de services interactifs à partir de plusieurs sources d'émission. Dans cet exemple, trois fournisseurs de services sont proposés : un service de jeux interactifs, un service d'achat en ligne et un service général (météo, info-trafic…). Ils peuvent ainsi insérer leurs scènes interactives dans les médias. Ces services peuvent contenir des informations statiques (texte, images, vidéos…), mais aussi des liens (Internet, adresse e-mail, numéro de téléphone…) vers les plateformes des services (serveurs de vote, de vidéo à la demande…), des sites Internet, des forums en ligne… Ces liens introduisent la notion de retour utilisateur et de connexion entre l'utilisateur final et le fournisseur de services. Les liens ainsi insérés dans les scènes interactives permettent une connexion individuelle (*unicast*) directe avec le service proposé. Celle-ci peut se faire facilement sur les téléphones mobiles à l'aide des connexions GSM, SMS, 3G… ou sur les télévisions fixes à l'aide d'une connexion Internet xDSL, Wifi, d'une ligne téléphonique...

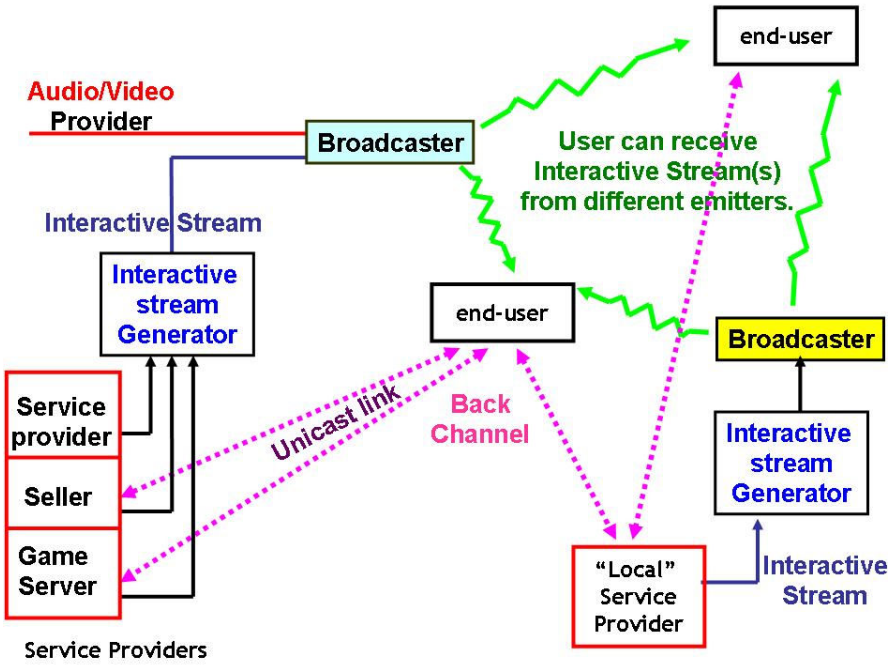


Figure 50 : Schéma d'implantation typique de la diffusion de contenu et d'interactivité associé incluant la chaîne de retour

La partie droite de la Figure 50 représente la possibilité de générer et diffuser des services interactifs complémentaires à partir d'un emplacement différent. Ces services interactifs complémentaires peuvent ainsi être adaptés à des lieux géographiques. Un exemple d'application est un musée disposant d'un service d'émission d'informations additionnelles en relation avec les œuvres exposées. Un service principal contiendrait des informations générales (plan du musé, répartition des différentes expositions en cours, horaires d'ouvertures…) et des services indépendants émis à proximité de chaque œuvre pour des informations complémentaires lorsque l'utilisateur du service passe à proximité.

2.3.3. Gestion des scènes interactives MPEG-4 BIFS

Le standard MPEG-4 est un standard basé sur une *conception objet* des documents multimédias et fournissant des mécanismes pour organiser et gérer spatiotemporellement les scènes interactives. Les différents objets médias sont ainsi animés dynamiquement en fonction de la description des évolutions spatiales et temporelles prédéfinies ou par l'intermédiaire d'un protocole de communication pour piloter la scène de manière distante.

Nous pouvons différencier deux types de construction de scènes interactives, les scènes interactives déployées et mises à jour en *direct* et les scènes interactives générées en mode *différé*. Nous avons développé les deux modes de fonctionnement pour chacun des services interactifs déployés. Nous avons dans un premier temps développé la scène interactive en mode différé afin de vérifier le fonctionnement « statique » du service et d'isoler les parties relatives à l'initialisation et les parties relatives au fonctionnement dynamique. Enfin, nous avons, à partir de ces éléments sélectionnés puis implantés, développé un mode de fonctionnement

dynamique à l'aide des commandes BIFS Update. L'Annexe 6 illustre l'implantation de ces deux modes de fonctionnement pour le service interactif d'information additionnelle sur la reconnaissance et le suivi de visage dans le média. Nous décrirons principalement l'implantation des services interactifs en direct.

Chaque service interactif spécifie la liste des descripteurs à obtenir pour pouvoir déclencher le service interactif dans le média dans un premier temps, puis une liste des descripteurs à obtenir pour maintenir à jour le service interactif dans le média. La plateforme gère le déploiement et le maintien de tous les services interactifs présents en fonction des résultats d'analyse par l'intermédiaire d'un composant de gestion des scènes interactives MPEG-4 BIFS (MPEG-4 *BIFS Manager)* Figure 51.

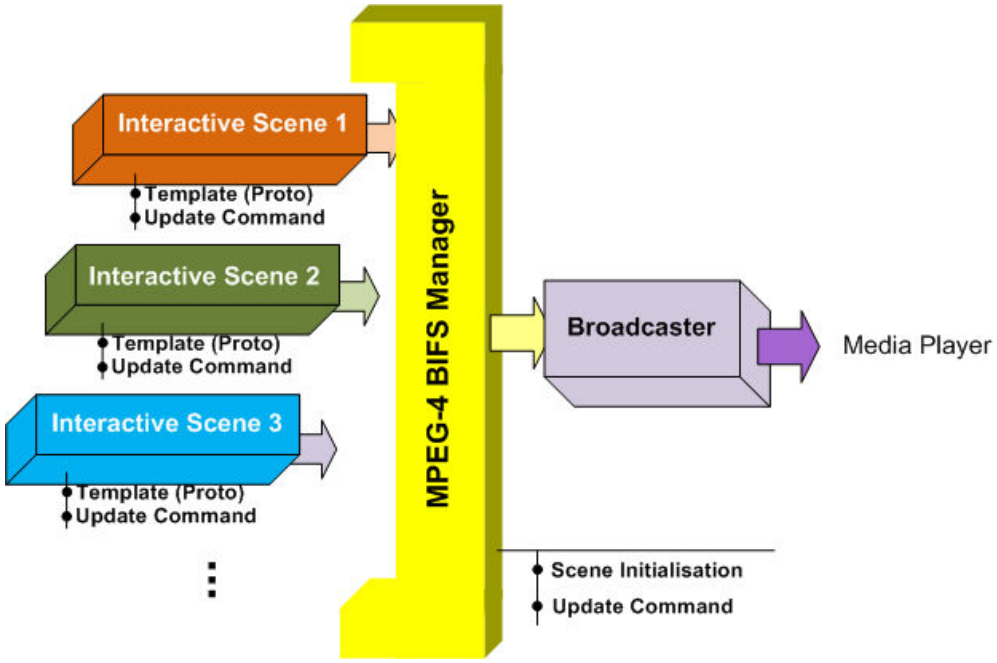


Figure 51 : Schéma d'implantation des scènes interactives associées aux services interactifs

Nous avons implanté une scène interactive pour chaque service interactif présent sur la plateforme. Les scènes interactives sont constituées d'un *template* généralement déclaré sous la forme d'un ensemble de *prototypes* pour être implanté et mis à jour de façon dynamique dans le média; et d'un ensemble de commandes MPEG-4 BIFS U*pdate* définissant le fonctionnement de la scène en fonction des évolutions du média. Un exemple de déclaration de *Prototype* est présenté dans l'Annexe 5.

Le composant MPEG-4 BIFS Manager est nécessaire pour la construction de la scène initiale composée de l'ensemble des objets présents dans les *template*s des services interactifs (champs de texte, boutons, images…). Ces objets sont ensuite liés de façon dynamique à des capteurs, des temporisations, des événements… à l'aide des commandes MPEG-4 BIFS U*pdate*. Les commandes MPEG-4 BIFS Update sont déclinées à travers trois commandes pour la modification des propriétés d'une scène :

- *REPLACE,* permet de remplacer tout ou partie de la scène ;
- *INSERT,* permet d'insérer un nouvel élément dans la scène ;
- *DELETE,* permet de supprimer un élément de la scène.

Ces commandes BIFS peuvent être appliquées à une ROUTE (lien entre des actions et des attributs), un NODE (groupe d'attributs), ou un événement. Nous avons implanté la définition de la scène pour chaque service interactif à l'aide de prototypes initialisés avec des objets vides et non affichés. En effet, ces commandes *Update BIFS* ne peuvent être appliquées qu'à des nœuds existants dans la scène. Il est ensuite possible à travers les commandes Update de mettre à jour la scène et d'instancier les objets en fonction de l'évolution du média.

La Figure 52 illustre les résultats obtenus avec un service interactif d'informations additionnelles sur « *Les travaux de l'Assemblée nationale* » sur La Chaîne Parlementaire (LCP).

Figure 52 : Exemple de résultats sur d'implantation d'un service interactif

Les informations pour le déclenchement du service sont l'apparition d'une personne reconnue dans la scène et les informations de maintien du service sont la position de cette personne dans la scène. Lorsque cette dernière information n'est plus disponible, le service est arrêté jusqu'à la prochaine détection.

Comme illustré Figure 51, le composant MPEG-4 BIFS Manager est connecté avec le *broadcaster RTP* [Concolato05] permettant l'encodage des commandes BIFS à la volée pour la mise à jour de la scène interactive en temps réel.

2.4. Accesseur de média

Les accesseurs média sont les premiers modules qui ont été développés sur la plateforme RAMSES afin d'accéder au contenu des documents multimédias. Ils sont implantés sur la plateforme comme des plugins permettant ainsi une évolution possible en termes de médias accessibles et de suivi d'évolution des formats. Les accesseurs média sont différenciés des autres plugins grâce à la présence du mot « *Media_Opener* » dans l'entête du champ de description de l'analyseur. Un second champ caractérise le type de média dont l'accesseur est capable d'analyser le contenu.

La Figure 53 illustre un exemple de média accesseur installé sur la plateforme dans le composant de gestion. On distingue ainsi « *Video_Analyzer* », « *AudioVisual_Analyzer* », et « *EPG_Analyzer* ».

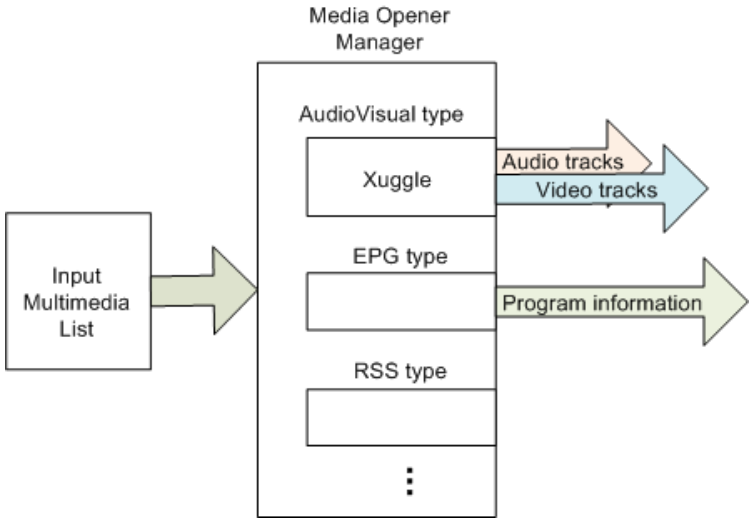


Figure 53 : Schéma d'implantation des accesseurs média

Ce composant permet d'obtenir la liste des médias accesseurs disponibles et de les installer en fonction des besoins d'analyse identifiés lors de la composition des analyseurs. Nous proposons par défaut sur la plateforme un accesseur de média vidéo fondé sur le projet open source [Xuggle], lui-même reposant sur le projet open source [FFMPEG]. La plateforme RAMSES peut ainsi fournir un accès à un grand nombre de formats vidéo image par image.

De plus, pour optimiser le fonctionnement global de la plateforme, celle-ci dispose d'un ensemble de fonctions *statiques* de base accessibles à tous les analyseurs telles que l'extraction d'histogrammes, la conversion de format de couleurs (HSV, YUV, 8bits…), l'extraction de l'équivalent des « coefficients DC » [Manerba08]… Ces fonctions étant très souvent appelées dans le domaine de l'analyse des images ; leur disponibilité permet aux analyseurs qui ne peuvent échanger directement des données (buffer, histogrammes…) de ne pas effectuer chacun de son côté les mêmes « opérations de base » et de concentrer les ressources de la plateforme sur les fonctions avancées d'analyse.

La Figure 54 illustre l'extraction de l'équivalent des « *coefficients DC* ». Ces coefficients sont accessibles directement dans le flux vidéo dans le cas d'un encodage MPEG-4 par exemple ou calculés à partir des valeurs moyennes des blocs de huit par huit pixels. Ces coefficients permettent d'effectuer des calculs sur une image 64 fois plus petite que l'original réduisant ainsi les temps de calcul. Ces images réduites sont essentiellement utilisées pour augmenter les performances de calcul dans la détection de changement de scène, l'analyse des couleurs dominantes, la détection de zones de mouvements...

Figure 54 : Illustration de la réduction d'une image d'origine (à gauche) à l'aide des « coefficients DC » correspondants (à droite)

Enfin, d'autres accesseurs média sont actuellement déployés sur la plateforme RMASES afin d'accéder simultanément à différentes sources d'information. Par exemple, un accesseur de flux d'informations sur les programmes diffusés sur les chaînes de télévision, *Electronic Program Guide* (EPG), a été ajouté dans le but de pouvoir accéder aux informations sur les programmes en cours de diffusion (titre, genre, durée…).

2.5. *Analyseurs de médias*

Les analyseurs média sont différenciés des autres « plugins » installés sur la plateforme RAMSES grâce à la présence du mot « Media Analyser Service » dans l'entête du champ de description de l'analyseur média.

Dans le but d'optimiser les performances globales de la plateforme RAMSES, il est nécessaire que les analyseurs média soient les plus « unitaires » possibles. Cela revient à :

- un minimum d'informations en sortie pour limiter l'analyse du média à l'extraction de l'information à fournir,
- un maximum d'information en entrée pour ne pas faire d'analyses supplémentaires sur le média qui peuvent être déjà effectuées par d'autres analyseurs. Enfin des analyseurs avec peu d'entrées / sorties et une fonctionnalité simple seront plus facilement sélectionnés et mis en œuvre sur la plateforme.

2.5.1. API des analyseurs multimédias

Les analyseurs média sont développés comme nous l'avons défini selon l'architecture en bundles (cf. Figure 47). La modélisation de l'analyseur multimédia a permis de définir l'ensemble des informations nécessaires à présenter à l'API de celui-ci telles que la description du but de l'analyseur, des descripteurs d'entrée, de sortie, des types de médias à analyser… Nous avons implanté les deux solutions, « bas niveau » et langage naturel contraint (CNL), pour la description des analyseurs.

Le Tableau 5 illustre un exemple de description pour un analyseur de détection de visages.

Tableau 5 : Exemple de description implantée pour un analyseur de détection de visages

```
Bas niveau :
        Entrées :       « /media/video/segment »
        Sorties :       « /media/video/segment/face »

                        « /media/video/segment/face/movingRegion/region »
Langage Naturel Contraint :
        Entrées :       « vidéo média, image »
        Sorties :       « visage »
        But :           « détecter les visages »
```

Afin de ne pas passer trop de temps sur l'implantation d'analyseurs, il a été décidé d'implanter des solutions d'analyseurs « simples » ou existants aussi bien pour la détection des changements de scène que pour les analyseurs d'un plus haut niveau comme la reconnaissance de visages. Les analyseurs ainsi intégrés pour fournir les éléments nécessaires à la description du contenu des médias vidéo et pour démontrer les capacités de la plateforme pour l'insertion automatique de services interactifs n'ont pas été optimisés. Ces analyseurs –modulaires– pourront être remplacés par la suite par des analyseurs équivalents reposant sur des algorithmes plus « évolués ».

Il a ainsi été implanté des analyseurs provenant aussi bien du commerce tel que [Verilook] que des analyseurs décrits dans la littérature [Cuce04, Ferman02]. La plateforme supporte et combine actuellement les analyseurs suivants :

- détecteur de visage (Verilook) ;
- détecteur de visages (OpenCV) ;
- reconnaissance des visages dans l'image (Verilook) ;
- détecteur de changement de scène inspiré de [Ferman02, Joyce06]
- algorithme de suivi d'objet inspiré de [Cuce04].

La vérification des entrées et sorties des analyseurs est effectuée après leur sélection si ceux-ci fournissent des résultats pertinents pour un fournisseur de service. Dès lors, dans le cas où le résultat de la vérification d'un analyseur n'est pas validé (condition d'entrée non validée par exemple), la plateforme signale une erreur en indiquant l'analyseur en défaut. Dans le cas où l'analyseur ne spécifierait pas correctement ces descripteurs fournis en sortie, celui-ci n'est tout simplement pas sélectionné pour l'analyse du média.

2.5.2. Analyseurs multimédias bas niveau

L'extraction descripteurs de bas niveau est une étape importante dans le modèle d'analyse présenté. Cette première étape concerne la segmentation spatiotemporelle d'un média permettant la liaison entre les médias (données vidéo, audio…) et les descriptions de haut niveau qui sont déduites des concepts détectés dans les médias.

Il a été développé un ensemble d'analyseurs « simples » pour la segmentation temporelle avec le détecteur de changement de scène par exemple et la segmentation spatiale avec le détecteur de visages ou encore le détecteur d'arrière plan de la scène.

La Figure 55 illustre un exemple des différents points d'intérêt pour la segmentation spatiotemporelle d'un média vidéo dans le cadre de l'insertion d'un service interactif d'informations additionnelles sur les « travaux de l'Assemblée nationale ».

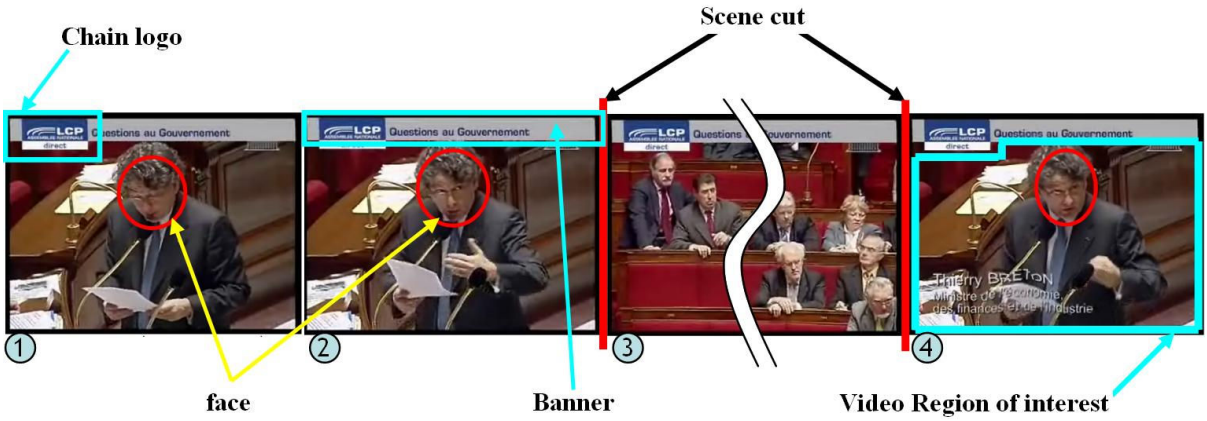


Figure 55 : Schéma illustrant les zones d'intérêt pour les scènes interactives dans le programme « Les travaux de l'Assemblée nationale » [LCP]

Dans cet exemple, les images du média vidéo sont analysées dans le domaine compressé [Gu02] afin d'optimiser les performances d'analyse sur les analyseurs de base. L'analyseur de détection de changements de scène repose sur l'analyse des histogrammes [Ferman02, Joyce06] pour la segmentation temporelle. La segmentation spatiale implantée combine la détection de visages, le suivi des objets détectés, la détection du fond d'écran de la scène...

La détection du fond d'écran est réalisée à l'aide de l'enregistrement des *zones invariantes* [Chien02]. La combinaison des résultats de cette méthode avec les résultats de détection de changements de scène permet à un analyseur de *détection de logo et bannières* de faire la distinction entre les *zones statiques* liées à :

- une scène (nom d'un interlocuteur par exemple),
- un programme diffusé (le titre par exemple ou le logo de l'émission),
- une chaîne de diffusion (le logo de la chaîne par exemple),
- …

La segmentation spatiale permet ainsi en combinaison avec la segmentation temporelle l'identification des *zones d'intérêt* du média en fonction des analyseurs présents sur la plateforme et les besoins des services interactifs.

2.5.3. Limitations des analyseurs

On observe sur l'image numéro trois de la Figure 55 qu'aucun des visages présents n'est détecté. Cela montre les limitations des analyseurs d'un plus haut niveau, c'est-à-dire fonctionnant sur la combinaison de résultats d'analyseurs. Les analyseurs de *détection des visages* utilisés par exemple définissent en effet des limites de fonctionnement en fonction de la taille des visages, le contraste et la luminance de l'image, l'orientation des visages…

Cet exemple illustre par ailleurs les possibilités de description liées aux résultats des analyseurs et à la combinaison de ceux-ci. Cependant, il n'est pas possible d'en déduire par exemple une absence de visages dans la scène.

2.5.4. Composition manuelle d'analyseurs

La composition d'analyseurs effectuée manuellement a été abordée dans un premier temps afin d'identifier les différentes contraintes liées à la combinaison d'analyseurs. La création d'un analyseur de détection des visages a dès lors été mise en œuvre. Celui-ci représente en effet un analyseur *complexe* disposant d'une littérature très riche sur ces 20 dernières années [Yang02, Kienzle05, Kim08...]. La détection des visages a été implantée dans le domaine compressé. La Figure 56 illustre un exemple de combinaison d'analyseurs pour la détection des visages. Cette combinaison met en œuvre une détection des zones d'objets en mouvement par soustraction de l'arrière plan de la scène [Chien02] et un détecteur de zones de couleurs de peau [Vezhnevets03]. On obtient, après combinaison, le résultat illustré Figure 56.

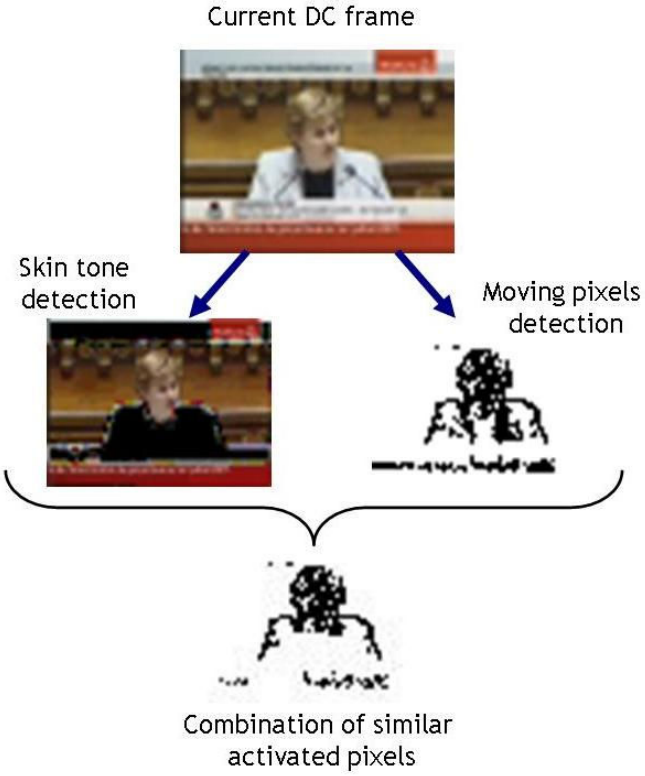


Figure 56 : Illustration de la combinaison des résultats d'analyse d'un détecteur de pixels de couleur de peau et d'un détecteur de mouvement de pixels

Pour résumer, la combinaison de ces résultats permet *la détection d'objets en mouvements ayant des couleurs dans la gamme des couleurs de peau*. Les limitations de cette « composition d'analyseurs » sont principalement l'obligation pour les visages d'être en mouvement pour être détectés, la caméra ne doit pas être en mouvement, il n'y aucune certitude que l'objet détecté soit effectivement un visage, etc. Il est nécessaire d'ajouter d'autres analyseurs pour améliorer les résultats : un analyseur de *correction des mouvements de la caméra*, un *détecteur de formes* pour éliminer les formes rectangulaires ou pyramidales par exemple, un filtre sur l'agglomération des zones détectées, etc.

Cet exemple nous a permis de démontrer la possibilité de construire des compositions d'analyseurs de bas niveau afin de fournir des informations d'un niveau plus élevé (depuis la couleur des pixels vers l'identification d'un objet). Les analyseurs ainsi développés dans cet exemple n'ont pas été supprimés et sont toujours utilisés pour la segmentation spatiotemporelle des médias vidéo. Cependant, nous avons choisi d'implanter un analyseur de détection des visages à partir d'un algorithme du commerce [Verilook] ayant des limitations plus acceptables et un taux de détection des visages plus élevé.

2.5.5. Résultats hétérogènes des analyseurs

Nous avons établi la possibilité de sélectionner les analyseurs en fonction de leurs domaines de fonctionnement. Dans ce contexte, la Figure 57 illustre les résultats de deux analyseurs de détection de visages différents. Le premier analyseur s'appuie

sur une librairie rendue publique par *Intel* il y a une dizaine d'années, *Open CV* [Bradski08]. Le second analyseur est un analyseur du commerce [Verilook].

