



HAL
open science

Système automatique de génération et de diffusion de cours et de conférences multimédias multilingues

Gilbert Sol, Olivier Poursac

► To cite this version:

Gilbert Sol, Olivier Poursac. Système automatique de génération et de diffusion de cours et de conférences multimédias multilingues. JRES (Journées réseaux de l'enseignement et de la recherche) 2007, Renater, Nov 2007, Strasbourg, France. hal-04802880v2

HAL Id: hal-04802880

<https://hal.science/hal-04802880v2>

Submitted on 29 Nov 2024

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

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



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Système automatique de génération et de diffusion de cours et de conférences multimédias multilingues

Gilbert SOL

ARTEMIS

Université Paris Diderot – Paris 7

2 place Jussieu

75 251 Paris Cedex 07

gilbert.sol@artemis.jussieu.fr

Olivier POURSAC

C.R.I

Université de la Sorbonne Nouvelle – Paris 3

1 rue Censier

75005 Paris

olivier.poursac@univ-paris3.fr

Résumé

ARTEMIS (Applications et Recherche en Télématique et Moyens d'Interconnexion de Systèmes) a initié, avec les étudiants du DESS ART (Applications des Réseaux et de la Télématique), de l'Université Paris Diderot – Paris7, en 2000, une recherche sur l'enregistrement et la génération automatique de cours, afin qu'ils soient rapidement consultables sur l'Internet. Une première étude (<http://www.artemis.jussieu.fr/projets2001>) s'est portée sur les standards de synchronisation et les logiciels de streaming. Dans un premier temps, nous avons retenu SMIL (Synchronized Multimedia Integration Language) et les produits de Realnetworks pour le streaming. Dès 2003, un premier système automatique est opérationnel (<http://www.artemis.jussieu.fr/projets2003>). En 2005, nous avons ajouté, en liaison avec l'université Sorbonne Nouvelle (<http://www.artemis.jussieu.fr/uee2005>), la possibilité de retransmettre et d'enregistrer dans une autre langue que celle d'origine. Depuis, plusieurs conférences ont été interprétées dans différentes langues, via le système mis en place. Pour les conférences, la notion de régie mobile a été traitée (<http://www.artemis.jussieu.fr/projets2005>). Le système présenté dans son état actuel peut à la fois être très compact (contenu dans un sac à dos) ou bien complètement distribué sur l'Internet mais en conservant sa principale caractéristique qui est l'automatisation de la génération de la mise en ligne de cours ou conférences multimédias consultables aussi bien en temps réel qu'en VOD.

Mots clefs

Multimédia, streaming, VOD, SMIL

1 Introduction

Des études montrent que nous ne retenons que 30 % d'un cours oral ou d'une conférence. La projection d'images, de schémas ou de synthèses de contenu, avec éventuellement

distribution imprimée, améliore certes l'efficacité. Néanmoins, l'ouverture sur l'Europe et le monde fait qu'un certain nombre de colloques ou séminaires se déroulent dans une langue différente de la langue maternelle ce qui s'ajoute à la difficulté de perception. De plus, l'éloignement et la contrainte d'un emploi du temps chargé, n'autorisent pas toujours la possibilité d'assister aux conférences souhaitées. Certes, le meilleur moyen est la diffusion en temps réel et la mise en ligne rapidement en VOD (Video On Demand). Mais, pour cela, il faut recréer l'ambiance de la conférence ou du cours, c'est-à-dire transmettre en synchronisation, les transparents électroniques, la vidéo et le discours du présentateur (voire son interprétation dans une ou plusieurs autres langues).

En VOD, une indexation adéquate va permettre un parcours non linéaire, beaucoup plus efficace. Une interprétation simultanée en plusieurs langues et une traduction des transparents électroniques apportent un confort non négligeable aux différents auditeurs. Toutes ces opérations doivent être automatisées afin de réduire les coûts de production et les temps de mise à disposition en VOD. De plus, elles doivent être complètement transparentes pour l'orateur afin de ne pas le perturber dans son exposé. Le système que nous proposons répond à toutes ces exigences. Cependant, il est conçu pour le traitement de conférences qui contribuent à la dissémination de la Recherche et de cours de type magistral qui entrent encore pour une bonne partie dans nos formations. Il ne prétend pas couvrir toutes les situations présentes dans l'acte d'enseignement bien qu'il puisse, dans certains cas convenir, soit tel qu'il est actuellement ou dans des évolutions futures.

2 Caractéristiques du système

Le système ainsi conçu est, entre autres, distribué, synchronisé, cohérent et évolutif.

2.1 Système distribué

Lors de colloques ou séminaires, les conférenciers ne sont pas tous dans un même lieu. Ils peuvent intervenir de sites

distants en utilisant des outils de visioconférence. De même, les auditeurs peuvent être répartis dans plusieurs salles, voire même suivre les exposés de chez eux ou de leur bureau. Dans le cas d'interprétations simultanées en plusieurs langues, les cabines de traduction risquent elles aussi de se trouver à distance. Car, soit les salles n'en sont pas équipées, soit le déplacement d'interprètes reste d'un coût relativement élevé. Grâce à l'Internet et à son architecture basée sur des agents communicant entre eux par le protocole OSUP à deux couches SUP (Smart UDP Protocole) et OUP (OTESA UDP Protocol), le système répond à ces diverses situations de façon synchronisée.