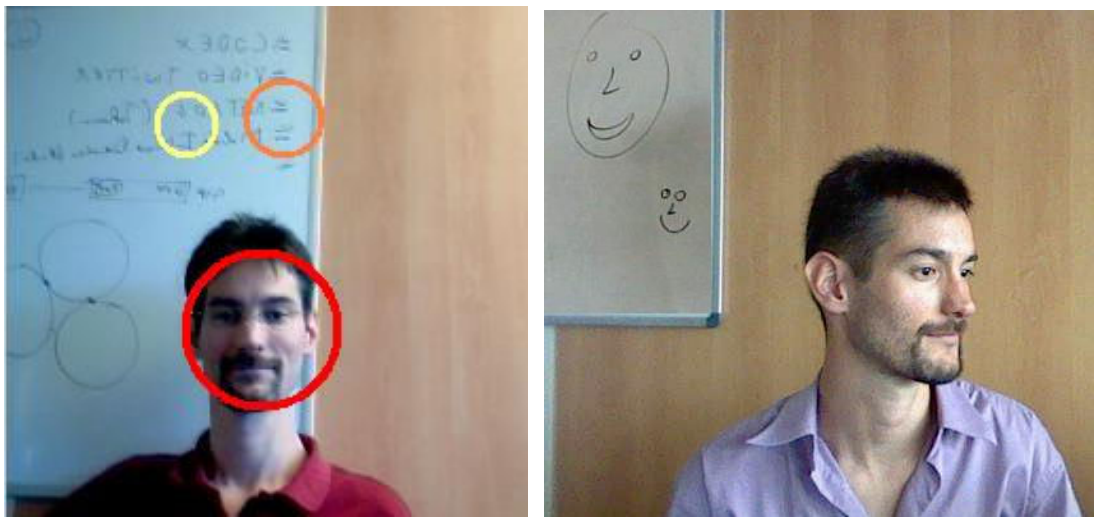


Figure 57 : Exemple de comparaison de résultats différents entre OpenCV (à gauche) et Verilook (à droite)

La Figure 57 illustre les différences pour ces deux analyseurs de même type dans des contextes à peu près similaires en termes de luminosité, contraste, sujet à détecter… La Figure 57 permet de considérer les résultats de ces deux analyseurs comme « insuffisants ».

Toutefois, ces résultats s'expliquent par les limitations des analyseurs eux-mêmes (luminance, contraste, orientation…) ainsi que par les performances (faux-positifs, vrai-positifs, faux-négatifs et vrai-négatifs) :

- l'algorithme d'analyse basé sur OpenCV (à gauche) détecte généralement tous les visages présents dans l'image, mais le taux de faux positifs est élevé,
- l'analyseur basé sur Verilook détecte seulement les « vrais » visages mais le taux de faux négatifs est élevé.

Dès lors, le choix de l'analyseur à sélectionner se fera en fonction des besoins des services interactifs en termes de quantité ou de qualité.

Par exemple, dans le cas d'une application qui affiche d'une manière visuelle des informations sur les visages détectés, la solution de Verilook sera préférée (affichage lorsque l'on est « sûr »). Dans le cas d'un service qui effectue une reconnaissance des visages avant d'afficher les informations, l'algorithme basé sur OpenCV pourra être sélectionné (les faux positifs n'étant pas reconnus dans la base de visages ne seront pas traités et moins de visages seront manqués).

2.6. *Contexte virtuel du document multimédia*

Afin d'éviter de multiplier les dépendances entre les analyseurs et les services, nous proposons de mettre en place un contexte virtuel pour le média à analyser

permettant d'assigner les descriptions issues des analyses et de les partager avec les analyseurs et les services. Ainsi, les résultats d'analyse de chaque analyseur sont stockés dans ce contexte virtuel. De manière symétrique, les analyseurs et les services ayant souscrit à des descripteurs spécifiques sont notifiés en cas de modification de ces descripteurs, et peuvent ainsi retrouver leurs valeurs.

2.6.1. Principe de fonctionnement

Analyseurs et *Services* sont liés sémantiquement à la description du contexte virtuel : cela permet à la fois de définir les règles pour déclencher les analyseurs et pour déclencher les services.

Afin que chaque analyseur puisse s'exécuter de manière correcte, le contexte virtuel multimédia permet à ces analyseurs d'accéder aux résultats précédents d'analyse du contenu dans ce contexte virtuel, ainsi qu'aux éléments du média nécessaires à l'analyse. Par ailleurs, partant du fait que les analyseurs de haut niveau s'appuient sur des résultats de bas niveau et sur des concepts de haut niveau obtenus par l'analyse du média, le système de notification permet ainsi de synchroniser l'analyse sémantique et le contexte virtuel.

2.6.2. Implantation

Le contexte virtuel du multimédia est un composant majeur dans l'implantation de l'analyse du multimédia, de la sélection et du maintien des services interactifs. Ce composant centralise les résultats des différents analyseurs média, créant ainsi une représentation « virtuelle » de l'état de la description du média. Nous avons implanté le *Design Pattern Observer* pour la gestion du contexte virtuel. Seuls les analyseurs média pour les résultats d'analyse et les accesseurs média pour la racine de la description ont un accès en écriture du contexte virtuel du document multimédia.

Tous les modules (autres analyseurs média, services interactifs…) peuvent se mettre à l'écoute de l'évolution du contexte virtuel qui est diffusé à travers l'émission d'« Event » comme l'illustre la Figure 58. Nous proposons de relier ces fonctions en exploitant la modularité des modèles présentés. Nous utilisons pour cela des technologies issues du domaine de la sémantique. L'interaction forte entre les modules qui en résulte permet de réutiliser de façon dynamique les informations d'un module pour configurer ou améliorer les fonctions ou modules qui y sont liés.

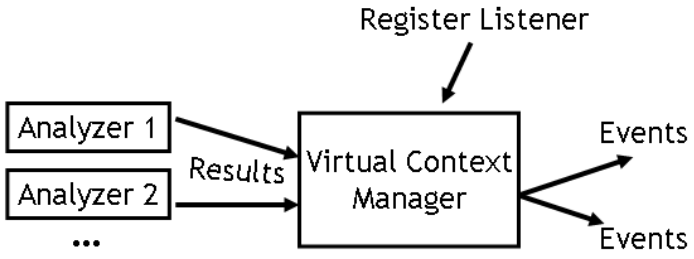


Figure 58 : Schéma d'exploitation du contexte virtuel

Ces « Event » contiennent des informations du type :

<Nom_du_Descripteur>

Type d'event : nodeAdded ou nodeRemoved ou nodeChanged

<\Nom_du_Descripteur>

Ce contexte virtuel permet donc, en fournissant une vue instantanée des événements en cours et des événements des étapes précédentes, d'améliorer les performances dans les phases de sélection des analyseurs et les phases d'analyses proprement dites. Par exemple, les résultats déjà obtenus lors de la détection et des reconnaissances de personnes peuvent ainsi être utilisés en priorité lors de la détection et des reconnaissances suivantes.

2.6.3. Gestion des accès concurrents

Cette plateforme s'appuie sur un système modulaire où chaque analyseur est associé à un module. Chacun des analyseurs peut créer, modifier ou supprimer des descripteurs de bas niveau ou des concepts de haut niveau directement dans le contexte virtuel multimédia. Chaque analyseur est également lié au cycle de vie des descripteurs. Il apparaît ainsi qu'il est nécessaire de maintenir une cohérence globale du contexte virtuel dans le cas notamment d'accès concurrents de plusieurs analyseurs. Par exemple, lorsqu'un analyseur demande la suppression d'un descripteur, il est nécessaire de vérifier d'abord qu'aucun autre analyseur ou service n'est attaché à ce descripteur.

Cette gestion concurrente des accès contient également des règles permettant d'assurer l'efficacité de la plateforme, en permettant l'arrêt d'un analyseur lorsqu'il n'est plus nécessaire.

2.6.4. Minimisation du nombre d'analyseurs actifs

Un point important étudié est de pouvoir réduire au minimum le nombre d'analyseurs actifs et exécutés à tout instant. Nous avons identifié que le principal point ici concerne l'optimisation du déploiement des analyseurs. La première proposition est donc de n'utiliser que les analyseurs nécessaires, sous-entendu nécessaires pour compléter les valeurs des descripteurs contenus dans le contexte virtuel. La deuxième proposition est de déterminer le moment où un analyseur devient inutile et de le supprimer.

Afin de ne déployer que les analyseurs nécessaires, la principale difficulté est de déterminer le sous-ensemble contextuel, ou domaine d'analyse, en fonction des résultats déjà obtenus. Par exemple, si un analyseur a permis de détecter que la scène est située sur une plage, il est plus pertinent de chercher à détecter des bateaux, des palmiers ou des surfeurs plutôt que des bureaux ou des camions.

Afin de supprimer les analyseurs inutiles, nous relions chaque analyseur au cycle de vie du descripteur associé. La Figure 59 illustre un exemple de séquence d'analyse. Cet exemple s'appuie sur des analyseurs média générant des descripteurs MPEG-7.

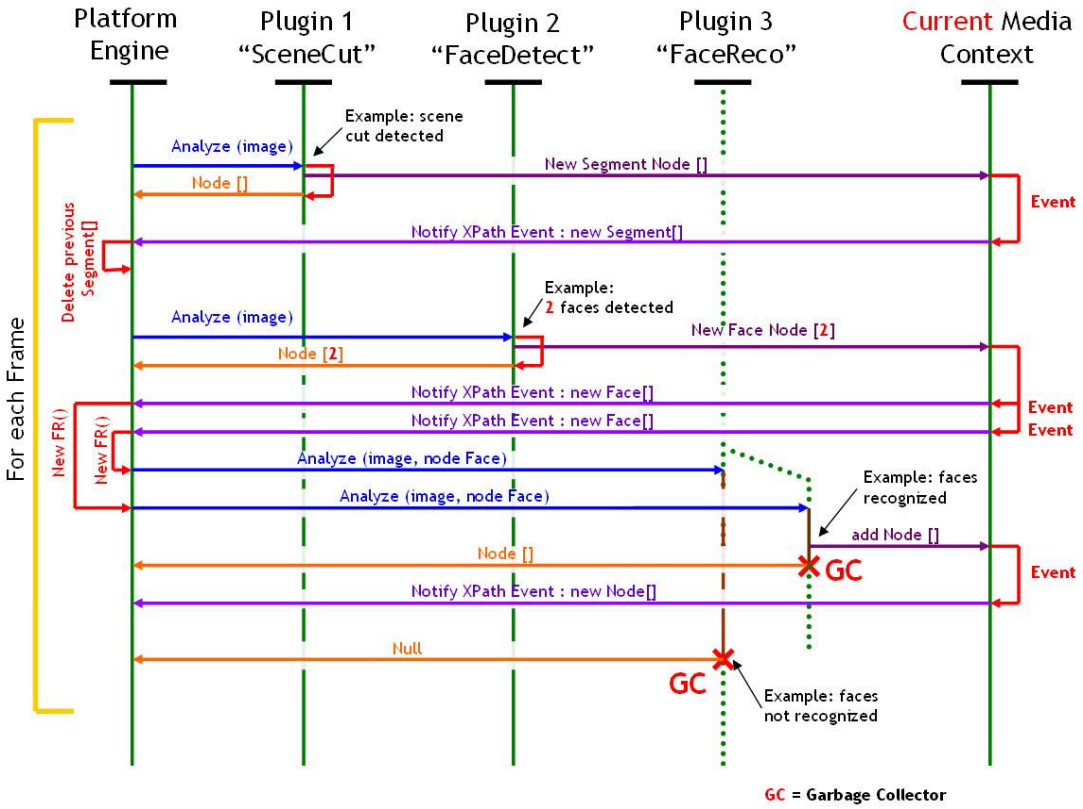


Figure 59 : Diagramme d'une séquence d'analyse

On vérifie ici que chaque instance de l'analyse de reconnaissance de visage (*Plug-in 3*) est déployée et attachée à chaque nouveau descripteur de visage (*Face node*). De plus, cet analyseur de reconnaissance de visage est stoppé une fois le résultat d'analyse obtenu (*face recognized*). Le descripteur de visage est alors retiré. Ainsi l'analyseur de reconnaissance de visage est activé uniquement lorsque des visages ont été détectés. En revanche les analyseurs de détection de scène ou de détection de visage sont toujours activés car ils ne dépendent pas des conditions en entrée.

Enfin, dans son implantation, le contexte virtuel ne rend visible que les descripteurs actifs, mais garde en mémoire l'ensemble des descripteurs qui ont été générés. Cela permet à la plateforme de fournir une description complète du multimédia à la fin de l'analyse. Les descripteurs décrivant une personne qui vient de disparaître de la scène par exemple ne sont plus visibles dans le contexte virtuel. Cependant la génération de la synthèse de la description du média contiendra toutes les phases d'apparition/disparition de la personne dans les scènes du média.

2.7. Composant de sélection pour une architecture dynamique et adaptative

La plateforme doit maintenant tenir compte des modules présents et s'adapter en fonction des besoins des services interactifs, des documents multimédias à analyser, des analyseurs disponibles…

Un champ lexical définit par un ensemble de noms, adjectifs, verbes… appartenant à une même catégorie syntaxique et liés par leur domaine de sens.

Un concept est défini par un mot (simple ou composé) ainsi que par l'ensemble du champ lexical qui lui est associé.

Le Langage Naturel Contraint (CNL) est défini comme la division de phrases en une somme de triplets « sujet »+ « verbe » + « complément ». Cela permet, dans le cadre des web services par exemple, de simplifier l'analyse de la requête qui peut être ainsi constituée de plusieurs requêtes « simples ». « Détecter les personnes qui se promènent sur la plage » peut ainsi être découpé en triplets : « détecter les personnes dans la scène » + « la scène est une plage ».

2.7.1. Etat de l'art

Le domaine de la sémantique propose un ensemble d'outils permettant des solutions pour l'analyse de concepts objectifs ou subjectifs. Les ontologies et les moteurs d'inférences par exemple permettent de raisonner sur des concepts et non plus de se limiter à la comparaison de chaînes de caractères (mots clés).

Cette évolution permet de considérer non plus un mot-clé, mais l'ensemble du champ lexical auquel il se rapporte, de proposer la correction des fautes de syntaxe, de considérer les synonymes… Ainsi par exemple, pour la recherche d'une « *voitrue* », le système pourrait proposer dans un premier temps la correction par « voiture », puis considérer le synonyme « véhicule » et associé un champ lexical associé « conducteur, route, camion, conduire, Porche… ».

Dans ce contexte, le *framework* Jena fournit un environnement de programmation pour l'implantation de standards W3C de description tels que RDF, RDF-S et OWL, ainsi que l'implantation d'un langage de requêtes standardisé SPARQL et enfin un moteur d'inférence de base.

2.7.2. Contribution

Nous avons sélectionné le *framework* Jena afin de disposer des outils de description sémantique ainsi que d'un moteur d'inférence pour raisonner sur des concepts et non plus sur la comparaison de chaînes de caractères. La Figure 60 illustre l'implantation des outils sémantiques dans la plateforme RAMSES.

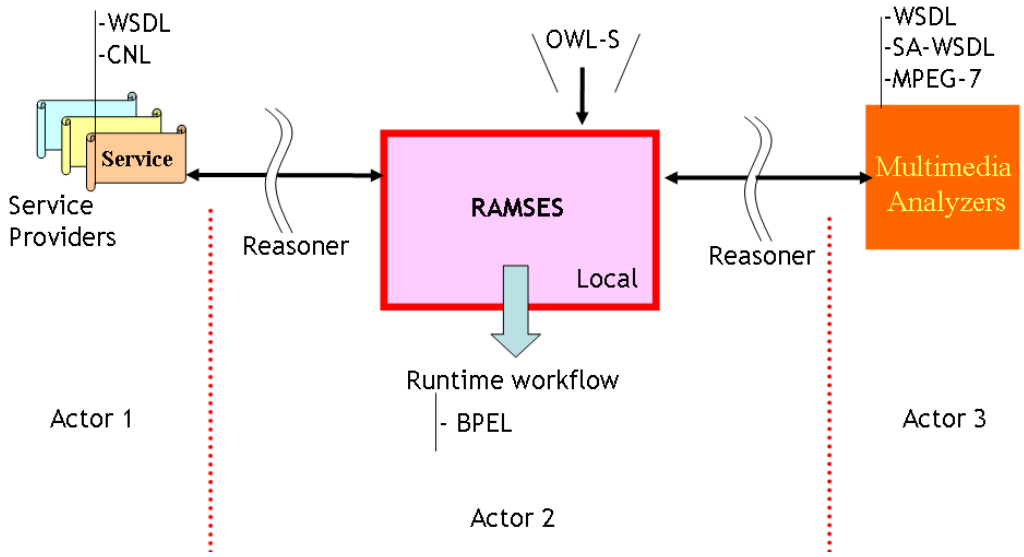


Figure 60 : Schéma d'implantation de la validation du service et de la sélection des analyseurs

Les descriptions des analyseurs de médias par exemple sont définies à l'aide de document WSDL (W3C). Les analyseurs média sont considérés de la même façon que des « Web Services » par la plateforme. Il est dès lors possible de raisonner sur l'ensemble des propriétés d'un analyseur média exprimées en langage naturel contraint (CNL).

2.8. Description d'un document multimédia

2.8.1. Granularité de description d'un document multimédia

Dans le but d'assurer l'interopérabilité, les modules d'analyse des médias reposent sur le même standard de description. Le standard MPEG-7 fournit une description formelle de la structure et du contenu d'un multimédia audiovisuel [Troncy03]. Cependant, MPEG-7 n'est pas compatible avec l'utilisation de moteurs d'inférences et ne spécifie pas non plus comment créer des liens vers des ontologies externes [Nack05]. Dès lors, comme cela est introduit dans [Arndt07], nous proposons une implantation du standard MPEG-7 pour l'utilisation des descripteurs de bas niveau et des ontologies fondées sur OWL pour décrire les concepts d'un haut niveau d'abstraction. La Figure 61 illustre le schéma d'implantation des outils de description de la plateforme.

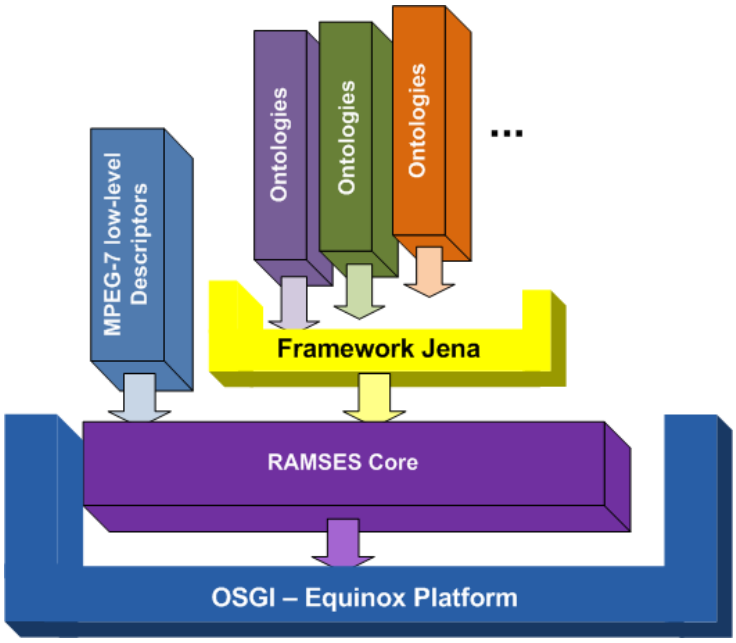


Figure 61 : Schéma d'implantation des capacités de description

L'implantation du *framework* Jena pour l'hébergement des capacités de description des concepts de haut niveau permet l'extension des « connaissances » de la plateforme en fonction des domaines à analyser.

2.8.2. Description de bas niveau MPEG-7

Les descripteurs de bas niveau MPEG-7 sont utilisés pour relier les descriptions de haut niveau au contenu « physique » des médias analysés. Nous avons sélectionné le standard MPEG-7 car celui-ci permet la description du contenu d'un multimédia audiovisuel en fournissant un ensemble complet de descripteurs de bas niveau standardisés.

Le standard MPEG-7 permet de décrire les informations liées à la création et à la production du contenu (producteur, titre, date…), au contenu du multimédia (audio, vidéo, image…), ou encore aux usages (les copyrights, le niveau de contrôle parental…), etc. Nous avons dans un premier temps transposé une partie réduite des descripteurs audiovisuels de bas niveau du standard MPEG-7. La Figure 62 représente l'arbre de description bas niveau implanté à partir des recommandations du standard MPEG-7.

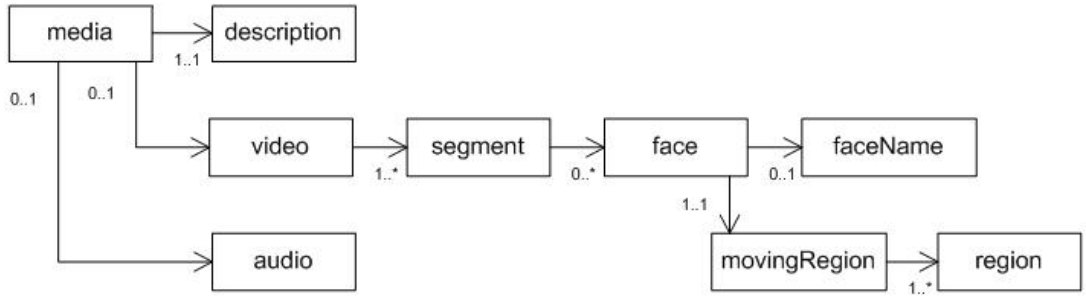


Figure 62 : Schéma de description réduit implanté sur la plateforme à partir des recommandations du standard MPEG-7

Cet ensemble réduit de descripteurs de bas niveau a permis de tester la représentation spatiotemporelle du contexte d'un document multimédia en mémoire.

2.9. Evolutions de l'implantation de la plateforme RAMSES

A partir des choix d'implantation définis ci-dessus, nous avons planifié les évolutions de la plateforme depuis la *composition manuelle des analyseurs* jusqu'à la composition et l'analyse automatique des documents multimédias. Nous avons défini à l'aide d'une représentation graphique les différentes fonctionnalités à implanter sur la plateforme RAMSES. La représentation graphique propose par ailleurs une évaluation à titre indicatif des fonctionnalités (temps réel, modularité, évolutivité et adaptabilité) associée aux étapes de déploiement. Nous sommes actuellement à l'étape trois sur quatre de cette planification.

2.9.1. Etape 1 : Composition manuelle des analyseurs

La Figure 63 représente l'architecture actuellement majoritairement utilisée dans les systèmes d'analyse basés sur la composition d'analyseurs.

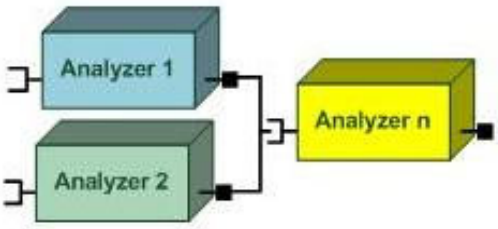
	**Temps Réel** : ●●●●●○ Modularité : ●○○○○○ Evolutivité : ●●○○○○ Adaptabilité : ○○○○○○

Figure 63 : Schéma illustrant le type de structure encore utilisé aujourd'hui pour combiner des analyseurs

Ce type d'architecture est considéré comme la solution d'implantation la plus rapide de par la liaison directe entre les différents analyseurs (échange d'information, de données…). Cette architecture offre un semblant de modularité et d'évolutivité lors de l'implantation de la plateforme de par le choix des analyseurs que l'on va combiner. Cependant, comme nous l'avons détaillé précédemment, il est difficile de remplacer un analyseur par un autre plus évolué de par le couplage fort entre les modules. Enfin, ce type d'architecture ne permet pas l'adaptation à des besoins. Il faut généralement refaire l'architecture du système.

2.9.2. Etape 2 : Plateforme statique et modulaire

La Figure 64 représente un schéma de la première étape dans la composition automatique d'analyseurs. Une API permet de définir les entrées et sorties de chacun des analyseurs. Une composition est une interface exprimant les liaisons entre les analyseurs sélectionnés et combinés manuellement. Les liaisons entre les analyseurs ne sont plus directes mais centralisées dans un contexte multimédia. La plateforme reçoit les analyseurs sous forme de plugins et gère l'appel des analyseurs et les résultats générés.

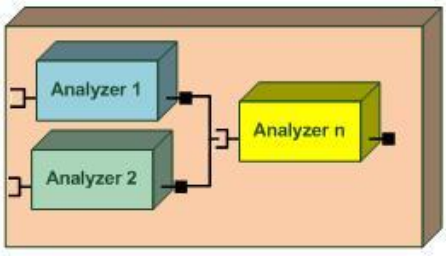
	Temps Réel : ●●○○○○ Modularité : ●●●●○○ **Evolutivité** : ●●●●●○ Adaptabilité : ●●○○○○

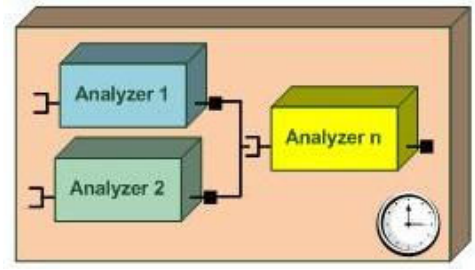
Figure 64 : Représentation d'une architecture modulaire mais statique

Ce type d'architecture est aujourd'hui accessible à travers des *frameworks* tels que celui spécifié dans le standard OSGI. L'évolutivité et la modularité sont accessibles

grâce à une architecture en plugins et la présence d'API pour interfacer les analyseurs. Il n'est pas possible de s'adapter de façon automatique aux besoins exprimés à travers la composition manuelle des analyseurs. Enfin, la modularité obtenue par la centralisation des résultats dans un contexte de description des documents multimédias se fait au détriment des performances d'analyse. Il est d'autant plus difficile de maintenir les contraintes de temps réel que le nombre d'analyseurs est élevé.

2.9.3. Etape 3 : Gestion des contraintes de temps réel

Cette étape illustrée Figure 65 est l'étape actuelle d'amélioration de la plateforme. Il s'agit ici d'implanter des technologies issues de la gestion des contraintes de temps réel telles que les sections critiques, la parallélisation, la prévention des blocages, la synchronisation des événements…



Temps Réel : ●●●●●○

Modularité : ●●●●○○

Evolutivité : ●●●●●○

Adaptabilité : ●●○○○○

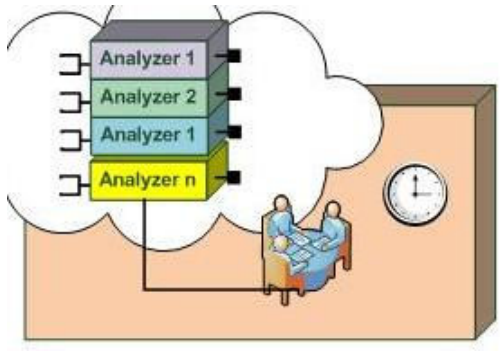
Figure 65 : Implantation des outils de gestion des contraintes de temps dans la plateforme RAMSES

La plateforme doit gérer avec précision les cycles d'analyse des différents analyseurs afin de maintenir les contraintes de temps. Ces contraintes peuvent être liées au temps disponible pour l'analyse d'une image pour les médias vidéo par exemple (fonction du nombre d'images par seconde).

Des fonctions avancées doivent dès lors être implantées par exemple pour maîtriser la *profondeur d'analyse* sur les différents concepts détectés en fonction des ressources disponibles, de la priorité des résultats pour le déploiement ou le maintien des services déployés…

2.9.4. Etape 4 : Plateforme dynamique et modulaire d'analyse

La Figure 66 illustre l'étape finale à atteindre. Le but ici est d'insérer la gestion dynamique de l'analyse des documents multimédias en sélectionnant et combinant de façon dynamique les analyseurs. Cette architecture pose deux problèmes majeurs : la résolution dynamique de la composition à créer pour répondre au besoin d'analyse et le maintien des contraintes de temps malgré cette étape supplémentaire de sélection dynamique des analyseurs.



Temps Réel : ●●●●●○
Modularité : ●●●●●○
Evolutivité : ●●●●●○
Adaptabilité : ●●●●○○

Figure 66 : Schéma d'une architecture dynamique et modulaire tenant compte des contraintes de temps

Il existe deux étapes successives dans cette architecture d'analyse. Dans un premier temps, la sélection automatique du type d'analyseurs à utiliser pour répondre au besoin d'analyse ; et dans un second temps, la sélection dynamique de l'analyseur qui correspond le mieux au contexte du multimédia pour sélectionner les meilleurs analyseurs. Le choix entre un détecteur de visage fonctionnant sur une image *infrarouge* et un détecteur de visage *classique* devra être effectué de façon dynamique en fonction des résultats d'analyse déjà obtenus.

La solution implantée dans la plateforme pour envisager la sélection automatique des analyseurs repose sur l'utilisation de moteurs d'inférence raisonnant sur des ontologies. Il est alors très difficile de maintenir les performances de fonctionnement en temps réel en fonction de la taille des ontologies présentes.

Nous avons dès lors limité dans un premier temps la sélection automatique des analyseurs à la création dynamique de la composition. Le choix de l'analyseur est effectué en fonction des requêtes des services interactifs. Le choix des analyseurs n'est pas modifié de façon dynamique durant l'analyse des médias en fonction de l'évolution des résultats d'analyse.

2.10. Validation et sélection des analyseurs

2.10.1. Sélection des analyseurs pertinents

Comme nous l'avons détaillé précédemment, la plateforme devra rapidement sélectionner un analyseur parmi plusieurs du même type (avec le même but). Dans le cas du suivi de visages par exemple, la plateforme pourra rapidement avoir à choisir entre des analyseurs « similaires » tels que ceux présents ci-dessous (liste non-exhaustive) :

- de suivi de visages [Turk02, Kim08…],
- de suivi de personnes [Siebel02],
- robuste de suivi d'objets [Tran07],
- de suivi d'objets déformables [Greminger08],

- de suivi de en mode différé [Wei07],
- …

Par ailleurs, outre le nombre d'analyseurs disponibles, les redondances entre analyseurs de même type sont à gérer pour un grand nombre de catégories telles que la détection de changement de scène (histogramme, filtre de gauss…), reconnaissance de visages (biométrie, Eigen faces…)…, et cela pour les analyseurs d'images, ou d'autres types de médias (audio, audiovisuel, texte…).

Afin de profiter de cette diversité, il sera possible d'étendre dans un premier temps le composant de sélection des analyseurs en tenant compte de façon dynamique du contexte de description du document multimédia. Par la suite, en fonction des résultats des analyseurs et de leurs performances, la plateforme pourrait sur la base d'un apprentissage maintenir une liste d'analyseurs à utiliser en « priorité » en fonction des combinaisons d'analyseurs et des besoins en description des services interactifs.

2.10.2. Composant de validation des analyseurs lors de leur insertion dans l'architecture

Cette fonction est utilisée à l'insertion d'un nouvel analyseur dans l'architecture. Elle vérifie que les capacités de description du nouvel analyseur s'intègrent avec les analyseurs déjà présents. Le rôle de cette fonction est de signaler les analyseurs qui ne peuvent être « reliés » à l'arbre de description d'un média. Un analyseur de reconnaissance des personnes sera signalé comme « inutile » dès lors que l'architecture ne supporte pas, en amont, d'analyseur capable de détecter des personnes ou des visages.

2.10.3. Composant de validation des services et sélection des analyseurs accès aux médias

Cette fonction (Figure 67) intègre la phase de validation, puis celle de sélection.

L'architecture vérifie dans un premier temps que le service qui vient d'être inséré pourra être déployé dans le média. Pour cela, il convient de s'assurer de l'existence des analyseurs nécessaires à la détection du contexte recherché, puis qu'il est possible d'accéder tant aux contenus des médias à analyser qu'aux médias à enrichir avec les scènes interactives.

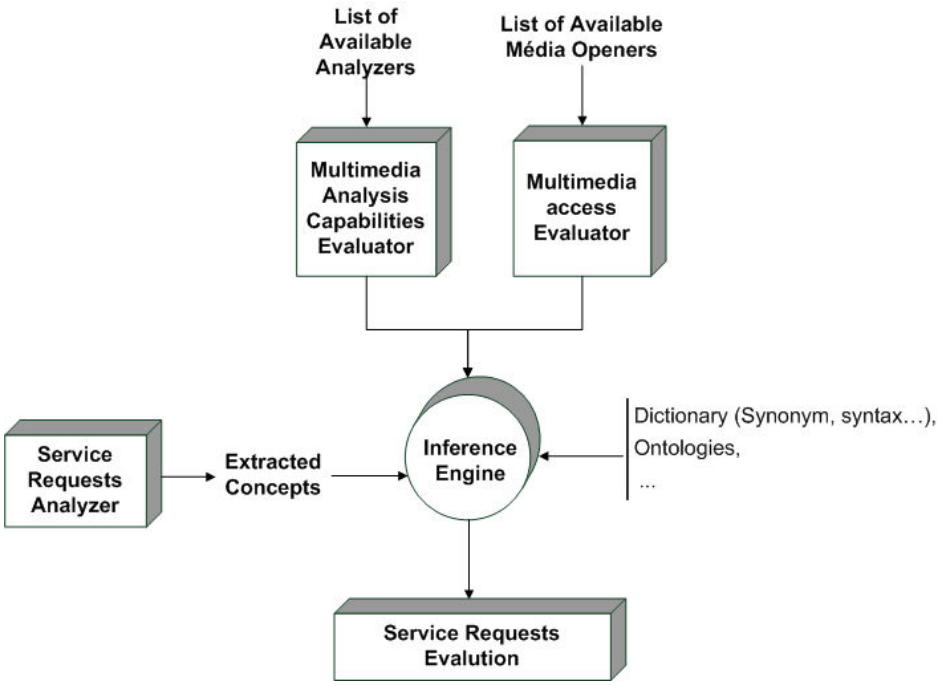


Figure 67 : Validation des requêtes des services interactifs

Enfin, l'architecture effectue une sélection des accesseurs média nécessaires pour accéder aux médias à analyser ainsi que de tous les analyseurs pertinents pour détecter le contexte du multimédia à sélectionner pour déployer les services interactifs.

La Figure 68 illustre les deux méthodes qui ont été développées pour la sélection des analyseurs pertinents vis-à-vis de chacune des requêtes de chacun des services interactifs. La première méthode fondée sur l'analyse de mots clés relatifs aux descripteurs MPEG-7 a été validée par le résultat illustré sur la droite de la Figure 68. La deuxième méthode à partir d'analyse à l'aide d'un moteur d'inférence des requêtes sous forme de langage naturel contraint (CNL) en fonction des descriptions des analyseurs média disponibles est implantée mais n'a pas encore été validée.

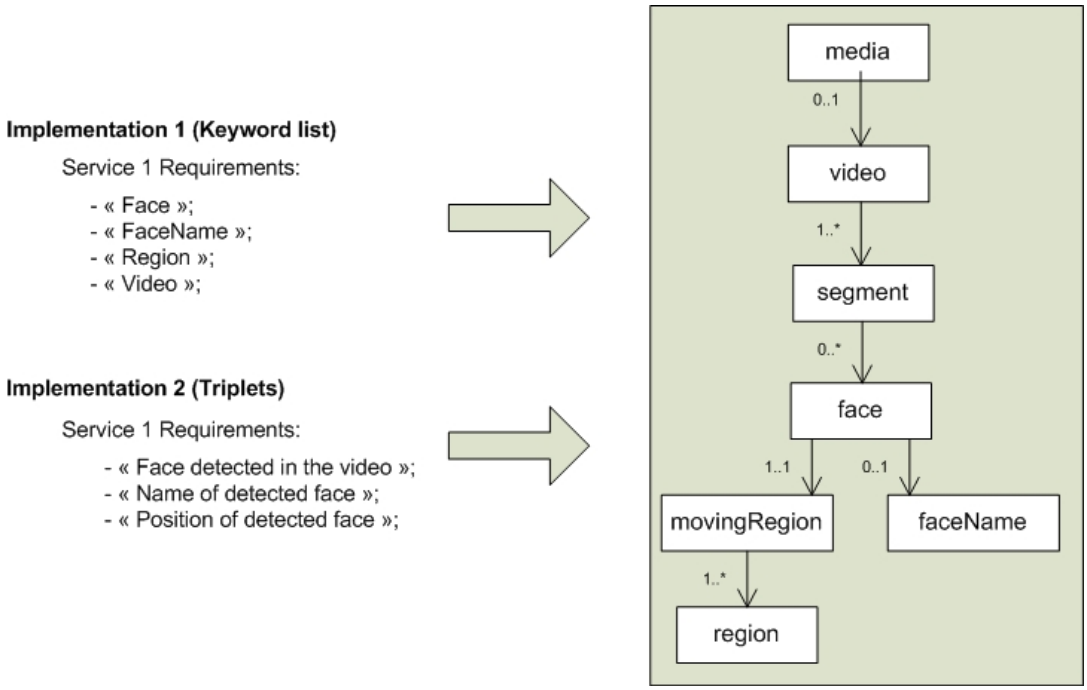


Figure 68 : Sélection des analyseurs

Dans cette seconde méthode, nous n'espérons pas une modification des résultats dans la fonction de sélection des analyseurs, mais une stabilité et une ouverture des possibilités d'analyse grâce à la suppression des limites imposées par les mots clés. Le fournisseur de services n'a plus à connaître MPEG-7 par exemple pour exprimer ses besoins en analyse.

2.10.4. Validation élémentaire du composant de sélection des analyseurs

Les figures ci-dessous illustrent les « traces » des résultats de la sélection des analyseurs pertinents suite à l'analyse de la requête du service interactif pour le cas des chemins de description MPEG-7. La requête du service interactif utilisé dans cet exemple concerne la détection, l'identification et le suivi des personnes dans la scène :

« MediaNode/VideoNode/SegmentNode/FaceNode/FaceNameNode »

« MediaNode/VideoNode/SegmentNode/FaceNode/RegionNode/MovingRegionNode »

L'Annexe 2 montre les résultats d'analyse de la plateforme. Les tableaux présentés démontrent :

- la capacité de description de la plateforme et d'extraction de la *liste des chemins de description possibles*,
- la capacité à extraire les *descripteurs* demandés ou fournis par les analyseurs média,

- la capacité à identifier la liste des *chemins d'analyse* en fonction des services interactifs présents,
- la capacité à sélectionner *les analyseurs pertinents* pour l'analyse du média.

Cet exemple illustre le fonctionnement de l'implantation, aujourd'hui limité à un petit nombre de descripteurs MPEG-7de bas niveau.

2.11. Composition et orchestration des analyseurs

L'avantage apporté par la composition d'analyseurs multimédias est de simplifier les moyens de combiner plusieurs analyseurs afin d'obtenir des résultats d'analyse de meilleure qualité, ou des résultats plus complets. La possibilité de pouvoir utiliser directement un analyseur de reconnaissance de visage sur une vidéo nécessite de pouvoir le combiner avec des algorithmes de détection et de suivi de visage. De même afin de pouvoir améliorer la reconnaissance des personnes, nous pouvons le combiner avec un analyseur de reconnaissance vocale. De la même façon, nous pouvons obtenir des résultats complémentaires en ajoutant des analyseurs *voix vers texte* (*speech to text*), des analyseurs de *segmentation des caractéristiques faciales* (*facial features segmentation algorithm*) [Hammal06, Eveno03] ou tout autre analyseur permettant d'enrichir les descriptions.

Cela permet ainsi au gestionnaire de la plateforme de se concentrer sur le niveau de description nécessaire pour la mise en place du service interactif, afin d'insérer ou combiner les analyseurs permettant d'obtenir cette description.

2.11.1. Les contraintes apportées par le temps réel

Notre proposition d'architecture RAMSES permet de fournir des services interactifs sur la base d'analyse des médias en entrée, cette analyse étant destinée à être faite à la volée, c'est-à-dire en temps réel. Le choix du mode hors-ligne ou temps réel influence directement la façon dont les analyses peuvent se dérouler, c'est-à-dire de manière asynchrone ou synchrone. Dans le cas de l'analyse *hors-ligne*, nous pouvons en effet consacrer du temps à améliorer la qualité des résultats à travers la multiplication des analyseurs, l'analyse en plusieurs passages et l'analyse avant/arrière. L'analyse en plusieurs passages permet de réutiliser comme point de départ lors des passages supplémentaires, des descriptions du média obtenues lors des passages précédents. A chacun de ces passages, la sélection des analyseurs peut être ré-effectuée afin de maximiser leur adéquation au contexte. L'analyse s'arrête lorsqu'il n'y a plus de nouveaux éléments permettant d'enrichir la description du média. L'analyse avant/arrière permet par exemple lors d'un suivi de visage ou d'objet, de revenir en arrière en partant du premier instant où le visage a été détecté afin de le suivre jusqu'au moment où il est réellement apparu, sans pour autant être détectable. L'analyse repart alors en avant à partir de ce point.

Dans le cas temps-réel, nous devons nous consacrer au compromis qualité / performance et faire des choix d'architecture qui exploiteront au mieux les performances élémentaires des analyseurs.

2.11.1.1. Mode temps-réel

Une des contraintes les plus difficiles à satisfaire est cette contrainte temps-réel : il convient de mettre en œuvre une architecture qui permette de parvenir au temps-

réel le plus naturellement possible. Il faut également exploiter des analyseurs qui soient les plus performants en termes de temps de calcul. Afin de ne pas dépendre directement des performances des analyseurs, nous nous sommes consacrés à mettre au point une architecture et des principes qui facilitent le passage au temps-réel : architecture modulaire afin de pouvoir mettre en œuvre les analyseurs les plus performants en fonction de l'évolution des analyseurs disponibles, la gestion de la mémoire et du partage du contexte entre les différents analyseurs, la minimisation des calculs effectués en sélectionnant et orchestrant au mieux ces analyseurs.

Nous mettons en œuvre d'une part, un contexte virtuel qui contient à tout instant l'état d'avancement de l'analyse globale. Par exemple, tant qu'une personne est détectée mais pas reconnue, alors qu'elle doit l'être, la plateforme maintient un état associé à cette reconnaissance. Il est indispensable que ce contexte soit disponible et modifiable par chaque analyseur à tout instant. Nous avons étudié la solution détaillée dans [Burnard01] pour maintenir de manière similaire un schéma de description XML (DTD). DTD. Cette référence montre qu'afin de maintenir un document XML valide, les nœuds parents doivent être modifiés avant les nœuds enfants. De manière similaire, lors de l'occurrence d'une nouvelle scène intégrant des personnes à décrire, cette nouvelle scène doit être créée et renseignée avant que les personnes soient décrites.

D'autre part, la plateforme est conçue pour adapter en permanence les analyseurs effectivement déployés en fonction des descripteurs qu'il est nécessaire de renseigner. Cela nécessite d'associer un état à chacun des descripteurs d'une part, et de mettre en œuvre un cycle de vie pour les analyseurs s'appuyant sur ces états pour activer ou désactiver de manière dynamique les analyseurs en fonction des besoins.

Pour cette analyse globale, l'horloge de référence de la plateforme est déterminée à partir de la plus petite période de temps issue des entrées multimédias. Cette unité de temps est exprimée en trame par seconde pour les médias audiovisuels ou en fréquence pour des médias audio purs.

2.11.1.2. Mise au point d'un cycle de vie des analyseurs

Les analyseurs sont « autonomes » et « responsables ». Le rôle de la plateforme est de pouvoir déployer les analyseurs en fonction des besoins des services interactifs pour la description du média, mais aussi en fonction des besoins des analyseurs pour leur propre fonctionnement. Un analyseur de reconnaissance de visage par exemple a besoin de l'image en cours d'analyse, mais aussi de l'emplacement du visage détecté pour fonctionner correctement. Un des objectifs de l'étape de composition est donc, à partir de l'arbre de description représentant l'expression des besoins des services interactifs, d'attacher un analyseur pour chaque nœud et de vérifier que les entrées de ceux-ci sont bien renseignées. Ainsi, chacun des analyseurs peut, à l'issue de l'analyse, assigner le résultat obtenu en tant que valeur à son descripteur associé.

De plus, cette méthode permet une gestion plus fine de la profondeur d'analyse. La Figure 69 reprend la Figure 68 en mettant en avant les relations de dépendances entre les descripteurs par l'intermédiaire des analyseurs.

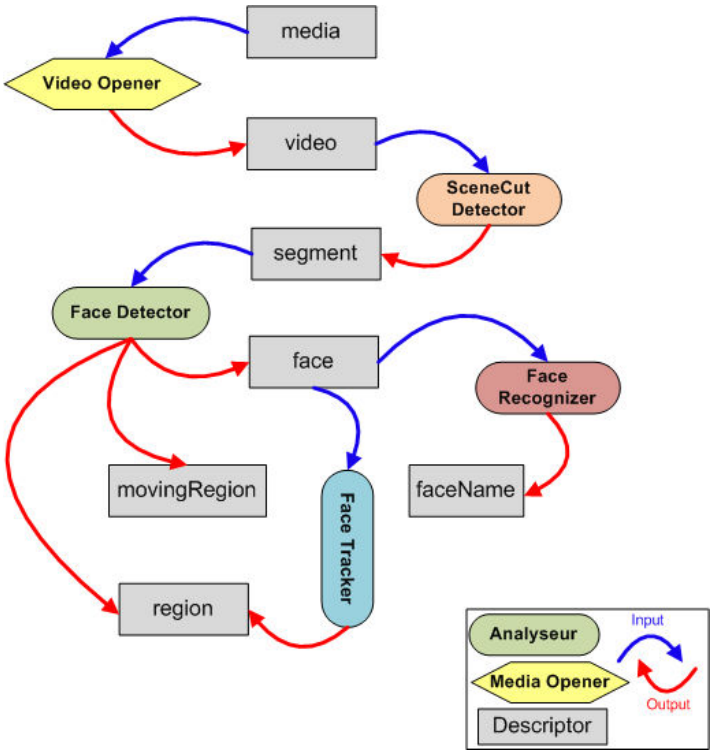


Figure 69 : Diagramme de gestion des analyseurs en fonction des dépendances dans une composition

La Figure 70 détaille les étapes de la composition par lesquelles chaque analyseur est déclenché en fonction des événements associés au contexte multimédia (création, mise à jour ou suppression de nœuds). En effet, si un nœud auquel est attaché un analyseur n'est plus actif, l'analyseur sera supprimé de la liste des analyseurs à composer. Par exemple, si un visage présent jusqu'alors, disparaît d'une scène, l'instance de l'analyseur suivi de visage sera alors supprimée pour les prochaines compositions. Le cycle de vie d'un analyseur (activation, déclenchement, suppression) dépend donc directement des avancées effectuées sur la description globale du document multimédia.

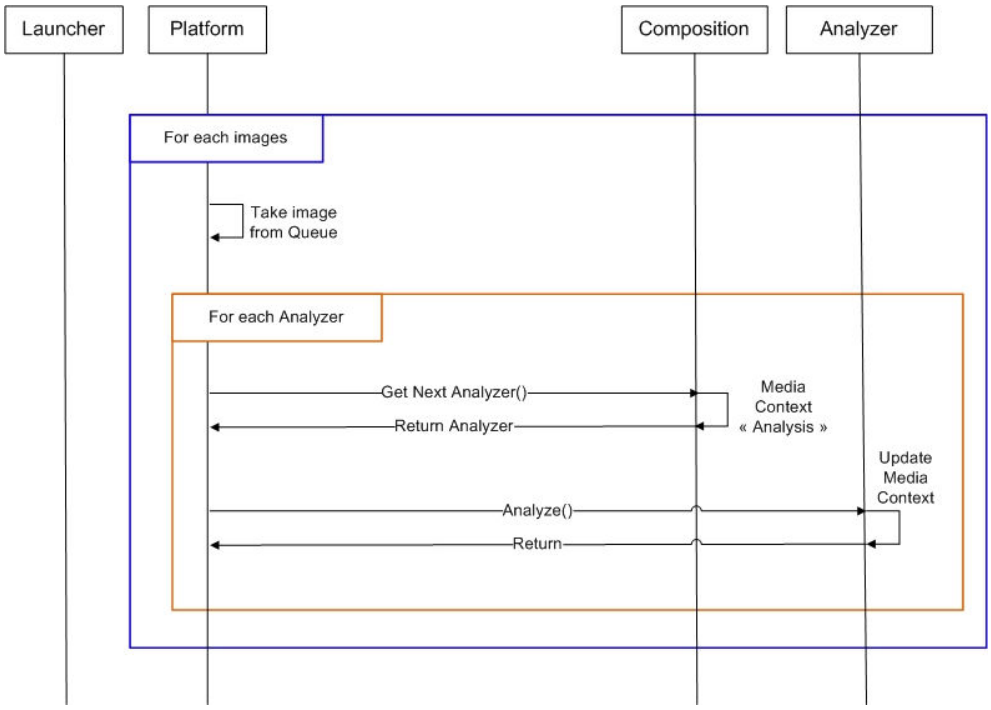


Figure 70 : Diagramme de séquences d'appel des analyseurs

2.11.2. Composant de génération de la composition des analyseurs

La composition d'analyseurs consiste à construire un chemin d'analyse exploitant les entrées et sorties des différents analyseurs en séquençant les analyses dans un ordre optimal permettant d'obtenir les résultats combinés de ces analyseurs. Cette composition peut être construite *a priori*, indépendamment de la plateforme, de manière statique, puis insérée dans la plateforme. Cette composition peut également exploiter une liste d'analyseurs présélectionnés et les combiner au mieux. Elle peut enfin introduire le composant de sélection dans la composition elle-même afin que les choix des analyseurs puissent être non pas fais une fois pour toute au début de l'analyse, mais dépendre des résultats d'analyse eux-mêmes. Nous décrivons ces différents cas ci-dessous.

2.11.2.1. Cas de la composition statique

La plateforme permet d'ajouter des analyseurs qui ont été composés entre eux manuellement. Le résultat de cette composition d'analyseurs peut être vu comme un analyseur lui-même par la plateforme. De cette manière la composition statique de plusieurs analyseurs peut être ajoutée en tant que "*bundle*" afin de pouvoir déclarer les entrées et sorties de ces « nouveaux » analyseurs composés.

La Figure 71 présente un exemple de composition d'analyseurs que nous avons mis en place sur la plateforme. Cette figure présente une vue statique des dépendances entre analyseurs. En termes de composition, cela montre que les analyseurs *Reconnaissance de Visage* (*Face Recognition)* et *Suivi de visage* (*Face Tracker)* sont dépendants des résultats de l'analyseur *Détection de Visage* (*Face Detector)*. Cela montre également que les entrées nécessaires dans ce cas sont la *Trame Vidéo en Cours* (*Current Frame*) et la *Notification de Changement de Scène*

(*Scenecut Descriptor status*). L'analyseur fournit alors en sortie la position des visages détectés et le nom des personnes reconnues.

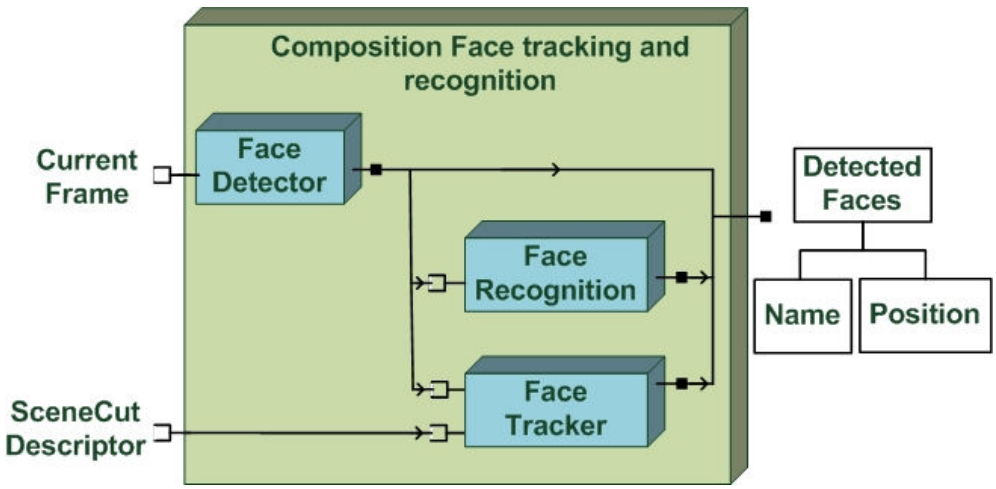


Figure 71 : Schéma de composition statique, exemple

Cette solution a l'avantage de permettre d'encapsuler des analyseurs complexes dans une solution d'analyse plus globale, de manière simple, et de construire des analyseurs de haut-niveau étape par étape.

2.11.2.2. Composition dynamique d'analyseurs déjà sélectionnés

Cette fonction a pour but de définir l'ordonnancement d'analyse à partir de l'ensemble des analyseurs sélectionnés dans un premier temps. La Figure 72 montre comment s'insère la détermination de cet ordonnancement dans le schéma global d'initialisation de la plateforme, depuis les opérations de détection des modules disponibles (analyseurs, services interactifs…) jusqu'à la mise en place de la composition. Dans dette dernière étape les opérations sont les suivantes :

- *affiner les choix des analyseurs* lorsque plusieurs analyseurs permettent de fournir les mêmes informations (nous verrons par la suite que les analyseurs redondants devront être conservés) ;
- *construire les chemins d'analyse*. Il s'agit ici d'identifier tous les analyseurs nécessaires pour remonter depuis chacun des analyseurs jusqu'à la racine de l'arbre de description. Si un analyseur permettant la reconnaissance des visages a été sélectionné, l'architecture vérifie qu'un détecteur de visage est présent et ainsi de suite jusqu'à la racine « piste vidéo » de la description ;
- *effectuer une pondération des chemins d'analyse*. Un chemin d'analyse qui apporte de l'information rapidement pour tous les services mais qui ne permet aucune décision d'insertion des services interactifs n'est pas pertinent. Il s'agit ici de différencier les chemins qui apportent des informations de ceux qui permettent de valider l'insertion de chacun des services interactifs.