2.2 Système synchronisé

Une synchronisation parfaite des divers médias (Vidéo, audio, images) paraît difficile en temps réel, étant donné que le support de transmission est l'Internet et que les diverses sources peuvent se trouver sur des serveurs distincts, voire même avoir été produites avec un certain décalage comme lorsque l'on utilise une ou plusieurs interprétations dans plusieurs langues. Cette synchronisation doit cependant rester dans des limites supportables. En VOD, elle pourra être améliorée, à posteriori, par un agent spécialisé et devenir quasi-parfaite si tous les médias sont sur le même serveur.

Cette exigence de synchronisation peut difficilement s'appuyer sur une synchronisation des horloges des différentes stations entrant dans l'architecture du système, étant donné leur répartition, dépendant de chaque cours ou colloque. Cette synchronisation s'effectue alors par le protocole OSUP. L'horodatage est fourni par un agent « horodateur », opérant en général sur le poste de l'orateur, qui donne le début et la fin de l'intervention ainsi que l'instant de chaque événement comme par exemple, le passage à l'image suivante. Chaque agent, se cale donc sur cet horodatage pour maintenir la correspondance entre l'horloge de l'horodateur et sa propre horloge. Ce qui lui assure une bonne synchronisation.

2.3 Système cohérent

Le système doit assurer de nombreuses tâches : horodatage, numérisation, transmission et visualisation des images, transmission et enregistrement de la vidéo et du son, transmission et enregistrement des interprétations dans les différentes langues, préparation du produit final visualisable en local ou en VOD, transfert des différentes versions dans les diverses langues sur les serveurs de VOD, gravure sur CDROM, etc. Le nombre de tâches que l'on peut imaginer est quasi-infini. Il s'agit donc d'assurer une parfaite cohérence du système pour qu'il puisse évoluer. Il est donc basé sur une architecture distribuée. Chaque tâche est dévolue à un « agent » qui intègre sa propre interface de configuration. La configuration, basée sur un fichier au format XML peut intervenir par défaut au démarrage de l'« agent » ou être reçue via le protocole OSUP.

Il est rare qu'un agent intervienne seul. Par exemple, lorsqu'un conférencier présente son exposé avec des

images comme support, le minimum de tâches à accomplir consiste à saisir, enregistrer et transmettre la vidéo et le son (agent « streameur »), à numériser les images (agent « numériseur »), à les horodater (agent « horodateur »), à les transmettre sur un serveur HTTP (agent « Téléchargeur temps réel »). Dans le cadre de colloques, plusieurs conférenciers interviennent et il n'est pas rare qu'il y ait une certaine fluctuation dans les horaires prévus. Il peut être convenable d'en avertir les internautes qui ne veulent suivre qu'une partie des interventions. Un agent se charge alors d'intervenir, avant la première conférence (initialisation) puis entre chaque conférence pour, par exemple, donner le thème, voire le plan de la conférence suivante et enfin, à la fin de la dernière conférence. Ses interventions peuvent se faire par l'envoi aux serveurs de pages HTML ou d'images qui seront visualisées sur le poste des auditeurs distants, en attendant la prochaine conférence. Un autre exemple concerne le poste de l'interprète distant. Il doit recevoir en temps réel, sans temps de latence, la vidéo et l'audio de l'orateur (agent « visioconférence H323 ») ainsi que les images traduites qui lui permettront une plus rapide interprétation (agent « visualiseur »). Son interprétation doit être aussi retransmise sans délais (agent « streameur »).

Afin d'éviter l'échange de trop nombreux messages via le protocole OSUP, les agents sont regroupés en containers. Seuls, les containers communiquent entre eux grâce au protocole OSUP, sur un port UDP (ou TCP). Les containers gèrent la couche de transport SUP et transmettent aux agents le contenu de la couche OUP. Par défaut, le contenu OUP est distribué à tous les agents qui se chargent d'interpréter ce qui les concerne. La communication éventuelle avec un seul agent est précisé dans un paramètre de nom « receiver ».

Comme pour les agents, la configuration des containers est réalisée à travers des fichiers au format XML, précisant la nature et les identifiants des divers agents ainsi que l'emplacement de leur fichier de configuration.

2.4 Système évolutif

L'architecture répartie en containers et agents assure au système une grande souplesse. En effet, les agents sont créés en fonction des besoins, avec leur propre organisation pour le fichier de configuration et leur propre interface de configuration. Ils peuvent être programmés sous n'importe quel langage et sur n'importe quel système d'exploitation. Eventuellement sous forme de DLL qui pourra être intégrée à un container. Cette architecture assure une évolutivité sans limite.