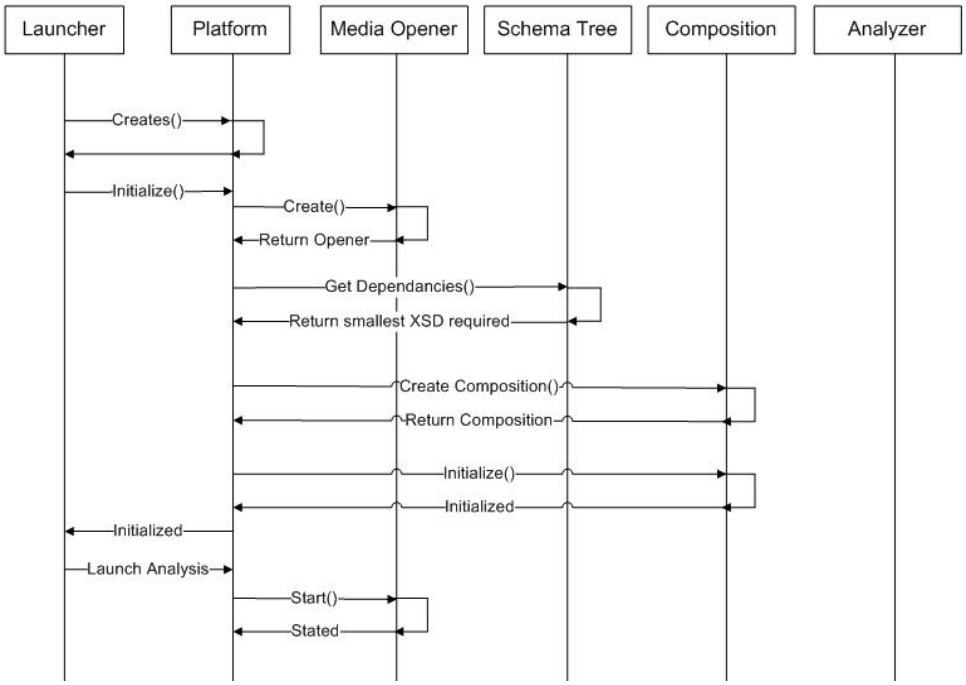


Figure 72 : Initialisation de la plateforme et phase de composition

La composition ainsi générée sera utilisée pour effectuer l'orchestration des analyseurs. Un chemin est « bloqué » et stoppe la composition d'analyseurs lorsque la plateforme ne trouve pas d'analyseur pour fournir le(s) descripteur(s) demandé(s).

2.11.2.3. Composition complètement dynamique d'analyseurs

Dans ce modèle de composition complètement dynamique des capacités d'analyse, chaque analyseur est choisi et activé en fonction des résultats d'analyse précédents. Dans ce cadre, la plateforme utilisera les propriétés de description des capacités d'analyse de la même manière que proposée dans les web services [Moulin05]. Cette sélection dynamique pourra être améliorée en introduisant des capacités sémantiques à la plateforme. Cette sélection pourra s'adapter au contexte d'analyse qui déterminera les analyseurs les plus appropriés ou au contraire les moins appropriés. Dans le cas où deux analyseurs de détection de visage sont disponibles, chacun ayant des contraintes explicites et différentes en termes de luminosité, le fait d'introduire un analyseur de luminosité permettra dynamiquement de choisir l'analyseur le plus adapté aux conditions réelles de l'image en cours. Les capacités sémantiques sont ici nécessaires pour faire le lien entre entrées, sorties et contraintes et tenir compte ainsi au mieux des possibilités opérationnelles des analyseurs.

2.11.3. Composant d'orchestration des analyseurs

Ce composant est appelé orchestrateur dans la suite de cette section.

L'implantation de l'orchestration (Figure 73) des analyseurs repose sur la liste des analyseurs sélectionnés pour effectuer l'analyse du média, sur le contexte virtuel et sur les contenus des médias fournis par les accesseurs média.

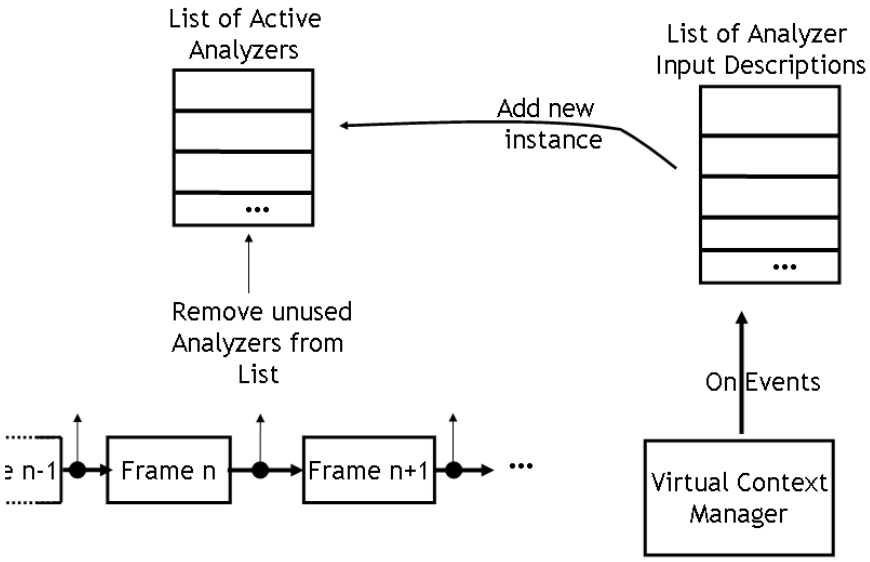


Figure 73 : Schéma d'orchestration des analyseurs

L'orchestration des analyseurs est synchronisée sur l'unité de temps la plus petite des accesseurs média. Dans l'exemple de l'accesseur média pour le média vidéo, il s'agit d'une image de cette vidéo.

Afin de limiter la consommation des ressources, l'orchestrateur ne déploie que les analyseurs qui sont utiles pour l'analyse de l'unité de temps du média en cours. Deux listes distinctes sont alors gérées : celle des analyseurs actifs constituée des analyseurs qui accèdent au contenu du média et fournissent des résultats d'analyse ; et celle des analyseurs qui ont été sélectionnés pour analyser le média, mais qui sont en attente d'être activés.

Nous avons implanté deux systèmes de synchronisation pour gérer la liste des analyseurs actifs. Dans un premier temps, après chaque image analysée, l'orchestrateur supprime de la liste des analyseurs actifs, les analyseurs qui signalent la fin de leur analyse. Par exemple, un suiveur de visage qui indique que le visage n'est plus dans la scène signale qu'il n'est plus « utile » et sera retiré de la liste des analyseurs actifs. Dans un deuxième temps, l'orchestrateur est en écoute des *Events* émis par le contexte virtuel pour vérifier si un analyseur peut être déployé. L'orchestrateur effectue pour cela une vérification auprès de tous les analyseurs présents dans la liste des analyseurs sélectionnés mais en attente d'être activés.

Cette double synchronisation permet de solliciter de façon immédiate les analyseurs en attente (*Events*). Cela permet dans un second temps aux analyseurs média de pouvoir libérer les ressources en ne les stoppant pas brutalement. Les analyseurs média ont dès lors la responsabilité de signaler quand ils peuvent être arrêtés et remis en attente (*useful flag*).

2.11.4. Validation élémentaire du composant de génération de la composition des analyseurs

La plateforme fonctionne de façon implicite pour les cas d'analyse linéaire comme dans les exemples de service interactifs proposés ici. En effet, le cycle de vie des

analyseurs correspond exactement aux cycles de vie des descripteurs qu'ils fournissent ainsi qu'aux cycles d'analyse du média par la composition et l'orchestration implantée. Le problème de composition non linéaire d'analyseurs s'est posé avec l'analyseur de suivi des visages. En effet, même si le descripteur de position d'un visage est présent et actif dans une image, il faut tout de même réactiver l'analyseur pour l'image suivante. Nous avons temporairement résolu ce problème en laissant à l'analyseur déployé le choix du moment de sa désactivation, à savoir lorsqu'il n'y a plus de visage à suivre pour l'analyseur de suivi de visage.

2.11.4.1. Validation simple

L'Annexe 2 présente les résultats d'analyse du média et illustre le fonctionnement de la plateforme en reprenant les *traces d'exécution de l'analyse des images* du média vidéo. Les tableaux présentés démontrent :

- le bon fonctionnement des analyseurs sélectionnés ;
- le déploiement des analyseurs en fonction de l'évolution du média ;
- l'arrêt des analyseurs une fois que ceux-ci ne sont plus "nécessaires" pour l'analyse en cours du média.

Cet exemple illustre le bon fonctionnement de la plateforme pour la gestion des appels des analyseurs par l'orchestrateur ainsi que la répartition des événements par le contexte virtuel.

2.11.4.2. Validation évoluée

Afin de pouvoir expérimenter et valider des compositions d'analyseurs plus complexes, nous avons préparé la plateforme pour l'implantation ultérieure d'une interface graphique de gestion de trois modes de composition. Cela permettra au gestionnaire de la plateforme d'avoir trois niveaux d'interaction sur la composition d'analyseurs proposée :

- le premier mode, ou *mode automatique*, permet de tracer les opérations effectuées lors de la composition, puis de l'orchestration en affichant que les analyseurs sélectionnés et leurs statuts (actifs ou désactivés) ;
- le deuxième mode, ou *mode semi-automatique*, permet au gestionnaire de la plateforme de participer à la sélection des analyseurs dans le cas où la plateforme se trouve face à un choix : par exemple, deux analyseurs de détection de visage sont disponibles et la plateforme ne dispose pas de critères de sélection entre ces deux modules ;
- le troisième mode, ou *mode manuel*, permet au gestionnaire de la plateforme de contrôler complètement la phase de composition afin de la corriger ou de la modifier. Il s'agira pour le gestionnaire d'indiquer les analyseurs à utiliser ainsi que leurs dépendances en termes de descripteurs pour les déclencher. De cette manière, la plateforme peut être vue comme un outil de test de génération de composition d'analyseurs permettant de concevoir des principes de compositions efficaces et évolutifs.

Dans ce *mode manuel*, si par exemple un analyser de reconnaissance de visage ne fournit pas de bons résultats, en raison de la qualité ou de la particularité du média en entrée, le gestionnaire de la plateforme pourra insérer un nouvel analyseur afin d'ajouter une étape de vérification. Si le résultat est alors considéré comme valable,

un nouvel analyseur de reconnaissance de visage est associé de manière persistante au descripteur *FaceNode*.

Face aux deux complexités résidant d'une part dans la capacité à décrire de façon suffisamment détaillée un document multimédia répondant au besoin de mise en place des services interactifs, et de pouvoir disposer des analyseurs permettant de construire cette description, il n'est pas absolument certain qu'il existe une composition des analyseurs disponibles répondant au besoin initial. C'est pourquoi nous recommandons la mise en place d'une interface de recommandation permettant au gestionnaire de la plateforme d'exprimer des retours sur les résultats d'analyse obtenus. Il serait ainsi possible par un système de « vote » de créer une liste de « favoris » ou d'analyseurs à proscrire.

2.12. Synthèse de l'architecture de déploiement

La Figure 74 synthétise les différents composants de l'architecture implantée. Le développement de cette architecture a été valorisé par le dépôt de quatre brevets, ceux-ci sont rappelés à la fin de ce document dans l'Annexe 1.

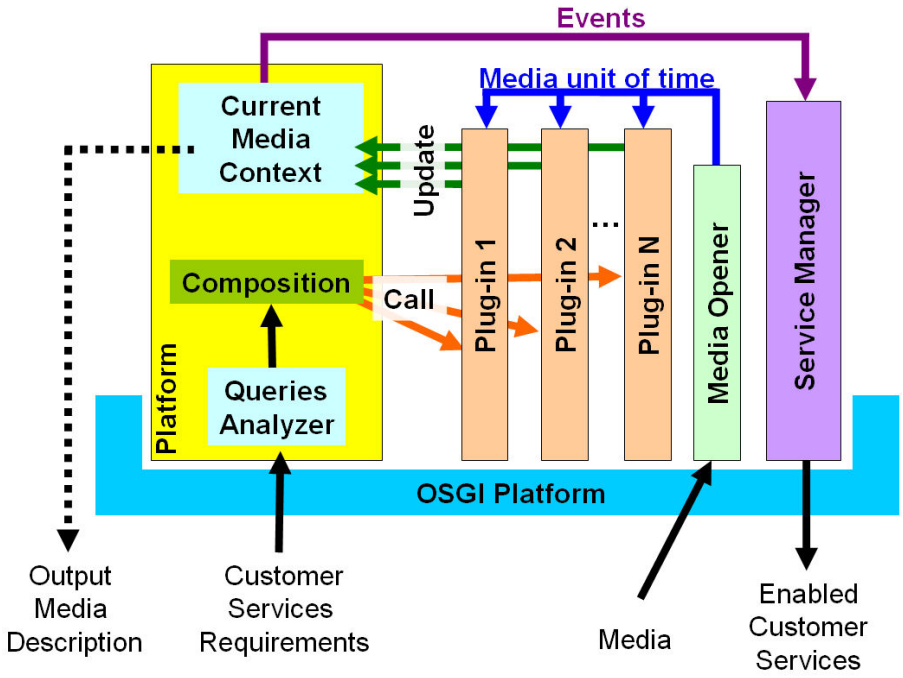
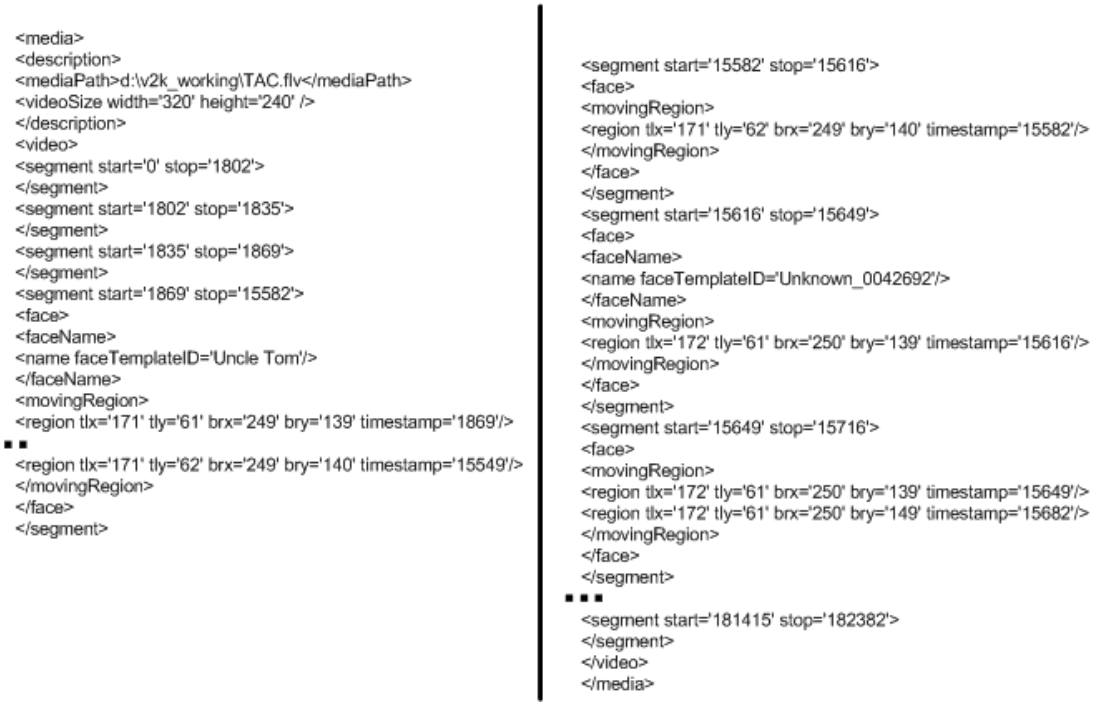


Figure 74 : Schéma simplifié de déploiement

2.12.1.1. Génération d'un fichier de description final

La Figure 75 illustre le fichier de description d'un média vidéo. Les *trois petits points* dans la figure représentent une suppression d'une partie de la description (*répétitions*) pour ne laisser qu'une partie représentative du résultat final de description bas niveau.


<media>
<description>
<mediaPath>d:\v2k_working\TAC.flv</mediaPath>
<videoSize width='320' height='240' />
</description>
<video>

<face>
<faceName>
<name faceTemplateID='Uncle Tom'/>
</faceName>
<movingRegion>
<region tlx='171' tly='61' brx='249' bry='139' timestamp='1869'/>
<region tlx='171' tly='62' brx='249' bry='140' timestamp='15549'/>
</movingRegion>
</face>

<face>
<movingRegion>
<region tlx='171' tly='62' brx='249' bry='140' timestamp='15582'/>
</movingRegion>
</face>

<face>
<faceName>
<name faceTemplateID='Unknown_0042692'/>
</faceName>
<movingRegion>
<region tlx='172' tly='61' brx='250' bry='139' timestamp='15616'/>
</movingRegion>
</face>

<face>
<movingRegion>
<region tlx='172' tly='61' brx='250' bry='139' timestamp='15649'/>
<region tlx='172' tly='61' brx='250' bry='149' timestamp='15682'/>
</movingRegion>
</face>

</video>
</media>

Figure 75 : Exemple de résultat de description bas niveau

La génération d'un fichier de description globale permet donc la réutilisation de cette analyse pour la recherche d'informations similaires ou complémentaires par d'autres outils.

2.12.2. Gain d'une plateforme adaptative

L'exemple représenté à l'aide des trois tableaux ci-dessous illustre la création manuelle d'une composition. Celle-ci met en œuvre deux analyseurs : un de détection de visages et l'autre de suivi des visages détectés. Les parties de la composition qui sont à modifier pour l'adapter aux besoins sont surlignées.

Tableau 6 : Spécification des packages des analyseurs média à utiliser

```
static private final List<String> MANDATORY_ANALYZERS
= Arrays.asList(
 "com.alblf.media_knowledge.scenecut.internal.SceneCutAnalyzerFactory",
"com.alblf.media_knowledge.facedetect.internal.FaceDetectAnalyzerFactory"
);
```

Le Tableau 7 illustre la création d'une liste vide d'analyseurs média, la création d'un contexte virtuel (MediaContext) et la création d'un conteneur pour recevoir l'image à analyser.

Tableau 7 : initialisation des composants de gestion des analyseurs

```
private final List<IVideoAnalyzer> _videoAnalyzers = new
CopyOnWriteArrayList<IVideoAnalyzer>() ;
```

```
private final MediaContext _mediaContext;
private BufferedImageHandler _previousHandler = null;
private int _pointer = 0;
```

Le constructeur de la composition contient la construction du contexte virtuel comme indiqué dans le Tableau 8 et demande à la plateforme la liste des *Factory* des analyseurs média sélectionnés.

Tableau 8 : Agrégation de la liste des analyseurs pertinents pour l'analyse du média

```
public CompositionBItv(final MediaContext mediaContext, final String[]
requiredPluginXPATH) {
        _mediaContext = mediaContext;
        _factories =
AnalyzersFactoriesRepository.getInstance().getFactoriesOf(requiredPluginXP
ATH);
}
```

La fonction de l'orchestrateur (Tableau 9) consiste à demander itérativement à la composition le prochain analyseur à exécuter sur l'image en cours On effectue ainsi une itération sur la liste des analyseurs que l'on souhaite exécuter sur chaque image jusqu'à la fin du média.

Tableau 9 : Algorithme de sélection des analyseurs à déployer en fonction de l'évolution du contexte du média

```
@Override
public IVideoAnalyzer getNextAnalyzer(final BufferedImageHandler handler) {
  IVideoAnalyzer analyzer = null;
  if (!handler.equals(_previousHandler)) {
    //next frame to analyze
    For (final IVideoAnalyzer videoAnalyzer : _videoAnalyzers) {
          if (!videoAnalyzer.isStillUseful()) {
                // Remove unusefull analyzers
                _videoAnalyzers.remove(videoAnalyzer);
           }
    }
    _pointer = 0; // get back to top list of analyzers
  }
  if (_pointer < _videoAnalyzers.size()) {
    analyzer = _videoAnalyzers.get(_pointer);
  }
  _pointer++;
  _previousHandler = handler;
  return analyzer;
}
```

Cette liste d'analyseurs à exécuter est choisie lors de la programmation de la composition. Le Tableau 10 illustre le véritable travail concernant la réalisation d'une composition d'analyses. La programmation se résume à initialiser le choix des analyseurs à utiliser et les actions à effectuer en fonction des différents résultats des analyses précédentes.

Tableau 10 : Modifications à apporter pour la gestion des événements et de l'orchestration pour l'analyse du média

```
@Override
public void initialize() {
  for (final IAnalyzerFactory factory : _factories) {
    if (MANDATORY_ANALYZERS.contains(factory.getClass().getName())) {
  final IAnalyzer createAnalyzer =
                    factory.createAnalyzer(_mediaContext.getRoot());
      if (createAnalyzer instanceof IVideoAnalyzer) {
          final IVideoAnalyzer videoAnalyzer =
                            (IVideoAnalyzer) createAnalyzer;
          _videoAnalyzers.add(videoAnalyzer);
      }
    }
   }
    _mediaContext.addListener( new IMediaContextListener() {
        @Override
        public void nodeAdded(final AbstractNode node) {
       if (node instanceof FaceNode) {      // Face detected
            final FaceNode faceNode = (FaceNode) node;
            // Adding Face Reco Analyzer
            for (final IAnalyzerFactory factory : _factories) {
               if (Arrays.asList(factory.getXPathsOutput()).contains(
      "MediaNode/VideoNode/SegmentNode/FaceNode/FaceNameNode")) {
                    LOGGER.debug("Connect a new facereco on the facenode");
                    final IAnalyzer faceAnalyzer = factory.createAnalyzer(faceNode);
                    if (faceAnalyzer instanceof IVideoAnalyzer) {
                       final IVideoAnalyzer videoAnalyzer =
                                          (IVideoAnalyzer) faceAnalyzer;
                       _videoAnalyzers.add(videoAnalyzer);
                    }
            }
            // Adding FaceTracker Analyzer
              if (Arrays.asList(factory.getXPathsOutput()).contains(
"MediaNode/VideoNode/SegmentNode/FaceNode/RegionNode/MovingRegionNod
e")) {
                  LOGGER.debug("Connect a new Face Tracker on the facenode");
                  final IAnalyzer faceAnalyzer =
                                       factory.createAnalyzer(faceNode);
                if (faceAnalyzer instanceof IVideoAnalyzer) {
                     final IVideoAnalyzer videoAnalyzer =
                                          (IVideoAnalyzer) faceAnalyzer;
                    _videoAnalyzers.add(videoAnalyzer);
                }
               }
        }
  }
}
```

Cet exemple met en avant le *gain de temps* obtenu par l'amélioration de la facilité de combinaison des analyseurs média. Nous avons ainsi pu tester facilement différentes combinaisons d'analyseurs afin d'augmenter de façon globale les performances des résultats d'analyse.

Les premiers essais d'utilisation d'analyseurs ont montré qu'il fallait entre une semaine et un mois pour mettre en place un outil d'analyse à partir d'outils sur étagère [Verilook] ou d'outils disponibles en open source, durée variable suivant que nous combinions ou non plusieurs analyseurs. Grâce aux outils de modélisation et de composition fournis par notre plateforme, cette durée est de un jour au maximum

(une demi-journée dans le cas de [Verilook]). Le facteur de gain en temps obtenu varie de 5 à 20.

2.13. Conclusion

Constant le besoin d'implanter un modèle évolutif et dynamique, nous avons proposé un système d'analyse qui se configure en fonction des services interactifs pour cibler les informations à analyser dans les documents multimédias.

Nous avons proposé, à partir de l'état de l'art sur les technologies disponibles, une implantation ou des recommandations pour l'implantation de chacune des contributions proposées. Ainsi, le standard OSGI a été choisi pour permettre la modularité de la plateforme. Ce standard a servi de guide pour l'implantation des modèles développés et la définition des API pour les analyseurs, les services interactifs, les accesseurs de médias...

Une implantation du contexte virtuel a ensuite été proposée. Celle-ci permet d'accéder aux éléments de description actifs du document multimédia. L'architecture réalisée maintient à jour la description d'un document multimédia selon les différents niveaux d'abstraction depuis les descripteurs de bas niveau MPEG-7 jusqu'aux concepts identifiés.

Nous avons ensuite relié ces *modules* en exploitant la modularité des modèles présentés. Nous avons pour cela implanté des technologies issues du domaine de la sémantique. L'interaction forte entre les modules qui en résulte permet de réutiliser de façon dynamique les informations d'un module pour configurer ou améliorer les fonctions ou modules qui y sont liés.

L'implantation de l'ensemble du modèle d'analyse est une façon opérationnelle d'envisager la réduction du gap sémantique par le déploiement d'un grand nombre d'analyseurs permettant la vérification des différents résultats des analyseurs en combinant les résultats d'analyses complémentaires.

Nous avons enfin présenté et discuté de premiers résultats validant les contributions proposées.

Nous avons par ailleurs introduit l'implantation du *framework* Jena permettant une extension de la capacité de description de la plateforme RAMSES. Le niveau d'abstraction accessible de la description d'un multimédia dépend de la capacité d'analyse des analyseurs présents ainsi que de l'ensemble des connaissances de la plateforme (capacité de description du média et des concepts). Il est dès lors possible de modifier les capacités d'analyse de la plateforme en intervenant sur les analyseurs présents ou en ajoutant des modules de description dans la plateforme (ontologies). En effet, l'implantation de moteurs d'inférences pour la sélection des analyseurs induit une liaison forte entre les capacités d'analyse des analyseurs média et la capacité de description de la plateforme.

Le chapitre suivant décrit les quatre services que nous avons déployés à partir de la plateforme RAMSES.

Chapitre 3

RAMSES : Exemples de services interactifs et discussion

Ce chapitre détaille l'objectif principal que nous visons pour le déploiement automatique de l'interactivité. Par conséquent, nous décrivons tout d'abord les différents services d'interactivité à implanter. Puis, nous détaillons l'architecture proposée. Enfin, nous présentons les différents résultats déjà disponibles.

3.1. Les services TV interactifs

Nous avons choisi de concevoir tout d'abord un service proposé aux téléspectateurs du programme TV dédié à la télétransmission des séances de discussions à l'Assemblée Nationale. Le but est de proposer des informations additionnelles contextuelles au contenu même des discussions. Les régions interactives insérées dans le média deviennent « visibles » et « actives » uniquement lorsque l'utilisateur souhaite interagir comme décrit dans [Royer07b].

Les services interactifs implantés sur la plateforme sont soit *statiques*, soit *dynamiques*. Nous considérons comme *statiques* les services interactifs permanents, c'est-à-dire liés au média, mais pas directement à l'évolution du contenu. Des informations générales sur l'organisation de l'Assemblée Générale, un forum, des jeux associés, chat… sont des exemples de services *statiques*. Ces services sont activés ou non par l'utilisateur par le biais d'un bouton ou d'un menu permanent (accessible à tout moment). Ces services restent élémentaires car ils ne requièrent pas un niveau élevé de description du contenu et donc d'analyse sémantique du média.

Les services interactifs *dynamiques* sont liés directement à l'évolution du contenu du média. Ces services peuvent être par exemple :

- des informations complémentaires sur les personnes présentes dans la scène ;
- un résumé de l'évolution de la séance courante à l'Assemblée Nationale ;
- un résumé des séances précédentes sur des sujets similaires…

Les services interactifs *dynamiques* constituent le principal critère d'évaluation des capacités de la plateforme. En effets, la scène interactive s'adaptera dynamiquement au contenu média. Dans le cas d'informations complémentaires sur les personnes présentes dans la scène, le service interactif doit se mettre à jour en temps réel pour proposer des informations contextuelles à la scène (en fonction des personnes présentes, de leur position, etc.).

3.2. Environnement de test

Pour mettre en place une architecture de tests (Figure 77), nous nous sommes inspirés du schéma fonctionnel illustré Figure 76.

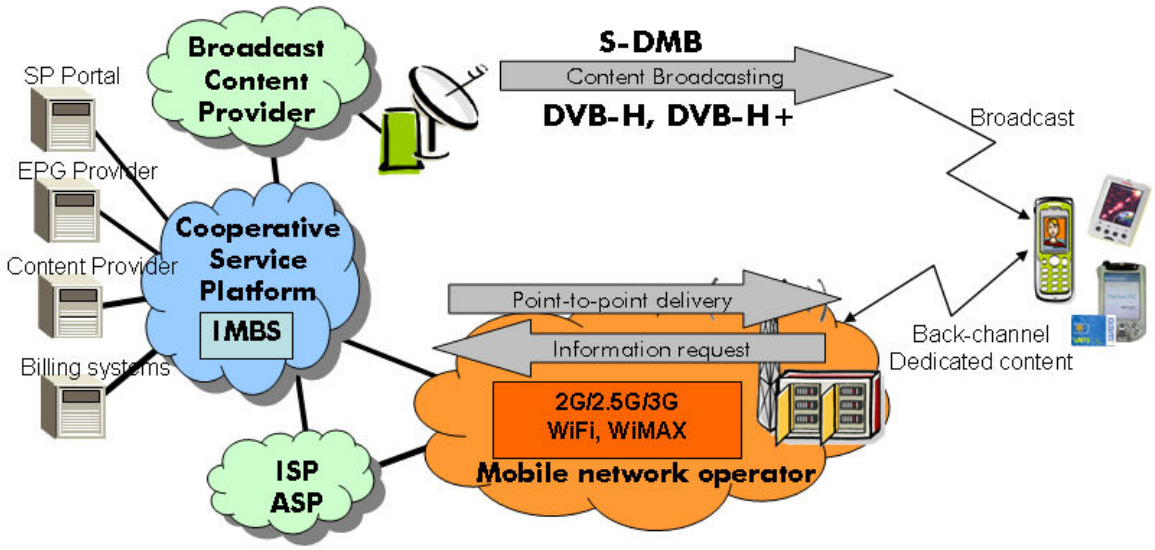


Figure 76 : Schéma d'architecture d'une chaîne de diffusion pour la télévision interactive mobile

Les flèches et les numéros représentés donnent le sens de lecture vis-à-vis de la propagation des flux et des décisions qui sont prises pour le déploiement des services interactifs.

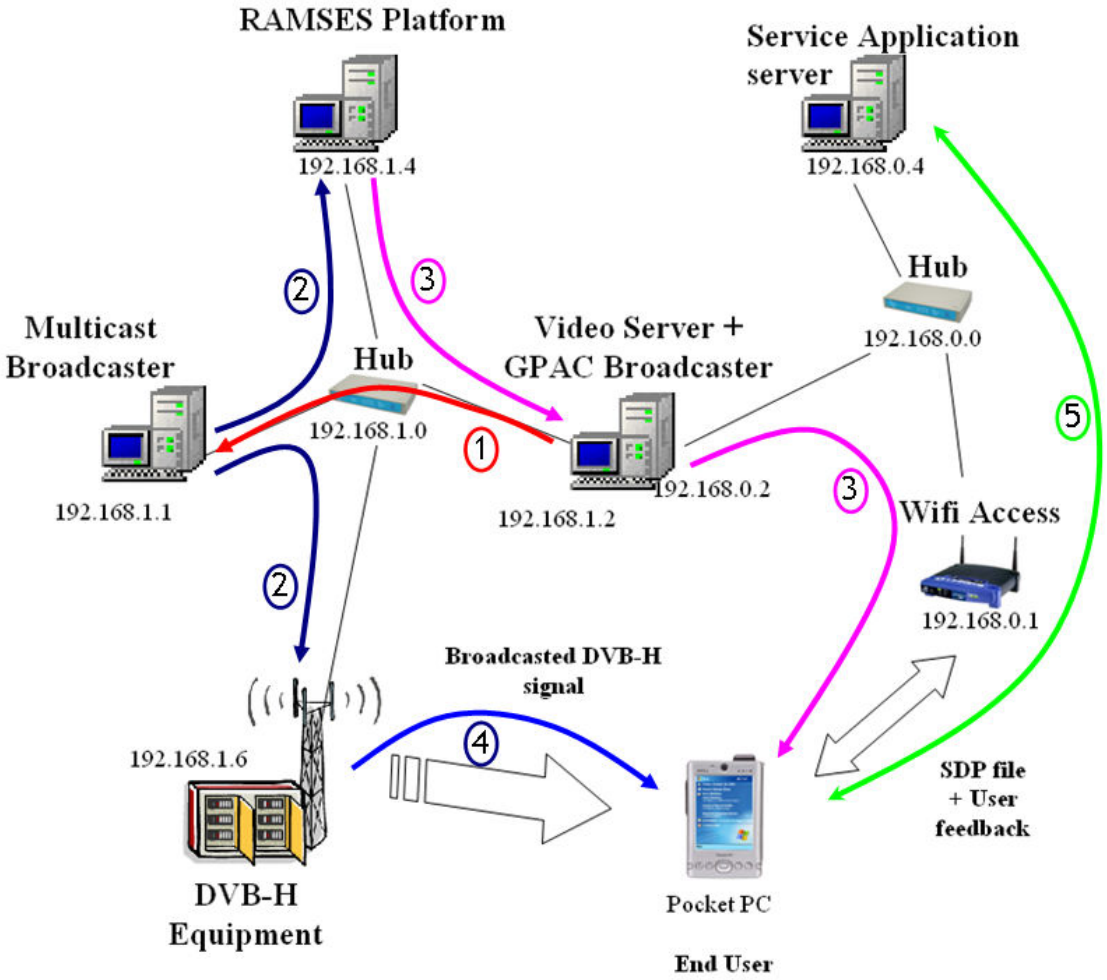


Figure 77 : Schéma d'architecture de la plateforme de diffusion

Dans le cadre d'une plateforme déconnectée du réseau Internet, nous avons implanté un outil de diffusion de flux audiovisuels (*Video Server*). La plateforme RAMSES reçoit ainsi les médias à analyser et à enrichir. L'insertion et le maintien des services interactifs déployés sont effectués par l'intermédiaire de l'outil de diffusion et d'encodage à la volée [Concolato05]. Enfin, l'utilisateur final reçoit d'une part le flux audiovisuel par l'intermédiaire d'un équipement de diffusion en [DVB-H], et d'autre part le flux d'interactivité correspondant aux services interactifs déployés à l'aide de la connexion Wifi.

3.2.1. Détails des appareils déployés

Cette section détaille les fonctions et relations entre les différentes machines représentées Figure 77 :

- *1) Darwin Streaming Server* (DSS) : permet la diffusion multicast ou unicast de documents audiovisuels à partir de plusieurs sources telles que la liste de médias vidéo contenus sur le serveur vidéo, un flux IP…

- *2) Mug - Multicast broadcaster*: Darwin Streaming Server est capable de diffuser des documents audiovisuels interactifs en mode *unicast* seulement. Nous avons ainsi inséré un programme de « réflexion » du flux diffusé [Highfield] en mode *unicast* pour le retransmettre en mode *multicast*.
- 3) La plateforme RAMSES, implantée sur un serveur, a reçu simultanément les paquets vidéo envoyés par le serveur vidéo. Les résultats de l'analyse permettent de sélectionner les services interactifs pertinents et l'envoi des commandes BIFS associées. L'encodeur à la volée de commandes BIFS permet d'envoyer les mises à jour des scènes interactives.
- 4) Un encapsulateur DVB-H (Mobileslice) connecté avec un modulateur (NN6-1161 DVB-T/H) provenant de chez [Enensys] permettant l'encapsulation du flux IP et sa diffusion vers les appareils mobiles.
- 5) La diffusion des services interactifs ainsi que les échanges entre les utilisateurs finaux et les services déployés le cas échéant se font par l'intermédiaire d'un routeur Wifi (*NETGEAR Broadband WiFi 54 Mb DG834G)*.

3.2.2. Limitations de la plateforme de test

Les limitations se situent principalement au niveau de l'implantation des spécifications du standard BIFS. En effet, l'encodeur utilisé implante une partie seulement de la norme. Certaines fonctions ne sont donc pas accessibles. Par exemple, dans le cadre de l'implantation des scènes BIFS, la sélection entre une scène jouée en différée ou une scène animée en direct est crucial car il n'est pas possible de « mixer » les deux modes. Ce cas est détaillé dans l'exemple de l'implantation du service *horloge* ci-dessous.

3.3. Service interactif horloge

Un service de type *horloge* a tout d'abord été déployé (Figure 78). Ce service est qualifié de « simple » puisqu'il n'est pas lié au contexte du média ni pour son déploiement, ni pour la mise à jour de la scène interactive. Le service est déployé de façon permanente dans le média. On observe sur les images ci-dessus le service en action en bas à gauche de la scène. Ce service est considéré comme « statique » car « autonome » une fois qu'il est déployé dans le média. Il est cependant toujours possible de le stopper via la plateforme en cas de besoin.

Cependant, en fonction des implantations du standard MPEG-4 BIFS dans l'encodeur à la volée, il est nécessaire de choisir entre le déploiement de scènes animées en direct ou animées en différé. En effet, les scènes BIFS animées en différé (pré-enregistrées) ne permettent pas de modifications en temps réel du contenu de leurs animations. Ces animations ajoutées de façon dynamique à la scène interactive en cours sont vues comme des éléments fermés qui ont leur propre « cycle de vie ». Pour ce service d'horloge, nous avons publié dans l'Annexe 7 le code exemple fourni par [Concolato05] pour la gestion de l'heure dans une scène BIFS. Nous avons ainsi déposé ce « composant interactif » sur le Service Application Server (Figure 77) afin de l'insérer de façon dynamique dans la scène interactive.

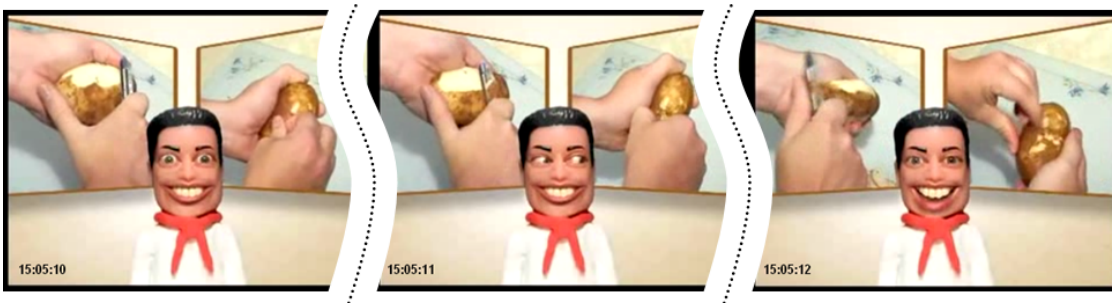


Figure 78 : Validation du service « horloge » : visible dans le coin inférieur gauche des images

Une autre solution pour ce service *horloge* consiste à déployer un service de mise à jour en temps réel de l'heure via des commande BIFS Update (on envoie la nouvelle heure toutes les secondes pour rafraichir l'affichage). Un second service simple a été implanté, celui-ci tient compte du contexte du média pour autoriser le déploiement.

3.4. Insertion conditionnelle d'un service statique

Dans ce cas, le service présente une scène plus « riche ». Il s'agit ici de l'insertion d'un jeu de puzzle (Figure 79), extrait des « goodies » proposés sur le site Internet en relation avec le programme télévisé [TAC]. Ce service permet dès lors de valider l'insertion conditionnelle de contenus additionnels.

Figure 79 : Validation d'un service « statique » : un bouton interactif Game permet d'accéder au jeu « puzzle »

La condition d'implantation du service interactif est exprimée par :

- "MediaNode/VideoNode/SegmentNode/FaceNode/FaceNameNode"

La photo de « TAC_Oncle_Tom » (cf. Figure 79) est fournie à la plateforme. De plus, la scène MPEG-4 BIFS associée au service interactif vérifie la valeur du descripteur du nom de la personne identifiée par l'analyseur de reconnaissance de visage et la compare à la chaîne de caractère « TAC_Oncle_Tom » pour décider de l'insertion du bouton interactif et du jeu interactif associé.

Une fois encore, le jeu est disponible en annexe pour la version « différée » permettant le fonctionnement d'un compteur (temps écoulé) et l'insertion d'une fonction (non implantée) vérifiant à chaque tour l'ordre des pièces pour savoir si le joueur à fini le puzzle.

Nous avons par la suite choisi de développer des services permettant un fonctionnement plus démonstratif de la plateforme aussi bien en temps réel qu'en différé en se basant sur l'ajout contextuel d'informations additionnelles.

3.5. Insertion d'un service interactif dynamique

Les séances des travaux de l'Assemblée nationale, tout comme les évènements politiques en général deviennent de plus en plus populaires en France depuis 2002 [Royer07a]. Dès lors, nous proposons d'enrichir de tels contenus (Figure 80).

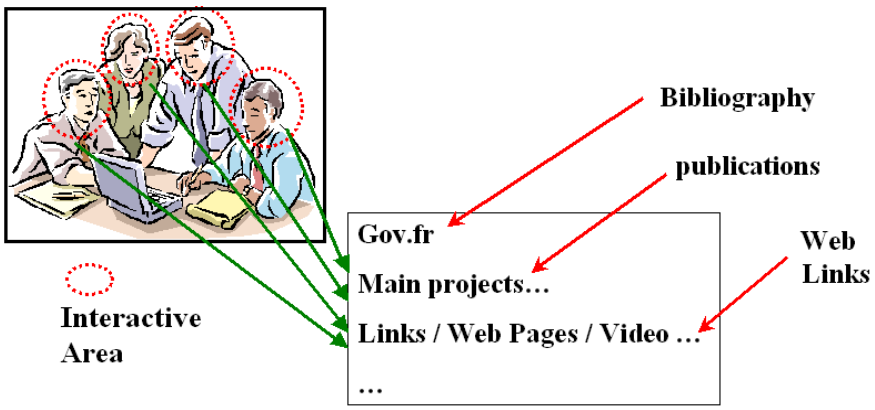


Figure 80 : Exemple de service interactif proposant des informations additionnelles contextuelles

Cet exemple permet une mise en valeur des capacités de la plateforme en termes de fonctionnement en temps réel sans mettre en défaut des analyseurs qui n'ont pas été optimisés. En effet, les séances de l'Assemblée nationale se passent toujours en intérieur au même endroit avec des mouvements de caméra limités.

Enfin, les séances sont diffusées et « enrichies » en « direct ». La Figure 81 illustre le délai de cinq minutes entre ce qui ce passe dans l'hémicycle et la diffusion réelle dans le programme de La Chaîne Parlementaire (LCP). Ce délai est nécessaire pour l'insertion manuelle d'informations additionnelles avant la diffusion sur les chaînes.

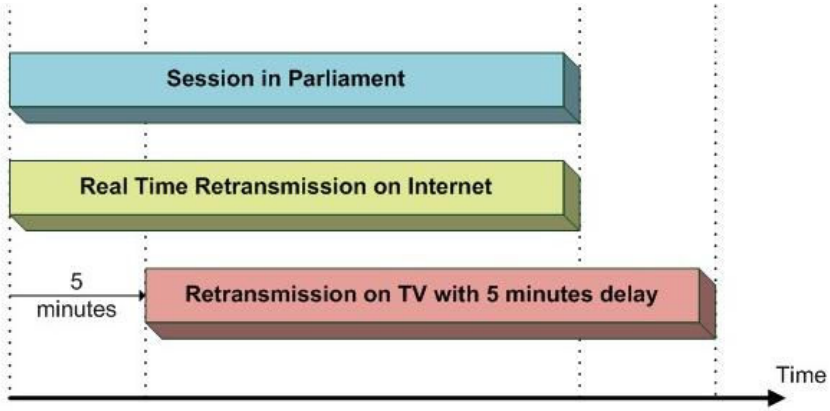


Figure 81 : Illustration des délais de retransmission après insertion manuelle d'informations additionnelles

Cet exemple permet ainsi une comparaison directe entre les résultats d'analyse manuelle et automatique du contenu. En raison du nombre limité d'analyseurs implantés sur la plateforme, il n'y aura dans un premier temps que la vidéo qui sera analysée pour extraire des informations du contenu.

3.5.1. Implantation des services interactifs dynamiques

La Figure 82 propose un aperçu de l'ensemble des services interactifs qu'il est possible de déployer dans ce type de contenu.

Figure 82 : Illustration des services interactifs possible liés à l'insertion d'informations additionnelles

On retrouve principalement dans cette figure des accès à des sources d'information sur le contenu lui-même au centre ou en général (flux d'information RSS), des zones pour régir tels que le forum en ligne de La Chaîne Parlementaire, des menus…

3.5.2. Insertion Inconditionnelle d'un service dynamique simple

Dans un premier temps, nous avons considéré un service simple (Figure 83) permettant l'affichage dynamique dans la scène de l'évolution du nombre de séquences cumulées dans le média.

Figure 83 : Validation d'un service « dynamique » : le numéro de séquence s'incrémente en fonction des détections de changement de scène

Il s'agit ici d'un service simple montrant l'utilisation d'un analyseur (détection des changements de scène) pour permettre la mise à jour de la scène interactive, constituée d'un compteur affiché en bas à droite de la scène (cf. Figure 83). La scène initiale est une scène interactive « vide » présentée dans l'Annexe 4. L'envoi des mises à jour de la scène interactive (*BIFS updates*) se fait ici en fonction des résultats d'analyse d'un analyseur.

3.5.3. Insertion d'un service dynamique complet

Nous avons développé un dernier service d'informations additionnelles sur les travaux de l'Assemblée nationale [Royer07a]. Ce service est le plus évolué et reprend toutes les fonctionnalités décrites précédemment dans un service d'insertion automatique d'informations additionnelles (Figure 84).

Figure 84 : Validation d'un service « dynamique » d'informations additionnelles : ajout de zone interactive sur les personnes identifiées

Ce service met en avant la validation de la combinaison d'analyseurs. Les conditions d'implantation exprimées sont :

- "MediaNode/VideoNode/SegmentNode/FaceNode/FaceNameNode",
- "MediaNode/VideoNode/SegmentNode/FaceNode/RegionNode/MovingRegionNode"

On notera que le cercle rouge dans ce dernier exemple a été rendu visible pour la démonstration, mais représente une zone active (sur click par exemple) non visible pour le confort visuel de l'utilisateur.

3.6. *Conclusion*

Pour une mise en perspective de ces résultats, rappelons qu'actuellement la mise en place de services interactifs statiques sur des programmes de TV nécessite de deux semaines à un mois afin de disposer des *templates* et de les instancier sur les émissions. Si la phase d'écriture des *templates* est capitalisable, il reste néanmoins toute une phase de saisie qui représente plus de la moitié de ce temps de déploiement qui nécessite opérationnellement des saisies durant l'émission. A l'aide de notre plateforme, lorsqu'elle dispose des analyseurs adaptés, la mise en place d'un service dynamique nécessite une demi-journée, hors conception du *Template*. Le facteur de gain en temps varie de 10 à 20.

Au travers de ces quatre services interactifs, de complexité croissante, nous avons démontré le fonctionnement effectif de la plateforme RAMSES. Les services interactifs ont bien apporté l'ensemble des informations nécessaires pour être pris en compte par la plateforme. Enfin, le fonctionnement de la combinaison des analyseurs pour le déploiement et le maintien des services a été validé.

3.7. Discussion

Si les exemples de services interactifs présentés précédemment démontrent d'une part le caractère opérationnel de la plateforme RAMSES et d'autre part la pertinence de nos contributions en termes de modélisation/ modularité / adaptabilité / temps réel…, il n'en reste pas moins qu'un certain nombre de limites existe comme nous nous proposons d'en discuter dans la suite.

3.7.1. Modélisation d'un service interactif

Seule la première implantation fondée sur des chemins de description MPEG-7 a été validée. Il reste donc à valider le fonctionnement pour des requêtes en langage naturel contraint, du type :

- Pour le service « statique » :

  « TAC_Oncle_Tom est détecté dans la scène »

  + « TAC_Oncle_Tom est une personne ».

- Pour le service « dynamique » :

  « Une personne est détectée dans la scène »

  + « la personne est reconnue »

  + « La personne est suivie ».

Comme précédemment indiqué, un service interactif est défini par les requêtes, le *template* de la scène interactive à insérer dans le média ainsi que par les ressources nécessaires le cas échéant (base de données de personnes à identifier…). Or, nous avons réparti manuellement l'ensemble de ces informations dans les différentes composantes de la plateforme : une base de photos de personnes connues, la liste des descripteurs à identifier, la génération des commandes MPEG-4 BIFS…

Il est nécessaire de faire évoluer l'implantation de la plateforme en termes de « réception » et de « traitement » des informations d'un service interactif dans le but de faciliter l'insertion du service interactif. Une fonctionnalité doit être introduite pour que le fournisseur de services puisse ajouter un service interactif sur la plateforme à l'aide d'un seul « package ».

3.7.2. Modélisation de l'analyse multimédia

Cette implantation reporte le problème du gap sémantique sur les analyseurs média. Celui-ci reste donc à résoudre afin d'augmenter les possibilités d'analyses des documents multimédias.

L'implantation de l'accès des analyseurs média au moteur d'inférence de la plateforme est une des solutions à mettre en œuvre afin de permettre aux analyseurs de raisonner sur les résultats d'analyse précédents obtenu et peut être ainsi pouvoir déduire des informations de plus haut niveau. La solution consiste à avoir une plateforme capable d'intégrer un très grand nombre d'analyseurs complémentaires pour identifier de façon la plus fiable possible des concepts subjectifs.

3.7.3. Modélisation de la description du document multimédia

L'implantation du modèle de description s'est limitée à celle d'une sous- partie du modèle de description de MPEG-7.

Pour généraliser le modèle de description, les fonctionnalités suivantes sont nécessaires : génération automatique d'un récapitulatif des capacités de description totale de la plateforme en fonction des analyseurs présents. Cela permettrait ainsi l'implantation de fonctions pour :

- l'extension automatique de ce modèle dès l'ajout de nouvelles capacités de description ;
- la vérification de la cohérence de la capacité de description de la plateforme à travers la suppression des doublons, la vérification de chemin de description unique…

3.7.4. Modélisation du contexte virtuel

L'exécution des analyseurs média est sérialisée dans la plateforme actuelle. Les fonctions à implanter sont les mécanismes liés à la programmation pour les systèmes « temps réel ». En effet, il est nécessaire de pouvoir gérer l'accès concurrentiel au contexte virtuel afin de permettre la parallélisation des analyseurs média.

Par ailleurs, des fonctions d'optimisation et de priorités doivent être ajoutées pour définir une hiérarchie dans l'ordonnancement des analyseurs média. Par exemple, une priorité peut être donnée en fonction de l'état d'avancement de l'analyse en cours sur des concepts permettant le déclenchement d'un ou plusieurs services interactifs plutôt que privilégier l'analyse d'informations générales à tous les services mais ne permettant aucun déclenchement direct.

3.7.5. Validation des analyseurs

Les modèles de description qu'ils soient bas ou haut niveau sont pour le cas de MPEG-7 et des ontologies prévus à l'origine pour être étendus de façon dynamique.

Cependant, cette capacité d'extension théorique n'est pas forcément vérifiée facilement en pratique. Cela remet dès lors en cause la capacité de la plateforme à intégrer des analyseurs dont les descripteurs en entrée / sortie ne sont pas déjà connus de la plateforme… Il y a donc un effort particulier à apporter sur les fonctions de vérification et d'intégration au modèle de description le cas échéant.

3.7.6. Génération de la composition des analyseurs

Les travaux futurs concernent la possibilité de modifier de façon dynamique la sélection des analyseurs média en fonction du contexte du média durant l'analyse de celui-ci.

Cependant, la composition automatique est aujourd'hui limitée à la composition « linéaire », séquentielle des analyseurs, sans tenir compte des boucles ou branchements non-séquentiels issues de la sélection des analyseurs.

De même, elle ne tient pas compte du cas plus général où la phase sélection elle-même ferait partie intégrante de la phase d'orchestration. Ceci pourrait permettre

d'optimiser la sélection des analyseurs en fonction de critère provenant de paramètres résultant des analyses elles-mêmes.

D'une façon générale, la plateforme RAMSES a apporté la preuve du concept de la possibilité de mettre en place des services interactifs sur des contenus vidéo de type TV en levant, au moins partiellement, les verrous de modularité, d'adaptabilité, d'extensibilité et de dynamicité.

3.8. Conclusion

Nous avons, à travers ces résultats, démontré le fonctionnement d'un ou plusieurs services interactifs simultanément mettant globalement en œuvre la plateforme.

D'après l'ensemble des observations effectuées, Il ressort que la plateforme RAMSES permet une réduction de la complexité de l'ensemble des domaines introduits dans ce document grâce à la possibilité de réunir les expertises de ces domaines à travers une implantation ouverte. Ainsi, chaque acteur de la plateforme peut apporter son expertise et profiter des avancées effectuées dans les autres domaines. Un fournisseur d'analyseurs média par exemple, peut profiter de l'expertise d'un fournisseur d'accesseurs média, tandis qu'un fournisseur d'analyseurs haut niveau peut profiter de l'expertise de fournisseurs d'analyseurs de bas niveau...

Enfin, la présence d'un système ouvert sur les ontologies permet la collaboration pour la définition d'un « standard » de description commun dans un premier temps, puis facilite à chacun des acteurs dans un second temps l'accès aux autres domaines. Un fournisseur de service par exemple n'a ainsi pas besoin de connaître le standard de description MPEG-7 pour spécifier ses besoins et profiter de la plateforme.

Chapitre 4

Conclusion et perspectives

4.1. Conclusion générale

La plateforme RAMSES développée et opérationnelle permet l'insertion automatique des services interactifs de façon dynamique et adaptative en fonction des cas d'application. L'implantation sur un modèle d'architecture modulaire assure ainsi l'adaptation et facilite l'accès tant aux applications qu'aux évolutions des outils, normes ou formats à tous les niveaux de la plateforme.

Les enjeux de l'insertion de services dans les documents multimédias sont multiples. Cela ouvre en effet des possibilités telles que relier des médias entre eux, les relier à de la connaissance, à des relations, à des émotions… afin de les conserver, les retrouver, ou de les partager. La possibilité de le faire de façon automatique permet de répondre à l'explosion de la diffusion des médias audiovisuels aussi bien sur Internet qu'a travers les bouquets de chaînes télévisées dans le monde. Enfin, l'extraction d'informations complémentaires à partir de sources multiples augmente les capacités d'analyse, et donc, les capacités décisionnelles du système.

Nous avons pu identifier un grand nombre de verrous principalement technologiques durant les recherches effectuées. Dans un premier temps, l'absence des propriétés conjointes de modularité, ouverture et complétude dans les systèmes actuels conduisent aujourd'hui à « réinventer la roue » pour chaque nouvelle implantation avec des spécifications propres ou pour le moins empêchent la réutilisation et la combinaison des avancées technologiques existantes. Enfin, les limites actuelles dans l'analyse des contenus des documents multimédias, en termes d'extraction d'informations et d'outils de description, restreignent les possibilités dans la chaîne décisionnelle d'insertion de services interactifs. Il est dès lors impossible d'implanter la chaîne complète d'automatisation de la génération de *Rich Media* depuis l'accès aux médias jusqu'à leur distribution vers l'utilisateur final.