3 Exemples de configuration

3.1 Configuration monoposte

La configuration la plus simple est celle ne faisant intervenir qu'un seul poste qui va assurer l'ensemble des tâches. Elle favorise la mobilité et peut, avec le matériel nécessaire, être contenue dans un sac à dos ou une valise. Matériellement, elle est constituée, d'une caméra

numérique avec un pied stable (semi-professionnel avec en particulier réglage du niveau horizontal par bulle d'eau), d'un mélangeur audio avec des entrées et sorties XLR pour pouvoir s'interconnecter avec n'importe quelle régie, d'un minimum de deux microphones sans fil, un microphone-cravate et un microphone-main (possibilité d'en avoir quatre), éventuellement d'un vidéoprojecteur et d'un micro-ordinateur. Le tout est transporté dans un sac à dos et coûte moins de 10 000 euros. L'ensemble du matériel peut aussi tenir dans une valise rigide pour des transports par avion. Depuis plus de trois ans, nous l'utilisons pour l'enregistrement de séminaires ou de colloques en France (par exemple pour les Assises du CSIESR à Seignosse) ou à l'étranger (Mauritanie, Maroc, Roumanie, ...)

Dans cette configuration, un seul container est utilisé. Actuellement deux applications intègrent ce container. Il s'agit de ARTS (*Aristote Real Time Slide*)¹, sous licence CeCILL² programmé en JAVA et SmartMacro qui est une macro-commande de Powerpoint en VBA (Visual Basic pour Applications). Dès le lancement, pour ARTS, ou l'ouverture de Powerpoint pour Smartmacro, le container est initialisé ainsi que les agents contenus, sans message afin de ne pas troubler la concentration du conférencier. En cas d'erreurs, celles-ci sont enregistrées dans un fichier. Cependant, l'interface de configuration intègre une commande de vérification qui peut être lancée avant le début de la conférence et qui vérifie l'existence des différents fichiers et la disponibilité des différents serveurs. Le container contient :

- un agent « horodateur » qui enregistre dans un fichier de « timing » XML au format OTESA, les temps de début et fin de la conférence et les instants de numérisation des images. Dans le cas d'utilisation de Powerpoint avec Smartmacro, les titres des diapositives et leur commentaire sont aussi enregistrés dans le fichier de timing.
- Un agent « numériseur » qui assure la numérisation des diapositives de l'orateur ainsi que celles traduites éventuellement dans les différentes langues, ou la numérisation de l'écran dans le cas d'ARTS, à intervalles réguliers paramétrables (en général toutes les 5 ou 10 s) ; la numérisation n'a lieu que s'il y a eu modification de l'affichage.
- Un agent « téléchargeur temps réel » qui transmet les images numérisées aux différents serveurs web en SSH (Secure Shell), protocole de transfert de fichiers, sécurisé, pour la consultation multimédia en temps réel.
- Un agent « streameur », (non encore implanté dans ARTS) pour la numérisation et la compression des sources audiovisuelles et leur transmission éventuelle vers des serveurs de

streaming. Dans la version mobile, le micro-ordinateur récupère la source vidéo via le port IEEE 1394 (*Firewire*). Ceci évite le rajout d'une carte PCMCIA de numérisation vidéo. Le son est numérisé depuis le mélangeur par l'interface audio du micro-ordinateur. Cependant, toute autre configuration matérielle est supportée à travers les fichiers de configuration du codeur de streaming. Actuellement, seul, le « *Producer Plus* » de la société Realnetworks est opérationnel. Mais d'autres agents supportant les produits Microsoft (*Windows media*) ou QuickTime sont en cours d'étude.

- Un agent « smileur » qui prépare tous les fichiers pour une consultation locale ou en VOD en SMIL (*Synchronized Multimedia Integration Language*). Il opère à partir des modèles de présentation référencés dans le fichier de configuration. Dans le cadre de Smartmacro sous Powerpoint, il s'appuie sur les titres des diapositives pour créer automatiquement un menu pour un parcours non linéaire de l'exposé. Le conférencier peut lui-même choisir les points de reprise en regroupant sous le même titre les diapositives entrant dans un même chapitre. Pour différencier les diapositives d'un même chapitre, il lui suffira d'intégrer des sous-titres non pris en compte par l'agent. Les commentaires peuvent être aussi traités et, par exemple, être visualisés en sous-titrage pour une éventuelle consultation par des malentendants. Cet agent s'appuie sur un éditeur qui peut être utilisé en autonome, pour du traitement à posteriori et l'ajout éventuel d'autres médias ou une interprétation dans une autre langue voire même une vidéo en langue des signes. Le produit final est visualisable grâce au « Real Player » de base gratuit et disponible sur tous les systèmes. Un agent créant un produit en « Flash » est en cours de réalisation.
- Un agent « graveur » qui enregistre tous les fichiers pour une présentation locale sur CDROM. Dès la fin de chaque présentation, le conférencier peut garder une copie multimédia chapitrée de son intervention. Il est possible de transmettre ces fichiers, par un agent « téléchargeur » sur un serveur ou un graveur haut débit pour distribuer ces produits aux participants dès la fin du colloque.