Nous avons apporté la preuve des solutions avancées en implantant des modules opérationnels. Dans un premier temps dans le cadre d'une application interactive dite statique, nous avons démontré la validité de l'implantation des modèles de spécification des besoins par les fournisseurs de services. Un deuxième service interactif a permis de valider l'adaptabilité de la plateforme aux nouvelles spécifications et de vérifier dans le même temps la capacité de la plateforme à maintenir le fonctionnement du service interactif déployé de façon dynamique à travers l'envoi de notifications. Enfin, le déploiement simultané de ces deux applications a permis la validation du fonctionnement global de la chaîne complète depuis l'accès au média jusqu'à l'enrichissement de celui-ci en passant par la chaîne décisionnelle.

4.2. Perspectives

Les caractéristiques principales de la plateforme ont été développées dans un objectif de pérennité à travers l'évolutivité et l'adaptabilité vis-à-vis des innovations en cours dans les différents domaines traités tels que l'analyse de flux et la prise de décisions conditionnelles en temps réel. RAMSES offre donc une plateforme générique implantant des outils de base dans un premier temps, mais favorisant l'accès aux expertises dans chacun de ces domaines.

Si les initiatives de télévision interactive sont de plus en plus nombreuses, le problème du passage à l'échelle n'est pas encore maîtrisé. RAMSES y apporte un début de solution.

Citons également les services à base de communication vidéo interpersonnelle qui offrent un autre contexte d'application des résultats de cette thèse.

Dans ces deux cas d'application, l'intérêt d'Alcatel-Lucent est manifeste, de part son portefeuille de produits incluant des solutions de communication voix, vidéo, des solutions de conférence ainsi que des solutions et des applications dans le domaine de l'IPTV et de la télévision mobile.

Cela conduit à des innovations potentiellement valorisables à terme par les unités d'affaire du Groupe Alcatel-Lucent.

Bibliographie

[Ahmed97]
M. Ahmed, S. Yamany, E. Hemayed, S. Ahmed, S. Roberts, A. Farag, 3D reconstruction of the human jaw from a sequence of images, in Proceedings of Computer Vision and Pattern Recognition 1997, IEEE Computer Society Conference on, p.646-653, San Juan, Puerto Rico, June 1997.

[Ahn08]
L. V. Ahn, L. Dabbish, Designing Games With a Purpose, Communications of the ACM, Vol. 51, No. 8, p. 58-67, august 2008, issn 0001-0782.

[AhnVideo]
L. v. Ahn, Human Computation video, http://video.google.com/videoplay?docid=-8246463980976635143, Google TechTalks, length: 51min31s, July 26, 2006, retrieved on May 2009.

[Alatan98]
A. A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, T. Sikora, Image Sequence Analysis for Emerging Interactive Multimedia Services-The European COST 211 Framework, IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, No. 7, November 1998.

[Albiol05]
A. Albiol, L. Torres, E.J. Delp, Fully automatic face recognition system using a combined audio-visual approach, IEE Proc.-Vis. Image Signal Process., vol. 152, No. 3, June 2005.

[Amintoosi07]
M. Amintoosi, F. Farbiz, M. Fathy, M. Analoui, N. Mozayani, QR Decomposition-Based Algorithm for Background Substraction, in Processing of International Conference on Acoustics, Speech and Signal 2007 (ICASSP 2007), Vol.1, p.1093-1096, April 2007, isbn 1-4244-0727-3.

[Antoniou04]
G. Antoniou, F. Van Harmelen, A semantic Web Primer, MIT Press, April 2004, ISBN 0-262-01210-3.

[Antoniou05]
G. Antoniou, C. V. Damasio, B. Grosof, I. Horrocks, M. Kifer, J. Maluszynski, P. F. Patel-Schneider, Combining Rules and Ontologies. A survey, Reasoning on the Web with Rules and Semantics (REWERSE) Project deliverable, Copyright © REWERSE 2005, No. I3-D3, March 2005.

[Arlen81]
M. J. Arlen, Thirty Seconds, publisher: Farrar Straus & Giroux, New York, 1980.

[Arndt07]
R. Arndt, R. Troncy, S. Staab, L. Hardman, Adding Formal Semantics to MPEG-7: Designing a Well-Founded Multimedia Ontology for the Web, Department of Computer Science, University of Koblenz. Technical Report. January 2007.

[Arsenault08]
A. H. Arsenault, M. Castells, The Structure and Dynamics of Global Multi-Media Business Networks, International Journal of Communication (IJoC), Vol. 2, p. 707-748, July 2008.

[Auer07]
S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, DBpedia: A Nucleus for a Web of Open Data, 6th International Semantic Web Conference (ISWC 2007), Busan, Korea, November 2007.

[Ayache06]
S. Ayache, G. Quénot, J. Gensel, S. Satoh, Using topic concepts for semantic video shots classification, in Proceedings of the 5th international conference on Image and Video Retrieval (CIVR 2006), Vol. 4071, p. 300-309, Tempe, AZ, USA, July 13-15, 2006, isbn 3-540-36018-2.

[Ayache07]
S. Ayache, G. Quénot, J. Gensel, Image and video indexing using networks of operators, Image and Video Processing, Vol. 2007, No. 3, p.1-13, November 2007, issn 1687-5176.

[Bara05]
T. Bara, E. Papaioannou, N. Ioannidis, MELISA Multiplatform E-Publishing for Leisure and Interactive Sports Advertising, Melisa Workshop, Athens, Greece, November 2005.

[Bechhofer04]
S. Bechhofer, R. Volz, WonderWeb OWL Ontology Validator, Internet web site, http://www.mygrid.org.uk/OWL/Validator, retrieved on July 2009.

[Beek06]
M. ter Beek, A. Bucchiarone, S. Gnesi, A Survey on Service Composition Approaches: From Industrial Standards to Formal Methods, Consigno Nazionale delle Ricerche, Technical report, Istituto di Scienza e Technologie dell'Informazione "Alessandro Faedo", May 2006.

[Bennett02]
P. N. Bennett, S. T. Dumais, E. Horvitz, Probabilistic combination of text classifiers using reliability indicators: Models and results, in Proceedings of the 25th annual international ACM conference on Research and development in information retrieval (SIGIR), p. 207-214, Tampere, Finland, 2002.

[Bickmore04]
T. Bickmore, J. Cassell, Social Dialogue with Embodied Conversational Agents. In J. V. Kuppevelt, L. Dybkjaer, N. Bernsen, Natural, Intelligent and Effective Interaction with Multimodal Dialogue Systems, New York: Kluwer Academic, 2004.

[Boni04]
A. Boni, E. Launay, T. Mienville, P. Stuckmann, Multimedia broadcast multicast service - technology overview and service aspects, Fifth IEE International Conference on 3G Mobile Communication Technologies 2004 (3G '04), p. 634-638, 2004.

[BPEL4People_Wiki]
BPEL4People web page on Wikipedia, Wikipedia®, Internet web site, http://en.wikipedia.org/wiki/BPEL4People, retrieved on August 2009.

[BPEL4People07]
A. Agrawal, M. Amend, M. Das, M. Ford, C. Keller, M. Kloppmann, D. König, F. Leymann, R. Müller, K. Plösser, R. Rangaswamy, A. Rickayzen, M. Rowley, P. Schmidt, I. Trickovic, A. Yiu, M. Zeller, WS-BPEL Extension for People (BPEL4People), Version 1.0, June 2007, Copyright © 2007 Active Endpoints Inc., Adobe Systems Inc., BEA Systems Inc., International Business Machines Corporation, Oracle Inc., and SAP AG.

[Bosca06]
A. Bosca, F. Corno, G. Valetto, R. Maglione, On-the-fly Construction of Web Services Compositions from Natural Language Requests, Journal of Software, p.40-50, Vol. 1(1), ISSN: 1796-217X, 2006.

[Bouilhaguet]
F. Bouilhaguet, J.C. Dufourd, S. Boughoufalah, C. Havet, "Interactive broadcast digital television. The OpenTV platform versus the MPEG-4 standard framework," , In Proceedings of The IEEE International Symposium on Circuits and Systems 2000 (ISCAS'00), Geneva, vol.3, p.626-629, 2000.

[Bradski08]
G. R. Bradski, A. Kaehler, Learning OpenCV, Computer Vision with the OpenCV Library, published by O'reilly, p.575, September 2008.

[Bray98]
T. Bray, RDF and Metadata, xml.com, June 09, 1998, Internet web page, http://www.xml.com/pub/a/98/06/rdf.html, retrieved on August 2009.

[Brelot03]
M. Brelot, Représentation orientée-objets de scènes visuelles pour la composition flexible, Ph.D. Thesis, Institut National Polytechnique de Grenoble, May 1999.

[Brickley07]
D. Brickley, L. Miller, FOAF Vocabulary Specification 0.91, Namespace Document 2 November 2007 - OpenID Edition, Internet web page, http://xmlns.com/foaf/spec/, retrieved on August 2009.

[Burke]
D. Burke, Become an Early Rejector!, A Guide to Interactive TV, Copyright © 2000 White Dot, Internet web page, http://www.whitedot.org/issue/iss_story.asp?slug=shortSpyTV, retrieved on July 2009.

[Burnard01]
L. Burnard, C. M. Sperberg-McQueen, Guidelines for Text Encoding and Interchange (P4). http://www.tei-c.org/P4X/index.html, The TEI Consortium, 2001.

[Burred06]
J. J. Burred, A. Röbel, X. Rodet, An Accurate Timbre Model for Musical Instruments and its Application to Classification, First Workshop on Learning the Semantics of Audio Signals, Athens, December 2006.

[Cassell01]
J. Cassell, Embodied Conversational Agents: Representation and Intelligence in User Interfaces, AI Magazine, Vol. 22 No. 4, p.67-84, 2001.

[Ceefax]
Ceefax web site, http://www.ceefax.tv/, ©2008 ceefax.tv, registered trademark of the BBC Corporation, retrieved on July 2009.

[Chellappa95]
R. Chellappa, C. L. Wilson, S. Sirohey, Human and Machine Recognition of Faces: a Survey, Proceedings of the IEEE, Volume 83, No. 5, p.705-740, May 1995.

[Cheng07]
O. Cheng, J. Dines, M. M. Doss, A Generalized Dynamic Composition Algorithm of Weighted Finite State Transducers for Large Vocabulary Speech Recognition, in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing 2007 (ICASSP 2007), Vol. 4, p. 345-348, Honolulu, Hawaii, USA, April15-20, 2007.

[Chiariglione]
MPEG Chiariglione Homepage, Internet web site, http://www.chiariglione.org/mpeg/, retrieved on July 2009.

[Chien02]
S-Y. Chien, S-Y Ma, L-G Chen, Efficient Moving Object Segmentation Algorithm Using Background Registration Technique, IEEE Transaction on Circuits And System for Video Technology, Vol.12, No.7, July 2002.

[CNIL]
French independent administrative authority for data protection and the liberties, www.cnil.fr

[Concolato02]
C. Concolato, J. C. Dufourd, Comparison of MPEG-4 BIFS and some other multimedia description languages, Workshop and Exhibition on MPEG-4, WEPM 2002, San Jose, California, USA, 2002.

[Concolato05]
C. Concolato, J. Le Feuvre, MPEG-4 BIFS and XMT Tutorial, Internet Web page, http://gpac.sourceforge.net/tutorial/bifs_intro.htm, retrieved on June 2009.

[Concolato07]
C. Concolato, Descriptions de scènes multimédia : représentations et optimisations, Ph.D. thesis Informatique et Réseaux, Département Traitement du Signal et Images, ENST, p.227, 2007.

[Correira98]
P. Correira, F. Pereira, The Role of Analysis in Content-Based, Video Coding and Indexing, Signal Processing, vol. 66, no. 2, p.125-142, 1998.

[Cuce04]
H. I. Cuce, A. E. Cetin, Mean-shift tracking of moving objects using multidimensional histograms, Signal and Data Processing of Small Targets 2004. Edited by Drummond, Oliver E. Proceedings of the SPIE, vol. 5428, p. 70-77, 2004, doi. 10.1117/12.553388.

[Cuenca-Grau06]
B. Cuenca-Grau, I. Horrocks, O. Kutz, U. Sattler, Will my ontologies fit together?, in Proceedings of the 2006 International Workshop on Description Logics (DL2006), vol. 189, Windermere, Lake District, UK, May 30 - June 1, 2006.

[Dang08]
E. Dang, France Télévisions innove avec un service de télévision interactive sur ADSL lancée à l'occasion des Elections Municipales, Communiqué de presse, France Television Interactive, March 14, 2008.

[DarwinStreamS]
Darwin Srtreaming Server, Copyright © 2009 Apple Inc., Internet web site, http://developer.apple.com/opensource/server/streaming/index.html, retrieved on July 2009.

[Delezoide06]
B. Delezoide, Modèles d'indexation multimédia pour la description automatique de films de cinéma, Ph.D. Thesis, Université Pierre et Marie Curie, Paris, France, April 2006.

[Dimitrova03]
N. Dimitrova , Multimedia Content Analysis: The Next Wave, in Proceedings of Image and Video Retrieval: Second International Conference, CIVR 2003, Urbana-Champaign, IL, USA, Vol. 2728, p.415-420, ISBN 978-3-540-40634-1, 2003.

[Dong07]
H. Dong, F.K. Hussain, E. Chang, Application of Protégé and SPARQL in the Field of Project Knowledge Management, Second International Conference on Systems and Networks Communications 2007 (ICSNC'07), p.74-74, 25-31, Aug. 2007.

[DublinCoreWiki]
Dublin core web page on Wikipedia, Wikipedia®, Internet web site, http://en.wikipedia.org/wiki/Dublin_core, retrieved on July 2009.

[Dufourd05]
J-C Dufourd and Olivier Avaro and Cyril Concolato, An MPEG Standard for Rich Media Services, IEEE MultiMedia, Vol. 12, No. 4, p.60-68, IEEE Computer Society, Los Alamitos, CA, USA, October-December 2005.

[Dufresne96]
R. J. Dufresne, W. J.Gerace, W. J.Leonard, J. P.Mestre, L. Wenk, Classtalk: A Classroom Communication System for Active Learning, paper Published in Journal of Computing in Higher Education, Vol 7, p.3-47, 1996.

[Dung95]
P. M. Dung, On the acceptability of arguments and its fundamental role in non monotonic reasoning, logic programming and n-person games, Artificial Intelligence, Vol. 77, issue. 2, p. 321–357, September 1995.

[DVB-H]
Global Mobile TV DVB-H, http://www.dvb-h.org/, retrieved on July 2009.

[Eclipse]
Eclipse Site, http://www.eclipse.org/, retrieved on May 2009.

[Eliens02]
A. Eliëns, Z. Huang, C. Visser, A platform for embodied conversational agents based on distributed logic programming. In Proceedings of AAMAS 2002 WORKSHOP: Embodied conversational agents - let's specify and evaluate them, 2002.

[Enensys]
Enensys, Internet web site, http://www.enensys.com/, retrieved on July 2009.

[Equinox]
Equinox Eclipse Project, OSGi R4 core framework specification, Internet web site, http://www.eclipse.org/equinox/, retrieved on May 2009.

[Euro08]
Euro 2008, Communiqué de presse, 19 juin 2008, Levallois, Mediametrie-eStat

[Euzenat03]
J. Euzenat, N. Layaïda, V. Dias, A semantic framework for multimedia document adaptation, In Proceedings of the 18th International Joint Conference on Artificial Intelligence IJCAI'2003, pages 31-36, Morgan Kauffman, August 2003.

[Eveno03]
N. Eveno, A. Caplier, and P-Y Coulon, Jumping Snakes and Parametric Model for Lip Segmentation, International Conference on Image Processing (ICIP'03), Barcelona, Spain, September 2003.

[Facebook]
Facebook web site, www.facebook.com, Facebook © 2009, retrieved on July 2009.

[Fan04]
J. Fan, N. Dimitrova, V. Philomin, Online face recognition system for videos based on modified probabilistic neural networks, International Conference on Image Processing, ICIP '04, p.2019-2022, Vol. 3, 24-27 Oct. 2004.

[Feldman97]
T. Feldman, An Introduction to Digital Media, Routledege, London, 1997.

[Ferman02]
A. Müfit Ferman, A. Murat Tekalp, R. Mehrotra, Robust Color Histogram Descriptors for Video Segment Retrieval and Identification, IEEE Transaction on Image Processing, Vol. 11, No.5, May 2002.

[FFMPEG]
FFmpeg homepage, Internet web site, http://ffmpeg.org/, retrieved on May 2009

[Flickr]
Flickr, Copyright © 2009 Yahoo!, Internet web site, http://www.flickr.com/, retrieved on July 2009.

[Franceschini07]
L. Franceschini, La Vidéo à la Demande en Europe, © Direction du développement des médias et Observatoire européen de l'audiovisuel (DDM), mai 2007, Internet web site, http://www.ddm.gouv.fr, retrieved on July 2009.

[Freebox]
Free homepage, Internet web site, http://freebox.free.fr/, © Iliad 2009, retrieved on May 2009.

[Freese07]
E. Freese, Enhancing AIML Bots using Semantic Web Technologies, Extreme Markup Languages 2007, Montréal, Québec, August 7-10, 2007.

[Ghilardi06]
S. Ghilardi, C. Lutz, F.Wolter, Did I damage my ontology? A case for conservative extensions in description logic, in Proceedings of 20th International Conference on Principles of Knowledge Representation and Reasoning, KR2006, p. 187–197, Lake District, UK, June 2-5, 2006.

[Gonzales08]
P. Gonzales, TMC et W9, les deux locomotives de la TNT, september 2008, article in Figaro newspaper. http://www.lefigaro.fr/

[GoogleLabel]
Google Image Labeler, http://images.google.com/imagelabeler/, retrieved on May 2009.

[Grammling06]
M. Grammling, OSGi web page on Wikipedia, Wikipedia®, Internet web site, http://en.wikipedia.org/wiki/OSGi, 2006, retrieved on July 2009.

[Greminger08]
M. A. Greminger, B. J. Nelson, A Deformable Object Tracking Algorithm Based on the Boundary Element Method that is Robust to Occlusions and Spurious Edges, in Proceedings of International Journal of Computer Vision 2008 (IJCV'08) p. 29-45, Vol. 78, No. 1, June 2008, issn 0920-5691.

[Gu02]
L. Gu, Video analysis in MPEG compressed domain, Ph.D. Thesis, University of Western Australia, September 2002.

[Gwap]
http://www.espgame.org/gwap/, Internet web site, retrieved on May 2009.

[Hammal06]
Z. Hammal, Facial Features Segmentation, Analysis and Recognition of Facial Expressions using the Transferable Belief Model. Ph.D. thesis, Laboratoire des Images et des Signaux dans le cadre de l'École Doctorale "Ingénierie pour la Santé, la Cognition et l'Environnement", France, Juin 2006.

[Hashimi06]
S. Al Hashimi, G. Davies, Vocal telekinesis: physical control of inanimate objects with minimal paralinguistic voice input, Proceedings of the 14th annual ACM international conference on Multimedia, Santa Barbara, CA, USA, session 2, p.813-814, 2006.

[Hausenblas07]
M. Hausenblas, F. Nack, Interactivity = Reflective Expressiveness, Multimedia, IEEE, p.1-7, Volume 14, Issue 2, April-June 2007.

[Herpel00]
C. Herpel and A. Eleftheriadis, Tutorial issue on MPEG-4, Signal Processing: Image Communication, vol. 15, nos. 4-5, 2000.

[Herrmann07]
M. Herrmann, M. A. Aslam, O. Dalferth, Applying Semantics (WSDL, WSDL-S, OWL) in Service Oriented Architectures (SOA), in Proceedings of the 10th International Protégé Conference, Budapest, Hungary, July 15-18, 2007.

[Highfield]
J. Highfield, UDP Packet Reflector, Internet web site, http://spirit.lboro.ac.uk/mug/mug.html, retrieved on July 2009.

[Hobbs85]
J.R. Hobbs, Granularity, In Proceedings of IJCAI'85, p. 432–435, San Francisco, 1985. Morgan Kaufmann.

[Hoogs03]
A. Hoogs, J. Rittscher, G. Stein, J. Schmiederer, Video Content Annotation Using Visual Analysis and a Large Semantic Knowledgebase, Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03), vol. 2, p. 327-334, June 2003.

[Horridge04]
M. Horridge, H. Knublauch, A. Rector, R. Stevens, C. Wroe, A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools - Edition 1.0, The University Of Manchester, August 27, 2004.

[Howard08]
B. Howard, Analyzing online social networks, Communications of the ACM, vol. 51, issue 11, p. 14-16, November 2008, ISSN:0001-0782

[HTML5]
I. Hickson, D. Hyatt, HTML 5, A vocabulary and associated APIs for HTML and XHTML, W3C Working Draft 23 April 2009, Internet web page, http://www.w3.org/TR/2009/WD-html5-20090423/, retrieved on August 2009.

[Huang04]
Z. Huang, A. Eliëns, C. Visser, Facial Expressions for Embodied Agents in STEP, Proceedings of AAMAS 2004 Workshop on Embodied Conversational Agents: Balanced Perception and Action, 2004.

[Huang07]
Z. Huang, W. Xuan, X. Chen, Spatial temporal geographic ontology, in Geoscience and Remote Sensing Symposium, 2007 (IGARSS'07), IEEE International, p.4627-4630, 2007.

[Informa]
M2 PRESSWIRE-19 March 2009-Informa Telecoms & Media: Informa Telecoms & Media reveals world broadband and IPTV statistics for 2008(C)1994-2009 M2 COMMUNICATIONS LTD, Internet web site, www.informatm.com, retrived in July 2009

[Isaac05]
A. Isaac, Conception et utilisation d'ontologies pour l'indexation de documents audiovisuels, Thesis, Paris IV University, Sorbonne, 2005.

[Ishaya07]
T. Ishaya, E. Eze, Context-Based Multimedia Ontology Model, Autonomic and Autonomous Systems, 2007. ICAS07. Third International Conference on, p. 2, 19-25 June 2007.

[Joyce06]
R. Joyce, B. Liu, Temporal Segmentation of video using frame and histogram Space, IEEE Transation on Multimedia, vol.8, No.1, February 2006.

[Junee09]
R. Junee, Zoinks! 20 Hours of Video Uploaded Every Minute!, The YouTube Blog, Wednesday, May 20, 2009, Internet web site, http://youtube-global.blogspot.com/2009_05_01_archive.html, retrieved on July 2009.

[Juszczyk05]
L. Juszczyk, Institut f¨ur Informationssysteme der Technischen Universit¨at Wien, Univ.Prof.Mag.rer.soc.oec.Dr.rer.soc.oec. Schahram Dustdar ,Vienna, May 2005, http://www.infosys.tuwien.ac.at/staff/lukasz/06_masterthesis_juszczyk.pdf

[Kakas90]
A. C. Kakas, P. Mancarella, Database updates through abduction, in Proceedings of the 16th International Conference on Very Large Databases (VLDB'90), p. 650–661, Brisbane, Australia, 1990, isbn 0-55860-149-X.

[Kienzle05]
W. Kienzle, G. Bakir, M. Franz, B. Scholkopf, Face Detection - Efficient and Rank Deficient. Advances in Neural Information Processing Systems 17, p. 673-680, 2005.

[Kim08]
M. Kim, S. Kumar, V. Pavlovic, H. Rowley, Face Tracking and Recognition with Visual Constraints in Real-World Videos, IEEE Conference on Computer Vision and Pattern Recognition (CVPR08), p. 1-8, 23-28 June 2008, isbn 978-1-4244-2242-5.

[Kubey02]
R. Kubey, M. Csikszentmihalyi, Television Addiction Is No Mere Metaphor, Scientific American, vol. 286, issue 2, ;286(2), Feb 2002.

[Lacot05]
X. Lacot, Introduction à OWL, un langage XML d'ontologies Web, Juin 2005.

[LBPWiki]
LittleBigPlanet web page on Wikipedia, Wikipedia®, Internet web site, http://en.wikipedia.org/wiki/LittleBigPlanet, retrieved on July 2009.

[LCP]
La Chaîne Parlementaire - Assemblée nationale, © 2008 LCP Assemblée nationale, Internet web site, http://www.lcpan.fr/

[Le07]
D. B. Le, Modèle d'édition de document multimédia. Masters thesis, Institut de la Francophonie pour l'Informatique, Hanoï, October 2007.

[Lee04]
W. - S. Lee, K. - A. Sohn, Face Recognition using Computer-Generated Database, in Proceedings of the Computer Graphics International (CGI'04), p. 561-568, 2004, isbn 0-7695-2171-1.

[Lee99]
W.-S. Lee, N. Magnenat-Thalmann, Generating a Population of Animated faces from Pictures, in Proceedings of Modelling People 1999, IEEE International Workshop on, p. 62-69, Kerkyra, Greece, February 1999.

[LeBonhomme08]
B. Le Bonhomme, M. Preda, F. Prêteux, From MPEG-4 Scene Representation to MPEG-7 Description, in M. Granitzer, M. Lux, M. Spaniol (Ed.), Multimedia Semantics - The Role of Metadata, Studies in Computational Intelligence, Vol.101, Springer-Verlag, Berlin, Germany, March 2008, isbn. 978-3-540-77472-3.

[LeGall91]
D. Le Gall, MPEG: a video compression standard for multimedia applications, Communications of the ACM, 1991, vol. 34, No.4, p. 46-58 April, 1991, issn. 0001-0782.

[Lemlouma05]
T. Lemlouma, N. Layaïda, Content Interaction and Formatting for Mobile Devices, In Proceedings of the 2005 ACM Symposium on Document Engineering, DocEng 2005, p. 98-100, ACM Press, November 2005.

[Lemuet08]
X. Lemuet, Guide des chaînes numériques, Association des Chaînes Conventionnées éditrices de Services, Mars 2008.

[Lester04]
J. Lester, K. Branting, B. Mott, Conversational Agents, The Practical Handbook of Internet Computing, Chapman & Hall, 2004.

[Little Big Planet]
Little Big Planet, copyrights 2007 Sony Computer Entertainment Europe, Internet web site http://www.littlebigplanet.com/, retrieved on July 2009.

[Liu05]
N. Liu, B. C. Lovell, Hand Gesture Extraction by Active Shape Models, Proceedings of the Digital Imaging Computing: Techniques and Applications (DICTA 2005), p.10, IEEE, 2005.

[Llach00]
J. Llach, L. Garrido, Visual segment tree creation for MPEG-7 description schemes, International Conference on Multimedia and Expo, ICME'2000, Vol. 2, p.907-910, 2000.

[Loof08]
A. Lööf, Internet usage in 2008 – Households and Individuals, Eurostats - Data in Focus, No 46, 2 December, 2008, issn. 1977-0340.

[Lu02]
L. Lu, H. J. Zhang, H. Jiang, Content Analysis for Audio Classification and Segmentation, IEEE Transaction on speech and audio processing, Vol. 10, No. 7, October 2002.

[Luo01]
J. Luo, A. Savakis, Indoor vs outdoor classification of consumer photograph using lowlevel and semantic features, in Proceedings of Image Processing (ICIP01), International Conference on, vol.2, p.745-748, Thessaloniki, Greece, October 2001.

[Lutz07]
C. Lutz, D. Walther, F. Wolter, Conservative extensions in expressive description logics, in Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'07), p.453-459, 2007.

[Manerba08]

F. Manerba, J. Benois-Pineau, R. Leonardi, B. Mansencal, Multiple Moving Object Detection for Fast Video Content Description in Compressed Domain, EURASIP Journal on Advances in Signal Processing (JASP), p. 1-15, Article ID 231930, Vol. 2008.

[Manjoo03]
F. Manjoo, Your TV is watching you, Salon Media Group Inc, May 8, 2003.

[Marchand-Maillet06]
S. Marchand-Maillet, E. Bruno, N. Moënne-Loccoz, Structured Multimedia Description for Simplified Interaction and Enhanced Retrieval, ECRIM News No.66, European Research Consortium for Informatics and Mathematics, issue: European Digital Library, p. 38-40, No. 66, July 2006.

[Marilly06]
E. Marilly, G. Delègue, O.Martinot, S.Betgé-Brezetz, Adaptation and Personalization of Interactive MobileTV Services, ICIN Conférence 2006.

[Martin98]
K. D. Martin, Y. E. Kim, Musical instrument identification: a pattern-recognition approach, in Proceedings of 136th Meeting of Acoustical Society of America (ASA'98), Vol. 104, Issue 3, p.1768-1768, Norfolk, Va, USA, September 1998.

[Martin04]
D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, K. Sycara, OWL-S: Semantic Markup for Web Services, W3C Member Submission 22 November 2004, Internet web page, http://www.w3.org/Submission/OWL-S/, retrieved on August 2009.

[Mazuel08]
L. Mazuel, N. Sabouret, Semantic relatedness measure using object properties in an ontology, In Proc. 7th International Semantic Web Conference (ISWC 2008), LNCS 5318, pp. 681-694, Springer-Verlag, 2008.

[McKee]
J. McKee, The 90-9-1 Principle, Internet web site, http://www.90-9-1.com/, website retrieved on July 2009.

[Mech98]
R. Mech, M. Wollborn, A Noise Robust Method For 2D Shape Estimation Of Moving Objects In Video Sequences Considering A Moving Camera, Signal Processing, Vol. 66, No. 2, p. 203-217, April 1998.

[MediaCabSat]
MédiaCabSat Studies, © 2009 Médiamétrie, Internet web site, http://www.mediametrie.fr, retrieved on July 2009.

[MediaInLife09]
Une consommation des médias boostée par les 13-17 ans et le développement des nouveaux supports d'écoute, communique de presse, Levallois, June 2009, Médiamétrie, Based on study Media In Life – Base Lundi-Dimanche Janv-Fév 2009, 00h-24h, Ensemble 13 ans et plus, toutes localisations, avec ou sans accompagnement, avec ou sans activité courante, Médiamétrie, http://www.mediametrie.fr/, 2009.

[Mediamat08]
Communiqué de presse, Médiamat Annuel 2008, Du 31 décembre 2007 au 28 décembre 2008 Durée d'écoute par individu de la télévision En heures et minutes - Jour moyen Lundi-Dimanche - de 3h00 à 3h00, Le 29 décembre 2008, Internet web site, www.mediametrie.fr, retrieved on July 2009

[Mei07]

T. Mei, X.-S. Hua, L. Yang, S. Li. VideoSense - Towards Effective Online Video Advertising, In Proceedings of the 15th ACM International Conference on Multimedia (ACM MM'07), p.1075-1084, Augsburg, Germany, September 2007.

[Mezaris04]
V. Mezaris, I. Kompatsiaris, M. G. Strintzis, Video Object Segmentation Using Bayes-Based Temporal Tracking and Trajectory-Based Region Merging, IEEE Transaction on Circuits and Systems for Video Technology, p.782-795, vol.14, No. 6, June 2004.

[MHP]
Multimedia Home Platform, © DVB 2008, Internet web site, http://www.mhp.org, retrieved on May 2009.

[ModNation]
Mod NationRacers game, copyrights 2009 Sony Computer Entertainment America Inc., Internet web site http://www.us.playstation.com/, retrieved on July 2009.

[Moulin05]
C. Moulin, M. Sbodio, Using Ontological Concepts for Web Service Composition, Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, p. 487–490, 2005.

[MPEG-1]
MPEG-1 Systems, ISO/IEC-11172-1: Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s, ISO/IEC JTC1/SC29/WG11, 1993.

[MPEG-2]
MPEG-2 Systems, ISO/IEC 13818-1: Generic Coding of Moving Pictures and Associated Audio Information, Second edition, December. 2000.

[MPEG-4]
ISO/IEC 14496-1:2001 — Information technology — Coding of Audio-Visual Objects — Part 1: Systems, ISO/IEC JTC1/SC29/WG11, Second edition, October. 2001.

[MPEG4-part10]
ISO/IEC 14496-10:2003 — Information technology — Coding of Audio-Visual Objects — Part 10: Advanced video coding, ISO/IEC JTC1/SC29/WG11, December. 2003.

[MPEG4-part11]
ISO/IEC 14496-11:2005 — Information technology — Coding of audio-visual objects — Part 11: Scene description and application engine, ISO/IEC JTC1/SC29/WG11, 2005.

[MPEG4-part20]
ISO/IEC 14496-20:2008 — Information technology — Coding of audio-visual objects — Part 20: Lightweight Application Scene Representation (LASeR) and Simple Aggregation Format (SAF), Second edition, Dec. 2008.

[MPEG7]
ISO/IEC 15938-1:2002 — Information technology — Multimedia content description interface — Part 1: System, July 2002.

[MPEG7Ov]
MPEG-7 Overview (v.10), ISO/IEC JTC1/SC29/WG11N6828, Palma de Mallorca, October 2004.

[Nack04]
F. Nack, J. van Ossenbruggen, L. Hardman, That Obscure Object of Desire: Multimedia Metadata on the Web, Part 1, IEEE MultiMedia, p. 38-48, Vol. 11, No. 4, Oct.-Dec. 2004, issn 1070-986X.

[Nack05]

F. Nack, J. van Ossenbruggen, L. Hardman, That Obscure Object of Desire: Multimedia Metadata on the Web, Part 2, IEEE MultiMedia, p. 54-63, Vol. 12, No. 1, Jan.-March 2005, issn 1070-986X.

[Naphade00]
M.R. Naphade, I. Kozintsev, T. Huang, Probabilistic semantic video indexing, In Proceedings of 14th Neural Information Processing Systems (NIPS), p.967-973, Denver, CO, USA, 2000.

[NDS]
[NDS] MediaHighway®, middleware graphic TV interface, NDS Group Ltd., Internet web site, http://www.nds.com/solutions/mediahighway.php

[Nefian02]
A. V. Nefian, L. H. Liang, X. X. Liu, X. Pi, K. Murphy, "Dynamic Bayesian networks for audio-visual speech recognition", EURASIP, Journal of Applied Signal Processing, vol. 2002, no 11, p. 1274-1288, 2002.

[Neto03]
E. L. Andrade Neto, E. Khan, J. C. Woods, M. Ghanbari, Player Classification in Interactive Sport Scenes Using Prior Information Region Space Analysis and Number Recognition, IEEE International Conference on Image Processing (ICIP) 2003, Barcelona, Spain, vol. III, p.129-132, September 2003.

[Norway09]
Continued decline in newspaper reading, Norwegian media barometer 2008, © Statistics Norway, April 2009, Internet web site, http://www.ssb.no/emner/07/02/30/medie/, retrieved on July 2009.

[Odell02]
J. J. Odell, Objects and Agents Compared, in Journal of Object Technology, vol. 1, no. 1, p.41-53 http://www.jot.fm/issues/issue_2002_05/column4, May-June 2002.

[Ofcom08]
The Communications Market 2008, Communications Market Reports, August 2008, Ofcom, www.ofcom.org.uk, retrieved on July 2009.

[OntologyWiki]
Ontology web page on Wikipedia, Wikipedia®, Internet web site, http://en.wikipedia.org/wiki/Ontology(information_science), retrieved on July 2009.

[OSGi]
OSGi Alliance Site, OSGi™ - The Dynamic Module System for Java™, Internet web site, http://www.osgi.org/Main/HomePage, Copyright © 2009 OSGi™ Alliance, retrieved on july 2009.

[OWL04]
OWL Web Ontology Language Overview, W3C, W3C Recommendation 10 February 2004, Internet web site, http://www.w3.org/TR/owl-features/, retrieved on July 2009.

[Papadopoulos08]
G. T. Papadopoulos, K. Chandramouli, V. Mezaris, I. Kompatsiaris, E. Izquierdo, M.G. Strintzis, A Comparative Study of Classification Techniques for Knowledge-Assisted Image Analysis, Ninth International Workshop on Image Analysis for Multimedia Interactive Services, 2008, p.4-7, WIAMIS '08. May 2008.

[Passin04]
T. B. Passin, Explorer's Guide to the semantic WEB, Manning Publications Co., p.304, ISBN 978-1-932394-20-6, March 2004.

[Park05]

U. Park, H. Chen, A. K. Jain, 3D Model-Assisted Face Recognition in Video, in Proceedings of the 2nd Workshop on Face Processing in Video, British Columbia, Canada, May 8-11, 2005.

[Perott02]
A. J. Perott , A. T. Lindsay, A. P. Parkes, Real-time Multimedia Tagging and Content-Based Retrieval for CCTV Surveillance Systems, Proceedings of SPIE, Volume 4862, Internet Multimedia Management Systems III, p.40-49 , July 2002.

[Pfeiffer96]
S. Pfeiffer, S. Fischer, W. Effelsberg, Automatic Audio Content Analysis, ACM Multimedia 96, Boston MA USA, ACM 0-89791-871-1/96/11, 1996

[Rabiner78]
L. Rabiner, R.W. Schafer, Digital Processing of Speech Signals, Englewood Cliffs, N.J.: Prentice Hall, 1978.

[Radhakrishnan08]
R. Radhakrishnan, C. Bauer, Robust video fingerprints based on subspace embedding, In proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008), 2008, p.2245-2248, April 2008, isbn 978-1-4244-1483-3

[RDFS_Wiki]
RDF Schema web page on Wikipedia, Wikipedia®, Internet web site, http://en.wikipedia.org/wiki/RDF_schema, retrieved on August 2009.

[Ross04]
S. Ross, E. A. Brownholtz, R. C. Armes, Principles and Architecture for a Conversational Agent, IUI 2004 International Conference on Intelligent User Interfaces, Madiera, Portugal, January 2004.

[Rowe00]
L. A. Rowe, The Future of Interactive Television, Computer Science Division, University of California at Berkeley, Rowe, 2000.

[Royer07a]
J. Royer, H. Nguyen, O. Martinot, M. Preda, F. Preteux, T. Zaharia, Interactive TV on Parliament Session, Proceedings SPIE Conference on Mathematics of Data/Image Pattern Recognition, Compression, Coding, and Encryption X with Applications, San Diego, CA, vol. 6700, p. 1-12, August 2007.

[Royer07b]
J. Royer, H. Nguyen, D. O Mishra, Automatic Generation of Explicitly Embedded Advertisement for Interactive TV: Concept and System Architecture, Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology, Mobility Conference 2007, session: Next generation communication service, p. 332-338, ACM 2007, Singapore, September 10-12, 2007.

[Royer08]
J. Royer, H. Nguyen, F. Preteux, O. Martinot, Multimedia Interactive Services Automation based on Contents Indexing, Bell Labs Technical Journal, p. 147-154, Vol. 13, No. 2, XP001514357 Wiley, CA, June 2008.

[RSSBoard]
RSS Advisory Board homepage, Internet web site, http://www.rssboard.org/, retrieved on May 2009.

[Rubicon08]

Online Communities and their Impact on Business: Ignore at Your Peril, Web users and web community, p.37, October 22, 2008, copyrights 2008 Rubicon Consulting, Inc., Internet web page http://rubiconconsulting.com/downloads/whitepapers/Rubicon-web-community.pdf, retrieved on July 2009.

[Rui07]
Y. Rui, G.-J. Qi, Learning Concepts by Modeling Relationships, in proceedings of International Workshop on Multimedia Content Analysis and Mining 2007 (MCAM'07), p.5-13, Weihai, China, June 30-July 1, 2007.

[Saltzman97]
J. Saltzman, Too much information, too little time – Column, USA Today (Society for the Advancement of Education). FindArticles.com, 12 Sept. 1997.

[Schallauer06]
P. Schallauer, W. Bailer, G. Thallinger, A Description Infrastructure for Audiovisual Media Processing Systems Based on MPEG-7, Journal of Universal Knowledge Management, Vol. 1 , Issue 1, p. 26-35, 2006.

[Scherp07]
A. Scherp, A Component Framework for Personalized Multimedia Applications, Carl von Ossietzky University of Oldenburg, School of Computing Science, Business Administration, Economics and Law, Department of Computing Science, Ph.D. Thesis, August 2006. Published February, 2007, by OlWIR, Oldenburg, Germany. ISBN 978-3-939704-11-9.

[Shao06]
B. Shao, L. M. Velazquez, N. Scaringella, N. Singh, M. Mattavelli, SMIL to MPEG-4 BIFS Conversion, Proceedings of the Second International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution, p. 77-84, 2006.

[Shao09]
G. Shao, Understanding the appeal of user-generated media: a uses and gratification perspective, Internet Research,Emerald Group Publishing Limited, p. 7-25, issue 1, vol. 19, 2009.

[Shin06]
J. Shin, D. Y. Suh, Y. Jeong, S. H. Park, B. Bae, and C. Ahn, Demonstration of Bidirectional Services Using MPEG-4 BIFS in Terrestrial DMB Systems, ETRI Journal, Volume 28, Number 5, October 2006.

[Shedroff94]
N. Shedroff, Information Interaction Design: A Unified Field Theory of Design, book chapter, Information Design, p.267-293, The MIT Press, June 18, 1999, isbn. 978-0262100694.

[Sidla06]
O. Sidla, Y. Lypetskyy, N. Brandle, S. Seer, Pedestrian Detection and Tracking for Counting Applications in Crowded Situations, IEEE International Conference on Video and Signal Based Surveillance, AVSS '06, p. 70, November 2006.

[Siebel02]
N. T. Siebel, S. Maybank, Fusion of Multiple Tracking Algorithms for Robust People Tracking, in Proceedings of the 7th European Conference on Computer Vision (ECCV 2002), Kobenhavn, Denmark, May 2002, vol. 4, p. 373-387, Springer Verlag, 2002, ISBN 3-540-43748-7.

[Sifakis02]
E. Sifakis, I. Grinias, G. Tziritas, Video Segmentation using fast Marching and Region Growing Algotithms, EURASIP Journal on Applied Signal Processing, vol. 2002, No. 4, p.379-388, January 2002.

[Signes00]
J. Signes, Y. Fisher, A. Eleftheriadis, MPEG-4's binary format for scene description, Tutorial issue on the MPEG-4 standard, vol. 15, no 4-5 (7 ref.), p.321-345, 2000.

[Sikora05]
T. Sikora, MPEG-7-based Audio Annotation, MPEG-7-based Audio Annotation for the Archival of Digital Video, 2005.

[Skalski]
P. Skalski, The Content Analysis Guidebook Online, Internet web site, http://academic.csuohio.edu/kneuendorf/content/index.htm, retrieved on July 2009.

[Smeulders00]
A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, Pattern Analysis and Machine Intelligence, IEEE Transactions on, p.1349-1380, Volume 22, Issue 12, Dec. 2000.

[SMIL]
Synchronized Multimedia Integration Language (SMIL 3.0), W3C Recommendation 01 December 2008, Internet web page, http://www.w3.org/TR/SMIL3/, retrieved on June 2009.

[Smith07]
J. R. Smith, The real Problem of Bridging the “Semantic Gap”, Multimedia Content Analysis and Mining (MCAM07), Lecture Notes in Computer Science, Vol. 4577, p.16-17, July 2007.

[Solis08]
B. Solis, J. Thomas, Internet web page, http://theconversationprism.com, retrieved on July 2009.

[Spore]
Spore game, http://www.spore.com, copyright 2009 Electronic Arts Inc., Internet web site, retrieved on July 2009.

[SVG]
W3C, Scalable Vector Graphics (SVG) 1.1 Specification [Recommendation], Internet web page, http://www.w3.org/TR/SVG11/, retrieved on August 2009.

[TAC]
TAC.TV, © Salambo Productions inc., Internet web site, http://www.tac.tv/index.php, retrieved on July 2009

[Tauberer06]
J. Tauberer, What Is RDF, xml.com, July 26, 2006, Internet web page, http://www.xml.com/pub/a/2001/01/24/rdf.html, retrieved on August 2009.

[Teinturier07]
B. Teinturier, Les Français et la Télévision, Vol. 4 of studies "Les enjeux du quotidien…", TNS-Sofres, France, 7 December, 2007, Internet web site, www.tns-sofres.com, retrieved on July 2009.

[Tessier07]
M. Tessier, M. Baffert, La presse au défi du numérique, Rapport au ministre de la culture et de la communication, La Documentation française, Paris, 2007.

[Tran03]

S.M. Tran, M. Preda, F. Prêteux, K. Fazekas, Exploring MPEG-4 BIFS features for creating multimedia games, Proceedings IEEE International Conference on Multimedia and Expo(ICME'03), Vol.1, p.429-432, Baltimore, WA, 2003.

[Tran07]
S. Tran, L. Davis, Robust Object Tracking with Regional Affine Invariant Features, ICCV 2007. IEEE 11th International Conference on Computer Vision, p.1-8, Rio de Janeiro, Brazil, 14-21 Oct. 2007.

[Tranchard94]
L. Tranchard, The systems part of MPEG-2 standard: objectives, problems and their solutions, Revue annuelle - LEP, p. 35-38, 1994, issn. 0750-6287.

[Troncy03]
R. Troncy, Integrating Structure and Semantics into Audio-visual Documents, in Second International Semantic Web Conference (ISWC 2003), Vol. 2870, p. 566–581, Sanibel Island, Florida, USA, October 20-23, 2003, isbn. 3-540-20362-1.

[Troncy07]
R. Troncy, W. Bailer, M. Hausenblas, M. Höffernig, VAMP: Semantic Validation for MPEG-7 Profile Descriptions, Proceedings of Multimedia Metadata Applications Workshop at I-MEDIA'07, Journal of Universal Computer Science, Sept. 2007.

[Turk02]
M. Turk, C. Hu, R. Feris, F. Lashkari, and A. Beall, TLA based face tracking, in Proceedings of International Conference on Vision Interface (VI'2002), p. 229-235, Calgary, Canada, May 27-29, 2002.

[TV-Anytime]
ETSI TS 102 822, Broadcast and On-line Services: Search, select and rightful use of content on personal storage systems ("TV-Anytime"); Internet web site, http://www.tv-anytime.org, retrieved on July 2009.

[UDDI]
Official community gathering place and information resource for the UDDI OASIS Standard, © 1993-2009 OASIS, Internet web page, http://uddi.xml.org/, retrieved on August 2009.

[Vercouter07]
L. Vercouter, S. J. Casare, J. S. Sichman, A. A. F. Brandao, An Experience on Reputation Models Interoperability based on a Functional Ontology, in Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), p. 617-622, Hyderabad, India, January 6-12, 2007.

[Verilook]
VerilookSDK, Copyright © 1998 - 2009 Neurotechnology, Internet web site, http://www.neurotechnology.com/, retrieved on july 2009.

[Viljoen03]
D. W. Viljoen, A. P. Calitz, N. L. O. Cowley, A 2-D MPEG-4 Multimedia Authoring Tool, in Proceedings of the 2nd international conference on Computer graphics, virtual Reality, visualisation and interaction in Africa (AFRIGRAPH '03), p. 151-160, Cape Town, South Africa, 2003, isbn. 1-58113-643-9.

[VRML97]
VRML97 Functional specification and External Authoring Interface (EAI), ISO/IEC 14772-1:1997 and ISO/IEC 14772-2:2004 — Virtual Reality Modeling Language (VRML), 1997.

[Vezhnevets03]

V. Vezhnevets, V. Sazonov, A. Andreeva, A Survey on Pixel-Based Skin Color Detection Techniques, in Proceedings of the 13th Conference on Graphicon 2003, p. 85-92, Moscow, Russia, September 2003.

[Wactlar02]
H. D. Wactlar, M. G. Christel, Digital Video Archives: Managing Through Metadata, In Building a National Strategy for Digital Preservation, Issues: Digital Media Archiving, Washington DC: Council on Library and Information Resources and the Library of Congress, April 2002.

[Wactlar99]
H. D. Wactlar, M. G. Christel, Y. Gong, Alexander G. Hauptmann, Lessons Learned from the Creation and Deployment of a Terabyte Digital Video Library, IEEE Computer, Vol. 32, No. 2, p. 66-73, 1999.

[Wendler99]
R. Wendler, LDI Update: Metadata in the Library, Harvard University Library Notes, no. 1294, p.4-5, July/August 1999.

[Wei07]
Y. Wei, J. Sun, X. Tang, H.-Y. Shum, Interactive Offline Tracking for Color Objects, IEEE 11th International Conference on Computer Vision (ICCV'07), p.1-8, Rio de Janeiro, Brazil, October 14-20, 2007.

[WikiESP]
ESP Game web page on Wikipedia, Wikipedia®, Internet web site, http://en.wikipedia.org/wiki/ESP_Game, retrieved on May 2009.

[WikiOSGi]
OSGi web page on Wikipedia, Wikipedia®, Internet web site, http://en.wikipedia.org/wiki/OSGi, retrieved on May 2009.

[WikipediaToday]
The current home page for Wikipedia, Internet web page, http://wikipedia.org/, Wikipedia®, retrieved in July 2009.

[Wiki_UGC]
User Generated Content web page on Wikipedia, Wikipedia®, Internet web site, http://en.wikipedia.org/wiki/User-generated_content, retrieved on June 2009.

[Wright06]
W. Wright, Will Wright and Spore, Game Developers Conference, Google Video, 2006, http://video.google.com/videoplay?docid=-2627744901843480 66&q=spore, retrieved on July 2009.

[WS-BPEL]
Web Services Business Process Execution Language Version 2.0, OASIS Standard, 11 April 2007, Copyright © OASIS® 1993–2007, Internet web page, http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html, retrieved on august 2009.

[WSDL]
E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001, Internet web page, http://www.w3.org/TR/wsdl, retrieved on August 2009.

[WSDL-S]
R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, M.-T. Schmidt, A. Sheth, K. Verma, Web Service Semantics - WSDL-S, Version 1.0, W3C Member Submission 7 November 2005, Internet web page, http://www.w3.org/Submission/WSDL-S/, retrieved on August 2009.

[WS-HumanTask]

A. Agrawal, M. Amend, M. Das, M. Ford, C. Keller, M. Kloppmann, D. König, F. Leymann, R. Müller, K. Plösser, R. Rangaswamy, A. Rickayzen, M. Rowley, P. Schmidt, I. Trickovic, A. Yiu, M. Zeller, Web Services Human Task (WS-HumanTask) Version 1.0. June 2007, Copyright © 2007 Active Endpoints Inc., Adobe Systems Inc., BEA Systems Inc., International Business Machines Corporation, Oracle Inc., and SAP AG.

[Wu08]
C. Wu, V. Potdar, E. Chang, Latent Semantic Analysis – The Dynamics of Semantics Web Services Discovery, Advances in Web Semantics I: Ontologies, Web Services and Applied Semantic Web, 2008, in Chang, E. and Dillon, T. and Meersman, R. and Sycara, K. (ed), Advances in Web Semantics I, p. 346-373. Heidelberg, Germany: Springer.

[XMethods]
Web Service Provider, Internet web site, http://xmethods.com/ve2/index.po, retrieved on August 2009.

[Xuggle]
Xuggle, Copyright© 2009 Xuggle Inc., Internet web site, http://www.xuggle.com, retrieved on May 2009.

[Yang02]
M. Yang, D. Kriegman, N. Ahuja, Detecting Faces in Images: A Survey, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No.1, p.34-58, January 2002.

[ZenithOptimedia07]
ZenithOptimedia Group Limited, Global ad market to accelerate in 2008 despite credit squeeze, "forecats December 2007", Internet web site, www.zenithoptimedia.com, retrieved in july 2009.

[Zhang06]
L. Zhang, H. Ai, Multi-View Active Shape Model with Robust Parameter Estimation, in Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Vol. 4, p. 469-468, 2006.

[Zhao02]
W. Zhao, R. Chellappa, A. Rosenfeld, P.J. Phillips. Face Recognition: A literature survey. Technical Report, CART-TR-948. University of Maryland, Aug. 2002.

[Zhou03]
C. Zhou, H. Tao, Dynamic Depth Recovery from Unsynchronized Video Streams, Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03), vol.2, p.351-359, June 2003.

Abréviations

AAC	« Advanced Audio Coding »
API	« Application Programming Interface »
AFX	« Animation Framework eXtension »
AIML	« Artificial Intelligence Markup Language »
API	« Application Programming Interface »
AVSR	« Audio-Visual Speech Recognition »
BIFS	« Binary Format for Scene description »
CIF	« Common Intermediate Format »
CNL	« Constraint Natural Language »
DC	« Direct Component (in frequency representation) »
DCT	« Discrete Cosine Transform »
DTT	« Digital Terrestrial Television »
DVB	« Digital Video Broadcasting »
DVB-H	« Digital Video Broadcasting – Handheld »
EDGE	« Enhanced Data Rates for GSM Evolution »
EPG	« Electronic Program Guide »
ES	« Elementary Stream »
ESG	« Electronic Service Guide »
GMC	« Global Motion Compensation »
GPRS	« General Packet Radio Service »
GUI	« Graphic User Interface »
IP	« Internet Protocol »
IPMP	« Intellectual Property Management and Protection »
ISO	« International Standardization Organization
LASeR	« Lightweight Application Scene Representation »
LTC	« Longitudinal Time Code »
M4IF	« MPEG-4 Industry Forum »
Mdat	« media data atoms »
MPEG	« Motion Picture Expert Group »
MPEG-2	« Audio-Video Standard ISOIEC 13818 »
MPEG-2 TS	« MPEG-2 Transport Stream »
MPEG-4	« Multimedia Standard ISOIEC 14496 »
MPEG-7	« Multimedia Content Description Interface ISOIEC »
MPEG-J	« Framework for MPEG Java API's »
OCI	« Object Content Information »

OD	« Object descriptor »
OWL	« Ontology Web Language »
PDA	« Personal Digital Assistant »
QCIF	« Quarter Common Intermediate Format »
QoS	« Quality of Service »
RSS	« Really Simple Syndication »
RTP	« Real Time Transport Protocol »
RTSP	« Real Time Streaming Protocol »
SMIL	« Synchronized Multimedia Integration Language »
SMPTE	« Society of Motion Picture and Television Engineers »
SVG	« Scalable Vector Graphics »
TCP	« Transmission Control Protocole »
TNT	« National digital terrestrial television for France »
TTS	« Text-to-speech »
UDP	« User Datagram Protocol »
UGC	« User Generated Content »
UMTS	« Universal Mobile Telecommunication System »
URI	« Uniform Resource Identifier »
VLBV	« Very Low Bitrate Video »
VOD	« Video On Demand »
VRML	« Virtual Reality Modeling Language »
VITC	« Vertical Interval Time Code »
W3C	« World Wide Web Consortium »
XML	« eXtensible Markup Language »
XMT	« eXtensible MPEG-4 Textual format »

Annexes

Annexe 1 – Liste des publications et brevets

Liste des publications

[Royer07a] J. Royer, H. Nguyen, O. Martinot, M. Preda, F. Prêteux, T. Zaharia, ***Interactive TV on Parliament Session***; Proceedings SPIE Conference on Mathematics of Data/Image Pattern Recognition, Compression, Coding, and Encryption X with Applications, San Diego, CA, August 2007, vol. 6700, p. 67000J:1-12.