Si la salle de conférence ou de cours est connectée à l'Internet et si les pare-feux (Firewalls) laissent passer les connexions SSH, un agent « téléchargeur » transmettra les fichiers « distants » créés par l'agent « smileur » sur les différents serveurs intervenant dans la VOD. Les conférences ou cours sont ainsi immédiatement disponibles en multimédia et organisés en chapitre, si Smartmacro a été utilisée.

¹ <http://www.aristote.asso.fr/OTESA/Doc/DocARTS/index.html>

² <http://www.cecill.info>

3.2 Configuration avec régie

L'université Denis Diderot – Paris 7 a installé une salle multimédia pour des cours entre plusieurs sites. Elle est équipée de deux vidéoprojecteurs, de microphones pour les étudiants et de 4 caméras pilotables à partir d'une régie contiguë. Un micro-ordinateur complète l'équipement. Smartmacro et ARTS sont installés sur le micro-ordinateur. La configuration diffère peu de la configuration monoposte à l'exception de l'agent « streameur » qui est installé sur un micro-ordinateur en régie. Les sources audiovisuelles sont gérées directement depuis la régie. Au moment du démarrage du cours, le container du poste professeur lance l'agent « streameur » de la régie via le protocole OSUP. La synchronisation s'effectue donc automatiquement sur le début. Par contre, pour pouvoir enregistrer les questions, l'arrêt du streaming est effectué manuellement (option prévue dans la configuration de l'agent « streameur »). L'agent « streameur », au moment du lancement, vérifie s'il y a des enregistrements en cours. S'il détecte des actions de streaming, il les termine automatiquement.

Sur le poste professeur, le container est toujours actif sur une configuration par défaut. Tous les cours ayant lieu dans cette salle, sont donc enregistrés et prêts à être mis sur un serveur pour une consultation ultérieure en VOD.

3.3 Utilisation d'un serveur de synchronisation

Dans les deux configurations précédentes, l'essentiel des opérations est effectué par le poste professeur. Si le système se complexifie, la capacité de traitement de ce poste risque de ralentir les échanges. Pour de grandes opérations, il est donc préférable de l'alléger au maximum. Pour cela, on utilise un serveur de synchronisation qui peut posséder tous les agents disponibles. Muni d'une base de données des clients potentiels, il peut intervenir dans plusieurs événements. La connexion s'effectue à travers le protocole OSUP par login et mot de passe éventuellement chiffré. Un gros avantage est de pouvoir implanter les derniers développements d'agents et d'en fournir les ressources aux clients référencés. Dans cette configuration, le poste professeur opère avec un container réduit aux agents indispensables (« horodateur », « numériser d'images », « téléchargeur d'images »). Au moment de l'initialisation, ce container se connecte au serveur de synchronisation et demande les ressources dont il a besoin, en transmettant le fichier de configuration de chaque agent et laisse à ces agents le soin d'agir à sa place dans toutes les tâches complexes comme le streaming, le téléchargement, la génération des fichiers locaux ou distants pour la VOD, La configuration des agents peut aussi se faire a priori, en interactif sur le serveur ou par téléchargement dans l'espace réservé au client. Cet espace contient aussi les modèles SMIL et les fichiers par défaut. Il est possible de constituer un réseau de serveurs de synchronisation qui communiquent entre eux par le protocole OSUP, pour pouvoir étendre sans limite la diffusion d'un événement.

4 Protocole OSUP

OSUP est la compression de SUP (*Smart UDP Protocol*) et OUP (*OTESA UDP Protocol*), les deux couches composant ce protocole.

Initialement, le protocole avait été conçu pour opérer au dessus de UDP (*User Datagram Protocol*) puisqu'il est basé sur l'envoi de messages. Cependant, sa structure est indépendante de la couche de transport et il est possible de l'implémenter sur TCP (*Transport Control Protocol*), en particulier lorsqu'il s'agit de passer des pare-feux (*Firewalls*) en choisissant, par exemple, le port 80 (port HTTP).

4.1 Couche SUP

La couche SUP est le support de trames pour le transport des commandes. Elle fonctionne en mode connecté du fait de la présence éventuelle de serveurs de synchronisation. La couche SUP est chargée de délivrer dans l'ordre à la couche OUP, les trames émises. Elle assure la gestion des acquittements, la segmentation et le chiffrement éventuel. Les trames SUP sont structurées en XML.

1.1.1 Structure générale des trames

Toutes les trames respectent la structure XML suivante :

```
<supMsg>
  <sender   address = 'adresse de l'expéditeur
             port   = 'Port d'écoute de l'expéditeur
             identifi er = /> 'Renseigné par l'application
  [<receiver identifi er = />] ' Container de destination
  <frame   serialNumber = 'sur 4 octets
             identifi er = 'Renseigné par l'application
             encryption = 'algorithme de chiffrement, renseigné
                par l'application, mais effectué par SUP
             ack = « Yes|No »>
  <command name = (éventuellement chiffrée)>
    <parameter [name = ]
                [type = string | xmlElement | xmlAttribut | ...]>
      Valeur = (éventuellement chiffré) : caractères spéciaux
      XML codés avec &...;
    </parameter>
    <parameter [name = ] [type = ]
      valeur
    </parameter>
    ....
  </command>
  <command name = (éventuellement chiffrée)>
    ...
  </command>
  ....
```

</supMsg>

L'attribut « *identifier* » de la balise « *sender* » servira de « *login* » lors d'une connexion à un serveur de synchronisation. La combinaison de « *address* », « *port* » et « *identifier* » identifie de façon unique dans le système, le container émetteur.

La balise « *receiver* » n'est utilisée que lorsque l'application réceptrice contient plusieurs containers sous le même port.

Le chiffrement s'effectue sur le nom des commandes et, éventuellement, sur les valeurs des paramètres (« *parameter* »), en fonction de l'algorithme choisi.

L'attribut « *identifier* » de la balise « *frame* » dépend uniquement de l'application. Il permet de vérifier qu'une trame particulière a bien été reçue puisque cet attribut est présent, sans modification dans la trame d'acquiescement.

Le numéro de trame, attribut « *serialNumber* » de la balise « *frame* » est choisi de façon aléatoire pour l'envoi de la première trame (de préférence inférieur à 1 000 000 000) puis est incrémenté de un pour chacune des trames suivantes pour que le récepteur puisse reconstituer l'ordre d'envoi.

Les attributs et la valeur des balises « *parameter* » sont gérés par la couche OUP.

1.1.2 Gestion des acquiescements

L'acquiescement d'une trame, lorsqu'il est demandé (attribut « *ack* » de la balise « *frame* » à « *yes* »), s'effectue par une trame d'acquiescement dont le nom de la commande (attribut « *name* » de la balise « *command* ») est « *OK* ». Les attributs « *identifier* » des balises « *sender* » et « *frame* », ainsi que l'attribut « *serialNumber* » de la balise « *frame* » sont ceux de la trame acquiescée. L'algorithme de chiffrement correspond à celui de la trame acquiescée.

Pour minimiser les effets de tentatives de déni de service, la couche SUP ignore les trames erronées : non-conformes au format, arrivées hors délai (paramétrable), de nom de commande inconnu, déjà acquiescées un nombre de fois (10 par défaut) paramétrable, etc.

1.1.3 Segmentation

Sous UDP, la longueur des datagrammes est limitée, non seulement par le protocole sous jacent (IP : *Internet Protocol*), mais aussi par les mémoires tampons implémentées dans les piles de protocole par les différents systèmes. Il semblerait que, sous Windows XP, la longueur maximum par défaut soit de l'ordre 8 200 octets. Or, lors de la transmission, par exemple, de configurations pour les divers agents, une trame peut dépasser cette longueur. Le protocole SUP utilise alors une trame dont la commande a pour nom « *SEGMENT* » (valeur de l'attribut « *name* » de

la balise « *command* »). Sa structure est conforme à la description suivante :

```
<supMsg>
  <sender address = 'adresse de l'expéditeur
    port = 'Port d'écoute de l'expéditeur
    identifier = /> 'de la trame initiale
  [<receiver identifier = />] 'Container de destination
  <frame SerialNumber = 'sur 4 octets en séquence
    identifier = 'de la trame initiale
    encryption = 'de la trame initiale : algorithme de
chiffrement
    ack = « Yes|No »/> 'de la trame initiale
```

'Partie segmentation

```
<command name = « SEGMENT »(éventuellement chiffré)>
  <parameter name = « firstSegment »>
    Numéro du premier segment (éventuellement chiffré)
  </parameter>
  <parameter name = « nbOfSegments »>
    nombre total de segments (éventuellement chiffré)
  </parameter>
  <parameter name = « contents »>
    segment de la trame initiale (éventuellement chiffré)
    caractères spéciaux XML codés avec &...;
  </parameter>
</command>
</supMsg>
```

La segmentation est gérée au niveau des paramètres (balises « *parameter* »). La couche SUP est chargée de reconstruire la trame initiale.

4.2 Couche OUP

La couche OUP est chargée de la gestion des commandes avec leurs divers paramètres. Elle vérifie en premier la présence du paramètre « *receiver* » dont la valeur, si elle existe, précise l'identifiant de l'agent auquel est destinée la commande. Ces commandes sont regroupées selon leurs fonctions.

1.1.4 Commandes de gestion

La couche SUP fonctionnant en mode connecté, les deux commandes « *CONNECT* » et « *DISCONNECT* » gèrent la connexion. Une troisième commande « *CONNECTTO* » remplace « *CONNECT* » lorsqu'un container désire recevoir d'un serveur de synchronisation les trames provenant d'un agent « *timeur* ». L'identifiant du container de cet agent est précisé comme valeur du paramètre (balise « *parameter* ») de nom (attribut « *name* »)

« *identifier* ».