[Royer07b] J. Royer, H. Nguyen, D. O Mishra, ***Automatic Generation of Explicitly Embedded Advertisement for Interactive TV: Concept and System Architecture***; Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology, Mobility Conference 2007, session: Next generation communication service, pp 332-338, ACM 2007, Singapore, September 10-12, ISBN:978-1-59593-819-0.

[Royer08] J. Royer, H. Nguyen, F. Prêteux, O. Martinot, ***Multimedia Interactive Services Automation based on Contents Indexing***, Bell Labs Technical Journal, p. 147-154, Vol. 13, No. 2, XP001514357 Wiley, CA, June 2008 , ISSN: 1089-7089.

Liste des brevets déposés

J. Royer, O. Martinot, G. Delegue, Modular architecture for Semantic Multimedia Analyzer, Patent number *FR 0754495*, Octobre 2006.

J. Royer, G. Delegue, Iterative Enrichment of Multimedia Description, Patent number *EPA 08290528.2*, Août 2008.

J. Royer, O. Martinot, F. Prêteux, Multimedia Service Enabler based on Dynamic Analysis Composer and Orchestrator, Patent number *FR 0856377*, September 2008.

A. Outtagarts, J. Royer, Automatic users identifications based on sensors and their behaviors, February 2009, waiting for patent number.

Annexe 2 – Résultats de la sélection des analyseurs

Tableau 11 : Résultats d'analyse de la capacité de description de la plateforme

Liste		Children
/media	REQUIRED	2
./media/description	REQUIRED	0
./media/video	OPTIONAL	1
../media/video/segment	ONE_OR_MORE	1
.../media/video/segment/face	ZERO_OR_MORE	2
..../media/video/segment/face/faceName	OPTIONAL	0
..../media/video/segment/face/movingRegion	REQUIRED	1
...../media/video/segment/face/movingRegion/region	ONE_OR_MORE	0

Tableau 12 : Liste de la capacité de description des analyseurs sélectionnés

List of available analyzers: /media/video/segment/face
List of available analyzers: /media/video/segment/face/movingRegion
List of available analyzers: /media/video/segment/face/movingRegion/region
List of available analyzers: /media/description
List of available analyzers: /media/video
List of available analyzers: /media/video/segment/face/movingRegion/region
List of available analyzers: /media/video/segment
List of available analyzers: /media/video/segment
List of available analyzers: /media/video/segment/face/faceName

Tableau 13 : Liste des descripteurs requis par les services interactifs

List of available Services: MediaNode/VideoNode/SegmentNode/FaceNode/FaceNameNode
List of available Services: MediaNode/VideoNode/SegmentNode/FaceNode/RegionNode/MovingRegionNode

Tableau 14 : Liste des chemins d'analyse pour l'analyse du média

/media	REQUIRED	2
./media/description	REQUIRED	0
./media/video	OPTIONAL	1
../media/video/segment	ONE_OR_MORE	1
.../media/video/segment/face	ZERO_OR_MORE	2
..../media/video/segment/face/faceName	OPTIONAL	0
..../media/video/segment/face/movingRegion	REQUIRED	1
...../media/video/segment/face/movingRegion/region	ONE_OR_MORE	0

Tableau 15 : Liste des analyseurs pertinents pour l'analyse du média

com.alblf.media_knowledge.scenecut.factory.SceneCutAnalyzerFactory
com.alblf.media_knowledge.facetrack.factory.FaceTrackerAnalyzerFactory
com.alblf.media_knowledge.facetrack.factory.FaceTimeoutAnalyzerFactory

com.alblf.media_knowledge.facereco.factory.FaceRecoAnalyzerFactory
com.alblf.media_knowledge.facedetect.factory.FaceDetectAnalyzerFactory
com.alblf.media_knowledge.core.platform.MediaDescriptionAnalyzerFactory

Annexe 3 – Résultats pour la gestion des appels des analyseurs en fonction de l'évolution du média

Tableau 16 : Résultats de l'analyse des 56 premières images de la vidéo

Time on class com.alblf.media_knowledge.core.internal.analyzers.MediaDescriptionAnalyzer	*47*
Time on class com.alblf.media_knowledge.facedetect.internal.FaceDetectAnalyzer	*16*
Time on class com.alblf.media_knowledge.scenecut.internal.SceneCutAnalyzer	*16*
Time on class com.alblf.media_knowledge.ui.internal.UIDebugAnalyzer	*46*
4 analyzers were applied to frame 2	
Time on class com.alblf.media_knowledge.facedetect.internal.FaceDetectAnalyzer	*0*
Time on class com.alblf.media_knowledge.scenecut.internal.SceneCutAnalyzer	*16*
Time on class com.alblf.media_knowledge.ui.internal.UIDebugAnalyzer	*0*
3 analyzers were applied to frame 3	
Time on class com.alblf.media_knowledge.facedetect.internal.FaceDetectAnalyzer	*0*
Time on class com.alblf.media_knowledge.scenecut.internal.SceneCutAnalyzer	*0*
Time on class com.alblf.media_knowledge.ui.internal.UIDebugAnalyzer	*0*
3 analyzers were applied to frame 4	
..... Il n'y a pas de changement jusqu'à l'image n°56...	
3 analyzers were applied to frame 55	

Tableau 17 : Résultats d'analyse pour la reconnaissance du visage détecté dans les images 56 à 58 de la vidéo

Time on class com.alblf.media_knowledge.scenecut.internal.SceneCutAnalyzer	*0*
Time on class com.alblf.media_knowledge.facedetect.internal.FaceDetectAnalyzer	*32*
FaceTrackerAnalyzer analyzeImage by FaceTracker	
FaceTrackerAnalyzer Tracker on sub Face image: TLP:x:171;y:61 BRP:x:249;y:139;	
FaceTrackerAnalyzer New Face position = TLP:x:166;y:56 BRP:x:244;y:144; timeStamp:1869	
Time on class com.alblf.media_knowledge.facetrack.internal.FaceTrackerAnalyzer	*31*
FaceRecoAnalyzer analyzeImage by FaceReco	
FaceRecoAnalyzer sub Face image Rect = width:78 height:78*	
FaceRecoAnalyzer Face position should not be up to date	
Time on class com.alblf.media_knowledge.facereco.internal.FaceRecoAnalyzer	*0*
Time on class com.alblf.media_knowledge.facetrack.internal.FaceTimeoutAnalyzer	*0*
Time on class com.alblf.media_knowledge.ui.internal.UIDebugAnalyzer	*0*
6 analyzers were applied to frame 56	

Time on class com.alblf.media_knowledge.scenecut.internal.SceneCutAnalyzer	0
Time on class com.alblf.media_knowledge.facedetect.internal.FaceDetectAnalyzer	15
FaceTrackerAnalyzer analyzeImage by FaceTracker	
FaceTrackerAnalyzer Tracker on sub Face image: TLP:x:173;y:61 BRP:x:251;y:139;	
Time on class com.alblf.media_knowledge.facetrack.internal.FaceTrackerAnalyzer	0
FaceRecoAnalyzer analyzeImage by FaceReco	
FaceRecoAnalyzer sub Face image Rect = width:78* height:78	
FaceRecoAnalyzer Face position should not be up to date	
Time on class com.alblf.media_knowledge.facereco.internal.FaceRecoAnalyzer	0
Time on class com.alblf.media_knowledge.facetrack.internal.FaceTimeoutAnalyzer	0
Time on class com.alblf.media_knowledge.ui.internal.UIDebugAnalyzer	0
6 analyzers were applied to frame 57	
Time on class com.alblf.media_knowledge.scenecut.internal.SceneCutAnalyzer	0
Time on class com.alblf.media_knowledge.facedetect.internal.FaceDetectAnalyzer	0
FaceTrackerAnalyzer analyzeImage by FaceTracker	
FaceTrackerAnalyzer Tracker on sub Face image: TLP:x:173;y:62 BRP:x:251;y:140;	
Time on class com.alblf.media_knowledge.facetrack.internal.FaceTrackerAnalyzer	0
FaceRecoAnalyzer analyzeImage by FaceReco	
FaceRecoAnalyzer sub Face image Rect = width:78* height:78	
FaceRecoAnalyzer not possible to extract Template: QualityCheckGrayscaleDensityFailed	
Time on class com.alblf.media_knowledge.facereco.internal.FaceRecoAnalyzer	32
Time on class com.alblf.media_knowledge.facetrack.internal.FaceTimeoutAnalyzer	0
Time on class com.alblf.media_knowledge.ui.internal.UIDebugAnalyzer	0
6 analyzers were applied to frame 58	

Tableau 18 : Résultats d'analyse pour la reconnaissance du visage détecté, analyse image après image tant que le visage est présent et non identifié

Time on class com.alblf.media_knowledge.scenecut.internal.SceneCutAnalyzer	0
Time on class com.alblf.media_knowledge.facedetect.internal.FaceDetectAnalyzer	0
FaceTrackerAnalyzer analyzeImage by FaceTracker	
FaceTrackerAnalyzer Tracker on sub Face image: TLP:x:169;y:62 BRP:x:247;y:140;	
Time on class com.alblf.media_knowledge.facetrack.internal.FaceTrackerAnalyzer	0
FaceRecoAnalyzer analyzeImage by FaceReco	
FaceRecoAnalyzer sub Face image Rect = width:78* height:78	
FaceRecoAnalyzer not possible to extract Template: QualityCheckExposureFailed	
Time on class com.alblf.media_knowledge.facereco.internal.FaceRecoAnalyzer	31
Time on class com.alblf.media_knowledge.facetrack.internal.FaceTimeoutAnalyzer	0

Time on class com.alblf.media_knowledge.ui.internal.UIDebugAnalyzer	*0*
6 analyzers were applied to frame 103	
Time on class com.alblf.media_knowledge.scenecut.internal.SceneCutAnalyzer	*0*
Time on class com.alblf.media_knowledge.facedetect.internal.FaceDetectAnalyzer	*16*
FaceTrackerAnalyzer analyzeImage by FaceTracker	
FaceTrackerAnalyzer Tracker on sub Face image: TLP:x:169;y:62 BRP:x:247;y:140;	
Time on class com.alblf.media_knowledge.facetrack.internal.FaceTrackerAnalyzer	*0*
FaceRecoAnalyzer analyzeImage by FaceReco	
FaceRecoAnalyzer sub Face image Rect = width:78 height:78*	
FaceRecoAnalyzer UncleTom match with template: score = 48	
Time on class com.alblf.media_knowledge.facereco.internal.FaceRecoAnalyzer	*78*
Time on class com.alblf.media_knowledge.facetrack.internal.FaceTimeoutAnalyzer	*0*
Time on class com.alblf.media_knowledge.ui.internal.UIDebugAnalyzer	*0*
6 analyzers were applied to frame 104	

Tableau 19 : Liste des analyseurs « restants » après l'identification du visage détecté, l'analyseur de reconnaissance est désactivé

Time on class com.alblf.media_knowledge.scenecut.internal.SceneCutAnalyzer	*0*
Time on class com.alblf.media_knowledge.facedetect.internal.FaceDetectAnalyzer	*15*
FaceTrackerAnalyzer - analyzeImage by FaceTracker	
FaceTrackerAnalyzer - Tracker on sub Face image: TLP:x:169;y:62 BRP:x:247;y:140;	
Time on class com.alblf.media_knowledge.facetrack.internal.FaceTrackerAnalyzer	*0*
Time on class com.alblf.media_knowledge.facetrack.internal.FaceTimeoutAnalyzer	*0*
Time on class com.alblf.media_knowledge.ui.internal.UIDebugAnalyzer	*0*
5 analyzers were applied to frame 105	
Time on class com.alblf.media_knowledge.scenecut.internal.SceneCutAnalyzer	*0*
Time on class com.alblf.media_knowledge.facedetect.internal.FaceDetectAnalyzer	*16*
FaceTrackerAnalyzer - analyzeImage by FaceTracker	
FaceTrackerAnalyzer - Tracker on sub Face image: TLP:x:169;y:62 BRP:x:247;y:140;	
Time on class com.alblf.media_knowledge.facetrack.internal.FaceTrackerAnalyzer	*0*
Time on class com.alblf.media_knowledge.facetrack.internal.FaceTimeoutAnalyzer	*0*
Time on class com.alblf.media_knowledge.ui.internal.UIDebugAnalyzer	*0*
5 analyzers were applied to frame 106	

Annexe 4 – Scène interactive MPEG-4 BIFS vide

Le Tableau 20 présente la scène BIFS utilisée pour le déploiement de services interactifs « animés » en temps réel. Le Tableau 21 illustre le fichier « **.sdp* » généré par l'encodeur [Concolato05] utilisé pour lancer la lecture sur le Player. Ce fichier permet la déclaration de l'envoi de mise à jour BIFS Update sur le port définit (*6000*).

Tableau 20 : Contenu d'une scène interactive vide pour la gestion des animations à la volée

```
InitialObjectDescriptor {
 objectDescriptorID 1
 audioProfileLevelIndication 254
 visualProfileLevelIndication 254
 sceneProfileLevelIndication 254
 graphicsProfileLevelIndication 254
 ODProfileLevelIndication 254
 esDescr [
  ES_Descriptor {
   ES_ID 1
   decConfigDescr DecoderConfigDescriptor {
    streamType 3
    decSpecificInfo BIFSConfig {
     isCommandStream true
     pixelMetric true
     #modify display size here
     pixelWidth 100
     pixelHeight 100
    }
   }
  }
 ]
}
#insert your interactive scene here
```

Tableau 21 : Fichier « **.sdp* » correspondant à la scène du Tableau 20 généré par l'encodeur BIFS [Concolato05]

```
v=0
o=GpacBroadcaster 3326096807 1117107880000 IN IP4 127.0.0.1
s=MPEG4Broadcaster
c=IN IP4 127.0.0.1
t=0 0
a=mpeg4-iod:"data:application/mpeg4-
iod;base64,AjUATwEBAQEBAywAAQAEFQENAAAAAAAAAAAAAAAAAFBjA4DIAMgAYQAA
QAAAPoAAAAAAAAAAAAAAw=="
m=application 7070 RTP/AVP 96
a=rtpmap:96 mpeg4-generic/1000
a=mpeg4-esid:1
a=fmtp:96 profile-level-id=1; streamType=3; mode=generic; objectType=1;
indexLength=16; randomAccessIndication=1
```

Annexe 5 – Exemple de scène interactive MPEG-4 BIFS

Le Tableau 22 illustre la déclaration d'un prototype pour la gestion des zones interactives pour l'accès aux informations additionnelles dans le service interactif « serve sénat ».

Tableau 22 : définition des accès aux variables du prototype et leurs initialisations

```
PROTO BOX_PROTO [
        exposedField SFVec2f Box_translation  0 0
        exposedField SFVec2f Text_Translation 0 0
        exposedField SFVec2f scale 1 1
        exposedField SFFloat rotation 0
        exposedField SFFloat width 3
        exposedField SFVec2f Box_size 0 00
        exposedField SFColor Box_Color 1 0 0     #Box Color
        exposedField SFColor Text_Color 0 1 0    #Text Color
        exposedField SFBool filled FALSE         #Not filled
        exposedField SFFloat transparency 0
        exposedField SFColor lineColor 0 0 1
        exposedField SFFloat lineWidth 2
        exposedField SFBool enable_center_Align TRUE
        exposedField SFBool isOver TRUE
        exposedField SFBool enabled_Filled TRUE
        exposedField MFString stringName []      #facename
        exposedField SFNode NamefontStyle NULL
        eventIn MFString active
        exposedField SFNode obj NULL
]
```

Le Tableau 23 illustre la définition des variables accessible depuis les commandes Update BIFS ainsi que leurs valeurs à l'initialisation de la scène interactive.

Tableau 23 : définition des accès aux variables du prototype et leurs initialisations

```
{
 DEF BOX_TRANSFORM Transform2D{
        translation IS Box_translation     #BOX position
        children [
         DEF PeopleSensor TouchSensor{isOver IS isOver}
         Shape {
              appearance Appearance {
               material Material2D {
                lineProps XLineProperties {
                     isCenterAligned IS enable_center_Align
                     width IS width                #Width line of BOX
                     texture CompositeTexture2D {
                     pixelWidth 32
                     pixelHeight 32
                     children [
                     Shape {
                       appearance Appearance {
                         material Material2D {
                           emissiveColor IS Box_Color
                           filled IS enabled_Filled
                         }
                       }
                        geometry Rectangle{ size IS Box_size }
                      }
                     ]
                    }
               }
              }
              geometry Rectangle {
               size IS Box_size              #Size of displayed Box
              }
             }
             DEF MY_TEXT Transform2D {
              translation IS Text_Translation      # texte position
              children [
               DEF Text_Shape Shape {
                appearance DEF TEXTAPP Appearance {
                       material Material2D {
                        emissiveColor IS Text_Color
                        filled TRUE
                       }
               }
                geometry DEF peopleName Text {
                       string IS stringName
                       fontStyle DEF FS FontStyle {
                          justify ["MIDDLE" "MIDDLE"]
                            size 14
                        }
               }
              }
             ]
            }
        ]
       }
```

Annexe 6 – Résultats de la génération d'une scène interactive pour le mode différé ou en direct

Tableau 24 : Initialisation complète d'un service interactif

```
InitialObjectDescriptor {
 objectDescriptorID 1
 audioProfileLevelIndication 254
 visualProfileLevelIndication 245
 sceneProfileLevelIndication 1
 graphicsProfileLevelIndication 1
 ODProfileLevelIndication 1
 esDescr [
 #BIFS Stream
  ES_Descriptor {
   ES_ID 1
   decConfigDescr DecoderConfigDescriptor {
    objectTypeIndication 1
    streamType 3
    decSpecificInfo BIFSv2Config {
     isCommandStream true
     pixelMetric true
     pixelWidth 512
     pixelHeight 384
    }
   }
  }
 #OD Stream
  ES_Descriptor {
   ES_ID 2
   decConfigDescr DecoderConfigDescriptor {
    streamType 1
   }
  }
 ]
}
# Noeud racine de la scène
DEF Main_OG OrderedGroup {
 children [
  #main Audio stream
  Sound2D {
   source AudioSource {
    url [od:10]
   }
  }
#Video Layout
  DEF LAYOUT OrderedGroup {
   children [
    #main video stream
    DEF MAIN_VIDEO Transform2D
    {
     children [
      Shape {
       appearance Appearance {
        texture MovieTexture {
         url [od:20]
        }
       }
       geometry Bitmap {}
      }
```

```
    ]
   }
  ]
 }
# Noeud racine pour les services interactifs
 DEF InteractiveScene OrderedGroup {
  children []
 }
]
}
RAP AT 0 {
 UPDATE OD [
  #ObjectDescriptor for Main Video
  ObjectDescriptor {
   objectDescriptorID 20
   esDescr [
    ES_Descriptor {
     ES_ID 20
     muxInfo MuxInfo {
      fileName "Tac_2.avi#1"
     }
    }
   ]
  }
  #ObjectDescriptor for Main Audio
  ObjectDescriptor {
   objectDescriptorID 10
   esDescr [
    ES_Descriptor {
     ES_ID 10
     muxInfo MuxInfo {
      fileName "Tac_2.avi#2"
     }
    }
   ]
  }
 ]
}
AT 0 {
 INSERTPROTO [
  PROTO MY_FACE_PROTO [
                exposedField SFVec2f Box_translation  0 0
                exposedField SFVec2f Text_Translation 0 0
                exposedField SFVec2f scale 1 1
                exposedField SFFloat rotation 0
                exposedField SFFloat width 3
                exposedField SFVec2f Box_size 0 00
                exposedField SFColor Box_Color 1 0 0
                exposedField SFColor Text_Color 0 1 0
                exposedField SFBool filled FALSE
                exposedField SFFloat transparency 0
                exposedField SFColor lineColor 0 0 1
                exposedField SFFloat lineWidth 2
                exposedField SFBool enable_center_Align TRUE
                exposedField SFBool isOver TRUE
                exposedField SFBool enabled_Filled TRUE
                exposedField MFString stringName []
                exposedField SFNode NamefontStyle NULL
                eventIn MFString active
                exposedField SFNode obj NULL
] {
   DEF MY_BOX Transform2D{
         translation IS Box_translation     #BOX position
```

```
        children [
        DEF PeopleSensor TouchSensor{isOver IS isOver}
              Shape {
               appearance Appearance {
                material Material2D {
                 lineProps XLineProperties {
                      isCenterAligned IS enable_center_Align
                      width IS width                          #Width line of BOX
                      texture CompositeTexture2D {
                       pixelWidth 32
                       pixelHeight 32
                       children [
                        Shape {
                         appearance Appearance {
                            material Material2D {
                               emissiveColor IS Box_Color   filled IS enabled_Filled }
                            }
                         geometry Rectangle{ size IS Box_size }
                        }
                       ]
                      }
                 }
                }
               }
               geometry Rectangle {
                size IS Box_size      #Size of displayed Box
               }
              }
                      DEF MY_TEXT Transform2D {

               translation IS Text_Translation      # texte translation from BOX position
               children [
                DEF Text_Shape Shape {
                 appearance DEF TEXTAPP Appearance {
                        material Material2D {
                         emissiveColor IS Text_Color
                         filled TRUE
                        }
                 }
                 geometry DEF peopleName Text {
                        string IS stringName
                        fontStyle DEF FS FontStyle {
                          justify ["MIDDLE" "MIDDLE"]
                          size 14
                        }
                 }
                }
               ]
              }
        ]
       }
 }
]
}
#Déclaration ici des composants qui doivent être Définis
AT 0 {
 INSERT AT InteractiveScene.children[0]
  DEF Service_1_Face_Track OrderedGroup{
  children[
        DEF Face0 MY_FACE_PROTO {}
        DEF Face1 MY_FACE_PROTO {}
        DEF Face2 MY_FACE_PROTO {}
        DEF Face3 MY_FACE_PROTO {}
```

```
        DEF Face4 MY_FACE_PROTO {}
        DEF Face5 MY_FACE_PROTO {}
        DEF Face6 MY_FACE_PROTO {}
        DEF Face7 MY_FACE_PROTO {}
        DEF Face8 MY_FACE_PROTO {}
        DEF Face9 MY_FACE_PROTO {}
  ]
 }
}
```

En mode Différé :

Tableau 25 : Commandes de mise à jour en fonction de l'avancement dans le média

```
AT 80 {
 REPLACE Face0.Box_translation BY 14 7
 REPLACE Face0.Box_size BY 78 78
 REPLACE Face0.Text_Translation BY 0 -49
}
AT 320 {
 REPLACE Face0.Box_Color BY 1 0 0 REPLACE Face0.stringName BY ["Uncle Tom"]
 REPLACE Face0.Box_translation BY 78 78
 REPLACE Face0.Box_size BY 0 -49
}
AT 360 {
 REPLACE Face0.Box_translation BY 14 12
 REPLACE Face0.Box_size BY 78 78
 REPLACE Face0.Text_Translation BY 0 -49
}
...
```

Pour une diffusion en direct, le Tableau 26 présente la liste des commandes *BIFS Update*.

Tableau 26 : Commandes de mise à jour à diffuser en fonction des évolutions du contenu du document multimédia

```
#_RTP_STREAM_SEND
AT D1 {
 REPLACE Face0.Box_translation BY 14 7
 REPLACE Face0.Box_size BY 78 78
 REPLACE Face0.Text_Translation BY 0 -49
}

#_RTP_STREAM_SEND
AT D1{
 REPLACE Face0.Box_Color BY 1 0 0 REPLACE Face0.stringName BY ["Uncle Tom"]
 REPLACE Face0.Box_translation BY 78 78
 REPLACE Face0.Box_size BY 0 -49
}
#_RTP_STREAM_SEND
AT D1{
 REPLACE Face0.Box_translation BY 14 12
 REPLACE Face0.Box_size BY 78 78
 REPLACE Face0.Text_Translation BY 0 -49
}
```

...

Annexe 7 – Service interactif Horloge

La version en mode différé à obtenue à partir de l'exemple publié par [Concolato05].

Tableau 27 : Code du service interactif horloge pour un fonctionnement en mode « différé »

```
InitialObjectDescriptor {
 objectDescriptorID 1
 audioProfileLevelIndication 255
 visualProfileLevelIndication 254
 sceneProfileLevelIndication 1
 graphicsProfileLevelIndication 1
 ODProfileLevelIndication 1
 esDescr [
  ES_Descriptor {
   ES_ID 1
   decConfigDescr DecoderConfigDescriptor {
    streamType 3
    decSpecificInfo BIFSConfig {
     isCommandStream true
     pixelMetric true
     pixelWidth 130
     pixelHeight 30
    }
   }
  }
 ]
}
OrderedGroup {
 children [
  Background2D {
   backColor 1 1 1
  }
  WorldInfo {
   info ["This shows script used to retrieve system time" "" "GPAC Regression Tests"
"(C) 2002-2004 GPAC Team"]
   title "Script Date() test"
  }
  Transform2D {
   children [
    Shape {
     appearance Appearance {
      material Material2D {
       emissiveColor 0 0 0
       filled TRUE
      }
     }
     geometry DEF TXT Text {
      string ["MPEG4 time on your system"]
      fontStyle FontStyle {
       justify ["MIDDLE" "MIDDLE"]
       size 18
      }
     }
    }
   ]
  }
  DEF TIMER TimeSensor {
   loop TRUE
```

```
  }
  DEF SC Script {
   eventIn SFTime set_time
   field SFNode str USE TXT
   url ["javascript:
         function set_time(value, text) {
           today = new Date();
           the_hour = today.getHours();
           the_minute = today.getMinutes();
           the_second = today.getSeconds();
           am_pm = 0;the_initials = 'a.m.';
           if ((the_hour >=2) && (the_hour <=11)) {
               am_pm = the_hour;
               the_initials = 'a.m.';
           }
           else if (the_hour == 0) {
              am_pm = 12;the_initials = 'a.m.';
           }
           else if (the_hour == 12) {
              am_pm = 12;
              the_initials = 'p.m.';
           }
           else if (the_hour >=13) {
              am_pm = the_hour - 12;
              the_initials = 'p.m.';
           }
           if (the_minute <=9)
              the_minute = '0' + (the_minute);
           if (the_second <=9)
              the_second = '0' + (the_second);
           str.string[0] = am_pm + ':' + the_minute + ':' + the_second + ' ' +
the_initials;
         }
         function initialize() {set_time(0, 0);}"
       ]
  }
 ]
}
ROUTE TIMER.cycleTime TO SC.set_time
```