La commande « *READY* » indique que la connexion a été réalisée. Elle contient une liste de paramètres, donnant des indications sur les capacités de traitement du récepteur comme, par exemples, les divers algorithmes de chiffrement supportés.

« *INIT* » initialise les échanges avec l'envoi éventuel des fichiers de configuration qui peuvent être modifiés, par la suite, grâce à une commande « *CONFIGURATION* ».

1.1.5 Commandes de type OTESA

Les autres commandes sont transmises directement aux couches supérieures, avec les différents paramètres qui correspondent en grande partie aux éléments de description des fichiers au format OTESA³. La liste actuelle est la suivante :

- « *SHOWBEGIN* » : pour indiquer le début d'un ensemble de conférences (correspond, par exemple, à une demi-journée ou à un cours en deux parties avec entre les deux, une pause).

- « *PRESEBEGIN* » : début d'une conférence (ou présentation).

- « *SLIDE* » : indique l'affichage d'une image qui vient d'être numérisée. C'est, soit une nouvelle numérisation d'écran dans le cadre de l'utilisation de ARTS, soit le passage à une nouvelle diapositive Powerpoint pour Smartmacro.

- « *SLIDSENT* » : commande émise par un agent « téléchargeur » pour indiquer que le fichier image a bien été entièrement téléchargé. Cette commande facilite la programmation des serveurs de synchronisation qui agissent dès la réception de cette commande sans avoir besoin de surveiller l'arrivée complète d'une image annoncée par « *SLIDE* ».

- « *PRESEND* » et « *SHOWEND* » : lors de la fin respectivement d'une conférence ou d'une présentation et de l'ensemble des conférences.

- « *ENDPROCESS* » : indique, dans le paramètre « *sender* », l'identifiant de l'agent qui a terminé sa tâche. Cette commande est très utilisée. En effet, certains agents n'entrent en action que lorsque d'autres ont terminé leur tâche. Il en est ainsi de l'agent « strimeur » qui ne peut reconstruire les cours multimédias que lorsqu'il possède le fichier de « *timing* » au complet et lorsque les enregistrements audiovisuels sont effectués. De même, l'agent graveur ne peut intervenir qu'après

que l'agent « strimeur » ait terminé. Un paramètre optionnel « *type* » permet de préciser le type d'agent ayant transmis la commande.

Les paramètres des commandes « *SHOWBEGIN* », « *PRESEBEGIN* », « *SLIDE* », « *PRESEND* », « *SHOWEND* », correspondent aux attributs des balises « *show* », « *presentation* » et « *slide* » définies dans les fichiers de « *timing* » au format OTESA.

1.1.6 Commande « *OTHER* »

Le protocole OSUP peut, certes, encore évoluer et des commandes indispensables peuvent s'avérer nécessaire. Cependant, pour garder une entière compatibilité entre les diverses applications, l'ajout de telles commandes ne doit se faire qu'en raison d'un absolu besoin. Néanmoins, pour ne pas figer les applications et les usages, une commande générique « *OTHER* » va pouvoir étendre l'utilisation de ce protocole. Le nom d'une nouvelle commande est donc mis comme valeur du paramètre (balise « *parameter* ») de nom (attribut « *name* ») « *command* » et son interprétation est laissée aux couches supérieures de l'application.

5 Exemples d'utilisation en Télé-interprétation

5.1 Conférence avec interprétation simultanée à distance.

(Expérience réalisée lors de l'Université Européenne d'été 2006 entre l'Université Denis Diderot – Paris 7 et L'Université de la Sorbonne Nouvelle – Paris 3).

Les conférenciers présentaient leur exposé depuis la salle de cours multimédia de l'université Denis Diderot à Jussieu, équipée du micro-ordinateur muni de Smartmacro avec, en particulier, les agents « numériseur » et « téléchargeur » activés. Deux interprètes, l'une en anglais, l'autre en espagnol, intervenaient depuis l'ESIT (Ecole Supérieure d'Interprètes et de Traducteurs) de l'Université de la Sorbonne Nouvelle située à Paris, Porte Dauphine.



Figure 1 - Poste d'une interprète

Dans les cabines (figure 1), chaque interprète recevait sur un écran vidéo, le conférencier via un codec H323. Le son était retransmis par ce codec et reçu par les interprètes, grâce à leur casque. Ici, la visioconférence s'impose pour obtenir un véritable temps réel sans latence. Nous travaillons actuellement, à la mise au point d'un poste autonome d'interprétation qui intégrerait la visioconférence sur micro-ordinateur, dans un premier temps avec des logiciels de type PVX, Netmeeting ou EKIGA⁴ puis, directement via un agent « visioconférence H323 ». Dans le cas d'interprètes sur plusieurs sites, il est nécessaire d'utiliser un pont de visioconférences MCU (*Multipoint Unit Control*).

Un second micro-ordinateur, connecté au serveur de diapositives via l'agent « visualiseur », permettait l'affichage des documents traduits. Chaque interprète reçoit ainsi, les transparents le concernant, en fonction de

sa langue d'interprétation (figure 2). Pour améliorer la rapidité d'affichage, un deuxième type d'agent « visualiseur » a été créé. Il s'agit d'un agent « visualiseur PPT ». Le fichier du conférencier ayant été traduit à priori, il est préalablement téléchargé sur le poste interprète. Le « visualiseur PPT » reçoit, via le protocole OSUP, les événements de changement de diapositive, contenant en paramètres, le numéro de diapositive et le nom du fichier dont elle est issue. L'agent « visualiseur PPT » interprète ces paramètres et affiche immédiatement l'image qui réside sur le disque local.

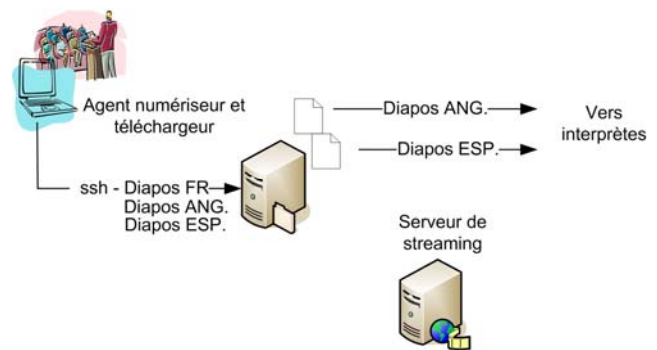


Figure 2 - Flux images

Sur ce même micro-ordinateur, un microphone permettait, via un agent « streameur », de transmettre au serveur de streaming l'interprétation dans la langue choisie. Chaque flux audio est numérisé au débit de 56 kbit/s et transmis au serveur de streaming (figure 3). Ces flux sont identifiés en fonction de leur langue d'interprétation :

- Audio-1-En.rm pour une version anglaise
- Audio-1-Sp.rm pour une version espagnole

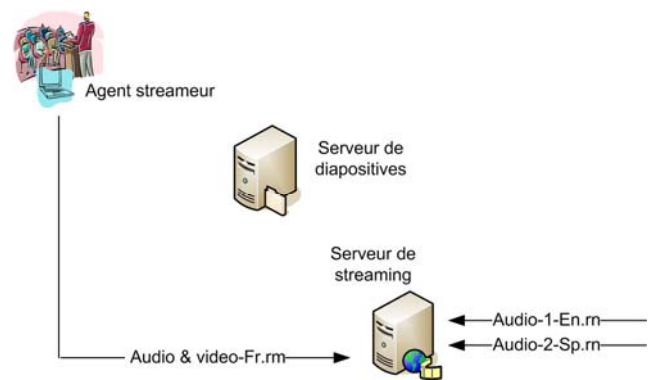


Figure 3- Flux audiovisuels

Les internautes, via un serveur web, choisissaient une langue (français, anglais ou espagnol). Ils pouvaient alors assister à la conférence, dans la langue choisie (audio et images) et voir l'orateur, via le « Realplayer » de base (gratuit) qui interprétait un fichier SMIL (figure 4).

⁴ EKIGA : logiciel libre de visioconférence utilisant les protocoles H323 et SIP (*Session Initiation Protocol*), <http://www.ekiga.org>

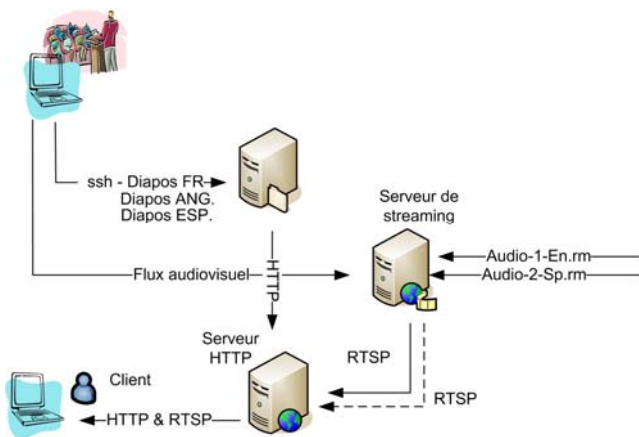


Figure 4 - Consultation temps réel

Le langage SMIL, recommandation du W3C (*World Wide Web Consortium*), respecte le format XML. Il synchronise des sources multimédias qui peuvent se situer sur des serveurs distants, différents. L'affichage s'effectue dans des « régions » préétablies sur des fonds d'écrans personnalisés : exemple sur la figure 5. Pour des sources audiovisuelles, il est possible grâce à « soundLevel », attribut de balise, de régler le volume du son. Pour une interprétation dans une langue étrangère, le volume du son de la source d'origine est mis à zéro ou à quelques pour cent si l'on veut entendre la voix initiale de l'orateur, alors que la langue d'interprétation est à 100 %.

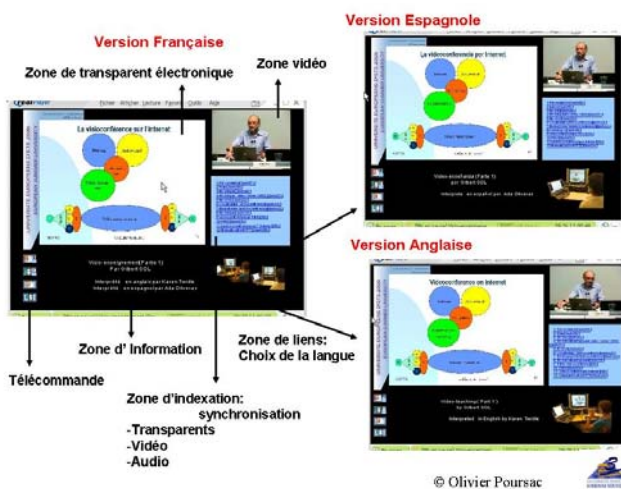


Figure 5 - Exemple d'affichage (SMIL)

Remarque : une interprétation du même type avait été faite en polonais, en juillet 2005, lors de l'université européenne d'été de 2005⁵.

⁵ <http://www.artemis.jussieu.fr/uee2005>

5.2 Conférence par visioconférence avec interprétation simultanée à distance.

(Expérience réalisée lors d'une conférence par visioconférence entre L'Université de la Sorbonne Nouvelle et L'Institut franco-japonais à Tokyo).

Les auditeurs japonais écoutent une conférence d'un orateur qui intervient en français depuis l'université de la Sorbonne Nouvelle à Paris. L'interprète français vers japonais, pour les auditeurs japonais, est à Tokyo. D'autres interprètes (anglais et espagnol) sont à l'ESIT à Paris. Pour une meilleure qualité, l'utilisation de deux codecs H323 s'avère nécessaire.

- Le premier codec est réservé à la transmission de la conférence de Paris vers l'interprète de Tokyo.
- Le second codec est réservé pour les interprètes de l'ESIT.

Sur la figure 6, les codecs sont notés visio-1 et visio-2

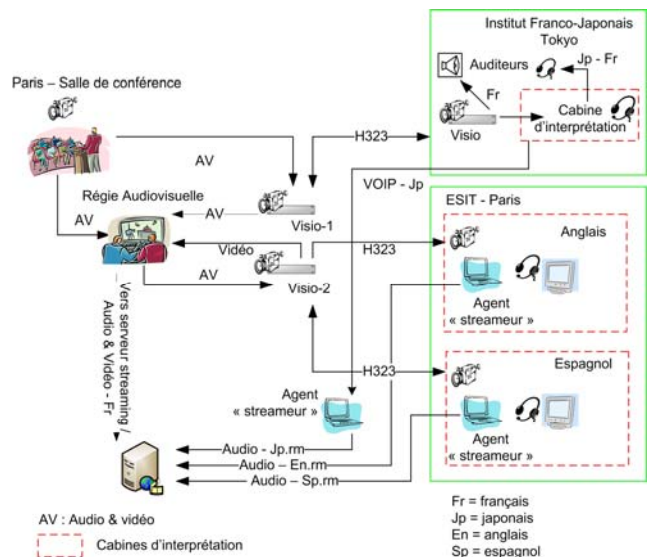


Figure 6 - Flux audiovisuels

L'intervention du conférencier français est retransmise en simultané au codec vidéo-1, pour sa transmission en H323 vers Tokyo, et à une régie audiovisuelle qui va assurer la retransmission en H323, via le codec video-2, vers le site de l'ESIT pour l'interprétation en anglais et Espagnol, avec le même dispositif que pour l'expérimentation précédente. Depuis la régie, le flux audio-visuel est récupéré sur un micro-ordinateur, par un agent « streameur » pour la mise à disposition des internautes via un serveur de streaming comme décrit en 5.1. Le flux audiovisuel reçu à Tokyo est retransmis dans la salle et dans la cabine d'interprétation. Les auditeurs peuvent ainsi choisir la version française ou japonaise tout en voyant l'intervenant sur grand écran. L'interprétation japonaise est récupérée en VOIP (voix sur IP) à Paris, via le logiciel EKIGA et retransmise à un agent « streameur » pour la mettre à disposition sur le serveur de streaming. L'organisation des flux images et l'architecture des serveurs restent identiques à celles précédemment décrites en 5.1.

Cet exemple associe l'interprétation à distance et la possibilité d'assister à une conférence dans une langue différente de celle de l'orateur, sans avoir besoin de se déplacer.

6 Conclusion

Ces deux exemples montrent l'intérêt du Système automatique de génération et de diffusion de cours et de conférences multimédias multilingues (SAGéDiC²M² voire SAGéDiC²M³ dans sa version mobile), adaptable à différentes situations. Les développements ultérieurs vont aussi prendre en compte le langage des signes pour malentendants. En effet, l'ESIT intègre aussi ce type d'interprétation dans ses formations. L'architecture du système associant des agents autonomes et éventuellement des codecs de visioconférence laissent une grande indépendance d'adaptation à des besoins spécifiques et surtout à une automatisation d'une mise en ligne pour une consultation multimédia en temps réel ou en différé.

